DECOMPOSITION AND SAMPLING METHODS FOR STOCHASTIC EQUILIBRIUM PROBLEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
MANAGEMENT SCIENCE AND ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
OPERATIONS RESEARCH

Vinayak (Uday) V. Shanbhag
December 2005

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Walter Murray
(Principal Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Peter W. Glynn

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Michael A. Saunders

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Gerd Infanger

Approved for the University Committee on Graduate Studies:

Abstract

This thesis presents some algorithms for equilibrium programming under uncertainty. We consider three equilibrium problems in which agents solve continuous stochastic nonlinear programs. The problems differ primarily in the structure of the agent problem. In each instance, we provide a sampling-based method for obtaining an equilibrium point. Convergence theory and computational results are also provided.

In Chapter 2, we consider stochastic Nash games. Such problems require the determination of a Nash equilibrium in which agents solve stochastic optimization problems. In particular, we require that agents solve two-stage stochastic quadratic programs (QPs). We show that the resulting equilibrium point may be obtained by solving a larger stochastic QP. A dual-decomposition method for solving this problem is suggested with two important modifications. First, we use QP duality to construct modified feasibility and optimality cuts. Second, we use inexact cuts that require only dual feasible solutions. Convergence of the resulting algorithm is proved. We also present a cut-sampling method that uses a sample of the cuts to construct probabilistic bounds on the optimal value. We demonstrate the performance of both methods on a two-stage stochastic QP test problem set. It is observed that using inexact cuts may reduce computational effort by as much as 40%. The sampling method results in solutions that are within 3% of the optimal solution when a fixed sample size of 5000 is used at each iteration.

Chapters 3 and 4 focus on mathematical programs with complementarity constraints (MPCCs). In Chapter 3, we present a new method for solving such problems, while in Chapter 4, we extend this method to stochastic MPCCs. A description of both chapters follows.

Interior point methods for mathematical programs with complementarity constraints (MPCC) were first proposed by Luo et al. [LPR96] in their monograph. They have been studied extensively over the past decade. However, existing theory supports convergence to first-order Karush-Kuhn-Tucker (KKT) points. In Chapter 3, we present a method that ensures that the iterates converge to a second-order KKT point. This is achieved by employing modified Newton and negative curvature directions to conduct a curvilinear search on an augmented Lagrangian merit function. Convergence theory for the method is provided. We also demonstrate that the curvilinear search approach competes well with a linesearch approach on a test problem set of quadratic programs with complementarity constraints (QPCCs).

We consider the generalization of the MPCC to the two-stage case under uncertainty and term the resulting problem a stochastic MPCC. Such problems arise in the study of Stackelberg equilibria under uncertainty. In Chapter 4, a new primal-dual method is described for this class of ill-posed stochastic nonlinear programs. The method relies on sampling to construct the linearized KKT system, which is subsequently solved using a scenario-based decomposition. Computational results from a test set of stochastic MPCCs are provided.

In Chapter 5, we construct a two-period spot-forward market under uncertainty. Such games

may be formulated using a Nash-Stackelberg framework. Such a structure results in a Nash-Stackelberg equilibrium (NSE). However, computing such equilibria or verifying their existence remains difficult. Instead, we propose a simultaneous stochastic Nash game that requires the solution of a stochastic complementarity problem. We prove that under certain conditions, the simultaneous stochastic Nash equilibrium (SSNE) is an NSE. We present a sampling-based iterative-decomposition method for solving such problems efficiently and provide convergence theory for the method. Scalability is demonstrated on a class of stochastic complementarity problems. We also provide some policy-based insights using a 6-node model of an electricity market.

Acknowledgments

Over the past five years, my advisor Professor Walter Murray has perfected the role of advisor, friend, and mentor. His advising has been deep and insightful and his expansive knowledge of optimization constantly amazes me. My objective has always been to meet the extraordinarily high standard of his own research. His warmth and kindness towards my entire family has made for a wonderful relationship. He has always been very understanding about changes on the personal front and has always provided me with unstinting support. I shall remain indebted for all that he has done for me.

I would like to thank Professor Peter Glynn for his guidance, support and advice on a variety of fronts. His encouragement and advice have been crucial to this research. His incisive comments and accent on rigor and clarity made an enormous difference to the quality of work. His inordinate ability to always ask the right questions has raised the level of the work significantly. It has been a privilege and honor to have worked with him.

Professor Michael Saunders has been instrumental in helping me with the algorithm design and implementation. His detailed comments on the theoretical and computational results, the language and the use of LaTeX have had a profound impact on the quality of this dissertation. He has been generous to a fault with his time and his suggestions unfailingly improve the performance of my code. I shall miss his warmth, kindness, and good humor. He has inspired a whole generation of students with his incomparable commitment to the field of optimization.

I would also like to thank Professor Gerd Infanger for his support throughout my stay here. He was one of the first professors, I met upon my arrival here and his suggestions were crucial in constructing the research problem. His knowledge of sampling methods and stochastic programming proved immensely helpful in my own research.

During the last few years, several other professors have been extremely helpful. I would like to thank Professor Benjamin Hobbs and Professor Shmuel Oren for their many conversations on electricity markets. I would also like to thank Professor Richard Cottle for teaching me optimization and complementarity theory. It was truly an honor to learn from him. Finally, I would also like to thank Professor M.A. Pai. His advice and mentorship has been enormously beneficial.

I would like to thank the staff at the Department of Management Science and Engineering at Stanford University, in particular Lorrie Papadakis, Juanita Winkelman, Roz Morf and Lori Cottle. The Systems Optimization Laboratory has been a wonderful place, largely due to the wonderful labmates, particularly, Michael Friedlander, Victor DeMiguel, Kien-Ming Ng, Maureen Doyle, Che-Lin Su, Kaustuv, Samantha Infeld, and Alexis Collomb.

Several people have had an enormous impact on me over the years. My aunt ("Nini") and uncle ("fulaji baba") cared for me as their own when I was an undergraduate student in Bombay. Their legacy has been carried on by my cousin Deepu, my brother-in-law Sameer and my wonderful nephew Akhil. Their support and kindness made our stay in California extremely memorable. My

brother Uttam and his wife Sofia have always been there for us. Uttam picked me up from JFK airport in New York City ten years ago and continues to stretch out constantly to help us along. His constant advice and encouragement has made the journey a little less arduous.

I would like to thank my mother-in-law Maya Joshi for being a wonderful caring presence. Her support has been crucial at various points in our lives. Both she and my brother-in-law Anjaneya Joshi have been enormously kind and supportive.

This dissertation is dedicated to my parents, my wife Aparna, and our son Aaranya. It is as much theirs as it is mine. My parents ("Aai" and "Baba") have sacrificed an immense amount over the years. I cannot imagine having progressed very far without their love, support, and guidance. Above all, I would like to thank Aparna for everything. She has sacrificed and suffered an incredible amount for this thesis. She had the unforgivably dubious privilege of starting her academic career while being faced with my hectic travel schedule. But her encouragement and good humor never waned and it was indispensable to my functioning. Aparna and Aaranya's love has made this thesis and all else possible.

Contents

List of Tables, Figures, and Algorithms **xi**

1 Introduction **1**
1.1 Stochastic Nash Equilibria . . . 2
1.2 A Second-Order Method for MPCCs . . . 3
1.3 Stochastic Stackelberg Equilibrium Problems . . . 5
1.4 Forward Contracting under Uncertainty in Electricity Markets . . . 6

2 A Sampling Method for Two-Stage Stochastic QPs **8**
2.1 Stochastic Nash Games . . . 9
2.1.1 Deterministic Games . . . 10
2.1.2 Introducing Uncertainty . . . 13
2.2 An Inexact L-shaped Method for Stochastic QPs . . . 17
2.2.1 The Two-Stage Stochastic Quadratic Program . . . 18
2.2.2 Some Properties of $\mathcal{Q}(x)$ . . . 19
2.2.3 Optimality and Feasibility Cuts . . . 21
2.2.4 The Master Problem and Subproblem . . . 23
2.2.5 Upper and Lower Bounds . . . 24
2.2.6 Introduction of Inexact Cuts . . . 26
2.2.7 Convergence Theory . . . 27
2.3 The Inexact-Cut-Sampling Algorithm . . . 30
2.3.1 Exact Cut Sampling . . . 30
2.3.2 Probabilistic Bounds . . . 34
2.3.3 Inexact Cut Sampling . . . 34
2.3.4 Stopping Rule and Confidence Intervals . . . 36
2.4 Computational Results . . . 37
2.4.1 A Test Problem Set . . . 37
2.4.2 Inexact L-Shaped Method for Stochastic QPs . . . 38
2.4.3 A Sampling-based Method . . . 40
2.5 Contributions and Future Research . . . 43

3 A Second-order Method for MPCCs **44**
3.1 Introduction . . . 44
3.1.1 Early research on MPCCs . . . 44
3.1.2 A Second-Order Method . . . 45
3.1.3 Some Definitions and Background . . . 46
3.2 An Interior Point Method for MPCCs . . . 49

3.2.1 The Barrier Problem . . . 50
3.2.2 The KKT Conditions of MPCC(γ, μ) . . . 51
3.2.3 The Linearized KKT Conditions . . . 51
3.2.4 A Negative Curvature Direction . . . 52
3.2.5 Globalization through a Curvilinear Search . . . 54
3.2.6 Termination Criteria . . . 56
3.3 Global Convergence to Second-Order Points . . . 56
3.3.1 Existence of Iterates . . . 58
3.3.2 Existence of Parameter Sequences . . . 59
3.3.3 Boundedness of Iterates . . . 64
3.3.4 Boundedness of α_k . . . 67
3.3.5 Global Convergence Results . . . 71
3.4 Implementation Details and Numerical Results . . . 77
3.4.1 Some Implementation Details . . . 78
3.4.2 A Test Problem Set . . . 79
3.4.3 Numerical Results . . . 79
3.5 Contributions and Future Research . . . 81

4 A KKT Sampling Method for Stochastic MPCCs 83
4.1 Introduction . . . 83
4.1.1 Early Research on Stochastic Programming . . . 84
4.1.2 Research on SMPCCs . . . 84
4.2 An Interior Point Method for MPCCs . . . 85
4.2.1 The Barrier Problem . . . 86
4.2.2 The KKT Conditions . . . 86
4.2.3 The Linearized KKT Conditions . . . 87
4.2.4 A Multi-phase Approach . . . 88
4.2.5 A Negative Curvature Direction . . . 90
4.2.6 Specifying a Steplength . . . 92
4.2.7 Termination Criteria . . . 92
4.2.8 Algorithm Statement . . . 93
4.3 Introducing KKT Sampling . . . 93
4.3.1 Solving the Sampled KKT systems . . . 94
4.3.2 Constructing a Sample . . . 95
4.3.3 Ensuring Global Convergence . . . 95
4.4 Numerical Results . . . 95
4.4.1 `SQPEC`gen: A Test Problem Set for Stochastic QPECs . . . 96
4.4.2 Computational Results . . . 97
4.5 Contributions and Future Research . . . 97

5 Forward Contracting under Uncertainty **101**
5.1 Introduction . . . 101
5.2 Modeling Spot-Forward Markets . . . 102
5.2.1 The Spot-Market Equilibrium . . . 103
5.2.2 A Nash-Stackelberg Equilibrium (NSE) . . . 104
5.2.3 A Simultaneous Stochastic Nash Equilibrium (SSNE) . . . 105
5.2.4 From SSNEs to NSEs . . . 107
5.2.5 A Numerical Comparison . . . 110
5.3 Application: Electricity Forward Markets . . . 112
5.3.1 Spot-Market Equilibrium . . . 112
5.3.2 A Nash-Stackelberg Equilibrium . . . 116
5.3.3 A Simultaneous Stochastic Nash Equilibrium . . . 117
5.3.4 Relaxing the Risk-neutrality Constraint . . . 117
5.4 A Decomposition-based Splitting Algorithm . . . 119
5.4.1 The DS Algorithm: Description and statement . . . 119
5.4.2 The DS Method: Convergence Theory . . . 121
5.4.3 Introducing Sampling . . . 123
5.4.4 Computational Experience . . . 124
5.5 An Electricity Market Model . . . 126
5.5.1 Description of the model . . . 126
5.5.2 Expected Profits . . . 128
5.5.3 Expected Spot-market Sales . . . 128
5.5.4 Forward Prices . . . 128
5.6 Contributions and Future Research . . . 129

Bibliography **131**

Tables, Figures, and Algorithms

Tables

2.1 Input parameters 38
2.2 Comparison of two update rules with the exact-cut L-shaped method 39
2.3 Performance of sampling method 40

3.1 Comparing linear-slow and linear-fast 80
3.2 Comparing curvilinear-slow and curvilinear-fast 81

4.1 Comparision of serial and parallel times 99
4.2 Comparision of `kktsampling` and `kktsorting` strategies 99

5.1 Computing SSNE and NSE with zero costs 111
5.2 Computing SSNE and NSE with random costs 111
5.3 Comparison of PATH with the DS method and Monte-Carlo-DS 124
5.4 Comparison of DS method with Monte-Carlo sampling, residual sorting and residual sampling 125
5.5 Specification of cost functions 127
5.6 Specification of cost functions 127

Figures

1.1 Stochastic Nash equilibrium 3
1.2 Stackelberg equilibrium 4
1.3 Stochastic Stackelberg equilibria 5
1.4 Stochastic Nash-Stackelberg equilibrium 7

2.1 Performance profiles for different updates in the ILS method 39
2.2 Confidence bounds from sampling method 41
2.3 Confidence bounds from sampling method 42

3.1 Performance profiles for major iterations required by four methods 80

4.1 Reduced Hessian and a basis for null-Space of hessian: K = 200, n = 2 91
4.2 The KKT sampling method: scalability and serial vs. parallel 98
4.3 Performance profiles 98

5.1 Finite generation capacities and quadratic costs of generation 115

5.2 Uncertainty in capacity and demand functions 118
5.3 Scalability: CPU time 125
5.4 Scalability: number of LCPs 125
5.5 Schematic of settings 127
5.6 Expected profits 128
5.7 Expected sales 129
5.8 Expected prices 130

Chapter 1

Introduction

The concept of an equilibrium has wide applicability across diverse fields. For instance, it may be used to articulate a notion of balance between reactants and products in a chemical reaction or supply and demand in an economic system. In a game-theoretic setting, the term refers to an outcome of a game between players in which no player has any incentive to deviate from his strategy. Our interest is in precisely such a class of equilibrium problems and we shall use the term predominantly in this context.

Equilibrium problems represent a generalization of optimization problems [NW99] to a setting with multiple players. The underlying optimization problem, a constrained nonlinear program (NLP), may be formulated as

NLP	$\underset{x,y}{\text{minimize}}$	$f(x,y)$
	subject to	$c(x,y) = 0,$
		$x, y \geq 0,$

where f and c are smooth functions and $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. In its most general form, an equilibrium problem, often referred to as a game, would imply a set of decisions x^i, where x^i solves the following parametric optimization problem:

NLP(x^{-i})	$\underset{x^i,y^i}{\text{minimize}}$	$f_i(x^i, y^i; x^{-i})$
	subject to	$c_i(x^i, y^i; x^{-i}) = 0,$
		$x^i, y^i \geq 0.$

The notation x_{-i} refers to a collection of decisions of all the other agents in the game, namely x^j where $j = 1, \ldots, N$ and $j \neq i$. This dissertation focuses on instances where the player's problem NLP(x_{-i}) is stochastic in nature. In particular, each player (hereafter referred to as an agent) would solve the problem

SNLP(x^{-i})	$\underset{x^i,y^i_\omega}{\text{minimize}}$	$\mathbb{E} f_\omega(x^i, y^i_\omega; x^{-i})$	
		$c_i(x^i, y^i_\omega; x^{-i}) = 0,$	$\forall \omega \in \Omega$
	subject to	$y^i_\omega \geq 0,$	$\forall \omega \in \Omega$
		$x^i \geq 0.$	

The issue of uncertainty in optimization problems has been studied extensively [BL97, RS03] and has recently gained immense relevance in the context of equilibrium problems.

We present algorithms for obtaining equilibrium points for several classes of such problems. We assume that the distribution of the random variables is discrete but with an arbitrarily large number of scenarios. This requires our algorithms to have an important property: they should be scalable in the sense that the computational effort to solve the problem should grow slowly (ideally linearly) with the number of scenarios in the distribution.

Our approach is similar in all the problems that we address: we attempt to define a stochastic optimization or complementarity problem[1] whose solution is an equilibrium point of the original problem. This equivalence, when available[2], provides two benefits: the ability to discuss the existence/uniqueness of a solution, and the means to construct globally convergent methods to obtain such points.

In practical settings, games may differ widely in structure and dynamics. In some settings, one may claim that agents behave in a Nash fashion: each agent maximizes his profit given that the decisions (often referred to as strategies) of all other agents are known.

Often, there may be a bias in the market and an agent (leader) may be aware of another agent's (follower's) optimization problem. This allows the leader to make decisions subject to the optimal reaction of the follower. Such a game is called a Stackelberg game.

Yet another variation is possible. This may be generalized to the case in which there are several Stackelberg leaders (such as in an airline or power market), each of whom is competing in a Nash game. An equilibrium among such players is called a Nash-Stackelberg equilibrium. We consider subclasses of each of these games in the next four chapters and describe them briefly in the remainder of this chapter.

1.1 Stochastic Nash Equilibria

The concept of Nash equilibrium is widely used in game theory. While its origins are in the field of economics, this notion of equilibrium is used extensively in other fields such as engineering, computer science, and biology. In a Nash setting, each agent maximizes his utility assuming that the strategies of other players are known. Our interest is in a class of *stochastic* Nash equilibria. Specifically, we assume that agents solve the problem $\mathrm{G}_i(x_{-i})$, where $x_{-i} \equiv (x^j)_{j \neq i}$.

The agent problems are two-stage stochastic optimization problems: agent i makes decision x^i in period 1 and recourse decisions y^i_ω in period 2 for each realization of uncertainty ω, where ω belongs to a sample-space Ω. We collectively refer to the second-stage decision of agent i as y^i. The optimization problem is then to minimize the expected cost subject to a set of first and second-level constraints, each parameterized by ω. Figure 1.1 shows the structure of such a game.

Definition 1.1 (G)**.** *The game G is an N-player game in which player i solves stochastic opti-*

[1] These shall be defined shortly.

[2] Such an equivalence may not be available in such settings, as seen in chapter 5.

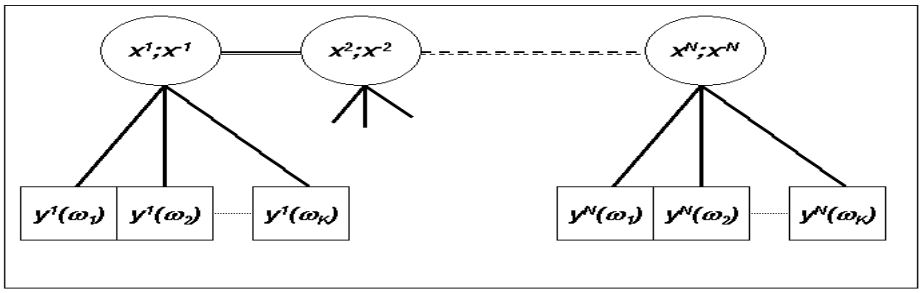

FIGURE 1.1 *Stochastic Nash equilibrium*

mization problem given by $G_i(x^{-i})$*:*

$$
\begin{array}{lll}
G_i(x^{-i}) & \underset{x^i, y^i_\omega}{\text{minimize}} & \mathbb{E} f^i_\omega(x^i, y^i_\omega; x^{-i}) \\
 & \text{subject to} & h(x^i) \leq 0 \\
 & & g(x^i, y^i_\omega) \leq 0, \quad \forall\, \omega \in \Omega.
\end{array}
$$

The stochastic Nash equilibrium is defined as follows.

Definition 1.2 (Stochastic Nash Equilibrium)**.** *The vector* $(x, y) = (x^i, y^i)_{i=1}^N$ *constitutes a Nash equilibrium for* G *if for each* $i = 1, \ldots, N$*,* (x^i, y^i) *solves the stochastic optimization problem* $G_i(x^{-i})$*.*

Under the assumption that G_i is a strictly convex quadratic program, we may reformulate the Nash equilibrium as a larger convex quadratic program, albeit a stochastic one. Such a problem has received less attention than the stochastic linear programming problem, which is the focus of the books by Infanger [Inf94] and Higle and Sen [HS96]. In chapter 2, we extend the work by Van Slyke and Wets [VSW69], Dantzig and Glynn [DG89], and Infanger [Inf94] in several ways. We prove the convergence of the L-shaped method for the quadratic case. Furthermore, this convergence theory is presented under the weaker assumption that the cuts are inexact [ZPR00]. To ease the computational burden, we suggest the usage of sampling of inexact cuts. The scalability of this inexact L-shaped method is demonstrated on a stochastic quadratic programming test problem set.

1.2 A Second-Order Method for MPCCs

The Stackelberg game was put forward by Von Stackelberg [Sta52]. Such a game is characterized by the existence of a leader and a follower. Moreover, the leader knows the optimization problem to be

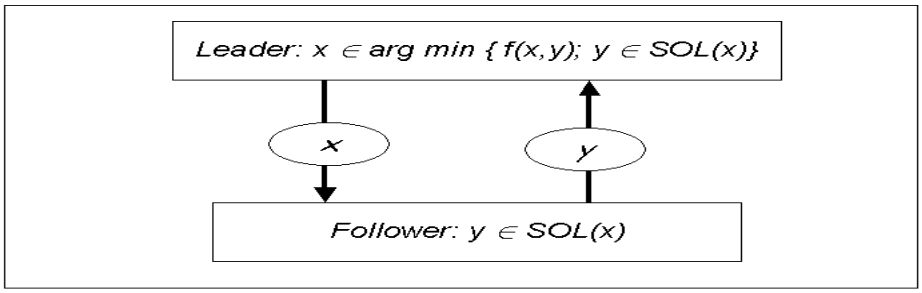

FIGURE 1.2 *Stackelberg equilibrium*

solved by a follower and thus makes decisions subject to an optimal response by the follower. The resulting equilibrium is called a Stackelberg equilibrium. The structure of such a game is shown in Figure 1.2.

Definition 1.3 (Stackelberg Equilibrium)**.** *If $SOL(x)$ represents the set of optimal solutions of the follower's problem, given a decision x, then a vector (x, y) constitutes a Stackelberg equilibrium if it solves the following problem:*

D	$\underset{x,y}{\text{minimize}}$	$f(x, y)$
	subject to	$g(x) \leq 0$
		$y \in SOL(x).$

Problem D is also called a mathematical program with equilibrium constraints [LPR96]. The constraint $y \in SOL(x)$ is called an equilibrium constraint, and under some assumptions we may replace it by a complementarity constraint $0 \leq y \perp F(x, y) \geq 0$. The notation $x \perp y$ means that $x_i y_i = 0, \forall i$. The resulting optimization problem is then called a mathematical program with complementarity constraints (MPCC).

The problem MPCC falls under a general nonlinear programming formulation, however, its constraints do not satisfy regularity conditions[3] that are commonly used to prove convergence of the algorithm. Significant effort has been poured into extending existing nonlinear programming algorithms to dealing with such ill-posedness [FLRS02, RB05, dMFNS05]. However, in all previous work, the obtained point can be only guaranteed to satisfy first-order optimality conditions. To ensure the satisfaction of second-order conditions of the solution, we need to enlist the use of second-order directions or directions of negative curvature. In chapter 3, we present an interior

[3] A common regularity condition that may be imposed is the linear independence constraint qualification [NW99]

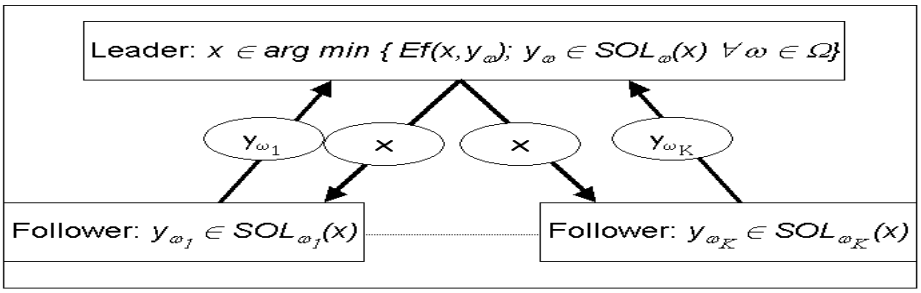

FIGURE 1.3 *Stochastic Stackelberg equilibria*

method that solves a sequence of regularized problems to obtain a second-order point and also discuss its associated global convergence theory. We then discuss the performance of the method on the `QPEC`gen [JR99] test problem set.

1.3 Stochastic Stackelberg Equilibrium Problems

A natural extension of the Stackelberg equilibrium problem is the consideration of uncertainty in the parameters of the equilibrium constraint. Figure 1.3 shows such a generalization. We refer to such an equilibrium as a *stochastic* Stackelberg equilibrium. It represents a generalization in which the follower's problem is parameterized by ω. Then, if the optimal solution of the follower is given by $y^\omega \in SOL(x;\omega)$, where x is the leader's decision and $SOL(.;\omega)$ represents the solution set of the follower under realization ω, we may define the equilibrium as follows.

Definition 1.4 (Stochastic Stackelberg Equilibrium)**.** *The vector (x, y) constitutes a Stackelberg equilibrium if it solves the following problem:*

$$
\begin{array}{lll}
\text{S} & \underset{x,y_\omega}{\text{minimize}} & \mathbb{E} f_\omega(x, y_\omega) \\
 & \text{subject to} & g(x) \leq 0 \\
 & & y_\omega \in SOL(x;\omega) \quad \forall\, \omega \in \Omega.
\end{array}
$$

Problem S is a stochastic mathematical program with equilibrium constraints (SMPEC). Again, by making suitable assumptions, one may reformulate the SMPEC as a stochastic mathematical program with complementarity constraints (SMPCC).

The solution of stochastic MPCCs has been studied by Shapiro and Xu [SX05] among others. Their approach employs sample-average approximation techniques [Sha03], which involve solving

a sampled instance of a stochastic MPCC. The theory provides convergence rates of the sampled solution to the true solution. It is assumed that such a sampled problem may be solved efficiently by existing methods. However, we contend that even reasonably sized problems may not be solved efficiently because there are no existing algorithms even for solving stochastic nonlinear programs.

An alternative formulation for the stochastic equilibrium problem has been proposed by Gurkan et al. [GOR99] in which the constraints are not posed as "almost-everywhere"[4] constraints but instead as expectation type constraints.

In interior methods, one obtains a solution to the original problem by using a Newton-method on a decreasing perturbation of the Karush-Kuhn-Tucker (KKT) system. But such a Newton step becomes expensive when the number of complementarity constraints is arbitrarily large. We introduce a decomposition of the KKT system such that the step may be obtained by solving a series of smaller scenario systems in parallel. Furthermore, exact Newton steps need not be taken early in the method, and we sample from the distribution to solve a reduced system. Such a technique is obviously useful for solving stochastic nonlinear programs, a class of problems for which few methods exist.

In Chapter 4, we present one of the first methods for stochastic nonlinear programming. We extend `QPEC`gen to `SQPEC`gen, a stochastic QPEC test problem set. This problem set generator is used to demonstrate the performance of our method.

1.4 Forward Contracting under Uncertainty in Electricity Markets

One particular generalization of Nash equilibrium problems is when some or all agents are Stackelberg players. An instance of such a problem is an imperfectly competitive market, as illustrated in Figure 1.4, where there are two Stackelberg leaders competing in a Nash fashion. The Nash-Stackelberg equilibrium may be defined as follows.

Definition 1.5 (Nash-Stackelberg Equilibrium)**.** *The vector* $(x, y) = (x^i, y^i)_{i=1}^N$ *constitutes a Nash-Stackelberg equilibrium if for each* $i = 1, \ldots, N$, (x^i, y^i) *solve the following Stackelberg problem:*

$\mathrm{D}_i(x_{-i})$	$\underset{x^i, y^i}{\text{minimize}}$	$f(x^i, y^i; x^i)$
	subject to	$g(x^i) \leq 0$
		$y^i \in SOL(x^i; x^{-i})$.

This definition may be extended to account for uncertainty.

Definition 1.6 (Stochastic Nash-Stackelberg Equilibrium)**.** *The vector* $(x, y) = (x^i, y^i)_{i=1}^N$ *constitutes a Nash-Stackelberg equilibrium, if for each* $i = 1, \ldots, N$, (x^i, y^i) *solves the stochastic Stackelberg problem* $D_i(x^{-i})$.

In Chapter 5, the setting of a spatial electricity market is considered. Agents are assumed to compete in a two-period market. In the first period, agents trade in forward contracts. These

[4] Such a formulation implies that there is one constraint corresponding to each realization of uncertainty.

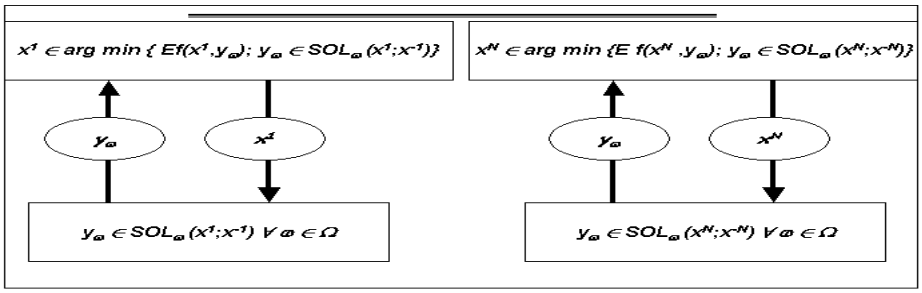

FIGURE 1.4 *Stochastic Nash-Stackelberg equilibrium*

contracts represent binding agreements to transact in power in the second period. Furthermore, they may make recourse-based spot-market decisions in the second period. The question, then, is to obtain an equilibrium point in forward and recourse-spot decisions under uncertainty.

These games are significantly harder to solve than the conventional Nash and Stackelberg problems. They represent agents competing in a Nash fashion in forward contracts while being Stackelberg leaders with respect to a spot market. The resulting equilibrium point is called a Nash-Stackelberg equilibrium point and requires the solution of an equilibrium problem with complementarity constraints (an EPCC).

In our setting, we solve a related simultaneous stochastic Nash problem. Equilibria of such games are shown to exist. Furthermore, under some conditions we can show that such equilibria are Nash-Stackelberg equilibria.

However, such an equilibrium is still not easy to obtain because it represents a solution of a stochastic mixed linear complementarity problem. We present a new sampling-based iterative-decomposition algorithm for such a class of problems and prove that it converges globally. Furthermore, we show that the algorithm scales well with the size of the discrete distribution. We also provide some policy-based insights for a 6-node electricity market model.

Chapter 2

A Sampling Method for Two-Stage Stochastic QPs

Planning under uncertainty was first discussed in the seminal paper by Dantzig [Dan55] in 1955. In the last decade, attention has shifted to stochastic integer, nonlinear, and equilibrium problems. An important question is whether growth in the size of the sample-space (under an assumption of a discrete distribution of uncertainty) results in a slow (say linear) growth in computational effort. We term such a property *scalable*. An important requirement in the development of stochastic programming algorithms is ensuring their scalability (see Birge and Louveaux [BL97] for an introduction and a review of stochastic programming).

In this chapter, we focus on two-stage stochastic quadratic programs with recourse. In recourse-based models, the computational burden lies in evaluating the expectation of the random recourse function, which represents the cost of recourse as a function of the first-stage decision x and is denoted by $\mathcal{Q}(x)$. Algorithms differ in spirit in the way they deal with the recourse function. For instance, deterministic decomposition methods build an outer-approximation of the recourse function. This approach finds its roots in the paper by Van Slyke and Wets [VSW69]. Significant effort has been poured into variants of this method (see the review by Ruszczynski [Rus03]).

The applicability of deterministic methods rests on the number of realizations in the second-stage being manageable in the presence of a discrete distribution. An alternate approach is to use statistical approximations. Such approaches construct a sample-mean of the recourse function by pre-sampling from the distribution [RS93, Sha03, GOR99].

There have been a few attempts at combining the ideas of sampling and decomposition methods. Such methodologies are often called interior sampling methods and were first discussed by Dantzig and Glynn [DG89]. Interior sampling methods tend to be restricted to cases in which the recourse function is convex. The underlying idea emerges from the L-shaped method of Van Slyke and Wets [VSW69] and involves building an outer-approximation of the recourse function in the first-stage problem (master). This is effected by the addition of cuts. The coefficients of the cuts represent an expectation of the dual solutions of each second-stage scenario problem. Dantzig and Glynn [DG89]

and Infanger [Inf94] use sample means of these cuts in conjunction with importance sampling. In addition, Higle and Sen [HS96] discuss a method that uses a single cut at each iteration to update the outer-approximation of the recourse function.

Our interest is in sampling-based L-shaped methods for the stochastic quadratic program (SQP), which may be be formulated as

SQP	$\underset{x}{\text{minimize}}$	$\frac{1}{2}x^T Q x + c^T x + \mathcal{Q}(x)$
	subject to	$x \geq 0,$

where the recourse function $\mathcal{Q}(x)$ is defined as $\mathcal{Q}(x) = \mathbb{E}_\omega \mathcal{Q}(x, \omega)$ and $\mathcal{Q}(x, \omega)$ is given by the value function[1] of

S_ω	$\underset{y}{\text{minimize}}$	$\frac{1}{2}y^T D_\omega y + d_\omega^T y$
	subject to	$By \geq b_\omega - Ax$
		$y \geq 0.$

We assume that Q and D_ω are symmetric positive-definite matrices.

We present an algorithm for solving stochastic convex quadratic programs with recourse. One motivation for studying such problems emerges from Nash equilibrium problems in which players solve stochastic quadratic programs. Earlier work on interior sampling methods has been restricted to the realm of linear programming. Our interior sampling method differs in several ways from previous approaches. First, we extend these methods to quadratic programs by including quadratic terms in the objective. Second, we enlist the help of *inexact* cuts in generating an approximation of the recourse function and prove the convergence of the L-shaped method for such inexact cuts.

The remaining part of this chapter is organized as follows. Section 2.1 motivates the study of stochastic quadratic programs through the stochastic Nash game.[2] Section 2.2 presents an L-shaped method [VSW69] with inexact cuts for solving two-stage stochastic QPs. We review the notion of inexact cuts [ZPR00] and prove the convergence of this method for the case of quadratic programming. Convergence of the algorithms and the associated bounds are also discussed. Section 2.3 introduces an inexact-cut sampling algorithm. Section 2.4 discusses computational results on a two-stage stochastic QP test problem set. In section 2.5, we conclude the study with a summary of the findings and some suggestions for future work.

2.1 Stochastic Nash Games

This section discusses the class of quadratic Nash games and shows that the resulting equilibrium problem may be solved as a larger stochastic quadratic program. In particular, our interest is in the determination of an equilibrium when agents solve strictly convex quadratic programs. We begin by considering the simple case in which the quadratic programs are deterministic, and subsequently

[1] In this context, the term *value function* refers to the objective function at the optimal solution.
[2] Throughout, we assume that agents solve quadratic programs unless mentioned otherwise.

introduce uncertainty into the structure.

2.1.1 Deterministic Games

We begin by defining the deterministic Nash game G_{ND} and the resulting Nash equilibrium. Note that $x_{-i} := \{x_j, j \neq i\}$. Such a definition requires the following assumption regarding the problem data.

Definition 2.1 (G_{ND})**.** *Consider a Nash game with N players in which each player is optimizing over an n-dimensional space with m linear inequality constraints (excluding nonnegativity bounds). Suppose the problem data is given by the matrices $Q \in \mathbb{R}^{p_1 \times p_1}$, $A \in \mathbb{R}^{p_2 \times p_1}$, $c \in \mathbb{R}^{p_1}$ and $b \in \mathbb{R}^{p_2}$, where $p_1 = N \times n$ and $p_2 = m \times N$. These matrices may be represented as*

$$Q = \begin{pmatrix} Q_{11} & Q_{12} & \cdots & Q_{1N} \\ Q_{21} & Q_{22} & \cdots & Q_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ Q_{N1} & \cdots & Q_{N,N-1} & Q_{NN} \end{pmatrix}, A = \begin{pmatrix} A_1 & & & \\ & \ddots & & \\ & & & A_N \end{pmatrix}, c = \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix},$$

where $Q_{ij} \in \mathbb{R}^{n \times n}$, $A_i \in \mathbb{R}^{m \times n}$, $c_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}^m$.

We make the following assumption regarding Q and its submatrices throughout the remainder of this chapter.

Assumption 2.2. *Q (and hence Q_{ii}) are assumed to be symmetric and positive-definite for $i = 1, \ldots, N$. We write this as $Q \succ 0$ and $Q_{ii} \succ 0$.*[3]

Definition 2.3. *Consider an N-player deterministic Nash game denoted by G_{ND}. Given $x = (x_1, \ldots, x_N)$, the set $SOL(A_i(x_{-i}))$ is the set of solutions of the problem $A_i(x_{-i})$:*

$A_i(x^{-i})$	$\underset{x_i}{\text{minimize}}$	$\frac{1}{2}x_i^T Q_{ii} x_i + c_i^T x_i + \sum_{j \neq i} x_i^T Q_{ij} x_j$
	subject to	$A_i x_i \geq b_i \quad : \lambda_i$
		$x_i \geq 0, \quad : \mu_i.$

We say that $(x_1^*, \ldots, x_N^*)$ is a Nash equilibrium for a deterministic Nash game if $x_i^* \in SOL(A_i(x_{-i}^*))$ for $1 \leq i \leq N$.[4]

Since $Q_{ii} \succ 0$ for all i, the agent problems are convex, and the first-order KKT conditions are sufficient. For general nonlinear programs, the necessary conditions require the assumption of an appropriate regularity condition or constraint qualification (see [NW99]). However, in the instance of linear and quadratic programming, the active constraints at a solution are always linear and the cone of feasible directions is an adequate representation of the feasible set. We refer the reader to Lemma 12.8 from [NW99] for more details and do not concern ourselves any further with the question of regularity conditions in this chapter.

[3] The notation $A \succ (\succeq) 0$ means that A is a positive definite (semidefinite) matrix.
[4] Note that λ_i and μ_i are multipliers associated with the relevant constraints.

The Karush-Kuhn-Tucker (or KKT) conditions for each player's problem $A_i(x^{-i})$ at x_i are given by

$$\begin{aligned} Q_{ii}x_i + c_i + \sum_{j\neq i} Q_{ij}x_j - A_i^T\lambda_i - \mu_i &= 0 \\ A_ix_i - b_i &\geq 0 \\ x_i &\geq 0 \\ [\lambda_i]_k[A_ix_i - b_i]_k &= 0, \quad k = 1,\ldots,m \\ [\mu_i]_j[x_i]_j &= 0, \quad j = 1,\ldots,n \\ \lambda_i &\geq 0 \\ \mu_i &\geq 0. \end{aligned}$$

These may be written compactly as

$$\begin{aligned} Q_{ii}x_i + c_i + \sum_{j\neq i} Q_{ij}x_j - A_i^T\lambda_i - \mu_i &= 0 \\ 0 \leq \lambda_i \perp A_ix_i - b_i &\geq 0 \\ 0 \leq \mu_i \perp x_i &\geq 0, \end{aligned} \tag{2.1}$$

where $u \perp v \implies [u]_j[v]_j = 0, \forall j$. Defining $A_ix_i - b_i$ as s_i, we have

$$0 \leq \begin{pmatrix} x_i \\ \lambda_i \end{pmatrix} \perp \begin{pmatrix} \mu_i \\ s_i \end{pmatrix} = \begin{pmatrix} Q_{ii} & -A_i^T \\ A_i & \end{pmatrix} \begin{pmatrix} x_i \\ \lambda_i \end{pmatrix} + \begin{pmatrix} c_i + \sum_{j\neq i} Q_{ij}x_j \\ -b_i \end{pmatrix} \geq 0. \tag{2.2}$$

By denoting x, s, λ and μ as

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}, \quad s = \begin{pmatrix} s_1 \\ \vdots \\ s_N \end{pmatrix}, \quad \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{pmatrix}, \text{ and } \mu = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix},$$

we may write the set of KKT conditions for all the agents as the following linear complementarity problem (LCP) [CPS92]:

LCP_D	$0 \leq \begin{pmatrix} x \\ \lambda \end{pmatrix} \perp \begin{pmatrix} \mu \\ s \end{pmatrix} = \begin{pmatrix} Q & -A^T \\ A & \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} + \begin{pmatrix} c \\ -b \end{pmatrix} \geq 0.$

This leads to the following equivalence result.

Proposition 2.4. *The vector x_* is a Nash equilibrium for G_{ND} if and only if $\left(x^*, \lambda^*\right)$ solves LCP_D.*

Proof. ($\Rightarrow$) The first-order conditions of each agent problem are sufficient and may be aggregated into LCP_D.

($\Leftarrow$) : If (x^*, λ^*) solves LCP_D, it follows that (x_i^*, λ_i^*) solves (2.2) for each i. But since $Q_{ii} \succ 0$, this is a sufficient condition for x_i^* being a minimizer of $A_i(x_{-i}^*)$.

■

By defining

$$M_D = \begin{pmatrix} Q & -A^T \\ A & \end{pmatrix} \text{ and } q_D = \begin{pmatrix} c \\ -b \end{pmatrix},$$

we may state the following result concerning the feasibility and solvability of an LCP.

Definition 2.5. *For given $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$, LCP(q,M) requires an $x \in \mathbb{R}^n$ that satisfies $0 \le x \perp Mx + q \ge 0$. Furthermore,*

1. *LCP(q,M) is said to be* **feasible** *if $Mx + q \ge 0$ for some $x \ge 0$ and*
2. *LCP(q,M) is said to be* **solvable** *if $x \perp Mx + q$ and $Mx + q \ge 0$ for some $x \ge 0$.*

Lemma 2.6. *If Q is symmetric positive definite, then the matrix M_D is positive semidefinite.*

Proof. If $x \neq 0$, we have

$$\begin{pmatrix} x \\ \lambda \end{pmatrix}^T \begin{pmatrix} Q & -A^T \\ A & \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = x^T Q x > 0.$$

Note that x could be zero with $\lambda \neq 0$ implying that $x^T Q x = 0$. ■

The existence of a solution to LCP_D is the subject of the next result.

Lemma 2.7. *Let M_D be positive semidefinite. Then if LCP_D is feasible, it is solvable.*

Proof. See [CPS92]. ■

An interesting observation is that if each agent solves an unconstrained QP, then the resulting matrix M_D is positive definite. In such a case, the resulting equilibrium point is unique.

Theorem 2.8. *If Q is positive definite, then the Nash game G_{ND} has an equilibrium point if a feasible solution to LCP_D exists. Moreover, if each agent's problem is an unconstrained quadratic program, then G_{ND} always has a unique equilibrium.*

We conclude this section with an equivalence between LCP_D and the following quadratic program.

Lemma 2.9. *The complementarity problem LCP_D represents the first-order conditions of the following convex quadratic program:*

$$\begin{array}{lll} \underset{x}{\text{minimize}} & \frac{1}{2}x^T Q x + c^T x & \\ \text{subject to} & Ax \ge b & : \lambda \\ & x \ge 0 & : \mu. \end{array}$$

Proof. Immediate. ■

This lemma, while obvious, has significant bearing on the rest of the discussion. In particular, we show later that when the agents solve two-stage stochastic quadratic programs , we may rewrite the optimality conditions as a large-scale stochastic quadratic program. Since the uncertainty in this problem lies entirely in the second stage, we may use an extension of the L-shaped method [VSW69] to solve such a problem.

2.1.2 Introducing Uncertainty

This section extends the agent problem from a single-stage QP to a two-stage stochastic QP with the uncertainty being resolved in the second stage. Agent i makes a first-stage decision x_i and then takes a second-stage (recourse) decision y_i^ω in the event of realization ω. We assume that ω has a discrete distribution with K realizations.[5] and that the probability of realization ω occurring is p^ω.

Definition 2.10. *Consider a stochastic Nash game in N agents in which agent i makes a first-stage decision given by $x_i \in \mathbb{R}^n$ and a second-stage decision given by $y_i^\omega \in \mathbb{R}^p$ for all $\omega \in \Omega$. Then, the resulting data for the first-stage problem is given by the matrices $Q \in \mathbb{R}^{p_1 \times p_1}$ and $c \in \mathbb{R}^{p_1}$, where $p_1 = N \times n$. These matrices may be represented as*

$$Q = \begin{pmatrix} Q_{11} & Q_{12} & \cdots & Q_{1N} \\ Q_{21} & Q_{22} & \cdots & Q_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ Q_{N1} & \cdots & Q_{N,N-1} & Q_{NN} \end{pmatrix} \text{ and } c = \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix}.$$

The second-stage data for each realization $\omega \in \Omega$ is given by the matrices $D^\omega \in \mathbb{R}^{p_1 \times p_1}$, $A^\omega \in \mathbb{R}^{p_2 \times p_1}$, $B \in \mathbb{R}^{p_2 \times p_1}$, $d^\omega \in \mathbb{R}^{p_3}$ and $b^\omega \in \mathbb{R}^{p_1}$, where $p_3 = p \times N$. The matrices may be represented as

$$D^\omega = \begin{pmatrix} D_{11}^\omega & & & \\ & D_{22}^\omega & & \\ & & \ddots & \\ & & & D_{NN}^\omega \end{pmatrix}, A^\omega = \begin{pmatrix} A_1^\omega & & \\ & \ddots & \\ & & A_N^\omega \end{pmatrix}, B = \begin{pmatrix} B_1 & & \\ & \ddots & \\ & & B_N \end{pmatrix},$$

$$d^\omega = \begin{pmatrix} d_1^\omega \\ \vdots \\ d_N^\omega \end{pmatrix} \text{ and } b = \begin{pmatrix} b_1^\omega \\ \vdots \\ b_N^\omega \end{pmatrix},$$

where $D_{ii}^\omega \in \mathbb{R}^{p \times p}$, $A_i^\omega \in \mathbb{R}^{m \times n}$, $B_i \in \mathbb{R}^{m \times p}$, $c_i \in \mathbb{R}^n$, $d_i^\omega \in \mathbb{R}^p$ and $b_i^\omega \in \mathbb{R}^m$.

In addition to Assumption 2.2, we make the following assumption regarding the matrices D_{ii}^ω.

[5] In keeping with conventional notation in numerical optimization, we use lower case to specify variables and upper case to specify matrices. Randomness in either is specified by using a superscript ω in keeping with the notation from [Inf94]

Assumption 2.11. *D_{ii}^{ω} are assumed to be symmetric and positive-definite for $i = 1, \ldots, N$ and for all $\omega \in \Omega$.*

Definition 2.12 (G_{NS})**.** *Consider an N-player stochastic Nash game denoted by G_{NS}. Given $x = (x_1, \ldots, x_N)$, the set $SOL(S_i(x_{-i}))$ is the set of optimal x_i of the problem $S_i(x_{-i})$:*

$$
\begin{array}{lll}
S_i(x_{-i}) & \underset{x_i, y_i}{\text{minimize}} & f_i(x_i, x^{-i}) + \mathbb{E} h_i(y_i^{\omega}; x_i, \omega) \\
 & & A_i^{\omega} x_i + B_i y_i^{\omega} \geq b_i^{\omega} \quad : p^{\omega} \lambda_i^{\omega} \\
 & \text{subject to} & x_i \geq 0 \quad : \mu_i \\
 & & y_i^{\omega} \geq 0. \quad : p^{\omega} \gamma_{\omega}^{i}, \quad \forall \quad \omega \in \Omega
\end{array}
$$

where

$$
\begin{aligned}
f_i(x_i; x^{-i}) &:= \frac{1}{2} x_i^T Q_{ii} x_i + c_i^T x_i + \sum_{j \neq i} x_i^T Q_{ij} x_j, \\
h_i(y_i^{\omega}; x_i, \omega) &:= \frac{1}{2} (y_i^{\omega})^T D_{ii}^{\omega} y_i^{\omega} + (d_i^{\omega})^T y_i^{\omega}.
\end{aligned}
$$

We say that $(x_1^*, \ldots, x_N^*)$ is a Nash equilibrium for a deterministic Nash game if $x_i^* \in SOL(S_i(x_{-i}^*))$ for $1 \leq i \leq N$.

The first-order optimality conditions for $i = 1, \ldots, N$ are

$$
\begin{aligned}
0 \leq x_i \perp \mu_i &= Q_{ii} x_i + c_i + \sum_{j \neq i} Q_{ij} x_j - \sum_{\omega} p^{\omega} (A_i^{\omega})^T \lambda_i^{\omega} \geq 0 \\
0 \leq y_i^{\omega} \perp \gamma_i^{\omega} &= D_{ii}^{\omega} y_i^{\omega} + d_i^{\omega} - B_i^T \lambda_i^{\omega} \geq 0, \forall \omega \in \Omega \\
0 \leq \lambda_i^{\omega} \perp s_i^{\omega} &= A_i^{\omega} x_i + B_i y_i^{\omega} - b_i^{\omega} \geq 0, \forall \omega \in \Omega.
\end{aligned}
$$

Since the sample-space has K realizations, agent i solves a quadratic program with a first-stage decision x_i and K second-stage decisions $y_i^j, j = 1, \ldots, K$. Using the notation z and z_i to denote

$$
z := \begin{pmatrix} z_1 \\ \vdots \\ z_N \end{pmatrix} \text{ and } z_i = \begin{pmatrix} z_i^1 \\ \vdots \\ z_i^K \end{pmatrix},
$$

we may define $s, c, \mu, \lambda, \gamma, b$ and y accordingly. Moreover, we define $\bar{x}_i$ as

$$
\bar{x}_i = \begin{pmatrix} x_i \\ \vdots \\ x_i \end{pmatrix}
$$

and the following matrices:

$$A_i = \begin{pmatrix} A_i^1 & & \\ & \ddots & \\ & & A_i^K \end{pmatrix}, A_i^P = \begin{pmatrix} p^1 A_i^1 \\ \vdots \\ p^K A_i^K \end{pmatrix}, A = \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_N \end{pmatrix}, A^P = \begin{pmatrix} A_1^P & & \\ & \ddots & \\ & & A_N^P \end{pmatrix},$$
$$D_i = \begin{pmatrix} D_{ii}^1 & & \\ & \ddots & \\ & & D_{ii}^K \end{pmatrix}, Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1N} \\ \vdots & \ddots & \vdots \\ Q_{N1} & \cdots & Q_{NN} \end{pmatrix}, \bar{B}_i = \begin{pmatrix} B_i & & \\ & \ddots & \\ & & B_i \end{pmatrix}, \quad d_i = \begin{pmatrix} d_i^1 \\ \vdots \\ d_i^K \end{pmatrix}.$$

Note that $D_i, \bar{B}_i$ and A are block-diagonal matrices. Then we may write the first-order conditions of optimality in terms of the agent problem i:

$$\begin{aligned} 0 \le x \quad &\perp \quad \mu = Qx + c - (A^P)^T \lambda \ge 0 \\ 0 \le y_i \quad &\perp \quad \gamma_i = D_i y_i + d_i - \bar{B}_i^T \lambda_i \ge 0 \\ 0 \le \lambda_i \quad &\perp \quad s_i = A_i \bar{x}_i + \bar{B}_i y_i - b_i \ge 0. \end{aligned} \tag{2.3}$$

Alternatively, using the notation z and z^j to denote

$$z := \begin{pmatrix} z^1 \\ \vdots \\ z^K \end{pmatrix} \text{ and } z^j = \begin{pmatrix} z_1^j \\ \vdots \\ z_N^j \end{pmatrix},$$

we may define $s, c, \mu, \lambda, \gamma, b$ and y accordingly to aggregate the variables and parameters by realization, as opposed to by agent. We define $\bar{x}$ as

$$\bar{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

and the following matrices:

$$A_P^j = \begin{pmatrix} p^1 A_1^j & & \\ & \ddots & \\ & & p_1 A_N^j \end{pmatrix}, A_P = \begin{pmatrix} A_P^1 \\ \vdots \\ A_P^K \end{pmatrix}, A^j = \begin{pmatrix} A_1^j & & \\ & \ddots & \\ & & A_N^j \end{pmatrix},$$
$$D_P^j = \begin{pmatrix} p^j D_{11}^j & & \\ & \ddots & \\ & & p^j D_{NN}^j \end{pmatrix}, Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1N} \\ \vdots & \ddots & \vdots \\ Q_{N1} & \cdots & Q_{NN} \end{pmatrix}, \quad \bar{B}_P^j = \begin{pmatrix} p^j B_1 & & \\ & \ddots & \\ & & p^j B_N \end{pmatrix},$$
$$d_P^j = \begin{pmatrix} p^j d_1^j \\ \vdots \\ p^j d_N^j \end{pmatrix}.$$

We may then rewrite the KKT conditions (2.3) as

$$\begin{aligned} 0 \le x \quad &\perp \quad \mu = Qx + c - A_P^T \lambda \ge 0 \\ 0 \le y^j \quad &\perp \quad \gamma^j = D_P^j y^j + d_P^j - (\bar{B}_P^j)^T \lambda^j \ge 0 \\ 0 \le \lambda^j \quad &\perp \quad s^j = A_P^j \bar{x} + \bar{B}_P^j y^j - b_P^j \ge 0. \end{aligned} \tag{2.4}$$

This system can be restated as

$$\text{LCP}_S \qquad 0 \le \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix} \perp \begin{pmatrix} \mu \\ \gamma \\ s \end{pmatrix} = \begin{pmatrix} Q & & -A_P^T \\ & D_P & -\bar{B}_P^T \\ A_P & \bar{B}_P & \end{pmatrix} \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix} + \begin{pmatrix} c \\ d_P \\ -b_P \end{pmatrix} \ge 0.$$

Proposition 2.13. *Consider the Nash game G_{NS} given by Definition 2.12. The vector x^* is a Nash equilibrium of G_{NS} if and only if (x^*, y^*, λ^*) solves LCP$_S$.*

Proof. Follows from Proposition 2.4. ■

Theorem 2.8 also holds in the stochastic case:

Theorem 2.14. *Consider the Nash game G_{NS} given by Definition 2.12. The Nash game G_{NS} has an equilibrium point if a feasible solution to LCP$_S$ exists. Moreover, if every agent solves an unconstrained quadratic program, then G_{NS} always has a unique equilibrium.*

Solving LCP$_S$ becomes more difficult as the number of realizations K grows. Unfortunately it is not easy to introduce separability into the problem structure. We approach the problem differently: By drawing a correspondence between LCP$_S$ and an equivalent stochastic quadratic program, we may then discuss how to solve the resulting stochastic QP efficiently. The following result makes this correspondence clear.

Lemma 2.15. *The linear complementarity problem LCP$_S$ represents the first-order conditions for the following equivalent quadratic problem EQP:*

$$\begin{array}{lll} \text{EQP} & \underset{x,y}{\text{minimize}} & \tfrac{1}{2}x^T Q x + c^T x + \tfrac{1}{2} y^T D_P y + d_P^T y \\ & & A_P x + \bar{B}_P y \ge b_P \qquad : \lambda_P \\ & \text{subject to} & x \ge 0 \qquad : \mu \\ & & y \ge 0, \qquad : \gamma. \end{array}$$

Proof. Immediately follows. ■

Corollary 2.16. *The KKT conditions of the following two-stage stochastic quadratic program SQP*

are given by LCP_S:

SQP	$\underset{x,y^\omega}{\text{minimize}}$	$\frac{1}{2}x^TQx + c^Tx + \sum_\omega p_\omega\{\frac{1}{2}(y^\omega)^TD^\omega y_\omega + (d^\omega)^Ty^\omega\}$	
		$A_\omega x + \bar{B}y^\omega \geq b^\omega$	$: \lambda_\omega$
	subject to	$x \geq 0$	$: \bar{\mu}$
		$y^\omega \geq 0$	$: \gamma^\omega, \quad \forall \omega \in \Omega.$

We may rewrite SQP using the following recourse-based formulation:

SQP_E	$\underset{x}{\text{minimize}}$	$\frac{1}{2}x^TQx + \mathbb{E}\mathcal{Q}(x,\omega)$
	subject to	$x \geq 0,$

where $\mathbb{E}$ denotes the expectation operator and $\mathcal{Q}(x,\omega)$ is a random recourse function:

$$\mathcal{Q}(x,\omega) := \min_{y^\omega}\{\frac{1}{2}(y^\omega)^TD^\omega y^\omega + (d^\omega)^Ty^\omega : A_\omega x + \bar{B}y^\omega - b^\omega \geq 0, \quad y^\omega \geq 0\}.$$

This leads to the following result.

Theorem 2.17. *Consider the Nash game G_{NS} in which agent i solves the stochastic quadratic program $S_i(x^{-i})$. Then the Nash equilibrium point $(x_1^*, \ldots, x_N^*)$ is given by the solution of a corresponding stochastic quadratic program SQP.*

Proof. Follows by earlier discussion. ■

2.2 An Inexact L-shaped Method for Stochastic QPs

Based on an approach suggested by Benders [Ben62], Van Slyke and Wets [VSW69] proposed a decomposition method for solving two-stage linear programs. The idea hinges on the specification of a master problem that minimizes the sum of first-stage cost and the cost of recourse $\mathcal{Q}(x)$ subject to first-stage constraints on x. The convexity of $\mathcal{Q}(x)$ allows us to work with an increasingly accurate outer-approximation of the recourse function. Cuts are obtained by solving the subproblems for all realizations of ω. Infanger [Inf92] suggested the use of sampling to avoid having to solve all the subproblems. Dantzig and Glynn [DG89] discussed importance sampling to obtain better second-stage cost estimates. This represents our point of departure.

In this chapter, we discuss the extension of the L-shaped method [VSW69] to stochastic quadratic programs. In addition, we use inexact cuts. The convergence of such an algorithm is proved in this chapter. Inexact cuts ease the computational burden. They were shown to result in a convergent algorithm for stochastic linear programming [ZPR00].[6] To allow for an easier comparison with stochastic linear programming, we restrict our discussion to stochastic quadratic programs. However, in the last subsection, we extend the results to stochastic convex programming.

[6] We postpone the discussion of sampling till the next section.

We begin by defining the stochastic quadratic program in section 2.2.1 and presenting some of the classical results on quadratic programming duality. Convexity of the recourse function is readily proved. Section 2.2.2 specifies the cuts that are necessary in the specification of the recourse function and feasibility of the second-stage problems. We then formulate the two problems of relevance in this L-shaped method: the master problem and the subproblem (section 2.2.4). Section 2.2.5 outlines the bounds and termination criteria for the L-shaped method. In section 2.2.6, we introduce inexact cuts and restate the bounds for such cuts. Finally, in section 2.2.7, we extend the convergence theory from [ZPR00] to account for two-stage stochastic quadratic programs.

Since our discussion from here onwards pertains only to stochastic quadratic programs, randomness shall be specified using the subscript ω (as opposed to the superscript ω as in the earlier section).

2.2.1 The Two-Stage Stochastic Quadratic Program

In this section, we consider the two-stage stochastic convex quadratic program SQP:

SQP	$\underset{x,y}{\text{minimize}}$	$\frac{1}{2}x^TQx + c^Tx + \sum_{j=1}^{K} p_j\{\frac{1}{2}y_j^TD_jy_j + d_j^Ty_j\}$	
	subject to	$A_jx + By_j \geq b_j$	$: \lambda_j,$ for $j = 1, \ldots, K$
		$x \geq 0$	$: \mu.$

We may use quadratic programming duality to construct a dual problem as the following result shows.

Lemma 2.18. *Let D be symmetric and positive semidefinite. Then the primal problem*

PQP	$\underset{y}{\text{minimize}}$	$\frac{1}{2}y^TDy + d^Ty$	
	subject to	$By \geq b - Ax$	$: z$
		$y \geq 0.$	$: \pi$

has the dual problem

DQP	$\underset{\pi,z}{\text{maximize}}$	$(b - Ax)^Tz - \frac{1}{2}\pi^TD\pi$	
	subject to	$-D\pi + B^Tz \leq d$	
		$z \geq 0,$	

and strong duality ($(b-Ax)^Tz^* - \frac{1}{2}(\pi^*)^TD\pi^* = \frac{1}{2}(y^*)^TDy^* + d^Ty^*$) holds at the optimal solution.[7]

Proof. Follows from Dorn duality [Dor61]. ■

As in linear programming, weak and strong duality results may be stated for this pair of problems [CPS92].

[7] Note that y^* is the optimal solution of PQP and (z^*, π^*) is the optimal solution of DQP.

Lemma 2.19. *The following extensions of weak and strong duality from linear programming may be made.*

1. *Weak Duality: If y is a feasible solution of (PQP) and (π, z) is a feasible solution of (DQP), then the following inequality holds:*

$$(b - Ax)^T z - \frac{1}{2}\pi^T Q\pi \leq \frac{1}{2}y^T Dy + d^T y.$$

2. *Strong Duality: If weak duality holds with equality then the solutions are optimal.*

Proof. See [Dor61]. ■

The second-stage primal problem 2P may be stated as

2P	$\underset{y}{\text{minimize}}$	$\sum_{i=1}^K p_i[\frac{1}{2}y_i^T D_i y_i + d_i^T y_i]$		
	subject to	$By_i \geq b - A_i x,$	$: p_i z_i$	for $i = 1, \ldots, K$
		$y_i \geq 0.$	$: p_i \pi_i$	for $i = 1, \ldots, K.$

The dual of the second-stage problem 2D is given by

2D	$\underset{\pi,z}{\text{maximize}}$	$\sum_{i=1}^K p_i((b_i - A_i x)^T z_i - \frac{1}{2}\pi_i^T D_i \pi_i)$	
	subject to	$-D_i\pi_i + B^T z_i \leq d_i,$	for $i = 1, \ldots, K$
		$z_i \geq 0.$	for $i = 1, \ldots, K,$

in which the objective function is parameterized by the first-stage decision x.

2.2.2 Some Properties of $\mathcal{Q}(x)$

In section 2.1, we introduced the recourse function $\mathcal{Q}(x)$. This function is defined as the optimal value of problem 2D, given a first-stage decision x. Problem 2D is decomposable into problems $2D_\omega$ defined by

$2D_\omega$	$\underset{\pi_\omega, z_\omega}{\text{maximize}}$	$p_\omega((b_\omega - A_\omega)^T z_\omega - \frac{1}{2}\pi_\omega^T D_\omega \pi_\omega)$	
	subject to	$-D_\omega\pi_\omega + B^T z_\omega \leq d_\omega,$	
		$z_\omega \geq 0.$	

The random recourse function $\mathcal{Q}(x, \omega)$ is the optimal value function of $2D_\omega$. We begin by proving some properties of the recourse function $\mathcal{Q}(x)$ and $\mathcal{Q}(x, \omega)$.

Lemma 2.20. *The random recourse function $\mathcal{Q}(x, \omega)$ is convex for all $\omega \in \Omega$.*

Proof. It suffices to show that the dual form of $\mathcal{Q}(x,\omega)$ is convex.

$$\begin{aligned}
\mathcal{Q}(\lambda x^1+(1-\lambda)x^2,\omega) &= \max_{z,\pi} \quad (b-A(\lambda x^1+(1-\lambda)x^2)^T z - \tfrac{1}{2}\pi^T D\pi \\
&= \max_{z,\pi} \lambda\{(b-Ax^1)^T z - \tfrac{1}{2}\pi^T D\pi\} + (1-\lambda)\{(b-Ax^2)^T z - \tfrac{1}{2}\pi^T D\pi\} \\
&\le \max_{z,\pi} \lambda\{(b-Ax^1)^T z - \tfrac{1}{2}\pi^T D\pi\} \\
&\quad + \max_{z,\pi}(1-\lambda)\{(b-Ax^2)^T z - \tfrac{1}{2}\pi^T D\pi\} \\
&= \lambda\mathcal{Q}(x^1,\omega) + (1-\lambda)\mathcal{Q}(x^2,\omega).
\end{aligned}$$

■

Lemma 2.21. *The recourse function $\mathcal{Q}(x)$ is convex over its effective domain C, where $C = \{x \in X : \mathcal{Q}(x) < \infty\}$.*

Proof. By lemma 2.20, the convexity of $\mathcal{Q}(x,\omega)$ follows:

$$\begin{aligned}
\mathcal{Q}(\lambda x^1+(1-\lambda)x^2) &= \mathbb{E}\mathcal{Q}(\lambda x^1+(1-\lambda)x^2,\omega) \\
&\le \mathbb{E}\{\lambda\mathcal{Q}(x^1,\omega) + (1-\lambda)\mathcal{Q}(x^2,\omega)\} \\
&= \lambda\mathcal{Q}(x^1) + (1-\lambda)\mathcal{Q}(x^2).
\end{aligned}$$

■

We adapt the following result from [HS96] to the quadratic case.

Lemma 2.22. *Suppose $\mathcal{Q}(x,\omega) < \infty$ with probability one for all $x \in X$, and $\mathbb{E}\{\|A_\omega\|\} < \infty$. Then*

1. *For almost every $\omega \in \Omega$, there exists an $M(\omega) < \infty$ such that $\mathbb{E}M(\omega) < \infty$ and $|\mathcal{Q}(x^1,\omega) - \mathcal{Q}(x^2,\omega)| \le M(\omega)\|x^1 - x^2\|$.*
2. *There exists an $M < \infty$ such that $|\mathcal{Q}(x^1) - \mathcal{Q}(x^2)| \le M\|x^1 - x^2\|$ for all $x^1, x^2 \in X$.*

Proof. Let C represent the set of dual feasible (π, z) and let $vert(C)$ represent the set of vertices of the dual feasible region.

1. By hypothesis, the primal and dual second-stage problems are feasible for all $x \in X$ with probability one. Given x^1 and x^2 in X, let

$$\{\pi^j(\omega), z^\omega\} \in \arg\max\{(b_\omega - A_\omega x)^T z^j - \frac{1}{2}(\pi^j)^T D\pi^j \mid \{\pi^j, z^j\} \in vert(C)\}, \quad j = 1,2.$$

   By the positive semidefiniteness of D and convexity of $\mathcal{Q}(x,\omega)$, we have

$$\begin{aligned}
\mathcal{Q}(x^1,\omega) - (A_\omega(x^2-x^1))^T z^1 &\le \mathcal{Q}(x^2,\omega) \\
\mathcal{Q}(x^2,\omega) - (A_\omega(x^1-x^2))^T z^2 &\le \mathcal{Q}(x^1,\omega).
\end{aligned}$$

Combining both of these, we obtain

$$-(A_\omega(x^1 - x^2))^T z^2 \leq \mathcal{Q}(x^1, \omega) - \mathcal{Q}(x^2, \omega) \leq -(A_\omega(x^1 - x^2))^T z^1. \tag{2.5}$$

With $M(\omega) := \max\{\|z\| \mid z \in vert(C)\} \cdot \|A_\omega\|$, it may be seen that

$$\mathbb{E}M(\omega) = \max\{\|z\| \mid z \in vert(C)\} \cdot \mathbb{E}\|A_\omega\| < \infty.$$

Therefore, we have

$$\begin{aligned} |Q(x^1, \omega) - Q(x^2, \omega)| &\leq \max\{|(A_\omega(x^1 - x^2))^T z^2|, |(A_\omega(x^1 - x^2))^T z^1|\} \\ &\leq \max\{\|z^T A_\omega\| \mid z \in vert(C)\} \cdot \|x^1 - x^2\| \\ &\leq \max\{\|z\| \mid z \in vert(C)\} \cdot \|A_\omega\| \cdot \|x^1 - x^2\| \\ &\equiv M(\omega), \text{ for } \omega \quad a.e. \end{aligned}$$

2. If we denote $\mathcal{Q}(x) := \mathbb{E}\mathcal{Q}(x, \omega)$ then we may use expectations on (2.5) to obtain

$$-\mathbb{E}(A_\omega(x^1 - x^2))^T z^2 \leq \mathbb{E}\mathcal{Q}(x^1, \omega) - \mathbb{E}\mathcal{Q}(x^2, \omega) \leq -\mathbb{E}(A_\omega(x^1 - x^2))^T z^1 \tag{2.6}$$

$$-\mathbb{E}(A_\omega(x^1 - x^2))^T z^2 \leq \mathcal{Q}(x^1) - \mathcal{Q}(x^2) \leq -\mathbb{E}(A_\omega(x^1 - x^2))^T z^1. \tag{2.7}$$

Denoting $\mathbb{E}M(\omega)$ as M, we have

$$|\mathcal{Q}(x^1) - \mathcal{Q}(x^2)| \leq M\|x^1 - x^2\|, \forall\, x \in X.$$

■

Before proceeding, we should note that the convexity of the recourse function is an essential property for approximating the second-stage problem by a series of cuts. Obviously, when the second-stage problem is not convex (for instance when there are integer variables in the second-stage), the approximation of the recourse function by cutting planes is not directly possible.

2.2.3 Optimality and Feasibility Cuts

Problem 2D is separable into the following subproblems:

$S_\omega(x)$	$\underset{\pi_\omega, z_\omega}{\text{maximize}}$	$p_\omega((b_\omega - A_\omega x)^T z_\omega - \frac{1}{2}\pi_\omega^T D_\omega \pi_\omega)$
	subject to	$-D_\omega \pi_\omega + B^T z_\omega \leq d_\omega$
		$z_\omega \geq 0.$

Each $S_\omega(x)$ is a convex quadratic program and its feasible region is dependent on ω unless one assumes that the linear and quadratic cost of recourses are identical for all random instances of ω. This is, in fact, the assumption made by several authors when dealing with multistage linear

programming. We may define the random recourse function $\mathcal{Q}(x,\omega)$ as

$$\mathcal{Q}(x,\omega) := \min_{y_\omega \geq 0} \{\tfrac{1}{2} y_\omega^T D_\omega y_\omega + y_\omega^T d_\omega : B y_\omega \geq b_\omega - A_\omega x\}.$$

By the duality theory of quadratic programming, $\mathrm{S}_\omega(x)$ and its dual have equal optimal values unless both are infeasible. Therefore, the random recourse function may be restated in terms of the dual problem:

$$\mathcal{Q}(x,\omega) := \max_{\pi_\omega, z_\omega \geq 0} \{-\tfrac{1}{2} \pi_\omega^T D_\omega \pi_\omega + z_\omega^T (b_\omega - A_\omega x) : B^T z - D\pi \leq d_\omega\}.$$

The convexity of $\mathcal{Q}(x,\omega)$ and its expectation $\mathcal{Q}(x)$ are established by the results from the previous section. We assume that the original large-scale QP is feasible. This implies that for at least one x, problem SQP_D is feasible with a finite solution. The finiteness of the original problem implies that the second-stage problem 2P is also finite for some x that is feasible in the first stage. By the duality theory of convex quadratic programming, the dual of the second-stage problem must be feasible. Unboundedness of the second stage is precluded by the finiteness of the primal. The feasible region of each instance of the second-stage problem is given by the polyhedron

$$\{(\pi_\omega, z_\omega) : -D_\omega \pi_\omega + B^T z_\omega \leq d_\omega, z_\omega \geq 0\}.$$

If $u_\omega = (\pi_\omega, z_\omega)$, then the region may be defined in terms of p extreme points $u_\omega^1, \ldots, u_\omega^p$ and q extreme rays $u_\omega^{p+1}, \ldots, u_\omega^{p+q}$. By the positive semidefiniteness of D_ω, we know that

$$\pi_\omega^T D_\omega \pi_\omega \geq 0, \forall \pi^\omega \implies \max_{\pi^\omega} -\pi_\omega^T D_\omega \pi_\omega \leq 0.$$

Therefore, the finiteness of the dual of the second-stage problem is implied by the following constraints on the first-stage decision x:

$$(z_\omega^j)^T (b_\omega - A_\omega x) \leq 0, \quad j = p+1, \ldots, p+q, \forall \omega \in \Omega. \tag{2.8}$$

These constraints are called feasibility cuts and may be appended to the problem. We define J^{fea} as the set of indices corresponding to the set of dual extreme rays corresponding to scenario ω:

$$J^{\mathrm{fea}}(x) = \{(j,\omega) : u_\omega^j \text{ represents an extreme ray}\}.$$

The feasibility cuts are identical to the two-stage linear case. They are appended to the master problem (to be defined in the next section) to ensure feasibility of the dual for every first-stage decision x.

In a linear program, when the optimal cost is finite, the optimal solution is either at an extreme point of the polyhedron or on a face. However, in the case of the latter, the linearity of the objective function implies that we may obtain an extreme point solution in both of the cases. In a QP, the optimal solution is not necessarily at an extreme point, but any extreme point will provide a lower

bound on the optimal solution in a maximization problem. Therefore

$$\theta \geq \sum_{\omega \in \Omega} p_\omega [(b_\omega - A_\omega x)^T z_\omega^j - \frac{1}{2}(\pi_\omega^j)^T D_\omega \pi_\omega^j] \quad \forall j \in J^{\text{opt}}(x), \tag{2.9}$$

where J^{opt} represents the set of indices corresponding to the set of dual solutions corresponding to each scenario. Formally, we have

$$J^{\text{opt}}(x) = \{j : \{\pi_{\omega j}, z_{\omega j}\} \quad \text{solves } S_\omega(x)\}.$$

These constraints are called *optimality cuts*. Given a solution $(\widehat{x}, \hat{\theta})$ from the master problem, we solve the quadratic scenario problems $\mathrm{S}_\omega(\widehat{x})$. If the primal is found to be infeasible or the dual is found to be unbounded, then the master problem is augmented with a feasibility cut. If all the dual problems are finite, we add an optimality cut. We may now define the master problem M, analogous to the specification in the seminal papers by Benders [Ben62] and Van Slyke and Wets [VSW69].

M	$\underset{x,\theta}{\text{minimize}}$	$\frac{1}{2}x^T Q x + c^T x + \theta$	
	subject to	$Ax = b$	
		$x \geq 0$	
		$\theta \geq \sum_\omega p_\omega [(z_\omega^j)^T (b_\omega - A_\omega x) - \frac{1}{2}(\pi_\omega^j)^T D_\omega (\pi_\omega^j)]$,	$j \in J^{\text{opt}}(x)$
		$0 \geq (z_{\omega^j})^T (b_\omega - A_\omega x)$,	$j, \omega \in J^{\text{fea}}(x)$.

2.2.4 The Master Problem and Subproblem

Recall that the recourse-based formulation poses the two-stage stochastic program as

$$\min_{x \geq 0} f_1(x) + \mathcal{Q}(x).$$

We approximate $\mathcal{Q}(x)$ by a series of optimality cuts. To ensure that the choice of x results in a feasible dual problem, we introduce feasibility cuts. We use Infanger's compact notation [Inf94] to specify the optimality and feasibility cuts:

M	$\underset{x,\theta}{\text{minimize}}$	$\frac{1}{2}x^T Q x + c^T x + \theta$	
		$Ax = b$	
		$x \geq 0$	
	subject to	$-(\bar{G}^j)^T x + \theta \geq \bar{g}^j$,	$j \in J^{\text{opt}}(x)$
		$-(G_\omega^j)^T x \geq h_\omega^j$,	$j, \omega \in J^{\text{fea}}(x)$,

where

$$\begin{aligned} G_\omega^j &:= (z_\omega^j)^T A_\omega, \\ g_\omega^j &:= (z_\omega^j)^T b_\omega - \frac{1}{2}(\pi_\omega^j)^T D_\omega \pi_\omega^j, \\ h_\omega^j &:= (z_\omega^j)^T b_\omega, \\ \bar{G}^j &:= \bar{G}(x_j), \\ \bar{g}^j &:= \bar{g}(x_j), \\ \bar{G}(x_j) &= \mathbb{E} G_\omega^j, \\ \bar{g}(x_j) &= \mathbb{E} g_\omega^j. \end{aligned}$$

The cuts in the master problem are obtained by solving subproblem $\mathrm{S}_\omega(x)$ for each realization of ω. Recall the definition of this subproblem from section 2.2.2:

$\mathrm{S}_\omega(x)$	$\underset{\pi_\omega, z_\omega}{\text{maximize}}$	$p_\omega((b_\omega - A_\omega x)^T z_\omega - \frac{1}{2}\pi_\omega^T D_\omega \pi_\omega)$
	subject to	$-D_\omega \pi_\omega + B^T z_\omega \le d_\omega$
		$z_\omega \ge 0.$

We shall reiterate the basic idea of the algorithm by discussing the kth iteration. The solution of the master problem M_k[8] gives us a solution x^k. Using the solution x^k, we proceed to form subproblems $\mathrm{S}_\omega(x^k)$ for each $\omega \in \Omega$, obtaining solutions $(\pi_\omega(x^k), z_\omega(x^k))$. If dual unboundedness is detected in any of the subproblems (implying primal infeasibility), we immediately switch to the master and add a feasibility cut. If all the dual problems are bounded, then we use the optimal dual solutions of the second stage to add an optimality cut to the master problem.

2.2.5 Upper and Lower Bounds

The L-shaped method relies on upper and lower bounds for its termination criterion. The lower bound is obtained by noticing that whenever an optimality cut is added to the master problem, we obtain a better outer approximation of the recourse function. The optimal value of the master problem provides a lower bound on the optimal solution (Lemma 2.23). The sets of optimality and feasibility cuts at iteration k are denoted by J_k^{opt} and J_k^{fea}, respectively. We define a function $\mathcal{Q}_k^L(x)$ as follows:

$$\mathcal{Q}_k^L(x) := \max_{j \in J_k^{\mathrm{opt}}} \bar{G}_j^T x + \bar{g}_j.$$

Lemma 2.23. $\mathcal{Q}_k^L(x) \le \mathcal{Q}(x), \forall k, x.$

[8] The master problem M_k refers to the master problem at the kth major iteration.

Proof. The recourse function $\mathcal{Q}(x)$ may be written as

$$\begin{aligned}\mathcal{Q}(x) &:= \mathbb{E}_\omega \mathcal{Q}(x,\omega)\\ &= \sum_\omega p_\omega \mathcal{Q}(x,\omega)\\ &\geq (\bar{G}^j)^T x + \bar{g}^j, \forall j \in J_k^{\text{opt}}.\end{aligned}$$

■

For a fixed value of x, the optimal solution of the master problem gives the value for θ as the maximum of all the optimality cuts added. In other words, we have the following result.

Lemma 2.24. *For a given x at the end of major iteration k, we have $\theta^*(x) = \mathcal{Q}_k^L(x)$.*

Proof. Follows immediately from the specification of problem M. ■

The master problem M is equivalent to

M^L	$\underset{x,\theta}{\text{minimize}}$	$\frac{1}{2}x^TQx + c^Tx + \mathcal{Q}_k^L(x)$
		$Ax = b$
	subject to	$x \geq 0$
		$x \in X^k$,

where

$$\begin{aligned}X^k := \{ &-(G_\omega^j)^T x \geq h_\omega^j, && j,\omega \in J^{\text{fea}}(x),\\ &-(\bar{G}^j)^T x + \theta \geq \bar{g}^j, && j \in J^{\text{opt}}(x)\}.\end{aligned}$$

The optimal value of the master problem gives a lower bound on the solution simply because not all the optimality cuts have been added. This may be proved formally:

Lemma 2.25. *If M is feasible at iteration k, we have*

$$c^Tx_k + \frac{1}{2}x_k^TQx_k + \theta_k \leq f^*,$$

where f^* is the true optimal solution. If M is infeasible at iteration k, then $f^* = +\infty$.

Proof. This follows from lemma 2.24 and $X_k \subseteq X_{k-1} \subseteq \ldots \subseteq X_1$. ■

Definition 2.26. *The lower bound at iteration k is given by*

$$L_k := c^Tx_k + \frac{1}{2}x_k^TQx_k + \theta_k,$$

where $\{L_k\}$ is a monotonically increasing sequence.

The optimal value of the master problem at the kth iteration is $c^T x_k + \frac{1}{2}(x_k)^T Q x_k + \mathcal{Q}^L(x_k)$. This gives the true cost of solution x_k and is therefore an upper bound to the optimal value, since x_k is feasible but not necessarily optimal. The upper bound to the optimal value may then be given by the following result.

Definition 2.27. *The upper bound at iteration k is given by*

$$U_k := \min\{U_{k-1},\, c^T x_k + \frac{1}{2} x_k^T Q x_k + \mathcal{Q}(x_k)\},$$

where $\{U_k\}$ is a monotonically decreasing sequence.

The algorithm terminates based on termination criterion T1.

Definition 2.28. *Termination criterion T1 is satisfied when $U_k - L_k \leq \epsilon$, where ϵ is a specified tolerance.*

2.2.6 Introduction of Inexact Cuts

Zakeri et al. [ZPR00] discuss the use of inexact cuts in an L-shaped method for stochastic linear programming. Since inexact cuts may be obtained by solving the dual problems to feasibility (and not optimality), there may be significant gain from a computational standpoint. In this section, we present convergence theory for an inexact-cut based L-shaped method for two-stage stochastic QPs.

We begin by defining an inexact optimality cut [ZPR00].

Definition 2.29. *Suppose each dual problem S_ω is solved to within a tolerance ϵ, resulting in the expected cut $(\bar{G}_I^j, \bar{g}_I^j)$. Then the ϵ-inexact cut is defined as*

$$(\bar{G}_I^j)^T x + \bar{g}_I^j + \epsilon > \mathcal{Q}(x).$$

In practice, we use an interior method to solve S_ω with an optimality tolerance of ϵ. We steadily reduce this tolerance by the use of an *inexactness* sequence $\{\epsilon_k\}$ that converges to zero. We term the L-shaped method with inexact cuts the ILS (Inexact L-Shaped) method. We need to restate the bounds for this algorithm. It can be seen that if we rewrite problem M using $(\bar{G}_I, \bar{g}_I)$, then the solutions to the master problem provide lower bounds with a gap of ϵ_k. In particular, we define the inexact-cut based master problem M^I as

$$\begin{array}{lll} \mathrm{M}^I & \underset{x,\theta^I}{\text{minimize}} & \frac{1}{2}x^T Q x + c^T x + \theta^I \\ & & Ax = b \\ & \text{subject to} & x \geq 0 \\ & & -(\bar{G}_I^j)^T x + \theta^I \geq \bar{g}_I^j, \qquad j \in J^{\mathrm{opt}}(x). \end{array}$$

Note that we assume that the primal problems are always feasible, allowing us to omit the feasibility cuts. Suppose we denote the inexact master problem by M_k^I and its solution by f_k^I ; then we may define the lower bound as follows.

Definition 2.30. *The lower bound at iteration k is given by*

$$L_k^I := c^T x_k + \frac{1}{2} x_k^T Q x_k + \theta_k^I,$$

where $\{L_k^I\}$ *is a monotonically increasing sequence.*

Normally, the upper bound is merely the expectation of the costs obtained from the second-stage problem in addition to the first-stage cost. However, this upper bound may not be valid when one uses a sample of the cuts.[9] We use a modified upper bound sequence in which we add the inexactness to the true upper bound.

Definition 2.31. *The upper bound at iteration k*[10] *is given by*

$$U_k^I := \min\{U_{k-1}^I, c^T x_k + \frac{1}{2} x_k^T Q x_k + (\bar{G}_I^k)^T x_k + \bar{g}_I^k + \epsilon_k\},$$

where $\{U_k^I\}$ *is a monotonically decreasing sequence.*

2.2.7 Convergence Theory

We begin with a statement of the L-shaped method for two-stage QPs.

1. $k = 1, \quad U_0^k = \infty, \quad L_0^k = -\infty$
2. Given u, ϵ, $\epsilon_k = \epsilon$
3. While $|U_I^k - L_I^k| > \delta$
   (a) Solve M_k^I to obtain θ_k and x_k
   (b) Update lower bound L_k^I
   (c) Solve $\mathrm{S}_\omega(x_k)$ to obtain (π^ω, z^ω)
   (d) If all $\mathrm{S}_\omega(x_k)$ are feasible, then construct $(\bar{G}_I^k, \bar{g}_I^k)$, otherwise add feasibility cut and go to (a)
   (e) Update lower bound U_k^I and add optimality cut $(\bar{G}_k, \bar{g}_k)$ to M_k
   (f) $\epsilon_{k+1} = \epsilon_k / u$
   (g) $k = k + 1$

In proving convergence of the ILS method, we follow the ideas of Zakeri et al. [ZPR00], which focused on Benders decomposition for linear programming. Our method extends these ideas to convex quadratic programming.

Lemma 2.32. *If the set of cuts* J_k^{opt} *is augmented by* $(\bar{G}_I^k, \bar{g}_I^k)$*, then* $(\bar{G}_I^k, \bar{g}_I^k)$ *is an* ϵ_k*-subgradient of* $\mathcal{Q}$ *at* x_k*.*

[9] We examine this issue more closely in the next section.

[10] One should note that the index j is used to specify a particular cut from the set of appended cuts at an iteration of the algorithm. The index k refers to a particular iteration of the L-shaped algorithm. In particular, after k iterations, the master problem can have as many as k optimality cuts.

Proof. Recall from the statement of the algorithm that the set of cuts is only updated if $(\bar{G}_I^k)^T x_k + \bar{g}_I^k + \epsilon_k > \mathcal{Q}(x_k)$. Since, we have $\mathcal{Q}(x) \geq (\bar{G}_I^k)^T x + \bar{g}_k^I$,

$$\begin{aligned} &\mathcal{Q}(x) - \mathcal{Q}(x_k) \geq (\bar{G}_I^k)^T (x - x_k) - \epsilon_k \\ &\quad \Longrightarrow \ \mathcal{Q}(x) \geq \mathcal{Q}(x_k) + (\bar{G}_I^k)^T (x - x_k) - \epsilon_k. \end{aligned}$$

■

Lemma 2.33. *Let U_k^I, L_k^I, x_k and θ_k^I be generated by the ILS method. Then, we have*

$$U_k^I - L_k^I \leq v_k - \theta_k^I + \epsilon_k,$$

where $v_k := (\bar{G}_I^k)^T x_k + \bar{g}_I^k$.

Proof. At iteration k, the inexact upper bound is bounded from above:

$$U_k^I \leq \frac{1}{2} x_k^T Q x_k + c^T x_k + \epsilon_k + (\bar{G}_I^k)^T x_k + \bar{g}_I^k.$$

Moreover, the lower bound is $L_k^I := \frac{1}{2} x_k^T Q x_k + c^T x_k + \theta_k^I$. The result follows with $v_k := (\bar{G}_I^k)^T x_k + \bar{g}_I^k$. ■

It is necessary to assume that the set of subgradients is bounded. In the linear case, we may prove this using complete recourse. However, such a proof depends on the fact that the optimal solution of each subproblem is a basic feasible solution. This may not occur in the case of quadratic programming.

Assumption 2.34. *The coefficient of the optimality cut $\bar{G}_I^k$ is bounded.*

We now show that the ILS method terminates with a δ-optimal solution in a finite number of iterations. If not, then the following two possibilities may occur:

1. There exists an m such that $\theta_k \geq v_k$ for all $k \geq m$.
2. There exists a subsequence $(x_{\sigma(k)}, \theta_{\sigma(k)})$ such that $\theta_{\sigma(k)} < v_{\sigma(k)}$.

Lemma 2.35. *If there exists an m such that for all $k \geq m$, $\theta_k \geq v_k$, then $U_k^I - L_k^I \to 0$.*

Proof. By invoking Lemma 2.33 and the hypothesis, we have for $k \geq m$,

$$U_k^I - L_k^I \leq v_k - \theta_k + \epsilon_k \leq \epsilon_k.$$

Since the inexactness sequence $\{\epsilon_k\} \to 0$, the result follows. ■

Lemma 2.36. *If there exists a convergent subsequence $\{x_{\tau(k)}, \theta_{\tau(k)}\}$ such that $\theta_{\tau(k)} < \mathcal{Q}(x_{\tau(k)})$ for $k \geq 1$, then*

1. $v_{\tau(k)} - \theta_{\tau(k)} \leq v_{\tau(k)} - v_{\tau(k-1)} - \bar{G}_{\tau(k-1)}^T (x_{\tau(k)} - x_{\tau(k-1)})$.
2. $\lim v_{\tau(k)} - v_{\tau(k-1)} = 0$.

3. $\liminf \bar{G}_{\tau(k-1)}(x_{\tau(k)} - x_{\tau(k-1)}) \geq 0$.

Proof. 1. The solution of the master problem at $\tau(k)$, given by $(x_{\tau(k)}, \theta_{\tau(k)})$, is feasible with respect to the cut added in iteration $\tau(k-1)$. To be specific, we have

$$\theta_{\tau(k)} \geq (\bar{G}_I^{\tau(k-1)})^T x_{\tau(k)} + \bar{g}_{\tau(k-1)}^I.$$

Therefore, it can be seen that

$$\begin{aligned} v_{\tau(k)} - \theta_{\tau(k)} &\leq v_{\tau(k)} - (\bar{G}_I^{\tau(k-1)})^T x_{\tau(k)} - \bar{g}_I^{\tau(k-1)} \\ &= v_{\tau(k)} - v_{\tau(k-1)} + v_{\tau(k-1)} - (\bar{G}_I^{\tau(k-1)})^T x_{\tau(k)} - \bar{g}_I^{\tau(k-1)} \\ &= v_{\tau(k)} - v_{\tau(k-1)} - (\bar{G}_I^{\tau(k-1)})^T (x_{\tau(k)} - x_{\tau(k-1)}). \end{aligned}$$

2. By the assumption that $\{(x_{\tau(k)}, \theta_{\tau(k)})\} \to \{(x^*, \theta^*)\}$, we have

$$\mathcal{Q}(x_{\tau(k)}) - \epsilon_k \leq v_{\tau(k)} \leq \mathcal{Q}(x_{\tau(k)}),$$

implying that $v_{\tau(k)} \to \mathcal{Q}(x^*)$.

3. Since $\lim v_{\tau(k)} = 0$ and

$$v_{\tau(k)} - v_{\tau(k-1)} - (\bar{G}_I^{\tau(k-1)})^T (x_{\tau(k)} - x_{\tau(k-1)}) \geq v_{\tau(k)} - \theta_{\tau(k)} > 0,$$

we have $\liminf (\bar{G}_I^{\tau(k-1)})^T (x_{\tau(k-1)} - x_{\tau(k)}) \geq 0$.

■

Lemma 2.37. *Suppose* $X = \{x \geq 0 \mid Ax = b\}$ *is bounded and* $\mathrm{dom}\mathcal{Q} = \mathbb{R}^n$. *If there exists a subsequence* $\{x_{\tau(k)}, \theta_{\tau(k)}\}$ *such that* $\mathcal{Q}(x_{\tau(k)}) < \theta_{\tau(k)}$ *for* $k \geq 1$, *then* $U_k^I - L_k^I \to 0$.

Proof. By the boundedness of X, the sequence $\{(x_{\tau(k)}, \theta_{\tau(k)})\}$ is also bounded. In particular, we may extract a subsequence $\{(x_{\tau(k)}, \theta_{\tau(k)})\}$ that is convergent to (x^*, θ^*). By showing that $U_{\tau(k)}^I - L_{\tau(k)}^I \to 0$, we have the result. By lemma 2.33, we have

$$\begin{aligned} 0 \leq U_{\tau(k)}^I - L_{\tau(k)}^I &\leq v_{\tau(k)} - \theta_{\tau(k)} + \epsilon_{\tau(k)} \\ &\leq v_{\tau(k)} - v_{\tau(k-1)} - (\bar{G}_I^{\tau(k-1)})^T (x_{\tau(k)} - x_{\tau(k-1)}) + \epsilon_{\tau(k)} \quad \text{by lemma 2.36, part 1.} \end{aligned}$$

By the boundedness of $\bar{G}_{\tau(k)}$, we have $(\bar{G}_I^{\tau(k-1)})^T (x_{\tau(k)} - x_{\tau(k-1)}) \to 0$. Moreover, by lemma 2.36 part 2, we have $\lim(v_{\tau(k)} - v_{\tau(k-1)}) \to 0$. This implies that $U_{\tau(k)}^I - L_{\tau(k)}^I \to 0$. Since $\{U_k^I - L_k^I\}$ is a decreasing sequence with a convergent subsequence, this implies that $U_k^I - L_k^I \to 0$. ■

Theorem 2.38. *If* X *is bounded and* $\mathrm{dom}\mathcal{Q} = \mathbb{R}^n$, *the ILS algorithm terminates in a finite number of iterations with a* δ*-optimal solution of SQP.*

Proof. By lemmas 2.33 and 2.37, we have $U_k^I - L_k^I \to 0$. There exists some K such that $U_K^I - L_k^I < \delta$, implying termination in K iterations. Let x_K be such that $U_K = \frac{1}{2} x_K^T Q x_K + c^T x_K + v_K + \epsilon_K$.

Then

$$\frac{1}{2}x_K^T Q x_K + c^T x_K + \mathcal{Q}(x_K) \leq \frac{1}{2}x_K^T Q x_K + c^T x_K + v_K + \epsilon_K \leq L_K + \delta,$$

implying that the optimal cost at x_K is within δ of the optimal value. ■

2.3 The Inexact-Cut-Sampling Algorithm

In the L-shaped method, the construction of a single optimality cut requires the solution of $|\Omega|$ subproblems (assuming the distribution is discrete). If $|\Omega|$ is large, it may be computationally burdensome. Instead, one may use a sample of observations to estimate the coefficients and intercept of an optimality cut.

For instance, we could sample the exact cuts (called exact-cut sampling) and construct confidence intervals for the optimal value of the stochastic program. We describe this strategy in section 2.3.1. Another possibility is to make the sampled cut inexact by reducing its intercept. We provide some details of this strategy in section 2.3.2.

2.3.1 Exact Cut Sampling

This section has its roots in the work by Dantzig and Infanger [DI95]. Recall that the optimality cut may be specified as

$$\theta \geq \bar{G}^T x + \bar{g},$$

where we suppress the iteration subscripts. If $f(y_\omega)$ is given by $f(y_\omega) := \frac{1}{2}y_\omega^T D y_\omega + d_\omega^T y_\omega$, then by weak duality, we have

$$f(y_\omega) \geq (G^\omega)^T x + g^\omega.$$

Therefore,

$$\begin{aligned}
&\theta + f(y_\omega) \geq (G^\omega)^T x + g^\omega + \theta \\
&\quad \Longrightarrow\ \theta \geq (G^\omega)^T x + g^\omega + (\theta - f(y_\omega) \\
&\quad \Longrightarrow\ \theta \geq \frac{1}{N}\sum_{\omega\in S}(G^\omega)^T x + \frac{1}{N}\sum_{\omega\in S} g^\omega + \frac{1}{N}\sum_{\omega\in S}(\theta - f(y_\omega)).
\end{aligned}$$

This implies that

$$\theta \geq \widehat{G}^T x + \widehat{g} - \phi, \tag{2.10}$$

where $\widehat{G}, \widehat{g}$ and ϕ are defined as

$$\widehat{G} := \frac{1}{N}\sum_{\omega\in S}(G^\omega)^T x, \tag{2.11}$$

$$\widehat{g} := \frac{1}{N}\sum_{\omega\in S} g^\omega \tag{2.12}$$

$$\text{and } \phi := \frac{1}{N}\sum_{\omega\in S}(\theta - f(y_\omega)). \tag{2.13}$$

In other words, ϕ is a sample average of random variables $f(y_\omega) - \theta$. This leads to the following definition.

Definition 2.39. *The probabilistic and pseudo cuts are defined as*

$$\theta \geq \widehat{G}^T x + \widehat{g} - \phi, \tag{2.14}$$

$$\textit{and } \theta \geq \widehat{G}^T x + \widehat{g}, \tag{2.15}$$

respectively.

We may use the pseudo cuts to construct a probabilistic master problem at iteration k of the probabilistic L-shaped method:

$$\begin{array}{lll}
\widehat{\text{M}} & \underset{x,\theta}{\text{minimize}} & \tfrac{1}{2}x^T Qx + c^T x + \theta \\
& & x \geq 0 \qquad\qquad : \gamma \\
& & -(\widehat{G}^1)^T x + \theta \geq \widehat{g}^1 \qquad : \lambda_1 \\
& \text{subject to} & -(\widehat{G}^2)^T x + \theta \geq \widehat{g}^2 \qquad : \lambda_2 \\
& & \vdots \\
& & -(\widehat{G}^k)^T x + \theta \geq \widehat{g}^k. \qquad : \lambda_k
\end{array}$$

At iteration k of this method, a random sample of size N_k is generated and denoted by S^k (where $|S^k| = N_k$). Then the coefficients $\widehat{G}^k$ and $\widehat{g}^k$ are computed to enable the construction of a pseudo cut:

$$-(\widehat{G}^k)^T x + \theta \geq \widehat{g}.$$

The expected second-stage costs at iteration k are $\mathbb{E}f(y_\omega(x_k))$, where y_ω minimizes the second-stage problem $S_\omega(x)$. However, the probabilistic L-shaped method uses a sample S_k at iteration k. Therefore a sample mean of the minimum second-stage costs at the kth iteration is given by

$$\hat{\theta}_k := \frac{1}{N_k}\sum_{w\in S^k} f(y_\omega).$$

The sample standard variance at iteration k may be computed as

$$\hat{\sigma}_k^2 := \frac{1}{N_k(N_k-1)} \sum_{w \in S^k} (f(y_\omega) - \hat{\theta}_k)^2.$$

Suppose the solution of the probabilistic master problem is denoted by $\widehat{x}_k$. Furthermore, an unbiased estimate of the total cost is given by

$$\widehat{v}^k := c^T \widehat{x}_k + \tfrac{1}{2} \widehat{x}_k^T Q \widehat{x}_k + \hat{\theta}_k, \tag{2.16}$$

where $\hat{\sigma}_k^2$ is an unbiased estimate of the variance of $\widehat{v}^k$. Since this is a probabilistic method, various termination criteria may be used. We use a criterion based on the number of major iterations taken by the method. Upon termination, one may choose x_k as the one corresponding to the minimum $\widehat{v}_k$:

$$\widehat{x}^* := \widehat{x}_l, \text{ where } l = \arg\min\{\widehat{v}_k\}.$$

The optimality conditions of the probabilistic master problem at a solution x of the master problem require the existence of λ and γ such that the following hold:

$$\begin{aligned} c + Qx - \sum_{j=1}^{K} \widehat{G}^j \lambda_j - \gamma &= 0 \\ \sum_{j=1}^{K} \lambda_j &= 1 \\ \gamma &\geq 0 \\ \lambda_j &\geq 0, \quad j = 1, \ldots, K. \end{aligned}$$

Theorem 2.40. *Let us define C, H, h and d as*

$$C = \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}, \quad H = \begin{pmatrix} -(\widehat{G}^1)^T & 1 \\ -(\widehat{G}^2)^T & 1 \\ \vdots & \vdots \\ -(\widehat{G}^K)^T & 1 \\ 1 & 0 \end{pmatrix}, \quad h = \begin{pmatrix} \widehat{g}^1 \\ \widehat{g}^2 \\ \vdots \\ \widehat{g}^K \\ 0 \end{pmatrix}, \quad d = \begin{pmatrix} c \\ 1 \end{pmatrix}.$$

Let $\widehat{v}^*$ and v^* be the solutions of the probabilistic and true master problems upon termination. Then

$$|\widehat{v}^* - v^*| \leq |\Delta|, \tag{2.17}$$

where

$$\Delta := \sum_{j=1}^{K} \lambda_j \phi_j, \quad \sum_{j=1}^{K} \lambda_j = 1, \quad \lambda_j \geq 0.$$

Proof. If the primal master problem is denoted by

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \tfrac{1}{2} w^T C w + d^T w \\ \text{subject to} \quad & Hw \geq h, \end{aligned}$$

then the dual problem (given by Dorn duality [Dor61]) is

$$\begin{aligned} \underset{s,u}{\text{maximize}} \quad & -\tfrac{1}{2} s^T C s + h^T u \\ \text{subject to} \quad & H^T u - Cs = h \\ & u \geq 0. \end{aligned}$$

Therefore, $\widehat{v}^*$ is given by

$$\widehat{v}^* = -\frac{1}{2} s^T C s + h^T u = -\frac{1}{2} s^T C s + \sum_{j=1}^{K} \lambda_j^T \widehat{g}^j.$$

By using the multipliers from the probabilistic master problem, we note that for the true master problem, we have

$$\begin{aligned} \tfrac{1}{2}(x^*)^T Q x^* + c^T x^* + \theta^* &= z^* \\ -(\widehat{G}^1)^T x^* + \theta^* &\geq \widehat{g}^1 - \phi_1 \qquad : \lambda_1 \\ &\vdots \\ -(\widehat{G}^K)^T x^* + \theta^* &\geq \widehat{g}^K - \phi_K \qquad : \lambda_K \\ x^* &\geq 0. \qquad : \gamma \end{aligned}$$

This leads to

$$\begin{aligned} v^* - \sum_{j=1}^{K} \lambda_j^T \widehat{g}^j + \sum_{j=1}^{K} \lambda_j \phi_j &\geq \tfrac{1}{2}(x^*)^T Q x^* + c^T x^* - \sum_{j=1}^{K} \lambda_j^T ((\widehat{G}^j)^T x^*) - \gamma^T x^* \\ v^* - \sum_{j=1}^{K} \lambda_j^T \widehat{g}^j + \sum_{j=1}^{K} \lambda_j \phi_j &\geq -\tfrac{1}{2}(x^*)^T Q x^* + (Qx^* + c - \sum_{j=1}^{K} (\widehat{G}^j)\lambda_j - \gamma)^T x^* \\ v^* - \sum_{j=1}^{K} \lambda_j^T \widehat{g}^j + \tfrac{1}{2}(x^*)^T Q x^* + \sum_{j=1}^{K} \lambda_j \phi_j &\geq 0. \end{aligned}$$

This implies that $|\widehat{v} - v^*| \leq |\Delta|$, where the random variable Δ is $\Delta := \sum_{j=1}^{K} \lambda_j \phi_j$. ■

The result by Dantzig and Infanger [DI95] requires that the variance of the second-stage cost is known or can be approximated closely by $\sigma_k^2(x_k)$. The same is true for the extension to the quadratic case.

2.3.2 Probabilistic Bounds

In this section, we use the distribution of Δ in prescribing confidence bounds for the solution as shown in [DI95]. In this analysis, the sample-size is always kept at N. In particular, if the true variance of the error is given by $(\sigma^*)^2$, then in the construction of each optimality cut, we obtain

$$(\sigma_j^*)^2 = \frac{1}{N-1}\sum_{i=1}^{N}(f(y_i(x_j^*) - \theta_j^*)^2.$$

Moreover,

$$\phi_j \sim \mathcal{N}(0, \frac{\sigma^*}{\sqrt{N}}) = \frac{\sigma^*}{\sqrt{N}}\mathcal{N}(0,1).$$

It should be noted that this requires the optimal solution and the true variance of the optimal solution. We may approximate this variance by the variance of the best available solution. Dantzig and Infanger [DI95] suggest the construction of two distributions that bound the distribution of Δ: a worst-case distribution and a conservative distribution. We discuss the worst-case distribution here and note that is constructed by

$$\Delta_w := \max_j \phi_j \geq \sum_{j=1}^{K} \lambda_j \phi_j,$$

since $\sum_{j=1}^{K} \lambda_j = 1$ and $\lambda_j \geq 0, \forall j$. Then

$$\mathbb{P}(\widehat{v}^* - \Delta_w \leq \widehat{v} - \Delta \leq v^*) \geq (1-\alpha),$$

since $\Delta_w \geq \Delta$.

2.3.3 Inexact Cut Sampling

The probabilistic cuts used in the earlier section were sampled exact cuts in that they represented sample means of exact cuts. The errors in the sampled cuts given by ϕ could be either positive or negative. As a result, constructing a valid lower bound to the solution was difficult. However, as mentioned earlier, we may construct confidence intervals associated with upper and lower bounds.

In this section, we modify the sampled inexact cuts by using the sample variance of the cut. Specifically, if the variance of the sampled cut at the kth iteration is given by $\hat{\sigma}_k^2$, then the value of the kth optimality cut at x is given by

$$(\widehat{G}_I^k)^T x + \widehat{g}_I^k, \tag{2.18}$$

and we define

$$\widehat{G}_I = \frac{1}{N}\sum_{j=1}^{N} G_I^j,$$
$$\widehat{g}_I = \frac{1}{N}\sum_{j=1}^{N} g_I^j - \beta\hat{\sigma}, \quad \beta \geq 1.$$

One would expect that in practice, such cuts would have a higher probability of being valid. As a result, the solution to the master problem would be a valid point estimate of a lower bound for SQP. The inexactness in the cuts defined above arises from solving the dual problems to feasibility and from reducing the intercept term of the cut. This leads to a notion of *probabilistic validity*.

Definition 2.41. *The α-inexact probabilistic cut $(\widehat{G}_I, \widehat{g}_I)$ is said to be probabilistically valid if*

$$\mathbb{P}(\widehat{G}_I^T x + \widehat{g}_I + \beta\hat{\sigma} + \epsilon > \mathcal{Q}(x)) \geq (1-\alpha).$$

For instance, to ensure that the cut is valid with a probability of 95%, we specify $\beta = 4$. The inexact version of the probabilistic master problem is denoted by $\widehat{\mathrm{M}}^I$:

$\widehat{\mathrm{M}}^I$	$\underset{x,\theta}{\text{minimize}}$	$\frac{1}{2}x^TQx + c^Tx + \theta$	
		$x \geq 0$	$: \gamma$
		$-(\widehat{G}_I^1)^Tx + \theta \geq \widehat{g}_I^1$	$: \lambda_1$
	subject to	$-(\widehat{G}_I^2)^Tx + \theta \geq \widehat{g}_I^2$	$: \lambda_2$
		$\vdots$	
		$-(\widehat{G}_I^k)^Tx + \theta \geq \widehat{g}_I^k.$	$: \lambda_k$

This allows us to construct a probabilistic lower bound.

Lemma 2.42. *The probabilistic lower bound at iteration k is given by*

$$\widehat{L}_k^I := c^Tx_k + \tfrac{1}{2}x_k^TQx_k + \theta_k^I,$$

where $\{L_k^I\}$ is a monotonically increasing sequence and x_k and θ_k^I are obtained from solving the master problem $\widehat{M}^I$.

Since $\widehat{L}_k^I$ is a random variable, we may obtain its variance by extending an analytical technique adopted by Infanger [Inf94]. By using the dual of the master problem, we find that the variance of the lower bound is given by the sum of the variances of each cut in the master problem weighted by the appropriate multiplier. Note that every cut in the master problem is obtained by solving a sample of subproblems, which provides a mean cost $\bar{z}_j$ with a variance of $\sigma^2_{\bar{z}_j}$. In particular, this is the variance in the cut and it allows us to state the variance of the sampled lower bound as follows:

$$\sigma^2_{\widehat{L}_k^I} := \sum_{j=1}^{k} (\rho_j^k)^2 \sigma^2_{\bar{z}_j}.$$

By monotonicity of the lower bounds, we may say that the last computed lower bound is indeed the largest in mean and is given by

$$\widetilde{L}_k^I := \mathcal{N}(\widehat{L}_k^I, \sigma^2_{\widehat{L}_k^I}).$$

Definition 2.43. *The upper bound at iteration k is given by*

$$\widehat{U}_k^I := \min\{\widehat{U}_{k-1}^I, c^T x_k + \frac{1}{2} x_k^T Q x_k + (\widehat{G}^k)^T x_k + \widehat{g}^k + \epsilon_k + \beta\hat{\sigma}_k\},$$

where $\{\widehat{U}_k^I\}$ is a monotonically decreasing sequence.

The upper bound is a random variable denoted by $\widetilde{U}_k^I$ because it represents the sum of the first-stage cost $f_x(x_k)$ and sample mean of the recourse cost $\widetilde{z}(x_k)$, the latter being a random variable. Therefore

$$\widetilde{U}_k^I = N(f_1(x_k) + \bar{z}(x_k), \sigma^2_{\bar{z}(x_k)}).$$

Since the random upper bounds need not be decreasing in means, we define a minimum $\widetilde{U}_k^I$ by

$$\widetilde{U}_k^{\min,I} = N((f_1(x_j) + \bar{z}(x_j), \sigma^2_{\bar{z}(x_j)}), \quad j = \arg\min_k \{\widehat{U}_k^I\}. \tag{2.19}$$

2.3.4 Stopping Rule and Confidence Intervals

Deterministic cutting-plane methods prescribe termination when the upper and lower bounds are within a tolerance. However, in our case the bounds are random in nature and we need to test the hypothesis that the sample means for the upper and lower bound are indeed different. A statistical technique for conducting such a test is called the Student's t-test and requires that the underlying distributions of the two populations being compared are normal and independent. We assume normality in the distributions but we do not have independence because the upper and lower bounds are constructed from the same samples.

We ensure independence between the two sample means by resampling for the lowest upper bound using the current x_k. Note that this resampling need not be done at every point but only when the sample means satisfy the null hypothesis. If the resampled data does satisfy the null hypothesis, we compute a confidence interval and terminate the algorithm.

Definition 2.44. *We define the t-test criterion t_k as*

$$t_k = \frac{\bar{z}_k^M - \overline{LB}_k}{\sqrt{b}} \quad \text{and} \quad b = \frac{(\sigma^2_{\widetilde{LB}_k} + \sigma^2_{\widetilde{UB}_k})}{n_k}.$$

The termination criterion T1 is satisfied when $t_k \leq 1.96$ for a 95% test.

The resulting 95% confidence interval is given by

$$\overline{LB}_k - 1.96\sigma_{\widetilde{LB}_k} \leq z^* \leq \overline{UB}_k + 1.96\sigma_{\widetilde{UB}_k}.$$

2.4 Computational Results

This section discusses the behavior of the algorithm on a two-stage stochastic QP test problem set [CW00].[11] It suffices to test the algorithm on such a class, given the equivalence between stochastic Nash games and two-stage stochastic QPs. Generally, test problem sets serve as proxies for real-life data and it is helpful, if not essential, for one to have the optimal value and solution available at the outset. Moreover, one would like to control various characteristics of the test problems that may prove crucial between obtaining a solution and failing. These would include issues such as the condition of the master and subproblems and the degeneracy of the problems. In addition, we have control on the size of both the first and second-stage decisions.

In section 2.4.1, we give a brief description of the problem set, focusing on the various specifications of the problem and provide the parameters of our problem generator. In section 2.4.2, we demonstrate the benefits of solving the dual problems to feasibility. Section 2.4.3 describes the performance of the sampling-based cutting-plane method. The method uses a fixed sample at every iteration and terminates after a fixed number of major iterations (if the sample means of the upper and lower bounds do not come within δ). The sampling method proves useful when the size of the sample space is large.

2.4.1 A Test Problem Set

We generate stochastic quadratic programs based on the following large-scale quadratic program:

F-QP	$\underset{x}{\text{minimize}}$	$\frac{1}{2}x^T Px + c^T x + \mathbb{E}_\omega \Psi(x,\omega)$
	subject to	$Ax \leq b,$

where $\Psi(x,\omega)$ is the optimal value of the subproblem

S-QP(ω)	$\underset{z}{\text{maximize}}$	$-\frac{1}{2}z^T Hz + z^T(S\omega - Tx)$
	subject to	$Wz \leq q(\omega).$

Chen and Womersely [CW00] describe an approach for generating random two-stage stochastic quadratic programs F-QP and S-QP.

Assumption 2.45. *The following assumptions hold for every problem:*

1. *P and H are symmetric positive definite matrices.*
2. *$\sum_{j=1}^{r} p(\omega_j) = 1$, where j represents the index of the appropriate subproblem.*
3. *Each of the r realizations $\omega_j, j = 1,\ldots,r$, has equal probability.*

Important benefits of any test problem set include knowledge of the optimal solution a priori as well as ability to control various aspects of the test problem such as the condition of the Hessian at

[11] We constructed a MATLAB version of this test problem set.

the first and second stage. Our problem set allows control on the number of active constraints with positive multipliers at the solution. In Table 2.1, we provide an incomplete list of the inputs to the problem set and refer the reader to [CW00] for a more exhaustive set. The algorithms have been implemented in MATLAB 7.0 and tested on a Linux-based (2 GB of RAM and 3.3 GHz) processor. We use Mex file interfaces from TOMLAB [Hol99] for the sparse QP solvers SQOPT [GMS97] and CPLEX. The barrier version of the latter is used to solve the QP subproblems.

We restrict the testing to problems in which the first stage has $m = 4$ constraints and $n = 5$ variables. The second-stage quadratic programs are also assumed to have $m_2 = 4$ constraints and $n_2 = 5$ variables. The number r of quadratic programs in the second stage is contingent on the number of random parameters `no-comps` and the number of realizations (ℓ) each random parameter is assumed to take. In particular, $r = \texttt{no-comps}^{\ell}$.

2.4.2 Inexact L-Shaped Method for Stochastic QPs

We compared two update strategies. The first started the inexactness sequence at 1e-3 and reduced it by a factor of 5 at every major iteration. The second strategy started the sequence at 1e-5 and reduced it by a factor of 2 at every major iteration. We compared these update strategies with an exact cut approach that specified an optimality tolerance of 1e-8 at every major iteration.

Table 2.2 compares the performance of the three methods. We solved problems with r as large as 65000. The ILS method is terminated if the upper and lower bounds fall within 1e-6 or the maximum number of major iterations is exceeded. We find that the first strategy always performs the best. In fact, for some problems, using inexact cuts saves nearly 40% in terms of computational effort compared to when using exact cuts. Note that we measure this effort in terms of the number of QP iterations.

Figure 2.4.2 shows the performance profiles of the 4 algorithms constructed on the basis of major iterations. The construction of such profiles is discussed by Dolan and Moré [DM02]. We summarize the construction of such profiles for a specific performance metric. Suppose we intend to provide a benchmark for computational time. Then, for solver s, we have the time $t_{p,s}$ to solve problem p, and we may define a performance ratio

$$r_{p,s} = \frac{t_{p,s}}{\min_{s\in\mathcal{S}} t_{p,s}}, \quad \forall s \in \mathcal{S}, p \in \mathcal{P}.$$

Note that $\mathcal{S}$ and $\mathcal{P}$ are the set of solvers and problems, respectively. Then, the performance profile

TABLE 2.1
Input parameters

Parameter	Definition
r	No. of second-stage quadratic programs
n, n_2	Size of x and z_ω
m, m_2	No. of linear constraints in first and second stage
t, a	Condition no. of P and H
μ_0, ν_0	No. of active constraints with positive/zero multipliers at x^*
m_0	Largest no. of active constraints at any of the r constraints

K	ϵ_0=1e-3, $\epsilon_{k+1} = \epsilon_k/5$			ϵ_0 = 1e-5, $\epsilon_{k+1} = \epsilon_k/2$			ϵ_0 = 1e-8, $\epsilon_{k+1} = \epsilon_k$		
	$U_k^I - L_k^I$	f_k	qpIter	$U_k^I - L_k^I$	f_k	qpIter	$U_k^I - L_k^I$	f_k	qpIter
243	2.26e-6	1.63e+2	6769	8.84e-6	1.63e+2	8611	2.09e-6	1.63e+2	11960
729	5.20e-6	1.35e+3	48122	3.45e-6	1.35e+3	52886	3.39e-6	1.35e+3	69172
1024	6.08e-6	6.54e+2	79939	3.95e-4	6.54e+2	143334	2.60e-6	6.54e+2	69473
2048	1.78e-4	8.16e+2	298894	2.29e-4	8.16e+2	310550	5.71e-5	8.16e+2	362801
4096	4.56e-6	5.52e+3	128343	8.24e-6	5.52e+3	156547	2.86e-6	5.52e+3	216419
6561	9.22e-6	8.73e+3	452254	9.32e-6	8.73e+3	481118	9.37e-6	8.73e+3	626814
16384	1.52e-7	7.88e+3	791463	1.21e-7	7.88e+3	904883	7.35e-8	7.88e+3	1256028
19683	4.01e-6	4.30e+2	2285399	2.43e-5	4.30e+2	2582340	2.29e-5	4.30e+2	2925343
32768	2.83e-6	6.23e+2	3050896	1.53e-5	6.23e+2	4840473	2.13e-5	6.23e+2	5612627
65536	8.75e-7	3.36e+3	5287912	1.43e-6	3.36e+3	5535908	9.22e-7	3.36e+3	6567603

TABLE 2.2
Comparison of two update rules with the exact-cut L-shaped method

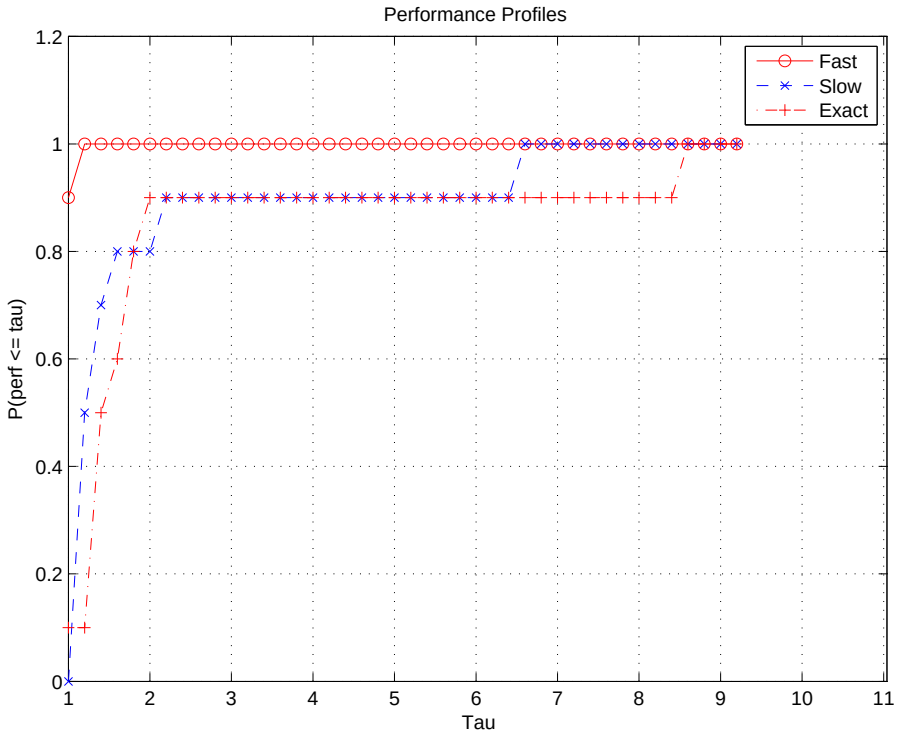

FIGURE 2.1 *Performance profiles for different updates in the ILS method*

TABLE 2.3
Performance of sampling method

K	f^*	N=1000		N=2,500		N = 5000	
		$\bar{z}_{LB}$	qpiter	$\bar{z}_{LB}$	qpiter	$\bar{z}_{LB}$	qpiter
15625	1.52e+3	1.52e+3	125571	1.49e+3	314681	1.50e+3	627455
16384	7.88e+3	7.64e+3	117360	7.57e+3	294124	7.83e+3	588041
16384	4.11e+2	4.09e+2	133785	4.02e+2	333729	4.04e+2	667167
16807	4.36e+4	4.37e+4	116802	4.25e+4	297148	4.24e+4	595825
32768	6.23e+2	6.00e+2	134305	6.17e+2	335131	6.16e+2	671115
46656	2.86e+2	2.90e+2	128331	2.85e+2	320330	2.86e+2	642430
59049	7.85e+3	7.66e+3	120485	7.72e+3	299979	7.65e+3	600952
65536	3.36e+3	3.36e+3	113944	3.29e+3	282442	3.30e+3	572170
65536	4.97e+3	4.77e+3	120972	4.88e+3	303125	4.95e+3	605597
78125	1.19e+3	1.17e+3	112622	1.18e+3	281680	1.17e+3	563376
117649	2.78e+3	2.73e+4	111247	2.77e+4	179562	2.71e+4	572044
131072	1.27e+3	1.19e+3	97636	1.24e+3	299735	1.27e+3	435338

for a particular solver, say s, is

$$\rho_s(\tau) := \frac{1}{n_p} size\{p \in \mathcal{P} : r_{p,s} \leq \tau\},$$

where n_p is the number of problems in the test problem set. Intuitively, the performance profile charts out a ratio that represents how well a solver does with respect to its competitors.

2.4.3 A Sampling-based Method

To accommodate large sample sizes, one may be satisfied with bounds on the optimal value of the optimization problem. By using a sample of the cuts at every major iteration, we construct probabilistic upper and lower bounds on the optimal value. The use of inexact cuts ensures that one may obtain an approximately valid point estimate of the lower bound.

Our sampling method terminates after a fixed number of major iterations. In the future, we plan to use more sophisticated termination criteria. At each major iteration, we obtain a random sample of size N. We compare the performance of the method under three settings of N: 1000, 2500 and 5000. Table 2.3 provides a point estimate for the lower bound and the number of QP iterations for each problem. The column f^* shows the optimal value for the problem.

We observe that the sampling method is significantly cheaper than the ILS method. For instance, if one considers the problem with $65,536$ realizations, the ILS method with the fastest update takes over 5 million QP iterations. The sampling-based method with $N = 1000$ results in a point estimate of 3.36e3 but requires approximately 113,000 QP iterations. If one considers a problem with 32,768 realizations, the sampling method wit $N = 5000$ saves nearly 80% in terms of computational effort. This comes at the cost of 1.12% gap between the point estimate of the solution and the true solution.

Our sampling method provides us with probabilistic lower and upper bounds. Moreover, by using the variances of the obtained cuts, we may construct confidence intervals on these bounds. Figures 2.2 and 2.3 compare the scaled bounds for $N = 1000, 2500$ and 5000. Note that the scaling is carried out with reference to the true solution.

As one would expect, the use of a larger sample results in tighter confidence bounds. Furthermore, it is observed that the point estimates of the solution (approximate lower bounds) are within 6.30%, 3.93% and 2.75% of the true solution for $N = 1000, 2500$ and 5000, respectively.

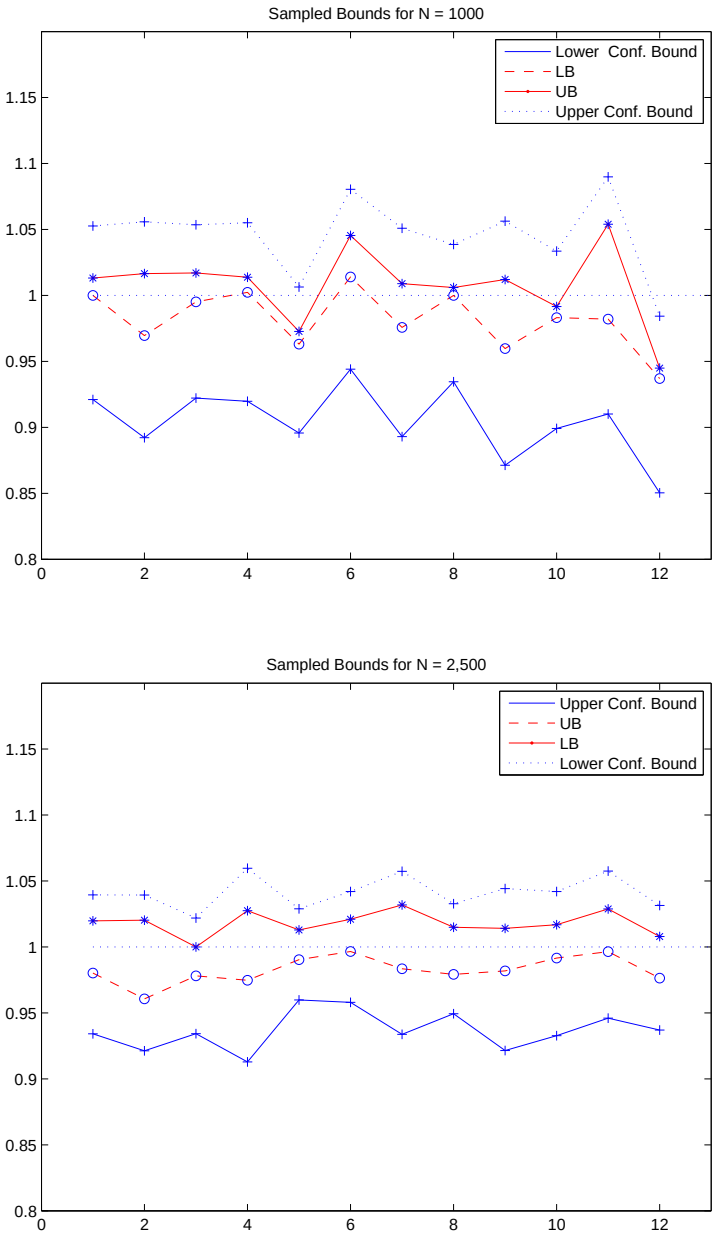

FIGURE 2.2 *Confidence bounds from sampling method*

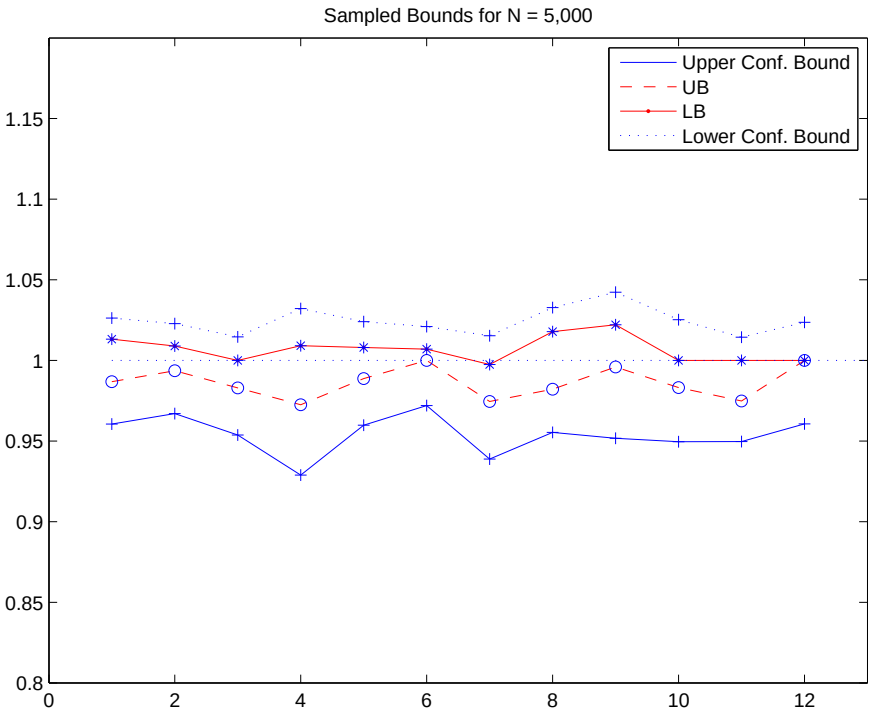

FIGURE 2.3 *Confidence bounds from sampling method*

2.5 Contributions and Future Research

We have presented a method for solving a class of stochastic quadratic programs. One instance of such problems is provided by a class of games in which agents solve two-stage stochastic convex quadratic programs. We show an equivalence of this class of problems to a larger two-stage stochastic quadratic program.

Zakeri et al. [ZPR00] discuss an inexact-cut method for linear programming. By using the duality theory of convex quadratic programming, we construct an inexact-cut based L-shaped method. The convergence of this algorithm is proved in section 2.2.7. An implementation of this method shows that using inexact cuts results in significant savings in computational effort.

Often the size of the number of realizations in the second-stage makes it difficult to compute an expectation of the recourse function. Dantzig and Infanger [DI95] suggest the sampling of cuts and prescribe probabilistic lower and upper bounds. These ideas are extended to the case of quadratic programming. By using the variance of the cuts, we modify the optimality cuts that are added to the master problem. This results in approximately valid cuts. We use these ideas to construct a sampling-based method that uses a fixed sample size and terminates after a specific termination criterion is satisfied. We find that such a method performs well on very large problems. In particular, we obtain point estimates of the solution that are less than 3% away from the true solution when using a sample of $N = 5000$ at each iteration. The ideas about inexact cuts need not be restricted to continuous programs but may be extended to other areas where cutting-plane methods are employed. For instance, cutting-plane methods are used in integer and convex programing and these ideas may be adapted to stochastic integer programming and stochastic convex programming. This is the subject of current research.

Chapter 3

A Second-order Method for MPCCs

3.1 Introduction

We consider the mathematical program with complementarity constraints (MPCC)

MPCC	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$
	subject to	$c(x,y,w) = 0,$
		$0 \leq y \perp w \geq 0,$

in which f and c represent the objective function and a vector of constraints. Both are assumed to be twice continuously-differentiable over $x \in \mathbb{R}^n$ and y and $w \in \mathbb{R}^m$. The complementarity constraint $y \perp w$ implies that either y_i or w_i (or both) is (are) zero for $i = 1, \ldots, m$. Such problems arise in the modeling of Stackelberg equilibria, traffic equilibria, and contact problems [LPR96]. Mathematical programs with complementarity constraints (MPCC) are discussed in detail in the books by Luo et al. [LPR96] and Outrata et al. [OKZ98]. One may rewrite the complementarity constraint as $Yw \leq 0$, where $Y = \text{diag}(y)$. However, the resulting nonlinear program lacks a strict interior, implying that the Mangasarian-Fromovitz constraint qualification (MFCQ) fails to hold at every feasible point.

3.1.1 Early research on MPCCs

Over the past few years, significant effort has been applied towards solving MPCCs. Fletcher et al. [FLRS02] reformulate the complementarity constraint as $Yw \leq 0$ and report promising results using sequential quadratic programming (SQP) methods. Anitescu [Ani05] provides global convergence theory for SQP methods.

Hu and Ralph [HR04] and Leyffer et al. [LLCN05] solve a sequence of nonlinear programs with penalized complementarity constraints. The latter work also provides convergence theory for the method. Regularization methods require a solution of a sequence of *regularized* problems involving the relaxed constraints $Yw \leq t_k$ with $\{t_k\} \to 0$. These regularized problems may be solved by

interior methods or by SQP methods. Luo et al. [LPR96] discussed an interior method for MPCCs. Subsequently, Liu and Sun [LS04a] and Ragunathan and Biegler [RB05] provided interior methods under weaker assumptions. DeMiguel et al. [dMFNS05] discuss a two-sided relaxation scheme and provide local convergence theory for an interior method coupled with such a relaxation scheme.

Several other algorithms have been proposed to solve the original MPCC with differing local and global convergence properties, such as [JR00, Pie01], penalty-based methods [LPR96], and others.

3.1.2 A Second-Order Method

One of the first primal-dual methods to obtain convergence to second-order points was by Moguerza and Prieto [MP03]. They focused on constrained nonlinear programs and discussed the performance of the algorithm on a set of test problems. Global convergence theory of the algorithm is provided under the assumptions of strict complementarity at the limit points and the compactness of primal and dual iterates. Subsequently Doyle [Doy03] presented a method for large-scale nonlinear programming that also employed negative curvature directions. Local and global convergence of the algorithm is also discussed. The dissertation by Ng [Ng03] presents a second-order primal-dual method for solving a class of discrete nonlinear programs.

The global convergence theory in [Doy03] leans on the work by Murray and Prieto [MP95a], which appears to be one of the few SQP algorithms that uses exact second-derivatives while showing convergence to second-order points. In [MP03], the authors present a primal-dual method and show convergence to second-order points. However, they consider well-posed nonlinear programs and assume that primal and dual iterates are bounded . We weaken both assumptions in this paper. In particular, our method is designed for mathematical programs with complementarity constraints and we only assume compactness of the non-slack primal variables. The convergence theory in this chapter uses the work by Murray and Prieto [MP95a] as a guide but has obvious major differences.

We propose a line-search based interior method for solving MPCCs. The important difference between our method and earlier algorithms for MPCCs is that we ensure convergence to *second-order points*. Such points satisfy the second-order sufficiency conditions, implying that they are local minimizers in a reduced space. In §3.2 we present our algorithm, focusing on the formulation, the specification of the Newton and negative curvature directions, the linesearch and the curvilinear search.

Section 3.3 discusses the convergence properties of the algorithm. We assume that the primal iterates (not including the slack variables) generated by the algorithm are in a compact set. We prove that the dual and slack variables are therefore bounded. In addition, we prove that the iterates generated by the algorithm converge to a second-order KKT point of MPCC.

In §3.4, we discuss some computational results for the method on the `QPEC`gen test problem set.

3.1.3 Some Definitions and Background

We provide a quick survey of some of the important concepts for nonlinear programs, using the following nonlinear problem (NLP) as the basis of our discussions:

NLP	$\underset{x}{\text{minimize}}$	$f(x)$	
	subject to	$c(x) = 0:$	λ
		$x \geq 0:$	$\sigma.$

The Lagrangian function defined as $\mathcal{L}(x) := f(x) - \lambda^T c(x) - \sigma^T x$ allows for the specification of the first-order Karush-Kuhn-Tucker (KKT) conditions.

Definition 3.1. *First-order KKT Conditions: The triple $(x^*, \lambda^*, \sigma^*)$ is a first-order KKT point for NLP if*

$$\begin{aligned} \nabla_x \mathcal{L}(x^*, \lambda^*, \sigma^*) &= 0 \\ c(x^*) &= 0 \\ 0 \leq x^* \perp \sigma^* &\geq 0. \end{aligned}$$

The second-order sufficient conditions are defined next.

Definition 3.2. *Second-order sufficient conditions (SOSC): The KKT point $(x^*, \lambda^*, \sigma^*)$ satisfies SOSC for NLP if it satisfies the first-order KKT conditions and if*

$$p^T \nabla_{xx} \mathcal{L}(x^*, \lambda^*, \sigma^*) p > 0$$

for all nonzero p such that $\nabla_x c(x^*)p = 0$ and also

$$p_j = 0 \text{ if } x_j^* > 0 \text{ and } p_j \geq 0 \text{ if } x_j^* = 0.$$

The specification of necessary optimality conditions requires the point to subscribe to a regularity condition. One such condition is the linear independence constraint qualification (LICQ).

Definition 3.3. *(LICQ): A point (x, λ, σ) satisfies the linear independence constraint qualification for NLP if*

$$\begin{pmatrix} \nabla c(x) \\ I_{\mathcal{A}} \end{pmatrix}$$

has full row-rank, where $\mathcal{A} = \{j : x_j = 0\}$ and $I_{\mathcal{A}}$ refers to the columns of the identity corresponding to the index set $\mathcal{A}$.

Since MPCCs do not satisfy the LICQ, a different tack is needed in specifying the stationarity conditions: one may define a relaxed form of MPCC denoted by RNLP. Using a form adopted in

[dMFNS05], we define RNLP at a feasible point $\bar{z} = (\bar{x}, \bar{y}, \bar{w})$ as

RNLP	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$			
		$c(x,y,w) = 0$			
	subject to	$y_j = 0,$	$w_j \geq 0,$	$j \in \mathcal{I}^y(\bar{z})$	
		$y_j \geq 0,$	$w_j = 0,$	$j \in \mathcal{I}^w(\bar{z})$	
		$y_j \geq 0,$	$w_j \geq 0,$	$j \in \mathcal{I}^y(\bar{z}) \cap \mathcal{I}^w(\bar{z}).$	

The index sets $\mathcal{I}^y(\bar{z})$ and $\mathcal{I}^w(\bar{z})$ are defined as $\{j : \bar{y}_j = 0\}$ and $\{j : \bar{w}_j = 0\}$, respectively. The crucial difference between RNLP and MPCC is in the treatment of complementarity constraints. When $\bar{y}_j = 0 < \bar{w}_j$, the constraint $w_j y_j = 0$ is replaced by $y_j = 0$ and $w_j \geq 0$. However, when $\bar{y}_j = 0 = \bar{w}_j$, $y_j w_j = 0$ is replaced by $y_j, w_j \geq 0$. In all these instances, the gradients of the active constraints are linearly independent. The term *relaxation* emerges from the fact that the feasible region of MPCC is contained in the feasible region of RNLP (i.e. $\mathcal{F}_{MPCC} \subseteq \mathcal{F}_{RNLP}$). Furthermore, we may define an MPCC-LICQ as follows.

Definition 3.4 (MPCC-LICQ)**.** *The MPCC satisfies an MPCC-LICQ at a feasible point $\bar{z}$ if RNLP satisfies the LICQ at $\bar{z}$.*[1]

This allows us to define a strong stationary point.

Definition 3.5 (Strong Stationarity)**.** *A point $(x^*, y^*, w^*, \lambda^*, \pi^*)$ is said to be a strong stationary point of MPCC if it is a KKT point of RNLP. In particular, it satisfies the first-order conditions:*

$$\begin{pmatrix} \nabla_x f(z^*) \\ \nabla_w f(z^*) \\ \nabla_y f(z^*) \end{pmatrix} - \begin{pmatrix} \nabla_x c(z^*) \\ \nabla_w c(z^*) \\ \nabla_y c(z^*) \end{pmatrix}^T \pi^* - \begin{pmatrix} \lambda_w^* \\ \lambda_y^* \end{pmatrix} = 0, \tag{3.1}$$

$$[\lambda_w^*]_j, [\lambda_y^*]_j \geq 0, \quad i \in \mathcal{I}^y(z^*) \cap \mathcal{I}^w(z^*) \tag{3.2}$$

$$c(z^*) = 0, \tag{3.3}$$

$$w_j^* [\lambda_w^*]_j = 0, \tag{3.4}$$

$$y_j^* [\lambda_y^*]_j = 0, \tag{3.5}$$

$$z^* \in \mathcal{F}_{MPCC}. \tag{3.6}$$

If z^* satisfies MPCC-LICQ, then (λ^*, π^*) is unique.

Next, we discuss the different types of complementary slackness or complementarity conditions that may be posed.

Definition 3.6. *The point (z^*, λ^*, π^*) satisfies strict complementarity slackness or MPCC-SCS if (z^*, λ^*, π^*) is a strong stationary point and $[y^*]_j + [\lambda_y^*]_j \neq 0$ and $[w^*]_j + [\lambda_w^*]_j \neq 0$, for all j.*

[1] Note that we use z to denote the triple vector (x, y, w).

Definition 3.7. *The point (z^*, λ^*, π^*) satisfies weak strict complementarity slackness or MPCC-WSCS if it is a strong stationary point of MPCC.*[2]

Our interest is in the solution of MPCC using nonlinear programming techniques. We therefore use the following formulation, similar to that used in [RB05]. Problem MPCC is equivalent to ENLP:

ENLP		
	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$
	subject to	$c(x,y,w) = 0,$
		$Yw \leq 0,$
		$y \geq 0,$
		$w \geq 0.$

The nonlinear program ENLP is ill-posed in that the constraints do not satisfy the Mangasarian Fromovitz constraint qualification (MFCQ) as observed by the fact that the complementarity constraints do not have a strict interior at any feasible point. However, such regularity conditions are crucial in proving the convergence of nonlinear programming methods. The stationarity conditions of ENLP are

$$\begin{pmatrix} \nabla_x f(z^*) \\ \nabla_w f(z^*) \\ \nabla_y f(z^*) \end{pmatrix} - \begin{pmatrix} \nabla_x c(z^*) \\ \nabla_w c(z^*) \\ \nabla_y c(z^*) \end{pmatrix}^T \pi + \begin{pmatrix} \bar{Y} \\ \bar{W} \end{pmatrix} \lambda_{cc} - \begin{pmatrix} \lambda_w \\ \lambda_y \end{pmatrix} = 0,$$
$$[\lambda_w^*]_j \geq 0, \quad [w^*]_j [\lambda_w^*]_j = 0,$$
$$[\lambda_y^*]_j \geq 0, \quad [y^*]_j [\lambda_y^*]_j = 0,$$
$$c(z^*) = 0,$$
$$z^* \in \mathcal{F}_{MPCC}.$$

Since ENLP is equivalent to MPCC, the assumptions of MPCC-LICQ, strong stationarity and second-order sufficiency can be used. Moreover, our objective is then to obtain a strong-stationary point of ENLP by using a barrier method. An important distinguishing feature of this method compared to that proposed by [RB05] is the incorporation of a curvilinear search that ensures convergence to second-order points. We may then specify the set of NLP multipliers Λ^* associated with ENLP as

$$\Lambda^* := \{(\lambda_{cc}, \lambda_w, \lambda_y, \pi) : \pi = \pi^*, \tag{3.7}$$
$$\bar{Y}\lambda_{cc} - \lambda_w = -\lambda_w^*, \tag{3.8}$$
$$\bar{W}\lambda_{cc} - \lambda_y = -\lambda_y^*, \tag{3.9}$$
$$\lambda_{cc}, \lambda_y, \lambda_w \geq 0\}. \tag{3.10}$$

The uniqueness of λ^* based on MPCC-LICQ implies that Λ^* is unambiguously specified. In [RB05],

[2] Only requires strict complementarity for multipliers associated with inequality constraints not related to complementarity constraints.

the authors show that by appropriately bounding the sequence of multipliers, one may approach a multiplier in Λ^* satisfying strict complementarity. This is important from the standpoint of constructing an algorithm that converges to a KKT point satisfying strict complementarity.

The second-order sufficiency conditions for an MPCC are of particular importance given our focus on second-order points. We may define the tangent cone $\mathcal{T}_{MPCC}(x^*, y^*, w^*)$ associated with a feasible point of an MPCC:

$$\begin{aligned}\mathcal{T}_{MPCC}(x^*) = \{\alpha(\Delta x^*, \Delta y^*, \Delta w^*) : \nabla c(x^*, y^*, w^*)(\Delta x^*, \Delta y^*, \Delta w^*) = 0,\\ \Delta w_i^* \geq 0 \text{ for all } i \text{ such that } w_i^* = 0,\\ \Delta y_i^* \geq 0 \text{ for all } i \text{ such that } y_i^* = 0\}.\end{aligned}$$

We now define the strong second-order sufficiency condition for MPCC.

Definition 3.8. *(MPCC-SSOSC) The point $(x^*, y^*, w^*, \pi, \lambda^*)$ is said to satisfy MPCC-SSOSC if $\Delta u^T \nabla^2_{zz} \mathcal{L} \Delta u > 0$ with $\Delta u \neq 0$ and $\Delta u \in \mathcal{F}$, where $\mathcal{F}$ is defined as*

$$\begin{aligned}\mathcal{F} = \{\Delta u \in \mathcal{T}_{MPCC} : \Delta y_i = 0 \text{ for all } i \text{ such that } y_i^* = 0, \quad (\lambda_y^* \neq 0),\\ \Delta w_i = 0 \text{ for all } i \text{ such that } w_i^* = 0, \quad (\lambda_w^* \neq 0)\}.\end{aligned}$$

3.2 An Interior Point Method for MPCCs

We begin by stating the regularized problem MPCC(γ):

MPCC(γ)	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$	
		$c(x,y,w)$	$= 0$
		$Yw + s^{cc} - \gamma e$	$= 0$
	subject to	$y - s^y$	$= 0$
		$w - s^w$	$= 0$
		s^y, s^w, s^{cc}	$\geq 0.$

For $\gamma > 0$, problem MPCC(γ) is a well-posed nonlinear program; it satisfies the common constraint qualifications such as the Mangasarian-Fromovitz constraint qualification. Specifically, the positivity of γ ensures that the regularized complementarity constraint set

$$\{(y, w) \mid Yw \leq \gamma e,\, y, w \geq 0\}$$

has a nonempty interior.

The idea is then to solve a sequence of problems MPCC(γ_k) with $\gamma_k \to 0$. Under some assumptions, the sequence of stationary points of MPCC(γ_k) converges to the strong-stationary point or the stationary point of MPCC(0). By using a log-barrier function, we rid ourselves of the inequality

constraints and solve a sequence of regularized barrier problems. By defining $d(.)$ and s as

$$d(x,y,w,s;\gamma) = \begin{pmatrix} Yw + s^{cc} - \gamma e \\ y - s^y \\ w - s^w \end{pmatrix} \text{ and } s = \begin{pmatrix} s^y \\ s^w \\ s^{cc} \end{pmatrix}, \tag{3.11}$$

we obtain a more compact representation of MPCC(γ):

MPCC(γ)	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$
		$c(x,y,w) = 0$
	subject to	$d(x,y,w,s;\gamma) = 0$
		$s \geq 0.$

3.2.1 The Barrier Problem

The crucial idea in interior methods for optimization is that the inequality constraints and nonnegativity bounds are kept strictly satisfied. One such interior method uses the logarithmic function to replace inequality constraints by *barrier* terms in the objective. These terms tend to infinity if the iterates tend towards the boundary. The resulting problem is parameterized by γ and the barrier parameter μ:

MPCC(γ,μ)	$\underset{z,s}{\text{minimize}}$	$f(z) - \mu \sum_j \ln[s]_j$	
	subject to	$c(z) = 0$	$: \beta$
		$d(z,s;\gamma) = 0$	$: \xi.$

Note that β and ξ represent the Lagrange multipliers associated with constraints $c(\cdot)$ and $d(\cdot)$.

The proposed interior method solves problem MPCC by approximately solving a sequence of regularized barrier subproblems MPCC(γ,μ). For each subproblem, the KKT conditions are linearized to provide primal and dual search directions. If the reduced Hessian is not positive definite, it may be modified resulting in a modified Newton direction. In such an instance, we may also obtain directions of negative curvature. A stepsize along the modified Newton directions is obtained by conducting a linesearch using an augmented-Lagrangian merit function. Alternately, if we have a direction of sufficient negative curvature, we may conduct a curvilinear search along the merit function. From either search, we obtain a stepsize enabling us to take a step to the new iterate. Then the parameters γ and μ are reduced unless a suitable termination criterion is satisfied. Before commencing the linesearch or curvilinear search, we may modify the penalty parameter to obtain sufficient descent.

3.2.2 The KKT Conditions of MPCC(γ, μ)

We may define a Lagrangian function $\mathcal{L}$ for the regularized barrier problem by

$$\mathcal{L}(z, s, \beta, \xi; \mu, \gamma) = f(z) - \mu \sum_j \ln s_j - \beta^T c(z) - \xi^T d(z, s; \gamma). \tag{3.12}$$

The first-order conditions for MPCC(γ, μ) may be stated as $\nabla \mathcal{L} = 0$. Define the associated residuals as follows:

$$\begin{aligned} r^z &:= \nabla_z \mathcal{L} = 0 \\ r^s &:= \nabla_s \mathcal{L} = 0 \\ r^c &:= c(z) = 0 \\ r^d &:= d(z, s) = 0. \end{aligned} \tag{3.13}$$

Note that

$$\nabla_s \mathcal{L} = -v - \nabla_s c^T \beta - \nabla_s d^T \xi$$

is simplified by defining a new variable v, where $Sv = \mu e$ and $S = \text{diag}(s)$. Let the associated residual be $r^v = Sv - \mu e$.

3.2.3 The Linearized KKT Conditions

Newton's method applied to (3.13) plus $r^v = 0$ results in the system

$$M \Delta p = -r, \tag{3.14}$$

where M, r and Δp are given by

$$M = \begin{pmatrix} \nabla_{zz}\mathcal{L} & \nabla_{zs}\mathcal{L} & -\nabla_z c^T & -\nabla_z d^T & \\ \nabla_{sz}\mathcal{L} & \nabla_{ss}\mathcal{L} & & -\nabla_s d^T & -I \\ \nabla_z c & & & & \\ \nabla_z d & \nabla_s d & & & \\ & V & & & S \end{pmatrix}, \; r = \begin{pmatrix} r^z \\ r^s \\ r^c \\ r^d \\ r^v \end{pmatrix} \text{ and } \Delta p = \begin{pmatrix} \Delta z \\ \Delta s \\ \Delta \beta \\ \Delta \xi \\ \Delta v \end{pmatrix}, \tag{3.15}$$

where $V = \text{diag}(v)$. We may eliminate Δv in each scenario-based subsystem to obtain

$$\begin{pmatrix} \nabla^2_{zz}\mathcal{L} & \nabla^2_{zs}\mathcal{L} & -\nabla_z c^T & -\nabla_z d^T \\ \nabla^2_{sz}\mathcal{L} & \nabla^2_{ss}\mathcal{L} + S^{-1}V & & -\nabla_s d^T \\ \nabla_z c & & & \\ \nabla_z d & \nabla_s d & & \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta s \\ \Delta \beta \\ \Delta \xi \end{pmatrix} = - \begin{pmatrix} r^z \\ r^s + S^{-1} r^v \\ r^c \\ r^d \end{pmatrix}. \tag{3.16}$$

A more compact representation is

$$\begin{pmatrix} \mathcal{H} & -\nabla h^T \\ \nabla h & \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} r^u \\ r_h \end{pmatrix},$$

where

$$\mathcal{H} = \begin{pmatrix} \nabla^2_{zz}\mathcal{L} & \nabla^2_{zs}\mathcal{L} \\ \nabla^2_{sz}\mathcal{L} & \nabla^2_{ss}\mathcal{L} + S^{-1}V \end{pmatrix}, \nabla h = \begin{pmatrix} \nabla_z c & \\ \nabla_z d & \nabla_s d \end{pmatrix}, r^h = \begin{pmatrix} r^c \\ r^d \end{pmatrix}, \tag{3.17}$$

$$\Delta\lambda = \begin{pmatrix} \Delta\beta \\ \Delta\xi \end{pmatrix}, \Delta u = \begin{pmatrix} \Delta z \\ \Delta s \end{pmatrix}, \text{ and } r^u = \begin{pmatrix} r^z \\ r^s + V^{-1}r^v \end{pmatrix}. \tag{3.18}$$

The search direction for v may be obtained by solving $S\Delta v = (r^v - V\Delta s)$. The system

$$\begin{pmatrix} \mathcal{H} & -\nabla h^T \\ \nabla h & \end{pmatrix} \tag{3.19}$$

is often called a KKT system and its invertibility is linked intimately to the rank of the Jacobian matrix ∇h and the invertibility of the reduced Hessian of the Lagrangian $\mathbf{Z}^T\mathcal{H}\mathbf{Z}$, where $\mathbf{Z}$ is a basis for the null-space of ∇h.

We may compute the primal and dual search directions efficiently by using the ideas from [Doy03]. Such a computation requires modifying the reduced Hessian $\mathbf{Z}^T\mathcal{H}\mathbf{Z}$ to ensure that it is positive definite. This may be achieved by computing the modified Cholesky factorization [GMW81] of $\mathbf{Z}^T\mathcal{H}\mathbf{Z}$. Such a factorization adds a sufficiently large nonnegative diagonal matrix E to $\mathbf{Z}^T\mathcal{H}\mathbf{Z}$ to ensure that $\mathbf{Z}^T\mathcal{H}\mathbf{Z} + E$ is positive definite. An important by-product of this factorization is that with a little more effort, one may obtain a direction of negative curvature.

3.2.4 A Negative Curvature Direction

Our intention is to compute points that satisfy first and second-order necessary conditions. To this end, it is necessary to use directions of negative curvature. In the case of unconstrained optimization, such directions Δp^c would only exist when the Hessian H is indefinite and would satisfy

$$(\Delta p^c)^T H \Delta p^c < 0. \tag{3.20}$$

A motivation for negative curvature directions (see [MS79, FGM95, MP95a]) may be seen from noticing that whenever the current iterate is a first-order KKT point (but not a minimizer), then the Newton direction would be zero (because the right-hand side of the linearized KKT system would be zero). Therefore, the algorithm would converge at a local saddle-point or maximizer. To

move away from such a point, we need a direction of negative curvature. Consider the example

E	$\underset{x,y}{\text{minimize}}$	$(x^3 - xy + y^3)$
	subject to	$0 \leq y + 2 \perp x - y \geq 0.$

The objective function has a saddle point at $(0,0)$ and a minimizer at $(\frac{1}{3}, \frac{1}{3})$. Both points are feasible but only $(\frac{1}{3}, \frac{1}{3})$ is a second-order point. This follows from noting that the reduced Hessian is indefinite at $(0,0)$. Since $y - x = 0$ at $(0,0)$, we have

$$\begin{aligned}\mathbf{Z}^T H \mathbf{Z} &= \begin{pmatrix}1\\1\end{pmatrix}^T \begin{pmatrix}6x & -1\\-1 & 6y\end{pmatrix} \begin{pmatrix}1\\1\end{pmatrix} \\ &= 6(x+y) - 2 \\ &= -2 \text{ at } (0,0).\end{aligned}$$

The negative curvature direction at $(0,0)$ is given by a Δp^c such that $-2(\Delta p^c)^2 < 0$. If we choose $\Delta p^c = \frac{1}{3}$, then the new iterate $(\widehat{x}, \widehat{y})$ is given by

$$\begin{pmatrix}\widehat{x}\\\widehat{y}\end{pmatrix} = \begin{pmatrix}x\\y\end{pmatrix} + \frac{1}{3}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}x^*\\y^*\end{pmatrix}$$

and the negative curvature direction leads to the local minimizer $(\frac{1}{3}, \frac{1}{3})$. (In this particular case, we know the minimizer so we can take an appropriate step size.)

A different approach may be followed by combining the Newton and negative curvature directions (see Murray and Prieto [MP95a], Moguerza and Prieto [MP03] and Doyle [Doy03]). In such an approach, the variables p_k are changed according to

$$p_{k+1} = p_k + \alpha_k^2 \Delta p_k + \alpha_k \Delta p_k^c, \tag{3.21}$$

where α_k is the steplength. The choice of an α_k is through a *curvilinear* search. The computation of Δp^c is based on the modified Cholesky factorization:

$$\mathbf{Z}^T H \mathbf{Z} + E = R^T R, \tag{3.22}$$

where R is a nonsingular upper triangular matrix and E is a nonnegative diagonal matrix. If $i = \arg\min\{H_{ii}\}$, then

$$R \Delta p_z^c = e_i. \tag{3.23}$$

However, the vector Δp_z^c is in a reduced space and Δp^c may be obtained as $\Delta p^c = \mathbf{Z} \Delta p_z^c$. Directions of negative curvature do not have a natural size associated with them and we may scale the direction so that $\|\Delta p^c\|_2 = \|\Delta p\|_2$.

In defining the direction of negative curvature, one question that has been left open is what

one should use for the multiplier estimate in the construction of the Hessian of the Lagrangian. In [MP95a], the authors use a (β'_k, ξ'_k) given by

$$\begin{pmatrix} \beta'_k \\ \xi'_k \end{pmatrix} = \begin{pmatrix} \beta_k \\ \xi_k \end{pmatrix} - \rho_{k-1} \begin{pmatrix} c_k \\ d_k \end{pmatrix}. \tag{3.24}$$

If ρ_k is assumed to be bounded, then for a sequence converging to a first-order KKT point,

$$\left\| \begin{pmatrix} \beta'_k - \beta_k \\ \xi'_k - \xi_k \end{pmatrix} \right\| \to 0, \text{ as } k \to \infty.$$

We shall use the same estimate in our construction of Δp^c.

3.2.5 Globalization through a Curvilinear Search

To ensure convergence from an arbitrary starting point, algorithms for general nonlinear programs are equipped with a *globalization* strategy. Such a strategy requires the specification of a merit function that represents a metric of progress for the optimization algorithm. For instance, in an unconstrained optimization problem, an appropriate merit function is simply the function value. In nonlinearly constrained problems, a suitable merit function is given by a measure of primal and dual infeasibilities. In particular, we use the augmented Lagrangian merit function. In the current setting, a merit function for MPCC(γ, μ) may be defined at an iterate (z, s, β, ξ) as

$$\mathcal{M}(z, s, \beta, \xi; \rho, \mu, \gamma) = \mathcal{L}(z, s, \beta, \xi; \mu, \gamma) + \frac{1}{2}\rho(\|c\|^2 + \|d\|_2^2). \tag{3.25}$$

Linesearch Conditions

Linesearch methods are well studied and form part of the globalization strategies in several NLP algorithms such as SNOPT [GMS05] and LOQO [VS99]. A backtracking linesearch seeks an $\widehat{\alpha} \in (0, 1]$ satisfying

$$\begin{aligned} \psi_L(\widehat{\alpha}) &\leq \psi_L(0) + \gamma_1 \widehat{\alpha} \psi'_L(0), \\ |\psi'_L(\widehat{\alpha})| &\leq \gamma_2 |\psi'_L(0)|, \end{aligned} \tag{3.26}$$

where

$$\psi_L(\alpha) := \mathcal{M}(z + \alpha\Delta z, s + \alpha\Delta s, \beta + \alpha\Delta\beta, \xi + \alpha\Delta\xi).$$

We also use the notation $\mathcal{M}(\alpha)$ to mean $\mathcal{M}(z + \alpha\Delta z, s + \alpha\Delta s, \beta + \alpha\Delta\beta, \xi + \alpha\Delta\xi)$. If we have $0 < \gamma_2 \leq \gamma_1 < 1$, we may be assured of the existence of such an α [GMW81]. However, this is predicated on the fact that the computed direction is a direction of descent for the merit function (3.25). If the original direction is not one of descent, it becomes one for a sufficiently large ρ, as we show later.

Curvilinear Search Conditions

We now discuss the curvilinear search, proceeding in a fashion similar to that of Doyle [Doy03]. Let the curvilinear search conditions be

$$\begin{aligned} z(\alpha) &:= z + \alpha^2 \Delta z + \alpha \Delta z^c \\ s(\alpha) &:= s + \alpha^2 \Delta s + \alpha \Delta s^c \\ \beta(\alpha) &:= \beta + \alpha^2 \Delta \beta \\ \xi(\alpha) &:= \xi + \alpha^2 \Delta \xi \end{aligned} \tag{3.27}$$

and let $\psi_C(\alpha)$ be defined as

$$\psi_C(\alpha) := \mathcal{M}(z + \alpha \Delta z^c + \alpha^2 \Delta z, s + \alpha \Delta s_c + \alpha^2 \Delta s, \beta + \alpha^2 \Delta \beta, \xi + \alpha^2 \Delta \xi).$$

There are several considerations in obtaining the stepsize α.

1. If one obtains a nonzero direction of negative curvature, it may be that no *linear* combination of Δp and Δp^c gives a direction of descent. If we have sufficient negative curvature, then the merit function may be reduced by conducting a curvilinear search.
2. If $\psi_C'(0) = \nabla_z \mathcal{M}(0)^T \Delta z^c + \nabla_s \mathcal{M}(0)^T \Delta s^c < 0$, then a step along the direction of negative curvature is indeed a direction of descent. If not, we use $-\Delta z^c$ and $-\Delta s^c$ instead.
3. If the obtained Δp^c is not a direction of sufficient negative curvature, we drop the direction and use the modified Newton direction. Moreover, we are guaranteed that $\psi_C''(0) < 0$ with a sufficiently large penalty parameter.
4. If the obtained direction is one of sufficient negative curvature, then we may conduct a curvilinear search as described next.
5. We define an $\alpha_{\max}$ by ensuring that s stays nonnegative. Such an $\alpha_{\max}$ is given by

$$\alpha_{\max} := \min_i \{\arg \min_{\alpha_i > 0} m_i(\alpha_i)\}, \text{ where } m_i(\alpha_i) = s_i + \alpha_i^2 [\Delta s]_i + \alpha_i [\Delta s^c]_i \geq 0.$$

6. Then by backtracking from $\alpha_{\max}$, we obtain an $\widehat{\alpha}$ that maintains $s > 0$ and satisfies

$$\begin{aligned} \psi_C(\widehat{\alpha}) &\leq \psi_C(0) + \gamma_1 (\widehat{\alpha} \psi_C'(0) + \frac{\widehat{\alpha}^2}{2} \psi_C''(0)) \\ \psi_C'(\widehat{\alpha}) &\geq \gamma_2 (\psi_C'(0) + \widehat{\alpha} \psi_C''(0)), \end{aligned} \tag{3.28}$$

   where $0 \leq \gamma_2 \leq \frac{1}{2} \leq \gamma_1 \leq 1$.

To summarize, we obtain modified Newton and negative curvature directions. We immediately obtain $\psi_C'(0) < 0$ because the sign of Δp^c may be switched. If the negative curvature direction does not result in sufficient negative curvature, we drop the direction and by raising the penalty parameter sufficiently, obtain $\psi_C''(0) < 0$. If Δp_k^c results in sufficient negative curvature, then a curvilinear search provides us with an improvement in the merit function.

3.2.6 Termination Criteria

Our method solves the barrier problems inexactly. For given values of γ and μ, the algorithm for solving the barrier problem is terminated when the following conditions are satisfied:

$$\begin{aligned} \|\nabla_z \mathcal{L}(z_k, s_k, \beta_k, \xi_k, v_k)\|_\infty &\leq \epsilon_k \\ \|\nabla_s \mathcal{L}(z_k, s_k, \beta_k, \xi_k, v_k)\|_\infty &\leq \epsilon_k \\ \|c(z_k, s_k, \beta_k, \xi_k, v_k)\|_\infty &\leq \epsilon_k \\ \|d(z_k, s_k, \beta_k, \xi_k, v_k; \gamma_k)\|_\infty &\leq \epsilon_k \\ \|S_k v_k - \mu_k e\|_\infty &\leq \epsilon_k \\ -\lambda_{\min}(\mathbf{Z}^T \nabla_z^2 \mathcal{L} \mathbf{Z}) &< \epsilon_k, \end{aligned} \tag{3.29}$$

where $\epsilon_k = \min(\gamma_k, \mu_k)$. In the following summary the primal and dual variables are denoted by p_k. Algorithm 1 provides a brief description of the steps of our method.

Algorithm 1: A second-order method for MPCCs

Initial: $(p_0; \mu_0, \rho_0, \gamma_0)$
$k = 1$
while $\epsilon_k > \epsilon$ **do**
  while $\|r_k\|_\infty \geq \epsilon_k$ **or** $-\lambda_{min}(\mathbf{Z}_k^T \mathcal{H}_k \mathbf{Z}_k) \geq \epsilon_k$ **do**
    Solve (3.15) to obtain $\Delta p_k = (\Delta z, \Delta s, \Delta \beta, \Delta \xi, \Delta v)$
    Solve (3.23) to obtain $\Delta p_k^c = (\Delta z^c, \Delta s^c)$
    Raise ρ_k to ensure sufficient descent
    Obtain α_k satisfying (3.28)
    Take step based on (3.27)
  $\gamma_{k+1} = u_k^\gamma \gamma_k$
  $\mu_{k+1} = u_k^\mu \mu_k$
  $k := k + 1$

3.3 Global Convergence to Second-Order Points

Our algorithm generates an infinite sequence $\{(z_k, s_k, \beta_k, \xi_k)\}$. In this section, we prove that the sequence converges to a second-order point $(z^*, s^*, \beta^*, \xi^*)$. In addition, the algorithm relies on barrier, penalty, and regularization parameter sequences for different tasks:

1. The barrier parameter sequence $\{\mu_k\}$ ensures that the primal iterates strictly satisfy inequality constraints and convert our inequality constrained problem to an equality constrained problem. Consequently, the stationarity conditions are expressible as a system of equations.

2. The regularization parameter sequence $\{\gamma_k\}$ ensures well-posedness at each step.

3. The penalty parameter sequence $\{\rho_k\}$ ensures that the combination of the Newton and negative curvature direction is one of descent with respect to the augmented Lagrangian merit function.

The next subsections prove several important intermediate results that are then invoked to prove our main result:

1. Existence of iterates: This set of results ensures that the primal and dual iterates are well-defined by showing that the Newton and negative curvature directions are always computable.

2. Existence of α_k and ρ_k: Given search directions, a curvilinear search is carried out to determine an acceptable steplength α. Moreover, by raising the penalty parameter to a finite value, we ensure that the directions obtained are indeed descent directions with regard to the merit function.

3. Boundedness of iterates: In [MP95a], the authors prove that the primal iterates generated by the algorithm lie in a compact set. In [MP03], the authors assume that both primal and dual iterates lie in a compact set. We take a middle ground in assuming that only the z_k are in a compact set. But, we prove that the slack variables s_k and the dual variables β_k and ξ_k are also bounded.

4. Sequence of search directions: We state some results concerning the sequence of search directions, in order to prove some properties of the steplength sequence.

5. Boundedness of steplengths: We prove that the the sequence $\{\alpha_k\}$ is bounded away from zero.

6. Global convergence results: We prove that the primal and dual feasibility conditions are satisfied in the limit, thus showing that the iterates converge to a first-order KKT point. Then, we invoke some results regarding the limit points of the negative curvature directions to claim that the iterates converge to a second-order point.

We begin with a statement of the assumptions.

Assumption 3.9.

A1. The subproblem termination criteria are satisfied in a finite number of iterations.

A2. The objective and constraint functions are thrice continuously-differentiable. Moreover, the functions and their first three derivatives are uniformly bounded on any compact set.

A3. Problem MPCC satisfies the MPCC-LICQ at any first-order KKT point.

A4. Strict complementarity holds at all KKT points of MPCC.

A5. The reduced Hessian of the Lagrangian is nonsingular at all first-order KKT points of MPCC.

A6. The primal iterates given by $\{z_k\} = \{(x_k, y_k, w_k)\}$ are bounded.

3.3.1 Existence of Iterates

In this section, we show that the primal and dual iterates are always computable. This requires showing that the Newton direction and negative curvature direction may always be obtained. Note that if the reduced Hessian of the Lagrangian is positive definite, then the negative curvature direction is zero.

The primal and dual iterates are obtained by solving the following linear system.

$$\begin{pmatrix} \nabla^2_{zz}\mathcal{L} & \nabla^2_{zs}\mathcal{L} & -\nabla_z c^T & -\nabla_z d^T \\ \nabla^2_{sz}\mathcal{L} & \nabla^2_{ss}\mathcal{L} + S^{-1}V & & -\nabla_s d^T \\ \nabla_z c & & & \\ \nabla_z d(\gamma) & \nabla_s d(\gamma) & & \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta s \\ \Delta\beta \\ \Delta\xi \end{pmatrix} = - \begin{pmatrix} r^z \\ r^s + S^{-1}r^v(\mu) \\ r^c \\ r^d \end{pmatrix}.$$

We also show where the dependence on γ and μ appears in the specification of the linear system. Showing that the system is nonsingular requires proving a result concerning the barrier problem MPCC(γ) for $\gamma > 0$.

Lemma 3.10. *If MPCC satisfies MPCC-LICQ at every feasible point of MPCC and $\nabla c(x)$ is of full row rank at every iterate, then MPCC(γ) for $\gamma > 0$ always satisfies LICQ at every feasible point $\bar{z} = (\bar{x}, \bar{y}, \bar{w})$ of MPCC(γ), where $\bar{y}$ and $\bar{w}$ are strictly positive.*

Proof. By MPCC-LICQ, $\nabla_z c(\bar{z})$ has full row rank. If the bounds on w and γ are strictly inactive, the only other active constraints are given by $d(\gamma)$, whose constraint normals are independent of $\nabla_z c(\bar{z})$. This can be concluded from the definition of $d(\gamma)$:

$$d(\gamma) = \begin{pmatrix} Yw + s_{cc} = \gamma e \\ y - s_y = 0 \\ w - s_w = 0 \end{pmatrix}.$$

■

The variables w and y are kept strictly positive by keeping the barrier parameter μ strictly positive (which in turn ensures that the slack variables s_w and s_y stay strictly positive).[3] The existence of the primal and dual iterates is ensured by the positive definiteness of the modified reduced Hessian of the Lagrangian $\mathbf{Z}^T\mathcal{H}\mathbf{Z}$, where $\mathbf{Z}$ is a basis for the null-space for ∇h. This modified M (see (3.15)) is denoted by $\bar{M}$:

$$\bar{M} := \begin{pmatrix} \bar{\mathcal{H}} & -\nabla h^T \\ \nabla h & \end{pmatrix}.$$

The following result proves that $\bar{M}$ is nonsingular.

Lemma 3.11. *Given a feasible point $\bar{z}$ of MPCC(γ, μ) with γ and $\mu > 0$. If $\bar{\mathcal{H}}$ satisfies $\mathbf{Z}^T\bar{\mathcal{H}}\mathbf{Z} \succ 0$, then $\bar{M}$ is nonsingular.*

[3] Here we assume that $y - s_y = 0$ and $w - s_w = 0$ are satisfied at every iteration.

Proof. The result follows from ∇h having full row rank and from lemma 16.1 from [NW99]. The full rankedness of ∇h follows from Lemma 3.10. ■

The following theorem formalizes the existence of primal and dual iterates at every feasible point of MPCC(γ, μ) with γ and μ positive.

Theorem 3.12. *By the nonsingularity of $\bar{M}$, the primal directions Δz and Δs and the dual directions $\Delta\beta$ and $\Delta\xi$ are well-defined. Furthermore, the negative curvature direction exists if $\mathbf{Z}^T\mathcal{H}\mathbf{Z} \not\succeq 0$ and is zero otherwise.*

3.3.2 Existence of Parameter Sequences

In this subsection, we show that the penalty parameter ρ and the steplength α are always well-defined. Our algorithm relies on a curvilinear search when the reduced Hessian of the Lagrangian is indefinite. Such a search combines a Newton direction (using a modified reduced Hessian) and a negative curvature direction. We also discuss the option of using only the Newton direction (termed a linear search). We begin by recalling the definitions

$$\begin{aligned} \psi_L(\alpha) &:= \mathcal{M}(z + \alpha\Delta z, s + \alpha\Delta s, \beta + \alpha\Delta\beta, \xi + \alpha\Delta\xi) \\ \psi_C(\alpha) &:= \mathcal{M}(z + \alpha\Delta z^c + \alpha^2\Delta z, s + \alpha\Delta s^c + \alpha^2\Delta s, \beta + \alpha^2\Delta\beta, \xi + \alpha^2\Delta\xi). \end{aligned} \tag{3.30}$$

The gradients of these functions are given by

$$\begin{aligned} \psi_C'(\alpha) &= \nabla\mathcal{M}(\alpha)^T \begin{pmatrix} \Delta z^c + 2\alpha\Delta z \\ \Delta s^c + 2\alpha\Delta s \\ 2\alpha\Delta\beta \\ 2\alpha\Delta\xi \end{pmatrix} \\ \implies \psi_C'(0) &= \nabla_z\mathcal{M}(0)^T\Delta z^c + \nabla_s\mathcal{M}(0)^T\Delta s^c, \end{aligned}$$

and

$$\begin{aligned} \psi_L'(\alpha) &= \nabla\mathcal{M}(\alpha)^T \begin{pmatrix} \Delta z \\ \Delta s \\ \Delta\beta \\ \Delta\xi \end{pmatrix} \\ \implies \psi_L'(0) &= \nabla_z\mathcal{M}(0)^T\Delta z + \nabla_s\mathcal{M}(0)^T\Delta s + \nabla_\beta\mathcal{M}(0)^T\Delta\beta + \nabla_\xi\mathcal{M}(0)^T\Delta\xi. \end{aligned}$$

Lemma 3.13. *The penalty parameter ρ_k is well-defined and finite for a linear search.*

Proof. The penalty parameter is only increased whenever the modified Newton direction is one of ascent with regard to the merit function. In this lemma, we prove that the penalty parameter is

raised to a $\bar{\rho}$ that is finite ($< \infty$). The gradient of the merit function is given by

$$\begin{aligned}
\nabla_z \mathcal{M} &= \nabla_z \mathcal{L} + \rho \nabla_z c^T c + \rho \nabla_z d^T d \\
\nabla_s \mathcal{M} &= \nabla_s \mathcal{L} + \rho \nabla_s c^T c + \rho \nabla_s d^T d \\
\nabla_\beta \mathcal{M} &= -c \\
\nabla_\xi \mathcal{M} &= -d.
\end{aligned} \tag{3.31}$$

We now consider the expression for $\psi_L'(0)$ based on (3.30):

$$\begin{aligned}
\psi_L'(0) &= \nabla_z \mathcal{M}^T \Delta z + \nabla_s \mathcal{M}^T \Delta s + \nabla_\beta \mathcal{M}^T \Delta\beta + \nabla_\xi \mathcal{M}^T \Delta\xi \\
&= \nabla_z \mathcal{L}^T \Delta z - \rho c^T \nabla_z c \Delta z - \rho d^T \nabla_z d \Delta z + \nabla_s \mathcal{L}^T \Delta s - \rho d^T \nabla_s d \Delta s - c^T \Delta\beta - d^T \Delta\xi \\
&= -\Delta z^T \nabla^2_{zz} \mathcal{L} \Delta z - \Delta s^T \nabla^2_{zs} \mathcal{L} \Delta z - \Delta s^T \nabla^2_{sz} \mathcal{L} \Delta z - \Delta s^T \nabla^2_{ss} \mathcal{L} \Delta s - \rho(\|c\|^2 + \|d\|^2) \\
&\quad + \Delta z^T \nabla c^T \Delta\beta + \Delta z^T \nabla d^T \Delta\xi + \Delta s^T \nabla d^T \Delta\xi - c^T(\beta + \Delta\beta) - d^T(\xi + \Delta\xi) \\
&= -\Delta p^T \bar{\mathcal{H}} \Delta p - \rho(\|c\|^2 + \|d\|^2) - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi) \\
&= \epsilon - \rho(\|c\|^2 + \|d\|^2),
\end{aligned}$$

where

$$\epsilon := -\Delta p^T \bar{\mathcal{H}} \Delta p - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi).$$

Then the negativity of $\psi_L'(0)$ can be ensured as the next three points show:

1. If the current iterate is feasible, then c and d are zero and $\epsilon < 0$ since $\Delta p^T \bar{\mathcal{H}} \Delta p > 0$. Therefore $\psi_L'(0) < 0$.
2. If the current iterate is infeasible and $\epsilon \leq 0$, then $\psi_L'(0) < 0$.
3. If the current iterate is infeasible and $\epsilon > 0$, then $\psi_L'(0) < 0$ for

$$\rho > \bar{\rho} := \frac{\epsilon}{\|c\|^2 + \|d\|^2}.$$

■

Lemma 3.14. *The penalty parameter ρ_k is well-defined and finite for a curvilinear search.*

Proof. When considering the curvilinear search, we need the first and second derivatives of ψ_C. The first derivative $\psi_C'(0)$ is given by

$$\begin{aligned}
\psi_C'(\alpha) &= \nabla \mathcal{M}(\alpha)^T \begin{pmatrix} \Delta z^c + 2\alpha \Delta z \\ \Delta s^c + 2\alpha \Delta s \\ 2\alpha \Delta\beta \\ 2\alpha \Delta\xi \end{pmatrix} \\
\implies \psi_C'(0) &= \nabla_z \mathcal{M}(0)^T \Delta z^c + \nabla_s \mathcal{M}(0)^T \Delta s^c.
\end{aligned}$$

Then if $\psi'_C > 0$, we merely need to replace Δz^c and Δs^c by $-\Delta z^c$ and $-\Delta s^c$. This does not alter the fact that these are negative curvature directions. Ensuring that $\psi''_C(0) < 0$ is a little more challenging. We define the second derivative of $\psi_c(\alpha)$ as follows:

$$\begin{aligned}
\psi''_C(\alpha) &= 2\nabla\mathcal{M}(\alpha)^T \begin{pmatrix} \Delta z \\ \Delta s \\ \Delta\beta \\ \Delta\xi \end{pmatrix} + \begin{pmatrix} \Delta z^c + 2\alpha\Delta z \\ \Delta s^c + 2\alpha\Delta s \\ 2\alpha\Delta\beta \\ 2\alpha\Delta\xi \end{pmatrix}^T \nabla^2\mathcal{M}(\alpha) \begin{pmatrix} \Delta z^c + 2\alpha\Delta z \\ \Delta s^c + 2\alpha\Delta s \\ 2\alpha\Delta\beta \\ 2\alpha\Delta\xi \end{pmatrix} \\
\implies \psi''_C(0) &= 2\nabla\mathcal{M}(0)^T \begin{pmatrix} \Delta z \\ \Delta s \\ \Delta\beta \\ \Delta\xi \end{pmatrix} + \begin{pmatrix} \Delta z^c \\ \Delta s^c \\ 0 \\ 0 \end{pmatrix}^T \nabla^2\mathcal{M}(0) \begin{pmatrix} \Delta z^c \\ \Delta s^c \\ 0 \\ 0 \end{pmatrix} \\
&= 2\nabla\mathcal{M}(0)^T \begin{pmatrix} \Delta p \\ \Delta\beta \\ \Delta\xi \end{pmatrix} + (\Delta p^c)^T \nabla^2_{pp}\mathcal{M}(0)\Delta p^c.
\end{aligned}$$

The first and second derivatives of $\mathcal{M}$ are

$$\begin{aligned}
\nabla\mathcal{M}(0) &= \begin{pmatrix} \nabla_p\mathcal{L} + \rho\nabla_p c^T c + \rho\nabla_p d^T d \\ -c \\ -d \end{pmatrix} \\
\text{and } \nabla^2\mathcal{M}(0) &= \begin{pmatrix} \nabla^2_{pp}\mathcal{L} + \rho(\nabla_p c^T\nabla_p c + \nabla_p d^T\nabla_p d) & -\nabla_p c^T & -\nabla_p d^T \\ \nabla_p c & & \\ \nabla_p d & & \end{pmatrix}.
\end{aligned}$$

Therefore,

$$\psi''_C(0) = (\Delta p^c)^T\nabla^2_{pp}\mathcal{M}\Delta p^c + 2(\nabla_p\mathcal{M})^T\Delta p + 2(\nabla_\beta\mathcal{M})^T\Delta\beta + 2\nabla_\xi\mathcal{M}^T\Delta\xi. \tag{3.32}$$

If $f_B(z,s) = f(z) - \mu\sum_j \ln s_j$, we may write the first term as follows:

$$\begin{aligned}
(\Delta p^c)^T\nabla^2_{pp}\mathcal{M}\Delta p^c &= (\Delta p^c)^T \left(\nabla^2 f_B - \sum_i \lambda_i\nabla^2 h_i + \rho\nabla h^T\nabla h + \rho\sum_i h_i\nabla^2 h_i\right)\Delta p^c \\
&= (\Delta p^c)^T \left(\nabla^2 f_B - \sum_i (\lambda_i - \rho h_i)\nabla^2 h_i + \rho\nabla h^T\nabla h\right)\Delta p^c \\
&= (\Delta p^c)^T \left(\mathcal{H} + \rho\nabla h^T\nabla h\right)\Delta p^c \\
&= (\Delta p^c)^T\mathcal{H}\Delta p^c + \rho\Delta p_z^c\mathbf{Z}^T\nabla h^T\nabla h\mathbf{Z}\Delta p_z^c \\
&= (\Delta p^c)^T\mathcal{H}\Delta p^c.
\end{aligned}$$

Therefore $\psi''_C(0)$ may be written as

$$\begin{aligned}\psi''_C(0) &= (\Delta p^c)^T \mathcal{H} \Delta p^c + \rho(\|\bar{c}\|^2 + \|\bar{d}\|^2) + 2g_B^T \Delta p \\ &\quad - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi) - 2\rho(\|c\|^2 + \|d\|^2) \\ &= (\Delta p^c)^T \mathcal{H} \Delta p^c) + 2g_B^T \Delta p - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi) - \rho(\|c\|^2 + \|d\|^2), \end{aligned} \tag{3.33}$$

where $g_B = \nabla_p(f(z) - \mu \sum_j \ln s_j)$. It is interesting to note that the specification of $\psi''_C(0)$ has a very similar structure to a similar result for second-order methods for SQP (see equation (3.4) in [MP95a]).

There are two possibilities to consider:

1. The current iterate is infeasible, implying that $\|d\| + \|c\| > 0$. Then we may define ϵ as

$$\epsilon := (\Delta p^c)^T \mathcal{H} \Delta p^c + 2g_B^T \Delta p - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi).$$

   Then the following specification of ρ ensures that $\psi''_C(0) < 0$:

$$\rho \geq \max(\bar{\rho}, 0), \quad \text{where } \bar{\rho} := \frac{\epsilon}{\|c\|^2 + \|d\|^2}.$$

2. The current iterate is feasible, implying that $\|d\| + \|c\| = 0$. We then have

$$\psi''_C(0) = (\Delta p^c)^T \mathcal{H} \Delta p^c + 2g_B^T \Delta p.$$

   The negativity of $\psi''_C(0)$ is ensured by making Δp a direction of descent for f_B.

■

In general, if the penalty parameter is updated to ensure sufficient descent, then the negative curvature direction may not be valid any longer and is set to zero. The previous result required a specification of ρ that ensured $\psi''_C(0)$ was negative but perhaps not sufficiently negative. Next, we strengthen this result by requiring that $\psi''_C(0)$ is *sufficiently* negative.

Lemma 3.15. *The minimum penalty parameter to ensure that $(\Delta p^c, \Delta p, \Delta\beta, \Delta\xi)$ is a direction of descent for the augmented Lagrangian merit function is given by $\bar{\rho}$:*

$$\bar{\rho} = \frac{\bar{w} + 2g_B^T \Delta p - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi)}{\|c\|^2 + \|d\|^2}. \tag{3.34}$$

Proof. From (3.33), we have

$$\psi''_C(0) = (\Delta p^c)^T \mathcal{H} \Delta p^c + 2g_B^T \Delta p - 2c^T(\beta + \Delta\beta) - 2d^T(\xi + \Delta\xi) - \rho(\|c\|^2 + \|d\|^2).$$

If ϖ is defined as

$$\varpi := \tfrac{1}{2} \Delta p^T \bar{\mathcal{H}} \Delta p + \|d\|^2 + \|c\|^2, \tag{3.35}$$

then at a feasible iterate, $\varpi = \frac{1}{2}\Delta p^T \bar{\mathcal{H}} \Delta p$. We may use this in constructing an upper bound for $\psi_C''(0)$ at a feasible iterate:

$$\begin{aligned}\psi_C''(0) &= (\Delta p^c)^T \mathcal{H} \Delta p^c + 2 g_B^T \Delta p \\ &\leq (\Delta p^c)^T \mathcal{H} \Delta p^c - \Delta p^T \mathcal{H} \Delta p \\ &= (\Delta p^c)^T \mathcal{H} \Delta p^c - 2\varpi.\end{aligned}$$

Therefore at a feasible iterate, we have that Δp does not have any components in the range space of ∇h and may be written as $\Delta p = \mathbf{Z} \Delta p_z$. Therefore $\varpi > 0$ because $\mathbf{Z}^T \mathcal{H} \mathbf{Z}$ has been modified to be positive definite and the penalty parameter need not be modified.

If $\|c\|^2 + \|d\|^2 > 0$, we set $\Delta p^c = 0$ and $\psi_C''(0) = -\varpi$ if

$$\bar{\rho} = \frac{\varpi + 2 g_B^T \Delta p - 2 c^T (\beta + \Delta\beta) - 2 d^T (\xi + \Delta\xi)}{\|c\|^2 + \|d\|^2}.$$

This implies that for $\rho \geq \bar{\rho}$, we have $\psi_C''(0) \leq -\varpi$. ■

Next, we prove that the steplength α is always well-defined.

Lemma 3.16. *The linesearch and curvilinear search conditions for an acceptable steplength $\widehat{\alpha}$ are*

$$\begin{aligned}\psi(\widehat{\alpha}) &\leq \psi(0) + \gamma_1 \widehat{\alpha} \psi'(0), \\ |\psi'(\widehat{\alpha})| &\leq \gamma_2 |\psi'(0)|\end{aligned} \tag{3.36}$$

and

$$\psi(\widehat{\alpha}) \leq \psi(0) + \gamma_1 (\widehat{\alpha} \psi'(0) + \frac{\widehat{\alpha}^2}{2} \psi''(0)) \tag{3.37}$$

$$\psi'(\widehat{\alpha}) \geq \gamma_2 (\psi'(0) + \widehat{\alpha} \psi''(0)), \tag{3.38}$$

where $0 \leq \gamma_2 \leq \frac{1}{2} \leq \gamma_1 \leq 1$. Based on the type of search employed, an acceptable $\widehat{\alpha}$ is guaranteed to exist.

Proof. We omit the proof for the linesearch and prove the existence of an $\widehat{\alpha}$ for the case of a curvilinear search. We begin by assuming that (3.37) does not hold. Then we define $\bar{\psi}(\alpha; \gamma_1)$ as

$$\bar{\psi}(\alpha; \gamma_1) \equiv \psi(\alpha) - \psi(0) - \gamma_1 (\alpha \psi'(0) + \frac{\alpha^2}{2} \psi''(0)),$$

where $\bar{\psi}(0; \gamma_1) = 0, \bar{\psi}(0; \gamma_1) = (1 - \gamma_1)\psi'(0) \leq 0$ and $\bar{\psi}''(0; \gamma_1) = (1 - \gamma_1)\psi''(0) < 0$. Since (3.37) is not satisfied, we have $\bar{\psi}(1; \gamma_1) > 0$ and there must exist an $\widehat{\alpha} \in (0, 1]$ for which $\bar{\psi}'(\widehat{\alpha}; \gamma_2) \geq 0$ for $\gamma_2 \in (\frac{1}{2}, 1)$ as specified in (3.38). If this were not the case, then $\bar{\psi}'(\alpha; \gamma_2) < 0$ for all $\alpha \in (0, 1]$. Moreover, from $\gamma_2 > \gamma_1$, we have that $\psi'(0) \leq 0$ and $\psi''(0) < 0$, implying that $\bar{\psi}'(\alpha; \gamma_1) < 0$ for all $\alpha \in (0, 1]$ and (3.37) would then hold, contradicting our assumption.

Assume that $\widehat{\alpha}$ is the smallest such α with $\bar{\psi}'(\alpha;\gamma_1)$ for all $\alpha \in (0,\widehat{\alpha})$. Then, we may claim that

$$\psi(\widehat{\alpha}) \leq \psi(0) + \gamma_2(\widehat{\alpha}\psi'(0) + \frac{\widehat{\alpha}^2}{2}\psi''(0)).$$

From $\psi'(0) \leq 0$ and $\psi''(0) < 0$ with $\gamma_1 < \gamma_2$, the first equation of (3.38) is satisfied at $\widehat{\alpha}$. By $\bar{\psi}'(\widehat{\alpha};\gamma_2) \geq 0$, the second equation of (3.38) is also satisfied. ■

The strategy for updating the penalty parameter is straightforward. If the obtained directions result in sufficient descent in the merit function, then $\rho_{k+1} = \rho_k$. Otherwise, the earlier two results show that $\rho_k = \max(\bar{\rho}, \rho_k)$.

3.3.3 Boundedness of Iterates

The proof of the next two results may be found in Lemma 3.5 from [MP95b] and Lemma 3.9 from [MP95a], respectively.

Lemma 3.17. *The first-order strong-stationary points for MPCC are isolated.*

Lemma 3.18. *There exists a positive constant β_H such that for any iteration we have*

$$-\varpi + \tfrac{1}{2}(\Delta p^c)^T \mathcal{H} \Delta p^c \leq -\beta_H(\|\Delta p\|^2 + \|\Delta p^c\|^3) \tag{3.39}$$

and

$$\psi_C''(0) \leq -\beta_H(\|\Delta p\|^2 + \|\Delta p^c\|^3). \tag{3.40}$$

Lemma 3.19. *If along a sequence S, we have $\|\Delta p_k^c\| \to 0$ for $k \in S$, then there exists a constant β_c such that for any iteration $k \in S$, $\|\Delta p_k^c\| = 0$ or $\|\Delta p_k^c\| > \beta_c$.*

Lemma 3.20. *There exists a constant M such that, for all k,*

$$\rho_k(\|c(z_k)\| + \|d(z_k, s_k;\gamma_k)\|) \leq M. \tag{3.41}$$

We may invoke the boundedness of the primal variables to prove that s_k, β_k and ξ_k are bounded. Moreover, we also show that s_k is componentwise bounded away from zero. Part (a) of this result is adapted from [LS04a].

Lemma 3.21. *Suppose $\rho_k = \bar{\rho}$ for $k \geq K$. Then the following hold:*

(a) The slack variables s_k are bounded above in norm.

(b) The multipliers $\lambda_k = (\beta_k, \xi_k)$ are bounded above in norm.

(c) Moreover, s_k is componentwise bounded away from zero.

Proof.

(a) Assume that s_k is not bounded. By the boundedness of ρ, we have $\rho = \bar{\rho}$ for $k \geq K$. Without loss of generality, we may assume that $\rho = \bar{\rho}$ for $k \geq 0$. At the $(k+1)$th iteration, we have

$$\mathcal{M}_{k+1} \leq \mathcal{M}_k.$$

At the beginning of the $(k+1)$th iteration, we specify $\mathcal{M}_{k+1}$ using an updated μ_k, viz. μ_{k+1}. Therefore, the merit function gets modified because of the reduction in the barrier parameter. If we denote the jth component of the kth iterate of s by $[s_k]_j$, then the modification in the merit function is given by

$$(\mu_k - \mu_{k+1}) \sum_{j=1}^{n} \ln[s_k]_j.$$

We may sum over the first k iterations to obtain

$$\mathcal{M}_{k+1} + \sum_{i=1}^{k} (\mu_i - \mu_{i+1}) \sum_{j=1}^{n} \ln[s_k]_j \leq \mathcal{M}_0.$$

Consequently, we may claim that

$$\begin{aligned} &f(z_{k+1}) - \mu_{k+1} \sum_{j=1}^{n} \ln[s_{k+1}]_j - \lambda_{k+1}^T h_{k+1} + \tfrac{1}{2}\bar{\rho}\|h_{k+1}\|^2 + \sum_{i=1}^{k} (\mu_i - \mu_{i+1}) \sum_{j=1}^{n} \ln[s_k]_j \\ \leq &f(z_0) - \mu_0 \sum_{j=1}^{n} \ln[s_0]_j - \lambda_0^T h_0 + \tfrac{1}{2}\bar{\rho}\|h_0\|^2. \end{aligned}$$

Suppose the qth component of s_k is unbounded: $[s_k]_q \to \infty$ as $k \to \infty$. We may divide the inequality by $([s_k]_q)^2$ and let k go to ∞. As a result, every term converges to zero except $\lim_{k\to\infty} \frac{\|h_k\|^2}{([s_k]_q)^2} \geq 1$. Therefore we have $\bar{\rho} \leq 0$ which is a contradiction. Therefore $\{s_k\}$ is bounded.

(b) To show the boundedness of λ_k throughout the algorithm, we note that from the first-order conditions, we have

$$\nabla_z f - \nabla_z h^T \lambda(\gamma_k, \mu_k) = 0. \tag{3.42}$$

Then

$$\begin{aligned} \nabla_z h \nabla h^T \lambda(\mu_k, \gamma_k) &= \nabla h \nabla_z f \\ \lambda(\mu_k, \gamma_k) &= (\nabla_z h \nabla_z h^T)^{-1} \nabla_h \nabla_z f \\ \|\lambda(\mu_k, \lambda_k)\| &\leq \|(\nabla_z h \nabla_z h^T)^{-1}\| \|\nabla_z h\| \|\nabla f\|. \end{aligned}$$

The boundedness of $\lambda_k = \lambda(\mu_k, \gamma_k)$ follows from the boundedness of z_k.

(c) It suffices to show that $\|S_k^{-1}\|$ is bounded from above, where $S_k = \text{diag}(s_k)$. From the

first-order conditions, we have

$$\begin{aligned}\mu_k S_k^{-1} e &= \nabla_s h_k^T \lambda_k \\ S_k^{-1} e &= \frac{\nabla_s h_k^T \lambda_k}{\mu_k} \\ \|S_k^{-1}\| &\leq \|\nabla_s h_k^T \lambda_k\|/\mu_k \\ &\leq \|\nabla_s h_k\| \|\lambda_k\|/\mu_k.\end{aligned}$$

Recall that

$$\nabla_s h = \begin{pmatrix} \nabla_s c \\ \nabla_s d \end{pmatrix}.$$

Moreover, c is independent of the slack variables, while

$$\nabla_s d = \begin{pmatrix} I & & \\ & -I & \\ & & -I \end{pmatrix}.$$

Therefore, we have

$$\|S_k^{-1}\| \leq \|\lambda_k\|/\mu_k.$$

Under the boundedness of the λ_k and the strict positivity of μ_k, we have $[s_k]_j > 0$ for all j. ■

It remains to show that the directions Δp and Δp^c are bounded in norm.

Lemma 3.22. *The Newton direction Δp and the negative curvature direction Δp^c are bounded in norm.*

Proof. From Theorem 3.12, Δp is well-defined. By the boundedness of the iterates, the boundedness of Δp follows. Furthermore, the negative curvature direction does not have a natural norm but is scaled with reference to the norm of the Newton direction. Therefore Δp^c is also bounded. ■

We refer the reader to [MP95a] for the proof of the following result.

Lemma 3.23. *For any iteration k_l in which the value of the penalty parameter ρ is modified,*

$$\rho_{k_l} \|h(z_{k_l})\| \leq N$$

and

$$\rho_{k_l} \|\Delta p_{k_l}\|^2 + \|\Delta p_{k_l}^c\|^3 \leq N,$$

for some constant N.

3.3.4 Boundedness of α_k

In an earlier section, we showed that the steplengths are well-defined. We now prove that they are bounded away from zero.

Lemma 3.24. *Assume that $\{z_k, s_k\}$ generated by the algorithm lie in a compact set and that $\|\Delta p_k\| \leq \beta^p$, $\|\Delta p_k^c\| \leq \beta^c$ and $\|h\| \leq \beta^h$, where $h = (c, d)$. For $0 \leq \theta \leq \alpha_k$,*

$$\psi_k'''(\theta) \leq 24\theta\Delta\lambda^T h + 2g_B^T\Delta p - 2c^T\Delta\beta - 2d^T\Delta\xi + O(\|\Delta p\|^2 + \|\Delta p^c\|).$$

Furthermore, we may claim that

$$\psi'''(\theta) \leq \beta_1(\|\Delta p\|^2 + \|\Delta p^c\|).$$

Proof. We suppress the subscript k to simplify the notation. Recall that

$$\begin{aligned}
\psi(\theta) := & f(z + \theta\Delta z^c + \theta^2\Delta z) \\
& - \mu\ln(s + \theta\Delta s^c + \theta^2\Delta s) \\
& + (\beta + \theta^2\Delta\beta)^T c(z + \theta\Delta z^c + \theta^2\Delta z) \\
& + (\xi + \theta^2\Delta\xi)^T d(z + \theta\Delta z^c + \theta^2\Delta z, s + \theta\Delta s^c + \theta^2\Delta s) \\
& + \tfrac{1}{2}\rho(\|c(z + \theta\Delta z^c + \theta^2\Delta z)\|^2 + \|d(z + \theta\Delta z^c + \theta^2\Delta z, s + \theta\Delta s^c + \theta^2\Delta s)\|^2).
\end{aligned}$$

We introduce the following notation:

$$\begin{aligned}
\psi_1(\theta) &:= f(z + \theta\Delta z^c + \theta^2\Delta z) - \mu\ln(s + \theta\Delta s^c + \theta^2\Delta s) \\
&:= f_B(p + \theta\Delta p^c + \theta^2\Delta p) \\
\psi_2(\theta) &:= (\beta + \theta^2\Delta\beta)^T c(z + \theta\Delta z^c + \theta^2\Delta z) \\
&\quad + (\xi + \theta^2\Delta\xi)^T d(z + \theta\Delta z^c + \theta^2\Delta z, s + \theta\Delta s^c + \theta^2\Delta s) \\
&= (\lambda + \theta^2\Delta\lambda)^T h(p + \theta\Delta p^c + \theta^2\Delta p) \\
\psi_3(\theta) &:= \tfrac{1}{2}\|c(z + \theta\Delta z^c + \theta^2\Delta z)\|^2 + \tfrac{1}{2}\|d(z + \theta\Delta z^c + \theta^2\Delta z, s + \theta\Delta s^c + \theta^2\Delta s\|^2 \\
&= \tfrac{1}{2}(\|h(p + \theta\Delta p^c + \theta^2\Delta p)\|^2).
\end{aligned}$$

The third derivative may be expressed as

$$\psi'''(\theta) := \psi_1'''(\theta) - \psi_2'''(\theta) + \rho\psi_3'''(\theta). \tag{3.43}$$

To simplify the expressions further, we denote $\bar{z} = (\Delta p^c + 2\theta\Delta p)$. The third derivatives of $\psi_1(\theta)$,

$\psi_2(\theta)$ and $\psi_3(\theta)$ are given by

$$\begin{aligned}
\psi_1'''(\theta) &:= 6\Delta p^T \nabla^2 f_B(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z} + \sum_i \bar{z}^T \nabla_i^3 f_B(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z}\bar{z}_i \\
\psi_2'''(\theta) &:= 6\theta \sum_j \Delta\lambda_j \bar{z}^T \nabla^2 h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z} + 12\theta\Delta\lambda^T \nabla h(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z} \\
&\quad + \sum_j \left((\lambda_j+\theta^2\Delta\lambda_j) \sum_i \bar{z}^T \nabla_i^3 h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z}\bar{z}_i \right) \\
&\quad + \sum_j \left(6\Delta p^T \nabla^2 h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z} \right) \\
&\quad + 6\Delta\lambda^T \nabla h(p+\theta\Delta p^c+\theta^2\Delta p\Delta p \\
\psi_3'''(\theta) &:= 3\sum_j (\nabla h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z})\bar{z}^T \nabla^2 h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z} \\
&\quad + 6\nabla h(p+\theta\Delta p^c+\theta^2\Delta p)^T \nabla h(p+\theta\Delta p^c+\theta^2\Delta p) \\
&\quad + \sum_j h_j(p+\theta\Delta p^c+\theta^2\Delta p) 6\Delta p^T \nabla^2 h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z} \\
&\quad + \sum_i \bar{z}^T \nabla_i^3 h_j(p+\theta\Delta p^c+\theta^2\Delta p)\bar{z}\bar{z}_i,
\end{aligned}$$

where $\nabla_i^3 h_j$ represents the matrix of derivatives $\left(\frac{\partial^3 h_j}{\partial x_i \partial x_l \partial x_k}\right)$ for a given value of i and all values of l and k. We may use the following results from the Taylor's expansions of h_j:

$$\begin{aligned}
h_j(p+\theta\Delta p^c+\theta^2\Delta p) &= h_j + \nabla h_j^T(\theta\Delta p^c+\theta^2\Delta p) \\
&\quad + \tfrac{1}{2}(\theta\Delta p^c+\theta^2\Delta p)^T \nabla^2 h_j(y_1)(\theta\Delta p^c+\theta^2\Delta p) \\
&= (1-\theta^2)h_j + \theta\nabla h_j^T \Delta p^c \qquad (3.44) \\
&\quad + \tfrac{1}{2}(\theta\Delta p^c+\theta^2\Delta p)^T \nabla^2 h_j(y_1)(\theta\Delta p^c+\theta^2\Delta p) \\
&= (1-\theta^2)h_j + \tfrac{1}{2}(\theta\Delta p^c+\theta^2\Delta p)^T \nabla^2 h_j(y_1)(\theta\Delta p^c+\theta^2\Delta p) \\
\nabla h_j(p+\theta\Delta p^c+\theta^2\Delta p)^T\bar{z} &= \nabla h_j^T\bar{z} + (\theta\Delta p^c+\theta^2\Delta p)^T \nabla^2 h_j \bar{z} \\
&= -2\theta h_j + (\theta\Delta p^c+\theta^2\Delta p)^T \nabla^2 h_j \bar{z} \\
\nabla h_j(p+\theta\Delta p^c+\theta^2\Delta p)^T\Delta p &= -h_j + (\theta\Delta p^c+\theta^2\Delta p)^T \nabla^2 h_j(y_2)\Delta p, \qquad (3.45)
\end{aligned}$$

for some $y_i = p+\theta_i\Delta p^c+\theta_i^2\Delta p$ for $\theta_i \in [0,\theta]$. We may then obtain the following bounds on third derivative terms:

$$\begin{aligned}
\psi_1'''(\theta) &\le 6M_1(\|\Delta p\|^2 + \|\Delta p\|\|\Delta p^c\|) + M_2(\|\Delta p\|+\|\Delta p^c\|)^3 \\
&\le 6M_1(\|\Delta p\|^2 + \tfrac{1}{2}(\|\Delta p\|^2+\|\Delta p^c\|^2)) + M_2(\|\Delta p\|+\|\Delta p^c\|)^3 \\
&\le M_3(\|\Delta p\|^2+\|\Delta p^c\|)^2
\end{aligned}$$

for some positive constants M_1, M_2 and M_3. We may also construct a bound for $\psi_2'''(\theta)$ using (3.45)

and $2ab \le a^2 + b^2$ as follows:

$$\begin{aligned}
-\psi_2'''(\theta) &= \sum_j \left((\lambda_j + \theta^2 \Delta\lambda_j) \sum_i \bar{z}^T \nabla^3 h_j (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} \bar{z}_i \right) \\
&\quad + \sum_j \left(6 \Delta p^T \nabla^2 h_j (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} \right) \\
&\quad + 6\theta \sum_j \Delta\lambda_j \bar{z}^T \nabla^2 h_j (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} + 12\theta \Delta\lambda^T \nabla h (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} \\
&\quad + 6\Delta\lambda^T \nabla h (p + \theta \Delta p^c + \theta^2 \Delta p) \Delta p \\
&= \sum_j \left((\lambda_j + \theta^2 \Delta\lambda_j) \sum_i \bar{z}^T \nabla^3 h_j (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} \bar{z}_i \right) \\
&\quad + \sum_j \left(6 \Delta p^T \nabla^2 h_j (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} \right) \\
&\quad + 6\theta \sum_j \Delta\lambda_j \bar{z}^T \nabla^2 h_j (p + \theta \Delta p^c + \theta^2 \Delta p) \bar{z} \\
&\quad + 12\theta \Delta\lambda^T (-2\theta h_j + (\theta \Delta p^c + \theta^2 \Delta p)^T \nabla^2 h_j \bar{z}) \\
&\quad + 6\Delta\lambda^T (-h_j + (\theta \Delta p^c + \theta^2 \Delta p)^T \nabla^2 h_j (y_2) \Delta p) \\
&\le M_4 ((\|\Delta p\|^2 + \|\Delta p^c\|)^3 + \|\Delta p\|^2 + \|\Delta p\| \|\Delta p^c\| + (\|\Delta p\|^2 + \|\Delta p^c\|)^2) + 24\theta \Delta\lambda^T h \\
&\le M_5 ((\|\Delta p\|^2 + \|\Delta p^c\|)^2) + 24\theta \Delta\lambda^T h,
\end{aligned}$$

for some constants M_4 and M_5. Finally, a bound for $\psi_3'''(\theta)$ may be obtained using similar ideas.

$$\begin{aligned}
\psi_3'''(\theta) &\le M_6 \|\bar{z}\|^2 (\|h\| + \|\bar{z}\| (\|\Delta p^c\| + \|\Delta p\|) + 12\theta h^2 + M_7 \|h\| \|\bar{z}\| (\|\Delta p^c\| + \|\Delta p\|) \\
&\quad + M_8 \|\bar{z}\|^2 (\|\Delta p^c\|^2 + \|\Delta p\|^2) + M_9 ((\|\Delta p\| + \|\Delta p^c\|)^2 + \|h\|)(\|\bar{z}\|^3 + \|\bar{z}\| \|\Delta p\|)) \\
&\le 12\theta \|h\|^2 + M_{10} \|h\| (\|\Delta p\|^2 + \|\Delta p^c\|^2) + M_{11} (\|\Delta p\|^4 + \|\Delta p^c\|^4),
\end{aligned}$$

where $M_6, M_7, M_8, M_9, M_{10}$ and M_{11} are positive constants. Then by Lemmas 3.23, 3.19, 3.20 and the boundedness of $\|\Delta p^c\|$, we have

$$\rho \psi_3'''(\theta) \le 12\theta\rho \|h\|^2 + M_{12} (\|\Delta p\|^2 + \|\Delta p^c\|),$$

for some constant $M_{12} > 0$. From (3.43), we have

$$\begin{aligned}
\psi'''(\theta) &\le M_3 (\|\Delta p\|^2 + \|\Delta p^c\|)^2 + M_5 ((\|\Delta p\|^2 + \|\Delta p^c\|)^2) + 24\theta \Delta\lambda^T h + \theta\rho \|h\|^2 \\
&\quad + M_{12} (\|\Delta p\|^2 + \|\Delta p^c\|) \\
&\le 24\theta \Delta\lambda^T h + \theta\rho \|h\|^2 + O(\|\Delta p\|^2 + \|\Delta p^c\|).
\end{aligned}$$

From (3.33), we have

$$\rho(\|h\|^2) = (\Delta p^c)^T \nabla_p^2 \mathcal{L} \Delta p^c) + 2 g_B^T \Delta p - 2 c^T \Delta\beta - 2 d^T \Delta\xi - \psi_C''(0).$$

Then, we may claim that

$$\begin{aligned}\psi'''(\theta) &\leq M_3(\|\Delta p\|^2 + \|\Delta p^c\|)^2 + M_5((\|\Delta p\|^2 + \|\Delta p^c\|)^2) + 24\theta\Delta\lambda^T h + \theta\rho\|h\|^2 \\ &\quad + M_{12}(\|\Delta p\|^2 + \|\Delta p^c\|) \\ &\leq 24\theta\Delta\lambda^T h + \theta(\Delta p^c)^T\nabla_p^2\mathcal{L}\Delta p^c) + 2g_B^T\Delta p - 2c^T\Delta\beta - 2d^T\Delta\xi - \psi_C''(0)) \\ &\quad + O(\|\Delta p\|^2 + \|\Delta p^c\|).\end{aligned}$$

■

Lemma 3.25. *There exists a value $\bar{\alpha}$ such that $\widehat{\alpha} \geq \bar{\alpha} > 0$, where $\widehat{\alpha}$ is the steplength computed by the algorithm.*

Proof. Since $\widehat{\alpha}$ satisfies the curvilinear search conditions, we have

$$\begin{aligned}\psi(\widehat{\alpha}) &\leq \psi(0) + \sigma(\widehat{\alpha}\psi'(0) + \tfrac{1}{2}\widehat{\alpha}^2\psi''(0)) \\ \psi'(\widehat{\alpha}) &\geq \eta(\psi'(0) + \widehat{\alpha}\psi''(0)),\end{aligned}$$

where $0 < \sigma < \frac{1}{2} < \eta < 1$. It was shown in [MP95a] if the above conditions are satisfied for an $\widehat{\alpha}$, the second condition cannot hold for any $\alpha \in (0, \widehat{\alpha})$. Therefore, we have

$$\psi'(\widehat{\alpha}) = \eta(\psi'(0) + \widehat{\alpha}\psi''(0)).$$

From the Taylor's expansion of $\psi'(\alpha)$ we have

$$\psi'(\widehat{\alpha}) = \psi'(0) + \widehat{\alpha}\psi''(0) + \tfrac{1}{2}\widehat{\alpha}^2\psi'''(\theta), \quad \theta \in [0, \widehat{\alpha}].$$

We may combine these two results to obtain

$$(1-\eta)\psi'(0) + (1-\eta)\psi''(0)\widehat{\alpha} + \tfrac{1}{2}\psi'''(\theta)\widehat{\alpha}^2 = 0. \tag{3.46}$$

Since $\psi'(0) \leq 0$ and $\psi''(0) < 0$ and a positive root of the equation (3.46) must exist, we have $\phi'''(0) > 0$. (3.46) has roots given by

$$\widehat{\alpha} := -(1-\eta)\frac{\psi''(0)}{\psi'''(\theta)} \pm \sqrt{(1-\eta)^2\frac{\psi''(0)}{\psi'''(\theta)}^2 - 2(1-\eta)\frac{\psi'(0)}{\psi'''(\theta)}}. \tag{3.47}$$

Since $(1-\eta)\psi'(0)/\psi'''(\theta) \leq 0$, the following bound holds:

$$\widehat{\alpha} \geq -2(1-\eta)\frac{\psi''(0)}{\psi'''(\theta)}.$$

From Lemma 3.18,

$$\widehat{\alpha} \geq -2(1-\eta)\frac{\psi''(0)}{\psi'''(\theta)} \geq 2(1-\eta)\frac{\beta_H(\|\Delta p\|^2 + \|\Delta p^c\|^3)}{\psi'''(\theta)}.$$

Furthermore, from Lemma 3.24, we have

$$\widehat{\alpha} \geq 2(1-\eta)\frac{\beta_H(\|\Delta p\|^2 + \|\Delta p^c\|^3)}{\psi'''(\theta)} \geq \frac{2(1-\eta)\beta_H}{\beta_1}\frac{(\|\Delta p\|^2 + \|\Delta p^c\|^3}{\|\Delta p\| + \|\Delta p^c\|},$$

for some $\beta_1 > 0$. Using results from Lemma 3.21 in [MP95a], we may show that the lemma follows. ■

3.3.5 Global Convergence Results

We are now ready to present some of the global convergence results for our interior method.

Theorem 3.26. *The sequence of Newton and negative curvature directions converge to zero in norm:*

$$\lim_{k\to\infty} \|\Delta p_k\| = 0 \quad \text{and} \quad \lim_{k\to\infty} \|\Delta p_k^c\| = 0. \tag{3.48}$$

Proof. From Lemma 3.23, we have

$$\rho_{k_l}\|\Delta p_{k_l}\|^2 + \|\Delta p_{k_l}^c\|^3 \leq N.$$

When $\rho_k \to \infty$, from the boundedness of α_k, we obtain the required result. If $\rho_k \to \rho^* < \infty$, then we have

$$\begin{aligned}\psi_{k+1} - \psi_k &= \alpha_k\psi_k' + \tfrac{1}{2}\alpha_k^2\psi''(\bar{\alpha}_k)\\ &\leq -\tfrac{1}{2}\alpha_k^2\beta_H(\|\Delta p_k\|^2 + \|\Delta p_k^c\|^3)\\ &\leq -\tfrac{1}{2}\bar{\alpha}^2\beta_H(\|\Delta p_k\|^2 + \|\Delta p_k^c\|^3).\end{aligned}$$

Assume that the result does not hold. Then we have $\psi_k \to -\infty$. But this is impossible by the compactness of the iterates and strict positivity of all slack variables. Therefore, the desired result follows. ■

In addition, we prove that the first-order KKT conditions hold in the limit.

Theorem 3.27. *Let $\{\zeta_k(\mu_k, \gamma_k)\}_{k=1}^{\infty}$ be the sequence of iterates generated by the algorithm satisfying*

$$\nabla_z \mathcal{L}_k = \mathcal{O}(\epsilon_k) \tag{3.49}$$

$$\nabla_s \mathcal{L}_k = \mathcal{O}(\epsilon_k) \tag{3.50}$$

$$c(z_k) = \mathcal{O}(\epsilon_k) \tag{3.51}$$

$$d(z_k, s_k) = \mathcal{O}(\epsilon_k) \tag{3.52}$$

$$S_k v_k = \mathcal{O}(\epsilon_k), \tag{3.53}$$

where $\epsilon_k = \min(\gamma_k, \mu_k)$. Then there exists a limit point ζ^* of the sequence $\{\zeta(\mu_k, \gamma_k)\}_{k=1}^{\infty}$ that satisfies the first-order and second-order conditions of MPCC.

Proof. Boundedness of the sequence $\{\zeta_k\}_{k=1}^{\infty}$ follows from the assumption of boundedness of $\{z_k\}$ and Lemma 3.21. Then $\{\zeta_k\}_{k=1}^{\infty}$ has a convergent subsequence $\mathcal{K}$ with a limit point ζ^*. In particular, we have

$$\lim_{k\in\mathcal{K},k\to\infty} \zeta(\mu_k,\gamma_k) = \zeta^*. \tag{3.54}$$

If we take limits with $k \to \infty$, $k \in \mathcal{K}$ in (3.49)–(3.53), and by using (3.54), the desired result follows. From Lemma 3.19, the negative curvature direction is nonzero only for a finite number of iterations. Therefore, there must exist an iteration index $\bar{K}$ such that for $k \geq \bar{K}$, $\|\Delta p^c\| = 0$. Consequently, $\lambda_{\min}(\mathbf{Z}^T\nabla^2\mathcal{L}_k\mathbf{Z}) > 0$ for all $k \geq \bar{K}$, implying that $\mathbf{Z}^T\nabla^2\mathcal{L}^*\mathbf{Z}$ is positive definite. ■

Theorem 3.28. *Under the assumptions of Theorem 3.27, the limit point $\{\zeta^*\}$ is a second-order point.*

Proof. We use the ideas of [MP03] in proving this result. In particular, we show that if the limit point is not a second-order point, then the merit function must be unbounded below, in contradiction with the compactness of the iterates.

From Lemma 3.15, we have $\psi_k''(0) \leq -\varpi_k + \frac{1}{2}(\Delta p_c^k)^T\mathcal{H}_k\Delta p_c^k$ at any iteration. We define a subsequence $\bar{\mathcal{K}}$ such that $\|\Delta p_k^c\| \geq \beta_c$ for $k \in \bar{\mathcal{K}}$, implying that the iterates are converging to a first-order KKT point that does not satisfy the second-order conditions (because Δp^c is nonzero at the limit point). Therefore, there exists some $\beta_C > 0$ such that

$$\psi_k''(0) \leq -\beta_C, \quad k \in \bar{\mathcal{K}}. \tag{3.55}$$

Let us now consider the change in the merit function for all iterations greater than K. Without loss of generality, the boundedness of the penalty parameter allows us to assume that $\rho_k = \bar{\rho}$ for $k \geq K$. Then for all $k \geq K$,

$$\mathcal{M}(\zeta_{k+1};\mu_{k+1},\gamma_{k+1},\bar{\rho}) \leq \mathcal{M}(\zeta_k,\bar{\rho}) + \tfrac{1}{2}\gamma_1\alpha_k^2\psi''(0) + \sum_{i=1}^{n}(\mu_k - \mu_{k+1})\ln[s_{k+1}]_i. \tag{3.56}$$

By the boundedness of s_k, there exists an $[\bar{s}_k]_i > 1$ for all $i = 1,\ldots,n$ such that $[s_k]_i \leq [\bar{s}_k]_i, i = 1,\ldots,n$. This allows us to rewrite (3.56) as

$$\mathcal{M}(\zeta_{k+1};\mu_{k+1},\gamma_{k+1},\bar{\rho}) \leq \mathcal{M}(\zeta_k,\bar{\rho}) + \tfrac{1}{2}\gamma_1\alpha_k^2\psi''(0) + \sum_{i=1}^{n}(\mu_k - \mu_{k+1})\ln[\bar{s}_{k+1}]_i. \tag{3.57}$$

Then by summing up the inequalities from k to r, we have

$$\begin{aligned}\mathcal{M}(\zeta_{k+1};\mu_{k+1},\gamma_{k+1},\bar{\rho}) &\leq \mathcal{M}(\zeta_r,\bar{\rho}) + \sum_{j=1}^{r} \tfrac{1}{2}\gamma_1\alpha_j^2\psi''(0) + \sum_{i=1}^{n}(\mu_r - \mu_{k+1})\ln[\bar{s}_{k+1}]_i \\ &\leq \mathcal{M}(\zeta_r,\bar{\rho}) + \sum_{j\in\bar{\mathcal{K}},j\leq k} \tfrac{1}{2}\gamma_1\alpha_j^2\psi''(0) + \sum_{i=1}^{n}\mu_r\ln[\bar{s}_{k+1}]_i \\ &\leq \mathcal{M}(\zeta_r,\bar{\rho}) + \sum_{j\in\bar{\mathcal{K}},j\leq k} \tfrac{1}{2}\gamma_1\alpha_j^2(-\varpi_k + \tfrac{1}{2}(\Delta p_k^c)^T\mathcal{H}_k\Delta p_k^c) \\ &\quad + \sum_{i=1}^{n}\mu_r\ln[\bar{s}_{k+1}]_i.\end{aligned}$$

By the compactness of the iterates, we have $\mathcal{M}(\zeta_{k+1};\mu_{k+1},\gamma_{k+1},\bar{\rho}) > \beta_M$. Then

$$\mathcal{M}(\zeta_r,\bar{\rho}) + \sum_{i=1}^{n}\mu_r\ln[\bar{s}_{k+1}]_i \geq \beta_C - \sum_{j\in\bar{\mathcal{K}},j\leq k} \tfrac{1}{2}\gamma_1\alpha_j^2(-\varpi_k + \tfrac{1}{2}(\Delta p_k^c)^T\mathcal{H}_k\Delta p_k^c).$$

But from Lemma 3.25, we know that $\alpha_k \geq \bar{\alpha} > 0$. Moreover, $\varpi > 0$ and

$$(\Delta p_k^c)^T\mathcal{H}_k\Delta p_k^c \;<\; 0$$

for all $k \in \bar{\mathcal{K}}$. This implies that as $k \to \infty, k \in \bar{\mathcal{K}}$, we have $\mathcal{M}(\zeta_r;\mu_r,\gamma_r,\bar{\rho}) \to \infty$. But this contradicts the compactness of iterates. Therefore our assertion (3.55) must be false and $\|\Delta p_k^c\| \to 0, k \in \bar{\mathcal{K}}$.

■

The final results in this section show that the multipliers converge to the multiplier associated with the strong-stationary point. We use the ideas of Leyffer et al. [LLCN05], albeit for an interior-relaxation approach.[4] In this section, we remove the constraints

$$w - s_w = 0 \quad \text{and} \quad y - s_y = 0$$

and keep the constraints $w \geq 0$ and $y \geq 0$ strictly inactive by using barrier functions. The following result from [FLRS02] states the first-order conditions for MPCC. To allow for easier reading, we use different notation for our last result.

Theorem 3.29. *Consider the following problem:*

MPCC	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$
	subject to	$c(x,y,w) = 0$
		$Yw \leq 0$
		$w \geq 0$
		$y \geq 0.$

[4] Leyffer et al. [LLCN05] provide convergence theory for an interior-penalty approach.

Suppose MPCC-LICQ holds at (x^*, y^*, w^*), a minimizer of MPCC. Then there exist multipliers satisfying the following system:

$$\begin{aligned}
\begin{pmatrix} \nabla_x f \\ \nabla_y f \\ \nabla_w f \end{pmatrix} - \begin{pmatrix} \nabla_x c \\ \nabla_y c \\ \nabla_w c \end{pmatrix}^T \beta - \begin{pmatrix} 0 \\ \psi_y \\ \psi_w \end{pmatrix} &= 0 \\
c(x, y, w) &= 0 \\
y &\geq 0 \\
w &\geq 0 \\
Yw &= 0 \\
Y\psi_y &= 0 \\
W\psi_w &= 0 \\
[\psi_y]_i, [\psi_w]_i &\geq 0, i \in \{i : w_i, y_i = 0\}.
\end{aligned} \tag{3.58}$$

Proof. See [FLRS02]. ■

The resulting barrier problem BP(γ, μ)[5] is given by

BP(γ, μ)	$\underset{x,y,w,s}{\text{minimize}}$	$f(x, y, w) - \mu \sum_j \ln s_j - \mu \sum_j \ln y_j - \mu \sum_j \ln w_j$
	subject to	$c(x, y, w) = 0 \quad : \beta,$
		$Yw + s = \gamma e \quad : \lambda.$

Its stationarity conditions are given by

$$\begin{aligned}
\begin{pmatrix} \nabla_x f \\ \nabla_y f \\ \nabla_w f \end{pmatrix} - \begin{pmatrix} \nabla_x c \\ \nabla_y c \\ \nabla_w c \end{pmatrix}^T \beta - \begin{pmatrix} 0 \\ \mu Y^{-1} e - W\lambda \\ \mu W^{-1} e - Y\lambda \\ \mu S^{-1} e - \lambda \end{pmatrix} &= 0 \\
c(x, y, w) &= 0 \\
Yw + s &= \gamma e.
\end{aligned}$$

The last result of this section shows that the first-order KKT point is a strong-stationary point of MPCC. It is adapted from [LLCN05].

Theorem 3.30. *Suppose that the algorithm generates an infinite sequence $(z^k, s^k, \beta^k, \lambda^k)$ satisfying the termination criteria for sequences $\{\mu^k\}$ and $\{\gamma^k\}$ converging to zero. The sequence $\{\epsilon_k\}$ is constructed on the basis of $\epsilon_k = \min(\gamma_k, \mu_k)$. Also, we assume that f and c are twice continuously differentiable in the neighborhood of the limit point $(z^*, s^*, \beta^*, \lambda^*)$. Also suppose that the regularized barrier problems MPCC(γ, μ) are always solvable. Then the following hold:*

1. *If $(z^*, s^*, \beta^*, \lambda^*)$ is a limit point of this sequence, then it is a feasible point of MPCC.*

[5] This barrier problem is similar to MPCC(γ, μ) but in this particular case w and y are kept strictly positive.

2. If MPCC-LICQ holds at $(z^*, s^*, \beta^*, \lambda^*)$, then $(z^*, s^*, \beta^*, \lambda^*)$ is a strong stationary point.

Proof. Throughout this proof, we use the compact notation of

$$u(\mu, \gamma) = (z, s, \beta, \lambda; \mu, \gamma).$$

If $\{u^k(\mu^k, \gamma^k)\}$ represents the set of iterates generated by the algorithm, then, by assumption, we have $\{u^k(\mu^k, \gamma^k)\} \to u^*(0, 0)$. This requires that the sequences $\{\mu^k\}, \{\gamma^k\}$ converge to zero. This implies that for k sufficiently large or $k \in \mathcal{K}$ and

$$\lim_{k \in \mathcal{K}} u^k = u^*,$$

we may conclude that the sequences $\{\nabla f^k\}_{k \in \mathcal{K}}$ and $\{\nabla c^k\}_{k \in \mathcal{K}}$ have limits and are therefore bounded.

1. *Feasibility of u^**: Since MPCC(γ, μ) is always solvable and that $\epsilon^k \to 0$ in (3.29), we have $c(u^*) = 0$. Moreover, $Y^* w^* = 0$ because $\gamma^k \to 0$. The nonnegativity of w^* and y^* is by definition because the barrier problem keeps the iterates strictly feasible. Thus, u^* is feasible for MPCC.

2. *Existence of multipliers*: We may define

$$[\psi_y^k]_i = \frac{\mu^k}{y_i^k} - w_i^k \lambda_i^k$$
$$[\psi_w^k]_i = \frac{\mu^k}{w_i^k} - y_i^k \lambda_i^k,$$

and $\eta^k = \|(\beta^k, \lambda^k, \psi_y^k, \psi_w^k)\|_\infty$. Furthermore, we may claim that if the stationarity conditions of the regularized problem are evaluated with $\mu = \gamma = 0$, we have

$$Y\psi_y = 0$$
$$W\psi_w = 0$$
$$Wy = 0.$$

We show that the sequence $\{\eta^k\}_{k \in \mathcal{K}}$ is bounded. We may claim that $\eta^k \geq \tau > 0$ for $k \in \mathcal{K}$ because ψ_w^k and ψ_y^k are bounded away from zero. Then the multiplier set may be normalized as follows:

$$\widehat{\beta}^k = \frac{\beta^k}{\eta^k}, \quad \hat{\lambda}^k = \frac{\lambda^k}{\eta^k}, \quad \widehat{\psi}_y^k = \frac{\psi_y^k}{\eta^k} \quad \text{and} \quad \widehat{\psi}_w^k = \frac{\psi_w^k}{\eta^k}.$$

We define $\mathcal{I}_w(u)$ and $\mathcal{I}_y(u)$ as

$$\mathcal{I}_w(u^*) := \{i : w_i^* = 0\}$$
$$\mathcal{I}_y(u^*) := \{i : y_i^* = 0\}.$$

Consider an index $i \notin \mathcal{I}_w(u^*)$. We have $w_i^* \to w_i^* > 0$, implying that $y_i^* \to 0$ and $\epsilon^k \to 0$. We proceed by contradiction and assume that $[\psi_w^k]_i \neq 0, k \in \mathcal{K}$. Then by definition,

$$\begin{aligned}
&[\psi_w^k]_i \neq 0 \\
\implies &\frac{\mu^k}{w_i^k} - y_i^k \lambda_i^k \neq 0 \\
\implies &\frac{w_i^k}{y_i^k}(\frac{\mu^k}{w_i^k} - y_i^k \lambda_i^k) \neq 0 \\
\implies &(\frac{\mu^k}{y_i^k} - w_i^k \lambda_i^k) \neq 0 \\
\implies &[\psi_y^k]_i \neq 0.
\end{aligned}$$

Therefore, we may claim that

$$|[\widehat{\psi}_w^k]_i| = \frac{|[\widehat{\psi}_w^k]_i|}{\eta_k} \leq \frac{|[\widehat{\psi}_w^k]_i|}{|[\widehat{\psi}_y^k]_i|} \leq \frac{y_i^k}{w_i^k} \to 0.$$

However, we also know that $[\widehat{\psi}_w^k]_i$ converges to zero when $i \in \mathcal{I}_w$, implying that $[\widehat{\psi}_w^k]_i$ converges to zero for all indices i. Similarly, we have $[\widehat{\psi}_y^k]_i \to 0$, for all indices i. Therefore, the multipliers $[\psi_w^k]_i$ and $[\widehat{\psi}_y^k]_i$ both converge to zero for inactive constraints.

The boundedness of $\{\eta_k\}_{k\in\mathcal{K}}$ may be proved by contradiction. Assume that there exists a subsequence $\mathcal{K}'$ in which $\{\eta_k\}_{k\in\mathcal{K}'} \to \infty$. Moreover, the sequences of gradients and Jacobians are convergent, allowing us to divide by η_k and take k to the limit. This implies that

$$\lim_{k\to\infty, k\in\mathcal{K}'} \|\frac{1}{\alpha_k}\nabla_x \mathcal{L}(u^k; \mu^k, \gamma^k)\| \leq \lim_{k\to\infty, k\in\mathcal{K}'} \frac{\epsilon^k}{\eta^k} = 0.$$

Hence

$$\begin{aligned}
\lim_{k\to\infty, k\in\mathcal{K}'} &\left[\frac{1}{\eta_k}\nabla f(u_k) - \nabla c(u^k)^T \widehat{\beta}^k - \begin{pmatrix} 0 \\ \widehat{\psi}_y^k \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \widehat{\psi}_w^k \end{pmatrix}\right] = 0 \\
&\implies \left[-\nabla c(u^*)^T \widehat{\beta}^* - \begin{pmatrix} 0 \\ \widehat{\psi}_y^* \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \widehat{\psi}_w^* \end{pmatrix}\right] = 0 \\
\implies &\left[-\nabla c(u^*)^T \widehat{\beta}^* - \sum_{i\in\mathcal{I}_{y^*}} [\widehat{\psi}_y^*]_i \begin{pmatrix} 0 \\ e_i \\ 0 \end{pmatrix} - \sum_{i\in\mathcal{I}_{w^*}} [\widehat{\psi}_w^*]_i \begin{pmatrix} 0 \\ 0 \\ e_i \end{pmatrix}\right] = 0.
\end{aligned}$$

However, since the limit point x^* satisfies MPCC-LICQ, we have $\widehat{\beta^*}$, $\widehat{\psi}_y^*$ and $\widehat{\psi}_y^*$ equal to zero. However, since the sequences of normalized multipliers are convergent, we may claim that without loss of generality, $\eta_k = 1$ for all $k \in \mathcal{K}'$. Therefore, no such unbounded sequence exists and the multiplier sequences $\{\beta^k\}, \{\psi_w^k\}$ and $\{\psi_y^k\}$ for $k \in \mathcal{K}$ are all bounded with limit

points.

3. *Strong-stationarity:* We now prove that the limit point of the sequence is indeed a strong stationary point. This requires proving that the limit point u^* satisfies (3.58). By the continuity of f and c, we may claim that

$$\begin{pmatrix} \nabla_x f^k \\ \nabla_y f^k \\ \nabla_w f^k \\ \end{pmatrix} - \begin{pmatrix} \nabla_x c^k \\ \nabla_y c^k \\ \nabla_w c^k \\ \end{pmatrix}^T \beta^k - \begin{pmatrix} 0 \\ \mu^k (Y^k)^{-1} e - W^k \lambda^k \\ \mu^k (W^k)^{-1} e - Y^k \lambda^k \\ \mu^k (S^k)^{-1} e - \lambda^k \end{pmatrix} \leq \epsilon^k \to 0$$

$$\implies \begin{pmatrix} \nabla_x f^* \\ \nabla_y f^* \\ \nabla_w f^* \\ \end{pmatrix} - \begin{pmatrix} \nabla_x c^* \\ \nabla_y c^* \\ \nabla_w c^* \\ \end{pmatrix}^T \beta^* - \begin{pmatrix} 0 \\ \psi_y^* \\ \psi_w^* \\ 0 \end{pmatrix} = 0.$$

The first part of this proof has shown that the limit point is a feasible point of the MPCC. Therefore, we need to show that

(a) $W^* \psi_w^* = 0$ and $Y^* \psi_y^* = 0$: This may be rewritten as

$$w_i^* [\psi_w^*]_i = 0, \quad y_i^* [\psi_y^*]_i = 0.$$

If $i \in \mathcal{I}_{w^*}$, then $w_i^* [\psi_w^*]_i = 0$. Therefore, it suffices to consider indices $i \notin \mathcal{I}_{w^*}$ and show that $[\psi_w^*]_i = 0$. Let us consider a subsequence $\mathcal{K}'' \in \mathcal{K}$ in which $[\psi_w^k]_i \neq 0$. Then, using the same argument as earlier, we may claim that $[\psi_y^k]_i \neq 0$. This leads to

$$\lim_{k \to \infty, k \in \mathcal{K}''} \frac{|[\psi_w^k]_i|}{|[\psi_y^k]_i|} = \lim_{k \to \infty, k \in \mathcal{K}''} \frac{y_i^k}{w_i^k} = 0,$$

since $[y_i^k] \to 0$ and $w_i^k > 0$ and $[\psi_y^k]_i$ is bounded for $k \in \mathcal{K}''$. This can only imply that $\lim_{k \in \mathcal{K}''} [\psi_w^k]_i = 0$, giving us the required result. The same holds for ψ_w^*.

(b) $[\psi_y^*]_i, [\psi_w^*]_i \geq 0, i \in \{i : w_i^*, y_i^* = 0\}$: This follows by noting that if $i \in \mathcal{I}_{w^*} \cap \mathcal{I}_{y^*}$, then $w_i^* = y_i^* = 0$. Therefore, $\lim_{k \in \mathcal{K}} w_i^k \lambda_i^k = 0$ and $\lim_{k \in \mathcal{K}} y_i^k \lambda_i^k = 0$. Therefore, we have

$$[\psi_w^*]_i = \lim_{k \in \mathcal{K}} [\psi_w^k]_i = \lim_{k \in \mathcal{K}} \frac{\mu^k}{w_i^k} - y_i^k \lambda_i^k = \lim_{k \in \mathcal{K}} \frac{\mu^k}{w_i^k} \geq 0.$$

Similarly, $[\psi_y^*]_i \geq 0$ for $i \in \mathcal{I}_{w^*} \cap \mathcal{I}_{y^*}$.

■

3.4 Implementation Details and Numerical Results

We implemented the barrier method using MATLAB 6.5 on a Pentium 4 with 512 MB of RAM running Windows XP. In this section, we discuss various aspects of the implementation. We begin

by discussing some algebraic details of the implementation. Next, we discuss four implementations of the barrier method. This is followed by a description of the numerical results.

3.4.1 Some Implementation Details

In this section, we first discuss some details pertaining to the implementation. In particular, we shall discuss how one obtains a basis for the null-space of the Jacobian of the constraints viz. **Z**. Then we provide some insight into how we modify the Hessian of the Lagrangian to ensure that the reduced Hessian is positive definite. Finally, we briefly describe the four methods that we compare in the next section.

We use a QR factorization to obtain a basis for the null-space of ∇h (see (3.19):

$$\nabla h^T = Q \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

Then

$$\nabla h Q = (Q^T \nabla h^T)^T = \begin{pmatrix} R^T & 0 \end{pmatrix}.$$

If $\nabla h \in \mathbb{R}^{m \times n}$, then the last $n-m$ columns of Q give a basis for the null-space of ∇h. An alternate approach for obtaining a basis for the null-space is described in [Doy03, GMS05].

Recalled that the reduced Hessian of the Lagrangian is modified to ensure its positive definiteness. Given a symmetric matrix H, the modified Cholesky factorization [GMW81] is given by

$$H + E = LDL^T,$$

where E is a nonnegative diagonal matrix, L is a nonsingular lower triangular matrix and D is a positive diagonal matrix. In other words, the modified Cholesky factorization produces a posititive definite matrix that differs from the original matrix in some of its diagonal elements. If H is sufficiently positive definite, this factorization reduces to the Cholesky factorization with $E = 0$. Therefore, in our implementation, we may use E to construct a modified reduced Hessian that is positive definite. Furthermore, if $E \neq 0$, we may obtain a direction d such that $L^T d = e_q$, where $q = \arg\min_j H_{jj}$. This direction is a direction of negative curvature and approximates the eigenvector corresponding to the minimum eigenvalue of H [GMW81]. We implemented the linear search and the curvilinear search in our barrier method:

1. Linear: This implementation did *not* use negative curvature directions and instead modified the reduced Hessian of the Lagrangian. This was to ensure that the Newton direction associated with the linearized KKT system was a descent direction for the merit function. Globalization of the method was provided through a backtracking line search.

2. Curvilinear: This implementation used a curvilinear search that determined a step size based on the modified Newton and the negative-curvature directions.

We remove the need to update the regularization parameter separately by keeping $\gamma = \mu$. Two

different update strategies are compared: Strategy `slow` requires that

$$\mu = \begin{cases} \frac{\mu}{2} & \text{if } \mu > 1e^{-4} \\ \frac{\mu}{1.2} & \text{otherwise} \end{cases} \tag{3.59}$$

while `fast` implies that

$$\mu = \begin{cases} \frac{\mu}{4} & \text{if } \mu > 1e^{-4} \\ \frac{\mu}{1.2} & \text{otherwise .} \end{cases} \tag{3.60}$$

Using each of the update strategies, we are left with four implementations:

1. LS: Linear search with a `slow` barrier update
2. LF: Linear search with a `fast` barrier update
3. CS: Curvilinear search with a `slow` barrier update
4. CF: Curvilinear search with a `fast` barrier update

3.4.2 A Test Problem Set

We tested the method on the `QPECgen` [JR99] test problem set. In [JR99], Jiang and Ralph presented a problem generator for deterministic quadratic programs with equilibrium constraints. Such problems possess a quadratic objective with polyhedral *first-level* constraints and complementarity *second-level* constraints.[6]

The quadratic program with complementarity constraints (QPCC) is given by

QPCC	$\underset{x,y}{\text{minimize}}$	$\frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix}$
	subject to	$Gx \leq a$
		$0 \leq y \perp Nx + My + q \geq 0,$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. The generator has some very desirable features in that the user can control problem size, degeneracy, condition number of P, monotonicity of P and convexity of f. We tested our implementations on a set of 20 problems from `QPECgen`. The specifications of the problems are $n = 2$ and $m = 5k$, where $k = 1, \ldots, 20$. The set of problems satisfies strict complementarity at the lower level.

3.4.3 Numerical Results

Figure 3.1 shows the performance profiles of the four algorithms constructed on the basis of major iterations. Performance profiles are discussed by Dolan and Moré [DM02] and summarized in

[6] Actually, the second-level constraints are affine variational inequalities, but we restrict the discussion to affine complementarity constraints.

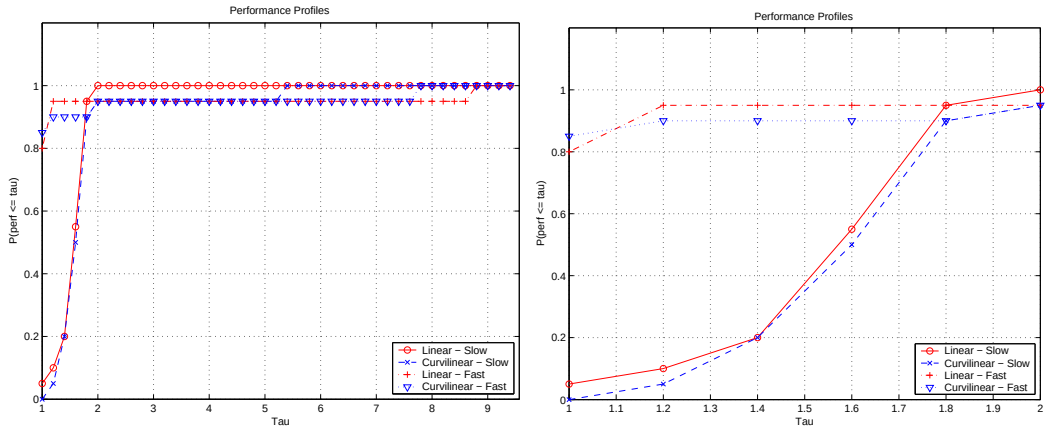

FIGURE 3.1 *Performance profiles for major iterations required by four methods*

TABLE 3.1
Comparing linear-slow and linear-fast

Dimension		Linear-Slow			Linear-Fast		
n	m	major	$\|kkt\|$	ρ	major	$\|kkt\|$	ρ
10	5	30	1.0e-4	1.0e+4	19	5.5e-5	1.0e+4
10	10	33	8.6e-5	1.0e+4	22	5.0e-3	1.0e+4
10	15	38	1.2e-5	8.1e+9	23	1.7e-6	1.8e+10
10	20	39	9.2e-6	6.1e+9	23	1.3e-5	1.5e+7
10	25	31	8.6e-5	1.0e+4	17	9.4e-5	1.0e+4
10	30	29	8.7e-5	1.0e+4	17	5.7e-5	1.0e+4
10	35	38	4.9e-5	3.5e+9	22	3.5e-5	6.7e+6
10	40	38	1.7e-5	3.8e+9	42	4.7e-5	1.0e+4
10	45	38	1.9e-5	2.7e+9	23	5.8e-6	1.0e+4
10	50	39	3.6e-5	2.4e+9	25	1.7e-5	1.8e+7
10	55	39	5.3e-6	2.2e+9	30	8.7e-5	2.4e+11
10	60	39	4.6e-5	2.0e+9	24	9.7e-5	1.0e+4
10	65	38	4.4e-5	1.9e+9	23	2.9e-5	4.1e+9
10	70	39	5.4e-5	1.8e+9	24	3.5e-5	3.3e+7
10	75	38	3.7e-6	1.6e+9	25	5.3e-5	2.2e+8
10	80	40	7.9e-5	1.5e+9	26	9.5e-5	4.9e+7
10	85	38	7.2e-5	1.4e+9	37	3.4e-6	2.8e+9
10	90	38	3.9e-5	1.4e+9	25	4.4e-5	1.0e+4
10	95	40	2.0e-5	1.3e+9	26	4.6e-5	2.8e+8
10	100	40	3.6e-5	1.2e+9	29	9.6e-5	5.7e+6

chapter 2.

The graph on the left in Figure 3.1 shows that the fast updates would solve 80% of the problems in the fewest major iterations. Moreover, when we consider methods that performed within a factor of 2 of the best method ($\tau = 2$), we find that the LS method is the only one that qualifies. Generally, it appears that the linesearch approaches do slightly better than the curvilinear approaches but this distinction may be a result of implementation issues. We also observe that the fast updates do not solve a small percentage of the problems even when we relax τ to 5. Tables 3.1 and 3.2 show detailed computational results of all the methods. It may be observed that the faster updates do much better in general. Also, the penalty parameters stay within $1e{+}11$ for the most part. The curvilinear search method compares well with the linear search for most problems. In some cases (such as $m = 85, 95$), the curvilinear search methods outperform the linesearch approach. We also note that the number of major iterations with the slow update is never more than double the

TABLE 3.2
Comparing curvilinear-slow and curvilinear-fast

Dimension		Curvilinear-Slow			Curvilinear-Fast		
n	m	major	$\|kkt\|$	ρ	major	$\|kkt\|$	ρ
10	5	30	1.0e-4	1.0e+4	19	5.5e-5	1.0e+4
10	10	33	8.6e-5	1.0e+4	23	3.9e-5	2.7e+10
10	15	38	1.2e-5	8.1e+9	26	5.1e-5	1.0e+4
10	20	39	9.2e-6	6.1e+9	23	1.3e-5	1.5e+7
10	25	31	8.6e-5	1.0e+4	17	9.4e-5	1.0e+4
10	30	29	8.7e-5	1.0e+4	17	5.7e-5	1.0e+4
10	35	38	4.9e-5	3.5e+9	22	3.5e-5	6.7e+6
10	40	16	3.9e+6	1.8e+14	73	3.4e-5	1.0e+4
10	45	38	1.9e-5	2.7e+9	23	5.8e-6	1.0e+4
10	50	39	3.6e-5	2.4e+9	25	1.7e-5	1.8e+7
10	55	39	5.3e-6	2.2e+9	30	8.7e-5	2.4e+11
10	60	39	4.6e-5	2.0e+9	24	9.7e-5	1.0e+4
10	65	38	4.4e-5	1.9e+9	23	2.9e-5	4.1e+9
10	70	39	5.4e-5	1.8e+9	24	3.5e-5	3.3e+7
10	75	34	2.9e-6	2.4e+9	24	1.4e-6	4.7e+7
10	80	40	7.9e-5	1.5e+9	26	9.5e-5	4.9e+7
10	85	34	5.9e-5	2.1e+9	32	3.6e-5	1.0e+4
10	90	38	3.9e-5	1.4e+9	25	4.4e-5	1.0e+4
10	95	35	8.9e-5	1.9e+9	24	7.8e-2	9.1e+8
10	100	40	3.6e-5	1.2e+9	29	9.6e-5	5.7e+6

iterations taken by the fast update.

Another observation is that the number of major iterations does not change significantly for a specific method. For instance, the Linear-Fast method takes 19 iterations for a problem with $m = 5$, but for the problem with $m = 100$, the number of major iterations is 29. (However a major iteration takes much more effort as the problem size increases.)

3.5 Contributions and Future Research

We present a regularization-based interior method for mathematical programs with complementarity constraints. Through the use of modified Newton and negative curvature directions, the algorithm generates a sequence of iterates that converges to second-order points. Existing algorithms for MPCCs only allow for convergence to first-order points.

The convergence of the algorithm is proved under weaker assumptions than in [MP03]. In particular, we only require that the non-slack primal variables be in a compact set and prove that the remaining dual and slack variables are bounded.

We also discuss a MATLAB-based implementation's results on the `QPECgen` test-problem set. We compare the performance of the curvilinear search with the linear search for fast and slow updates of the barrier parameter sequence. We observe that the performance of the curvilinear search compares well with that of the linear search. Specifically, for a particular update rule, the curvilinear search outperforms the linear search on only one of twenty test problems. For nearly 80% of the problems, the performance is identical.

There are several issues that we leave for future research:

- Theoretical issues: First, the work by Murray and Prieto [MP95a] proves that all iterates are in a compact set. This appears to be possible in the current setting and shall be investigated further. Furthermore, the current analysis has been carried out for a single barrier and penalty parameter. Ideally, there should be a vector of penalty and barrier parameters (see [Doy03]).

Third, questions of ill-conditioning have been left unanswered and shall be a focus of future work.

- Algorithmic issues: The implementation needs to be tested on a broader class of MPCCs, in particular the `macMPEC` test problem set [Ley00]. Second, a more detailed study of the results is required. In particular, we plan to examine specific problems for which the curvilinear and linesearch approach result in differing performance. Third, currently the construction of the basis for the null-space is achieved through a QR factorization. However, there are several better methods based on the sparse LU factorization (see [Doy03, GMS05]) that we plan to investigate further.

Chapter 4

A KKT Sampling Method for Stochastic MPCCs

4.1 Introduction

We consider the stochastic mathematical program with complementarity constraints (SMPCC)

SMPCC	$\underset{x,y_\omega,w_\omega}{\text{minimize}}$	$\mathbb{E}f_\omega(x, y_\omega, w_\omega)$
	subject to	$c_\omega(x, y_\omega, w_\omega) = 0,$
		$0 \leq y_\omega \perp w_\omega \geq 0 \quad \forall \quad \omega \in \Omega,$

where $f_\omega(.)$ and $c_\omega(.)$ represent the objective function and vector of constraint functions. The parametrization ω corresponds to realization ω, where $\omega \in \Omega$, the sample space. It is assumed that f_ω and c_ω are twice continuously differentiable over x, y_ω and w_ω, which lie in $\mathbb{R}^n, \mathbb{R}^m$ and $\mathbb{R}^m$ respectively. We assume that the probability space is given by $(\Omega, \mathcal{F}, \mathbb{P})$ and $\mathbb{P}$ is a discrete distribution.

The complementarity constraint $y_\omega \perp w_\omega$ implies that either $[y_\omega]_i$ or $[w_\omega]_i$ (or both) is (are) zero for $i = 1, \ldots, m$. Such problems arise in the modeling of Stackelberg equilibria, traffic equilbria and contact problems under uncertainty [LPR96]. The specification of uncertainty is through an "almost-sure" formulation in which there is one constraint for every realization ω of uncertainty, where $\omega \in \Omega$. The deterministic problem, known as a mathematical program with complementarity constraints (MPCC), is discussed in detail in the monographs by Luo et al. [LPR96] and Outrata et al. [OKZ98].

The solution of SMPCCs is plagued by two difficulties. First, the optimization problem does not have any feasible point that satisfies the inequality constraints strictly. This implies that the Mangasarian-Fromovitz constraint qualification (MFCQ) (and therefore the linear-independence constraint qualification (LICQ)) fails at every feasible point. Second, we make no assumption regarding the size of the discrete distribution $\mathbb{P}$. Therefore, the number of constraints in SMPCC may be arbitrarily large implying, that a direct application of NLP-based methods may not be sensible.

4.1.1 Early Research on Stochastic Programming

The last two decades have seen significant advancement in stochastic programming research (see Birge and Louveaux [BL97]). Consider a general two-stage stochastic programming problem in which a decision x is made in the first stage and a recourse decision y_ω is made in the second stage, contingent on x and the realization ω. The first-stage problem may be formulated as

$$\min_{x \in X} \quad (f(x) + \mathcal{Q}(x)),$$

where $x, X, f(x)$ and $\mathcal{Q}(x)$ represent the first-stage decision, the first-stage feasible region, first-stage cost and the second-stage recourse function, respectively. The recourse function $\mathcal{Q}(x)$ is the expectation of the random recourse function $\mathcal{Q}(x, \omega)$, which is the optimal value of an ω-scenario problem.

Under assumptions of convexity, we may approximate the recourse function $\mathcal{Q}(x)$ by a set of cuts. Van Slyke and Wets [VSW69] proposed such a method to solve stochastic linear programs. However, each cut required the solution of $|\Omega|$ dual problems. To alleviate the burden of computing $|\Omega|$ dual problems to add a single cut, Dantzig and Glynn [DG89], Higle and Sen [HS91] and Infanger [Inf94] suggested the use of sampling to construct the cuts (see Chapter 2 for more details).

One of the restrictions of such a class of methods is the reliance on convexity of the recourse function. In the case of two-stage stochastic nonlinear programming methods, convexity of the recourse function cannot be assumed. One possible approach is through the use of sample-average approximation (SAA) methods [Sha03]. Such methods rely on solving sampled versions of the problem to obtain estimators. Furthermore, under some assumptions, one may claim convergence of the estimators to the true solution at an appropriate rate. One of the shortcomings of this approach is that it assumes that large instances of the problem may be solved efficiently by existing methods. This does not appear to be the case for nonlinear programming.

4.1.2 Research on SMPCCs

The SMPCC rerpresents an instance of a stochastic equilibrium problem. Recently, there have been several efforts to address such problems. Lin and Fukushima [LCF03] discuss a smoothing method for stochastic programs with LCP constraints. Shapiro et al. [Sha03] discuss an SAA method. This work is an extension of earlier SAA research on stochastic programs and provides theory to show convergence of optimal values and solutions for a sampled problem to those of the true problem. It does *not* address the question of how one may solve the sampled problem efficiently.

We present a method that can deal with ill-posedness[1] in the constraints and the large number of constraints. To address the former, we rely on ideas discussed in Chapter 3. Dealing with the latter requires extending stochastic programming to the area of two-stage stochastic nonlinear programming.

The solution of SMPCCs requires the solution of stochastic nonlinear (albeit ill-posed) programs. Any suggested method should be scalable in that the computational burden should grow slowly

[1] Ill-posedness in this context refers to the fact that complementarity constraints do not satisfy the common regularity conditions at any feasible point.

with the sample-size. We propose a primal-dual method with some important modifications. First, the linearized KKT systems are sparse and nearly scenario-separable. This property is utilized in constructing a parallelizable method for obtaining the Newton steps. Second, we may take inexact Newton steps by using a sample of the distribution to construct a step. The sparsity of the reduced Hessian allows us to compute negative curvature directions efficiently. The use of such directions allows proof of convergence to a local minimizer.

The rest of this chapter is organized into five sections. In §4.2, we discuss our interior-point algorithm, focusing on how one may obtain the Newton and negative curvature directions. The use of sampling within the method is discussed in §4.3. The algorithm has been applied to a set of stochastic quadratic programs with complementarity constraints and the findings are discussed in §4.4.

4.2 An Interior Point Method for MPCCs

We begin by restating the problem SMPCC as a regularized problem SMPCC(γ):

$$
\begin{array}{lll}
\text{SMPCC}(\gamma) & \underset{x,y,w}{\text{minimize}} & \mathbb{E} f_\omega(x, w_\omega, y_\omega) \\
 & & c_\omega(x, y_\omega, w_\omega, \gamma) = 0 \\
 & & Y_\omega w_\omega + s^{cc}_\omega - \gamma e = 0 \\
 & \text{subject to} & y_\omega - s^{y}_\omega = 0 \\
 & & w_\omega - s^{w}_\omega = 0 \\
 & & s^{y}_\omega, s^{w}_\omega, s^{cc}_\omega \geq 0 \quad \forall \omega \in \Omega,
\end{array}
$$

where γ is a positive scalar parameter. By defining $d_\omega(.)$ as

$$
d_\omega(x, y_\omega, w_\omega, s_\omega; \gamma) = \begin{pmatrix} Y_\omega w_\omega + s^{cc}_\omega - \gamma e \\ y_\omega - s^{y}_\omega \\ w_\omega - s^{w}_\omega \end{pmatrix}, \tag{4.1}
$$

we obtain a simpler formulation:

$$
\begin{array}{lll}
\text{SMPCC}(\gamma) & \underset{x,y,w}{\text{minimize}} & \mathbb{E} f_\omega(x, w_\omega, y_\omega) \\
 & & c_\omega(x, y_\omega, w_\omega, \gamma) = 0 \\
 & \text{subject to} & d_\omega(x, y_\omega, w_\omega, s_\omega; \gamma) = 0 \\
 & & s_\omega \geq 0, \quad \forall \omega.
\end{array}
$$

For $\gamma > 0$, problem SMPCC(γ) is a well-posed nonlinear program; it satisfies the Mangasarian-Fromovitz constraint qualification. Specifically, the positivity of γ ensures that the interior of the inequality constraint is non-empty.

The idea is to solve a sequence of problems SMPCC(γ_k), where $\gamma_k \to 0$ (as in Chapter 3). Under some assumptions, the sequence of stationary points of SMPCC(γ_k) converges to a strong-stationary

point or a stationary point of SMPCC(0).

4.2.1 The Barrier Problem

The crucial idea in interior methods for optimization is that the inequality constraints and nonnegativity bounds are kept strictly satisfied. One such interior method uses the logarithmic function to replace inequality constraints by *barrier* terms in the objective. These terms tend to infinity if the iterates tend towards the boundary. The resulting problem is parameterized by γ and μ, the barrier parameter, and is denoted by SMPCC(γ, μ):

$$\begin{array}{lll} \text{SMPCC}(\gamma,\mu) & \underset{x,y,w,s}{\text{minimize}} & \mathbb{E} f_\omega(x, y_\omega, w_\omega) - \mu \sum_\omega \sum_i (\ln[s_\omega^y]_i + \ln[s_\omega^w]_i + \ln[s_\omega^{cc}]_i) \\ & \text{subject to} & \begin{array}{r} c_\omega(x, y_\omega, w_\omega) = 0 \\ d_\omega(x, y_\omega, w_\omega, s_\omega; \gamma) = 0 \quad \forall \omega. \end{array} \end{array}$$

We replace x by x_ω for all $\omega \in \Omega$, with constraints that say all x_ω are identical. These coupling constraints are often termed *nonanticipativity* constraints. We shall be working in the framework of discrete probability distributions, where ω may take on a finite number of realizations, say K. We shall drop the subscript ω and use (x_i, y_i, w_i) to specify that the variables correspond to scenario i. If π_j represents the probability of scenario j, then problem SMPCC(γ, μ) may be stated as

$$\begin{array}{lll} \text{SMPCC}(\gamma,\mu) & \underset{x,y,w}{\text{minimize}} & \sum_j \pi_j f_j(x_j, y_j, w_j) - \mu \sum_j \sum_i (\ln[s_j^y]_i + \ln[s_j^w]_i + \ln[s_j^{cc}]_i) \\ & & \begin{array}{rl} c_j(x_j, y_j, w_j) = 0, & \text{for } j = 1, \ldots, K \\ d_j(x_j, y_j, w_j; \gamma) = 0, & \text{for } j = 1, \ldots, K \\ x_1 - x_j = 0, & \text{for } j = 2, \ldots, K. \end{array} \\ & \text{subject to} & \end{array}$$

Next, we present the first-order KKT conditions for SMPCC.

4.2.2 The KKT Conditions

By labeling the coupling constraints as $q_j(x_1, x_j) = x_1 - x_j$ and $z_j = (x_j, y_j, w_j)$ and $s_j = (s_j^w, s_j^y, s_j^{cc})$, one may rewrite the barrier problem with the appropriate multipliers:

$$\begin{array}{lll} \text{SMPCC}(\gamma,\mu) & \underset{z,s}{\text{minimize}} & \sum_j \pi_j f_j(z_j) - \mu \sum_k \ln[s_j]_k \\ & & \begin{array}{rl} c_j(z_j) = 0 & : \beta_j,\ j = 1, \ldots, K \\ d_j(z_j, s_j) = 0 & : \xi_j,\ j = 1, \ldots, K \\ q_j(z_1, z_j) = 0 & : \zeta_j,\ j = 2, \ldots, K. \end{array} \\ & \text{subject to} & \end{array}$$

We may define a scenario-Lagrangian function of the barrier problem as

$$\begin{aligned}\mathcal{L}_1(z_j, s_j, \beta_j, \xi_j, \zeta_j) &= \pi_1 f_1 - \mu \ln(s_1) - c_1^T \beta_1 - d_1^T \xi_1 - \sum_j q_j^T \zeta_j, \\ \mathcal{L}_j(z_j, s_j, \beta_j, \xi_j, \zeta_j) &= \pi_j f_j - \mu \ln(s_j) - c_j^T \beta_j - d_j^T \xi_j - q_j^T \zeta_j, \quad j \neq 1.\end{aligned} \tag{4.2}$$

By introducing vectors v_j satisfying $S_j v_j = \mu e$, we may state the first-order KKT conditions as

$$\begin{aligned}\pi_j \nabla_{z_j} f_j - \nabla_{z_j} c_j^T \beta_j - \nabla_{z_j}^T d_j^T \xi_j - \sum_j \nabla_{z_j} q_j^T \zeta_j = 0 = r_j^z, \quad j = 1 \\ \pi_j \nabla_{z_j} f_j - \nabla_{z_j} c_j^T \beta_j - \nabla_{z_j}^T d_j^T \xi_j - \nabla_{z_j} q_j^T \zeta_j = 0 = r_j^z, \quad j \neq 1 \\ \pi_j \nabla_{s_j} f_j - \nabla_{s_j} c_j^T - \beta_j \nabla_{s_j}^T d_j^T \xi_j - \sum_j \nabla_{s_j} q_j^T \zeta_j - z_j = 0 = r_j^s \\ c_j(z_j) = 0 = r_j^c \\ d_j(z_j, s_j) = 0 = r_j^d \\ q_j(z_j) = 0 = r_j^q \\ S_j v_j - \mu e = 0 = r_j^v.\end{aligned} \tag{4.3}$$

4.2.3 The Linearized KKT Conditions

Newton's method may be applied to the set of equations (4.3). This system grows with the number of scenarios but still follows a particular structure. In particular, if we denote the jth scenario (both primal and dual) step as Δp_j, we may rewrite the system as

$$\begin{pmatrix} M_1 & & & & -\bar{I} & \dots & -\bar{I} \\ & M_2 & & & \bar{I} & & \\ & & \ddots & & & \ddots & \\ & & & M_K & & & \bar{I} \\ \bar{I} & -\bar{I} & & & & & \\ \vdots & & \ddots & & & & \\ \bar{I} & & & -\bar{I} & & & \end{pmatrix} \begin{pmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_K \\ \Delta \zeta_2 \\ \vdots \\ \Delta \zeta_K \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \\ r_2^q \\ \vdots \\ r_K^q \end{pmatrix}, \tag{4.4}$$

where

$$M_j = \begin{pmatrix} \mathcal{H}_{z_j z_j} & \mathcal{H}_{s_j z_j} & -\nabla_{z_j} c_j^T & -\nabla_{z_j} d_j^T & \\ \mathcal{H}_{z_j s_j} & \mathcal{H}_{s_j s_j} & & -\nabla_{s_j} d_j^T & -I \\ \nabla_{z_j} c_j & & & & \\ \nabla_{z_j} d_j & \nabla_{s_j} d_j & & & \\ & V_j & & & S_j \end{pmatrix} \text{ and } \Delta p_j = \begin{pmatrix} \Delta z_j \\ \Delta s_j \\ \Delta \beta_j \\ \Delta \xi_j \\ \Delta \zeta_j \\ \Delta v_j \end{pmatrix}.$$

Note that $\mathcal{H}_{z_j z_j} = \nabla^2_{z_j z_j} \mathcal{L}_j$ and $\mathcal{H}_{z_j s_j}, \mathcal{H}_{s_j z_j}$ and $\mathcal{H}_{s_j s_j}$ are defined accordingly. By eliminating Δv_j in each scenario subsystem, we obtain the system

$$\begin{pmatrix} \mathcal{H}_{z_j z_j} & \mathcal{H}_{s_j z_j} & -\nabla_{z_j} c_j^T & -\nabla_{z_j} d_j^T \\ \mathcal{H}_{z_j s_j} & \mathcal{H}_{s_j s_j} + S_j^{-1} V_j & & -\nabla_{s_j} d_j^T \\ \nabla_{z_j} c_j & & & \\ \nabla_{z_j} d_j & \nabla_{s_j} d_j & & \end{pmatrix} \begin{pmatrix} \Delta z_j \\ \Delta s_j \\ \Delta \beta_j \\ \Delta \xi_j \end{pmatrix} = - \begin{pmatrix} r_j^z \\ r_j^s + S_j^{-1} r_j^v \\ r_j^c \\ r_j^d \end{pmatrix}. \tag{4.5}$$

A compact symmetrized representation may then be obtained:

$$\begin{pmatrix} -\mathcal{H}_j & \nabla h_j^T \\ \nabla h_j & \end{pmatrix} \begin{pmatrix} \Delta u_j \\ \Delta \lambda_j \end{pmatrix} = \begin{pmatrix} -r_j^u \\ r_h^j \end{pmatrix}, \tag{4.6}$$

where

$$\mathcal{H}_j = \begin{pmatrix} \mathcal{H}_{z_j z_j} & \mathcal{H}_{s_j z_j} \\ \mathcal{H}_{z_j s_j} & \mathcal{H}_{s_j s_j} + S_j^{-1} V_j \end{pmatrix}, \quad r_j^h = \begin{pmatrix} r_j^c \\ r_j^d \end{pmatrix}, \quad \Delta u_j = \begin{pmatrix} \Delta z_j \\ \Delta s_j \end{pmatrix},$$
$$\Delta \lambda_j = \begin{pmatrix} \Delta \beta_j \\ \Delta \xi_j \end{pmatrix} \text{ and } r_j^u = \begin{pmatrix} r_j^z \\ r_j^s + V_j^{-1} r_j^v \end{pmatrix}.$$

The steps in v_j may be obtained by solving

$$S_j \Delta v_j = (r_j^v - V_j \Delta s_j).$$

Since the coupling constraints involve only the terms in x_j, they are unaffected by the elimination process. A solution of (4.4) may be obtained by a multi-phase approach. This was discussed in a linear programming context by Liu and Sun [LS04b] and subsequently by Bastin et al. [Bas04]. This is the subject of the next section.

4.2.4 A Multi-phase Approach

Directly solving the system (4.4) becomes increasingly difficult as the distribution grows in size. Instead, we propose a multi-phase approach to solving the problem. It involves first obtaining scenario directions $\Delta \bar{p}_j$ for each scenario under the assumption that $\Delta \zeta_j = 0$ for all j. Then, we obtain the direction $\Delta \zeta$ by employing the ideas of Schur complements. Lastly, we re-solve to obtain the actual directions Δp_j. We may rewrite (4.4) as

$$\begin{pmatrix} M_1 & & & & -\nabla_{p_1} q^T \\ & M_2 & & & -\nabla_{p_2} q^T \\ & & \ddots & & \vdots \\ & & & M_K & -\nabla_p q_K^T \\ \nabla_{p_1} q & \nabla_{p_2} q & \dots & \nabla_{p_K} q & \end{pmatrix} \begin{pmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_K \\ \zeta + \Delta \zeta \end{pmatrix} = - \begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_K \\ r_q \end{pmatrix}, \tag{4.7}$$

where

$$\nabla q = \begin{pmatrix} \bar{I} & -\bar{I} & & \\ \vdots & & \ddots & \\ \bar{I} & & & -\bar{I} \end{pmatrix}.$$

Then, if we assume that $(\zeta + \Delta\zeta) = 0$, this system is separable and may be solved for each j to obtain $\Delta\bar{p}_j$:

$$\begin{pmatrix} M_1 & & & \\ & M_2 & & \\ & & \ddots & \\ & & & M_K \end{pmatrix} \begin{pmatrix} \Delta\bar{p}_1 \\ \Delta\bar{p}_2 \\ \vdots \\ \Delta\bar{p}_K \end{pmatrix} = \begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_K \end{pmatrix}. \tag{4.8}$$

The coupling constraints may then be evaluated as $\sum_j q_j(\Delta\bar{p}_j)$, resulting in the system

$$\begin{pmatrix} M_1 & & & & -\nabla_{p_1} q^T \\ & M_2 & & & -\nabla_{p_2} q^T \\ & & \ddots & & \vdots \\ & & & M_K & -\nabla_{p_K} q^T \\ \nabla_{p_1} q & \nabla_{p_2} q & \dots & \nabla_{p_K} q & \end{pmatrix} \begin{pmatrix} \Delta\bar{p}_1 \\ \Delta\bar{p}_2 \\ \vdots \\ \Delta\bar{p}_K \\ 0 \end{pmatrix} = \begin{pmatrix} \bar{r}_1 \\ \bar{r}_2 \\ \vdots \\ \bar{r}_K \\ \sum_j q_j(\Delta\bar{p}_j) \end{pmatrix}. \tag{4.9}$$

By adding (4.9) to (4.7) and recalling that $r_q = \sum_j q_j(p_j)$, we obtain

$$\begin{pmatrix} M_1 & & & & -\nabla_{p_1} q^T \\ & M_2 & & & -\nabla_{p_1} q^T \\ & & \ddots & & \vdots \\ & & & M_K & -\nabla_{p_K} q^T \\ \nabla_{p_1} q & \nabla_{p_2} q & \dots & \nabla_{p_K} q & \end{pmatrix} \begin{pmatrix} \Delta\bar{p}_1 + \Delta p_1 \\ \Delta\bar{p}_2 + \Delta p_2 \\ \vdots \\ \Delta\bar{p}_K + \Delta p_K \\ \zeta + \Delta\zeta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \sum_{\ j} q_j(\Delta\bar{p}_j - p_j) \end{pmatrix}. \tag{4.10}$$

We may obtain $\Delta\zeta$ by solving the system

$$-(\sum_j (\nabla_{p_j} q M_j^{-1} \nabla_{p_j} q^T))(\zeta + \Delta\zeta) = \sum_j q_j(p_j - \Delta\bar{p}_j). \tag{4.11}$$

In general, the matrices M_j may not be invertible unless we modify the scenario-based reduced Hessians appropriately.

Theorem 4.1. *If*

$$M = \begin{pmatrix} \mathcal{H} & A^T \\ A & \end{pmatrix},$$

and Z is a basis for the null-space of A, then M is invertible if $Z^T \mathcal{H} Z$ is positive definite and A has full row rank.

Proof. See theorem 16.6 in [NW99]. ■

Modification of the reduced Hessians may be carried out by using a modified Cholesky factorization of the entire reduced Hessian system. We assume that these reduced Hessians have been made positive definite when proving the next result.

Lemma 4.2. *If the matrices M_j are modified such that all $\mathbf{Z}_j^T \mathcal{H}_j^c \mathbf{Z}_j$ are positive definite, then the M_j are invertible for $j = 1, \ldots, K$.*

Proof. Follows from positive definiteness of $\mathbf{Z}_j^T \mathcal{H}_j^c \mathbf{Z}_j$ and full row-rank of ∇h_j (see (4.6)). ■

We may then represent the system as

$$B = \begin{pmatrix} -M & \nabla q^T \\ \nabla q & \end{pmatrix} \text{ where } M = \begin{pmatrix} M_1 & & \\ & \ddots & \\ & & M_K \end{pmatrix} \text{ and } \nabla q = \begin{pmatrix} \nabla_{p_1} q & \ldots & \nabla_{p_K} q \end{pmatrix}.$$

Lemma 4.3. *B is invertible if $\nabla q^T M^{-1} \nabla q$ is invertible.*

Proof. See [CPS92]. ■

Theorem 4.4. *Let M^c be given by $M + \delta I$ such that $\mathbf{Z}_q^T M^c \mathbf{Z}_q \succ 0$, where $\mathbf{Z}_q$ represents a basis for the null-space for ∇q. Then B is invertible.*

Proof. This follows immediately from lemma 4.2 and theorem 4.1. ■

This theorem allows us to claim that system (4.10) is always solvable, if suitably modified.

4.2.5 A Negative Curvature Direction

The true reduced Hessian of the system has a sparse arrowhead structure as shown in figure 4.1. The structure of this matrix is relevant because we intend to compute points that satisfy second-order conditions of optimality. To this end, it is necessary to use directions of negative curvature. In the unconstrained case, such directions would only exist when the Hessian matrix is indefinite and would satisfy

$$p^T H p < 0. \tag{4.12}$$

An eigenvector associated with a negative eigenvalue would provide such a direction. Alternatively, p can be computed using the modified Cholesky factorization [GMW81]. Recall that for an indefinite matrix H, the modified Cholesky factorization may be stated as

$$H + E = R^T R, \tag{4.13}$$

where R is a nonsingular upper triangular matrix. If $i = \arg\min\{H_{ii}\}$, then

$$R d^c = e_i \tag{4.14}$$

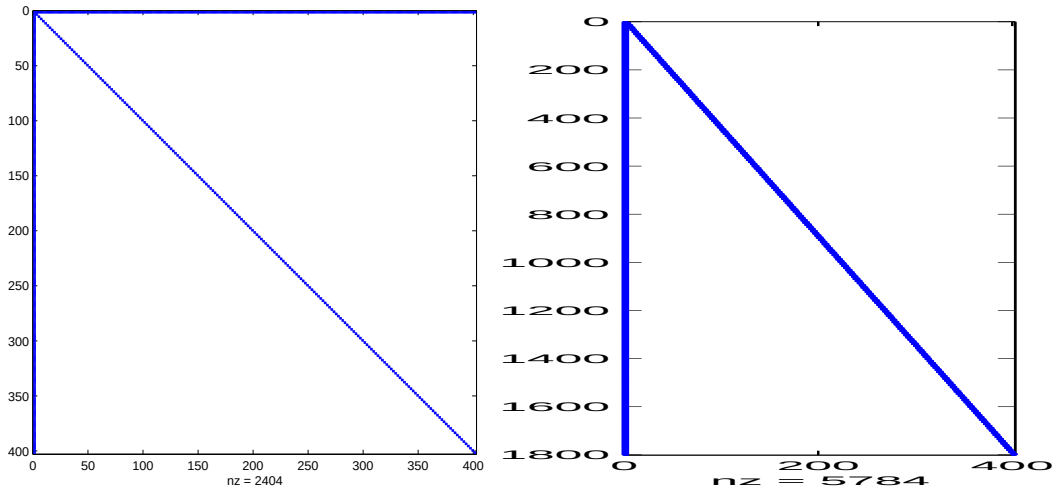

FIGURE 4.1 *Reduced Hessian and a basis for null-Space of hessian: $K = 200$, $n = 2$*

provides d^c, a direction of negative curvature. The reduced Hessian of the full KKT system may be written as

$$\bar{H} := \begin{pmatrix} U_0 & U_1^T & \dots & U_K^T \\ U_1 & W_1 & & \\ \vdots & & \ddots & \\ U_K & & & W_K \end{pmatrix}.$$

Lemma 4.5. *A modified Cholesky factorization of $\bar{H}$ can be obtained in the form*

$$\begin{pmatrix} U_0 & U_1^T & \dots & U_K^T \\ U_1 & W_1 & & \\ \vdots & & \ddots & \\ U_K & & & W_K \end{pmatrix} + \begin{pmatrix} E_0 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & E_K \end{pmatrix} = \begin{pmatrix} R_0^T & S_1^T & \dots & S_K^T \\ & R_1^T & & \\ & & \ddots & \\ & & & R_K^T \end{pmatrix} \begin{pmatrix} R_0 & & & \\ S_1 & R_1 & & \\ & & \ddots & \\ S_K & & & R_K \end{pmatrix}.$$

Moreover, we have

$$\begin{aligned} R_j^T R_j &= W_j + E_j, \qquad j = 1, \dots, K \\ R_j^T S_j &= U_j, \qquad\quad j = 1, \dots, K \\ U_0 + E_0 &= R_0^T R_0 + \sum_j S_j^T S_j. \end{aligned}$$

Note that the factors obtained in the previous Lemma are not the normal lower triangular Cholesky factors. Obtaining the scenario-based negative curvature directions is now straightforward. To ensure convergence from any starting point, a curvilinear search is carried out to determine the stepsize. In the current setting, we prescribe a single stepsize for each set of scenario variables. We do not discuss this any further in this chapter and instead refer the reader to chapter 3 for more details on the curvilinear search.

4.2.6 Specifying a Steplength

As mentioned in Chapter 3, nonlinear programming algorithms are equipped with a globalization strategy to ensure convergence from arbitrary starting points. Such a strategy requires the specification of a merit function that represents a metric of progress for the optimization algorithm. In nonlinearly constrained problems, a suitable merit function is given by a measure of primal and dual infeasibilities. In particular, we define the augmented Lagrangian merit function for two-stage stochastic nonlinear programming:

$$\mathcal{M}(z,s,\beta,\xi;\rho,\mu,\gamma) = \sum_{j=1}^{K}\left(\pi_j f_j - \mu\sum_i \ln[s_j]_i - c_j^T\beta_j - d_j^T\beta_j + \tfrac{1}{2}\rho(\|c_j\|^2 + \|d_j\|^2)\right) + \sum_{j=2}^{K}(\tfrac{1}{2}\rho\|q_j\|^2 - q_j^T\zeta_j). \tag{4.15}$$

A motivation for the use of negative curvature directions is provided in Chapter 3. We combine the modified Newton and negative curvature directions by using a steplength α obtained from a curvilinear search over the augmented Lagrangian merit function (4.15). Combining such directions with Newton directions, we obtain a new iterate in the form

$$\begin{pmatrix} p_{k+1} \\ \zeta_{k+1} \end{pmatrix} = \begin{pmatrix} p_k \\ \zeta_k \end{pmatrix} + \alpha_k^2 \begin{pmatrix} \Delta p_k \\ \Delta\zeta_k \end{pmatrix} + \alpha_k \begin{pmatrix} \Delta p_k^c \\ \end{pmatrix}, \tag{4.16}$$

where Δp_k^c is the direction of negative curvature.

4.2.7 Termination Criteria

Our method requires solving the barrier problems until the following termination criteria are satisfied:

$$\begin{aligned} \|\nabla_{z_j}\mathcal{L}(z_j^k, s_j^k, \beta_j^k, \xi_j^k, v_j^k)\|_\infty &\le \epsilon_k \\ \|\nabla_s\mathcal{L}(z_j^k, s_j^k, \beta_j^k, \xi_j^k, v_j^k)\|_\infty &\le \epsilon_k \\ \|c(z_j^k, s_j^k, \beta_j^k, \xi_j^k, v_j^k)\|_\infty &\le \epsilon_k \\ \|d(z_j^k, s_j^k, \beta_j^k, \xi_j^k, v_j^k;\gamma_k)\|_\infty &\le \epsilon_k \\ \|S_j^k v_j^k - \mu_k e\|_\infty &\le \epsilon_k \\ -\lambda_{\min}(\mathbf{Z}^T\mathcal{H}\mathbf{Z}) &\le \epsilon_k \\ \|q_j(z_1^k, z_j^k)\| &\le \epsilon_k, \end{aligned} \tag{4.17}$$

where ϵ_k is a stopping tolerance, with $\{\epsilon_k\} \to 0$. In practice, we may set $\epsilon_k = \min(\gamma_k, \mu_k)$. Algorithm 2 describes the steps of the primal-dual method for stochastic MPCCs.

Algorithm 2: A primal-dual method for stochastic MPCCs

Initial: $(p_0, \zeta_0; \mu_0, \rho_0, \gamma_0, \theta_0)$
$k = 1$
while $\|(r_k^P, r_k^D, r_k^{CC}, r_k^{NA})\|_\infty > \epsilon_1$ **or** $\lambda_{min}(\mathbf{Z}_k^T \mathcal{H}_k \mathbf{Z}_k) < \epsilon_2$ **do**
  while $\|(r_k^P, r_k^D, r_k^{CC}, r_k^{NA})\|_\infty > \tau\epsilon_1$ **or** $\lambda_{min}(\mathbf{Z}_k^T \mathcal{H}_k \mathbf{Z}_k) > -\tau_5\epsilon_2$ **do**
    Solve (4.4) in a multiphase fashion to obtain $(\Delta p, \Delta\zeta)$
    Raise ρ_k to ensure sufficient descent
    Obtain α_k satisfying Armijo criteria
    Take step $(p_{k+1}, \zeta_{k+1}) := (p_k + \alpha_k \Delta p_k, \zeta_k + \alpha_k \Delta\zeta_k)$
  $\gamma_{k+1} = u_k^\gamma \gamma_k$
  $\mu_{k+1} = u_k^\mu \mu_k$
  $k := k + 1$

4.2.8 Algorithm Statement

Algorithm 2 differs from a direct method in the fashion in which the direction is computed. Essentially, the multiphase approach of solving the linearized KKT system prevents the reduced Hessian of the Lagrangian from getting too large because it only works with the scenario systems. Moreover, the computation of the null-space basis may be done scenario-wise. This can be carried out in parallel as well for a scenario-based Jacobian matrix. However, there is still significant improvement that can be garnered. For instance, it is unnecessary to compute the entire linearized KKT system, far from the solution particularly because we are solving it for a fairly relaxed residual. We could obtain an incomplete solution that restricts the move to a subset of scenario variables or essentially makes a move in a subspace. This has an important advantage:

The incomplete solution is cheaper to compute because it is for a subset of the scenarios.

Using an incomplete solution has an impact on the linesearch technique that is adopted. Since the linearized KKT system is nearly separable, a parallelized linesearch may be adopted. This results in a steplength associated with each scenario variable. We do not use a parallel search in this thesis and instead use a single stepsize for all scenarios.

4.3 Introducing KKT Sampling

One of the observations already made is that at every major iteration, we take a series of steps along a combination of Newton and negative curvature directions so as to satisfy the subproblem termination criteria. However, these directions are the result of considering all the scenarios originating from what we consider the true distribution. This system represents a perturbation of the true system in two ways: in terms of the barrier and the regularization parameter.

In this section, we consider another perturbation within the same setting: one that results from taking a *sample* of the distribution. This sample would then be used to construct the KKT conditions and the directions would be taken so as to satisfy the linearization of the sampled system. Another perspective of such an approach is that we use an approximation of the random distribution in taking a step and then keep improving such an approximation as the solution is approached.

Several questions emerge from considering such an approach. What sample size does one start with and how rapidly does one increase the sample-size? Also, how do we ensure that the true residuals are decreasing rapidly enough, given that a particular scenario may not get sampled enough to ensure its residual is reduced sufficiently?

We begin by outlining the sampled KKT conditions and the resulting direction in section 4.3.1. In section 4.3.2, we discuss two techniques of sampling from the distribution. This is followed by a discussion of the impact of sampling on sufficient descent. Specifically, the failure of sufficient descent in the sampled system may be remedied by either increasing the penalty or increasing the sample size. Finally, in section 4.3.4, we propose a set of termination criteria that are also dependent on a function of the sample size. Note that we assume that the true distribution is discrete but may have a very large number of realizations.

4.3.1 Solving the Sampled KKT systems

The sampled KKT system requires a sample of $\bar{K}$ realizations, where $\bar{K} \leq K$. Since the reference scenario is 1 as specified by the coupling constraints, we always include this scenario. A sampled step would take linearizations for the sampled components and determine the sampled scenario directions. The directions corresponding to the unsampled scenarios will be set to zero. If one assumes without loss of generality that the first $\bar{K}$ scenarios appear in the sample, then $\Delta p_k, k \leq \bar{K}$ are defined by (4.18) and $\Delta p_k = 0, k > \bar{K}$.

$$\begin{pmatrix} M_1 & & & & -\bar{I} & \dots & -\bar{I} \\ & M_2 & & & \bar{I} & & \\ & & \ddots & & & \ddots & \\ & & & M_{\bar{K}} & & & \bar{I} \\ \bar{I} & -\bar{I} & & & & & \\ \vdots & & \ddots & & & & \\ \bar{I} & & & -\bar{I} & & & \end{pmatrix} \begin{pmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_{\bar{K}} \\ \Delta \zeta_2 \\ \vdots \\ \Delta \zeta_{\bar{K}} \end{pmatrix} = - \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{\bar{K}} \\ r_2^q \\ \vdots \\ r_{\bar{K}}^q \end{pmatrix}. \tag{4.18}$$

Since we set $\Delta p_j = 0, j > \bar{K}$ and $\Delta \zeta_j = 0, k > \bar{K}$, we may represent this solution as a solution to

$$\begin{pmatrix} \bar{M} & & \bar{B} & \\ & I & & \\ -\bar{B}^T & & & \\ & & & I \end{pmatrix} \begin{pmatrix} \Delta p_S \\ \Delta p_{NS} \\ \Delta \zeta_S \\ \Delta \zeta_{NS} \end{pmatrix} = - \begin{pmatrix} r_S \\ r_{NS} \\ r_S^q \\ r_{NS}^q \end{pmatrix}, \tag{4.19}$$

where $S = \{1, \dots, \bar{K}\}$ and $NS = \{\bar{K}, \dots, K\}$. Moreover, r_{NS} and r_{NS}^q are set to zero, and at the end of the major iteration, μ and γ are updated, and the sample size is also increased.

To ensure that the sample size is sufficiently large, we need to determine whether the *sampled* direction is a direction of descent for the original merit function (see section 4.2.6). If not, we proceed to increase the sample size or the penalty parameter ρ or possibly both.

4.3.2 Constructing a Sample

Given a distribution $\mathbb{P}$, the question arises of which scenarios to select from a sample space $|\Omega|$. The samples could be selected in accordance with their likelihood, as with Monte-Carlo sampling. However, such a technique does not utilize all information available to us. For instance, corresponding to each scenario is the residual of its KKT system, denoted by $\|r_j\|$. It may be preferable for the iterates to move along a direction that reduces the scenario residuals. Therefore, one could bias the distribution by using weights w_j, where

$$w_j := \frac{\|r_j\|}{\sum_j \|r_j\|}.$$

Clearly, along a trajectory of iterates in which all the scenarios have similar residuals, this strategy reduces to Monte-Carlo sampling and is referred to as `kktsampling`.

We may use a deterministic sampling strategy as well, in which we sort the scenarios by some parameter and select the first $\bar{K}$ scenarios. We denote this strategy as `kktsorting`.

A third possibility is to construct scenario-based augmented Lagrangian merit functions denoted by $\mathcal{M}_j$. The directions from the previous major iteration may be used to compute $\psi_j' := \nabla \mathcal{M}_j^T \Delta p_j$. These ψ_j' provide an indication of how much reduction can be made in a scenario-based merit function by a full step. These may again be used to bias the distribution in order to obtain a sample or be used to construct a deterministic sample. We do not employ such a strategy here but suggest it as a topic for further research.

4.3.3 Ensuring Global Convergence

Ensuring global convergence requires the use of a merit function. The modified Newton direction must be a direction of descent for the merit function. As discussed in Chapter 3, we may raise the penalty parameter to a suitable finite level. When the entire sample is used, the penalty parameter allows us to ensure that such a condition always holds.

However, for a step in the subspace defined by the sampled iterates, it may be that lack of *sufficient* descent could be cured by an increased sample size. Clearly, one would like to minimize the increase to ensure that the computational effort stays low. By the same token, if the penalty parameter is increased significantly, then the constraints residuals are made needlessly small early in the algorithm. These objectives are conflicting and have to be balanced in the implementation.

If the sufficient descent condition is satisfied, the linesearch may proceed.

4.4 Numerical Results

The algorithm of this chapter has been implemented in MATLAB 6.5 and tested on a Linux-based server (2 GB of RAM). The performance and scalability of the described barrier method is demonstrated on an extension of the `QPEC`gen [JR99] test problem set. In particular, we extend the `QPEC`gen MATLAB problem generator to the realm of *stochastic* quadratic programs with equilibrium constraints and refer to it as `SQPEC`gen.

4.4.1 SQPECgen: A Test Problem Set for Stochastic QPECs

In [JR99], Jiang and Ralph presented a problem generator for deterministic quadratic programs with equilibrium constraints. Such problems possess a quadratic objective with polyhedral *first-level* constraints and complementarity *second-level* constraints:

$$\text{QPCC} \qquad \begin{array}{ll} \underset{x,y}{\text{minimize}} & \frac{1}{2}\begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c \\ d \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} \\ \text{subject to} & Gx \leq a \\ & 0 \leq y \perp Nx + My + q \geq 0, \end{array}$$

where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$.[2] The generator has some very desirable features in that the user can control problem size, degeneracy, condition number of P, monotonicity of P and convexity of f. The stochastic generalization of the QPCC may be best illustrated by an application of equilibrium problems. The deterministic QPCC models a Stackelberg equilibrium in which the *leader* makes a decision "x" subject to equilibrium in the followers decisions "y". This equilibrium condition is

$$0 \leq y \perp Nx + My + q \geq 0.$$

One motivation for a stochastic QPCC arises from the possibility that the decisions made by followers are contingent on the realization of some uncertainty given by ω. Specifically, $\omega \in \Omega$ and for each realization ω, one may articulate an equilibrium constraint:

$$0 \leq y_\omega \perp N_\omega x + My_\omega + q_\omega \geq 0.$$

As a result, the follower's decision is based on ω. The objective function is then modified to an expectation function:

$$f(x,y) := \mathbb{E} f(x, y_\omega) := \mathbb{E}_\omega \left[\frac{1}{2} \begin{pmatrix} x \\ y_\omega \end{pmatrix}^T \begin{pmatrix} P_{xx} & P^\omega_{xy} \\ P^\omega_{yx} & P^\omega_{yy} \end{pmatrix} \begin{pmatrix} x \\ y_\omega \end{pmatrix} + \begin{pmatrix} c \\ d_\omega \end{pmatrix}^T \begin{pmatrix} x \\ y_\omega \end{pmatrix} \right].$$

If we assume that the distribution of ω is finite and takes on K realizations, then the expectation function may be rewritten as

$$\frac{1}{2} \begin{pmatrix} x \\ y_{\omega_1} \\ \vdots \\ y_{\omega_K} \end{pmatrix}^T \begin{pmatrix} P_{xx} & P_{xy_{\omega_1}} & \cdots & P_{xy_{\omega_K}} \\ P_{y_{\omega_1}x} & \pi_1 P_{y_{\omega_1}y_{\omega_1}} & & \\ \vdots & & \ddots & \\ P_{y_{\omega_K}x} & & & \pi_K P_{y_{\omega_K}y_{\omega_K}} \end{pmatrix} \begin{pmatrix} x \\ y_{\omega_1} \\ \vdots \\ y_{\omega_K} \end{pmatrix} + \begin{pmatrix} c \\ \pi_1 d_{\omega_1} \\ \vdots \\ \pi_K d_{\omega_K} \end{pmatrix}^T \begin{pmatrix} x \\ y_{\omega_1} \\ \vdots \\ y_{\omega_K} \end{pmatrix},$$

[2] Since we only use complementarity constraints in constructing the problem, we refer to the problem as a quadratic program with complementarity constraints (QPCC) (and not a quadratic program with equilibrium constraints (QPEC)).

where π_j is the probability of scenario j. The stochastic QPCC may be formulated as

SQPCC	$\underset{x,y_j}{\text{minimize}}$	$f(x,y)$
	subject to	$Gx \leq a$
		$0 \leq y_{\omega_j} \perp N_{\omega_j} x + M y_{\omega_j} + q \geq 0, \quad j = 1, \ldots, K.$

4.4.2 Computational Results

We present some aspects of the performance of the stochastic primal-dual code in this section. Two facets of the algorithm are discussed here:

1. Scalability: The growth in computational effort with the size of Ω. Also of interest is the difference between serial and parallel CPU time.

2. Performance profiles: We present performance profiles to demonstrate the difference in performance for three algorithms: `full-sample`, `kktsampling` and `kktsorting`. The first of these refers to a strategy in which the entire sample size is always used, while the second and third are described in section 4.3.2.

Any method for the solution of stochastic optimization problems needs to be relatively insensitive to growth in the size of the discrete distribution. This is of immense relevance when one models uncertainty using an "almost-sure" formulation. In primal-dual methods for the solution of such problems, the computationally burdensome step is the solution of the linearized KKT system. If one solves this directly, the effort may grow with the square of the number of realizations K.[3] Therefore, it is essential to introduce a decomposition to ensure that a direct solve is only carried out on the smaller scenario systems (which do not grow in size). Figure 4.2 shows that the growth in computational effort is linear. Specifically, it takes approximately 11 times as long to solve the 150 scenario case as it takes for the 10 scenario case. This ratio drops to a little less than 4 when one uses parallel CPU time. Note that we assume that two of the phases of the multiphase effort are carried out in parallel when measuring parallel CPU time.

The use of sampling in the early iterations results in some computational benefit. Figure 4.3 shows performance for the three methods discussed earlier. The construction of these profiles is discussed by Dolan and Moré [DM02] and is summarized in chapter 3. The profiles were constructed on the basis of serial CPU time. We find that the `kktsampling` method performs best while the `kktsorting` method is not significantly behind.

Tables 4.1 and 4.2 provide further details of the performances of the three algorithms.

4.5 Contributions and Future Research

We present one of the first methods for the solution of two-stage stochastic mathematical programs with complementarity constraints. In fact, even two-stage stochastic nonlinear programming has been a little-studied topic over the years. We believe that the customization of primal-dual methods

[3] Note that this assumes that the scenario-based KKT systems are sparse.

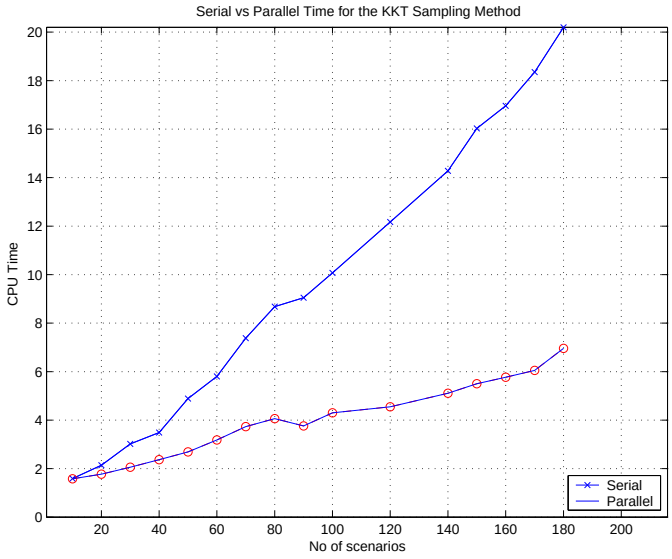

FIGURE 4.2 *The KKT sampling method: scalability and serial vs. parallel*

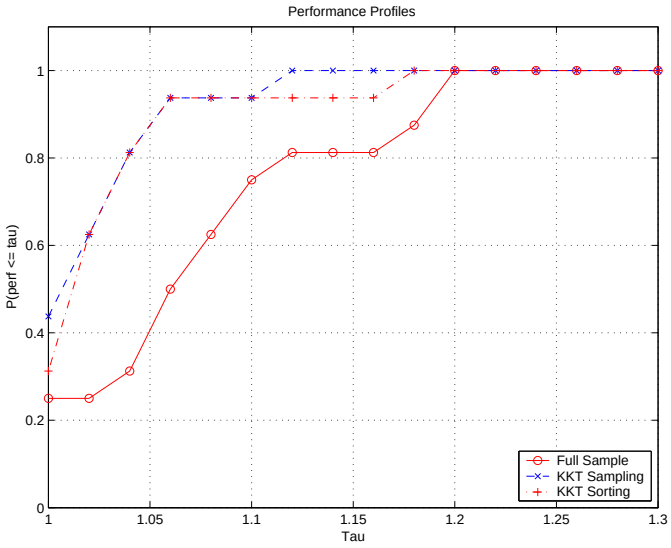

FIGURE 4.3 *Performance profiles*

TABLE 4.1
Comparision of serial and parallel times

$\lvert\ \Omega\ \rvert$	full-sample			
K	major	$\|kkt\|$	serial	parallel
10	94	1.3e-6	1.47	1.46
20	94	3.8e-6	2.26	1.82
30	94	8.3e-7	2.78	2.02
40	94	1.7e-6	3.73	2.48
50	105	2.5e-6	5.25	2.69
60	107	1.8e-6	5.95	3.23
70	133	1.1e-6	8.63	4.15
80	120	6.1e-6	8.87	3.96
90	119	8.1e-6	10.36	3.84
100	132	8.2e-6	11.81	4.75
120	118	1.1e-6	12.39	4.70
140	120	5.2e-6	15.11	5.33
150	118	3.9e-6	16.01	5.56
160	119	9.9e-7	18.16	6.00
170	134	6.3e-6	21.27	7.27
180	119	9.3e-7	20.02	6.74

TABLE 4.2
Comparision of `kktsampling` *and* `kktsorting` *strategies*

$\lvert\ \Omega\ \rvert$	kktsampling				kktsorting			
K	major	$\|kkt\|$	serial	parallel	major	$\|kkt\|$	serial	parallel
10	94	1.3e-6	1.59	1.58	94	1.3e-6	1.55	1.46
20	93	3.3e-6	2.14	1.77	93	3.3e-6	2.33	1.82
30	94	8.3e-7	3.02	2.06	94	8.3e-7	2.94	2.02
40	94	1.1e-6	3.49	2.37	94	1.1e-6	3.83	2.48
50	105	2.5e-6	4.89	2.69	105	2.5e-6	4.77	2.69
60	106	3.0e-6	5.80	3.18	106	3.0e-6	6.02	3.23
70	118	1.2e-6	7.38	3.73	118	1.2e-6	7.46	4.15
80	124	5.3e-6	8.68	4.06	124	5.3e-6	9.45	3.96
90	119	8.1e-6	9.05	3.76	119	8.1e-6	9.63	3.84
100	120	5.8e-6	10.07	4.30	120	5.8e-6	10.43	4.75
120	118	1.1e-6	12.17	4.55	118	1.1e-6	12.02	4.70
140	120	5.2e-6	14.28	5.11	120	5.2e-6	14.56	5.33
150	122	6.8e-6	16.03	5.50	122	6.8e-6	16.02	5.56
160	119	9.9e-7	16.96	5.77	119	9.9e-7	17.13	6.00
170	118	1.4e-6	18.35	6.05	118	1.4e-6	18.20	7.27
180	127	6.3e-6	20.20	6.96	127	6.3e-6	20.33	6.74

to handle stochasticity is of particular relevance because of the broader applicability of primal-dual methods. For instance, primal-dual methods have been used to solve complementarity problems and variational inequalities [FP03]. They have also been effective in obtaining solutions to nonlinear integer programs [Ng03].

We extend the ideas presented for deterministic MPCCs in chapter 3. An important modification is the multiphase approach for the solution of the linearized KKT system. This takes advantage of the structure of the problem and ensures that the computational effort grows slowly. A negative curvature direction is also obtained for each scenario, requiring us to use a curvilinear search to obtain an appropriate stepsize.

To further ease the computational burden, we may use a sample of the scenarios. There are different ways to construct such a sample and we discuss two approaches. The first biases the distribution using the residuals of the scenario KKT systems. The second is a deterministic method that sorts the scenarios based on the residuals and picks the top $\bar{K}$.

We construct a new test problem set based on stochastic QPCCs for the purpose of testing a MATLAB implementation. It is observed that the biased sampling approach using the KKT residuals performs better than the other methods. Also, we find that the computational effort grows linearly with the number of scenarios.

There are several issues that shall be the focus of future research:

- The current implementation does not employ directions of negative curvature. We need to extend the implementation to ensure convergence to second-order points.

- Currently, the linesearch employed specifies a single stepsize for all scenario variables. Conceivably, using different steplengths for each scenario would prove advantageous from a computational standpoint. We plan to extend the existing theory to allow for a parallelized linesearch.

- One of the shortcomings of the algorithm is that it requires solution of the linearized KKT conditions with all the scenarios, as the solution is approached. For large sample sizes, this is difficult. One possibility is to use a fixed sample size at each major iteration. In such a case, the `kktsampling` approach should perform well. However, these ideas need further research. In particular, can one relate the norm of the infeasibility in the KKT system to the fixed sample size employed at each iteration?

Chapter 5

Forward Contracting under Uncertainty

5.1 Introduction

In a Nash-Stackelberg game, agents compete in a Nash manner while being Stackelberg leaders with respect to a set of second-period players (called followers). In particular, first-period decisions are made contingent on an equilibrium in the second period. For instance, when agents have access to spot and forward markets, forward decisions are made contingent on spot-market equilibrium. In his seminal work, Allaz [All92] solves for a Nash-Stackelberg equilibrium in which participants in the forward market maximize profit subject to equilibrium in the spot market. Under assumptions of convexity in the spot market problem, the first-order optimality conditions are sufficient and the forward market problem is to maximize profit subject to these conditions.

The resulting optimization problem falls under the category of hard nonconvex problems called mathematical programs with complementarity constraints[1] or MPCCs [LPR96] (see appendix for a statement of such problems). But defining a Nash equilibrium in forward decisions is a far more difficult proposition because it requires solving for an equilibrium in which each agent solves an MPCC. The resulting equilibrium is called a Nash-Stackelberg equilibrium because it requires determining a Nash equilibrium in Stackelberg agents. Its solution is neither guaranteed to exist nor known to be unique. Algorithms for obtaining such problems are notoriously dependent on an initial solution [HMP00, YOA04] and currently have no global convergence theory.

In [YOA04], Yao et al. formulate an equilibrium problem with Stackelberg players using a comprehensive model of the electrical network. Such a problem is often referred to as an equilibrium problem with complementarity constraints or an EPCC. A common heuristic used to solve such problems is an iterative method that passes the decisions of one agent to the next in a round-robin fashion. Each agent's decisions are decided by solving an MPCC. Such a heuristic may not always converge. Moreover, the existence of equilibria in the original EPCC may not be taken for granted. Furthermore, the scalability of such a methodology to accommodate uncertainty in the system requires efficient methods to solve stochastic MPECs (see Chapter 4.)

[1] Complementarity constraints are generally of the form $0 \leq G(x) \perp H(x) \geq 0$, where $\perp$ implies that $[G(x)]_i[H(x)]_i = 0$ for all i.

From the earlier discussion, one may notice that the sequential decision-making leads to difficulty because it requires making first-period decisions subject to complicating nonconvex constraints. We propose a different approach that is based on each agent playing a *single-period* game in forward and recourse-spot decisions. We term such a game a *simultaneous stochastic Nash* (SSN) game because the decisions are made simultaneously. The games are constructed from stochastic programs; in particular, we make a recourse-spot decision for each realization of uncertainty in the spot-market. This has several consequences:

1. It introduces robustness in our decision-making process by introducing stochasticity in the second-stage parameters.

2. Questions of existence and uniqueness questions may be answered for such a formulation. More importantly, we may use these ideas to provide an important existence result for Nash-Stackelberg equilibria.

3. The resulting equilibria are obtainable by a scalable globally convergent method.

Excluding the introduction, this chapter has 6 sections. §5.2 introduces the forward market model proposed by Allaz [All92] and contrasts it with a complementarity-based formulation resulting from a simultaneous Nash game. §5.3 generalizes this framework to an oligopolistic two-period electricity market model. In particular, we provide existence theory for the equilibria associated with the simultaneous Nash game. The resulting stochastic complementarity problem is discussed under different risk-neutrality assumptions.

In §5.4, an iterative decomposition method is presented for solving such a class of stochastic equilibrium problems. An important characteristic of stochastic programming algorithms is their ability to scale well with the number of scenarios. We demonstrate this property for a set of problems in §5.5. We provide some computational evidence of the use of sampling within the algorithm. Since the discussion of forward markets has been carried out in the context of electricity markets, some insight is provided from a 6-node network in a variety of settings and different assumptions of stochasticity. We also show how the obtained equilibria are also Nash-Stackelberg equilibria (NSE).[2]

5.2 Modeling Spot-Forward Markets

Allaz [All92] provides a framework for analyzing spot-forward decisions in a general market. We provide a description of this model and the resulting Nash-Stackelberg equilibrium. Then, we provide a formulation for the SSN game and compare the solutions from each model. We use the context of electricity markets without loss of generality.

Consider a simple two-node market in which there are n generating firms operating at node 1. It is assumed that each firm sells as much as it generates. Each firm may sell its power at node 2

[2] Nash-Stackelberg equilibria refer to Nash equilibria between a set of agents, some of whom could be Stackelberg leaders with respect to some set of followers.

across a transmission line connecting the two nodes. The price of power at node 2 is given by

$$p_s^\omega := a_s^\omega - \sum_i g_i^\omega,$$

where g_i^ω is the generation level of firm i for a realization ω and a_s^ω is the random intercept of the price function. Specifically, ω lies in the sample-space Ω. The generation cost for firm i is assumed to be linear and is given by $c_i g_i^\omega$. The price of forwards is endogenously determined by a price function based on forward decisions of all the firms. It is defined by

$$p_f := a_f - \sum f_i,$$

where f_i is the forward transaction of firm i and a^f is the intercept of forward-price function. The expected spot and forward-market profit associated with firm i is denoted by π_i and is defined by

$$\pi_i := p_f f_i + \mathbb{E}\{p_s^\omega(g_i^\omega - f_i) - c_i g_i^\omega\}. \tag{5.1}$$

If we assume perfect foresight in the specification of forward prices, then we have $p_f = \mathbb{E}p_s^\omega$ and the resulting profit function π_i may be written as

$$\pi_i := \mathbb{E}_\omega\{(p_s^\omega - c_i)g_i^\omega\}.$$

5.2.1 The Spot-Market Equilibrium

In the spot-market, under realization ω, agent i maximizes his profit given forward positions f_i and the generation decisions of all other agents (given by g^{-i})[3], as shown by the following parametric optimization problem B_i^ω:

B_i^ω	$\underset{g_i^\omega \geq 0}{\text{maximize}} \;\; (p_s^\omega(g_i^\omega - f_i) - c_i g_i^\omega).$

We define the spot-market equilibrium as follows.

Definition 5.1. *Given a set of forward positions $(f_1, \ldots, f_n)$, the spot-market equilibrium is given by $(g_1^*, \ldots, g_n^*)$, where $g_i^* = (g_i^\omega)^*$ for all $\omega \in \Omega$, and $(g_i^\omega)^*$ solves B_i^ω. Furthermore, the generation decisions are in accordance with the risk-neutrality constraint: $a_f - \sum_i f_i^* = \mathbb{E}_\omega(a_s^\omega - \sum_i (g_i^\omega)^*)$.*

Since this is a convex problem in g_i^ω, the equilibrium point is given by the following stochastic mixed-linear complementarity problem LCP_1:

LCP_1	$0 \leq g_i^\omega \perp 2g_i^\omega + \sum_{j \neq i} g_j^\omega + f_i + (c_i - a_s^\omega) \geq 0, \qquad \forall i, \omega$ $a_f - \sum_j f_j = \mathbb{E}_\omega(a_s^\omega - \sum_j g_j^\omega).$

Note that in LCP_1, the first system refers to the first-order optimality conditions of the scenario-based spot-market problem. The equality constraint prescribes that the expected spot-price is equal

[3]The notation g^{-i} refers to the generation decisions of all agents except agent i.

to the forward price.

The following result proves that for every set of forward decisions, a unique spot-market equilibrium exists. Note that if the equilibrium problem could be written as an LCP, we could use some matrix properties to show existence and uniqueness. However, in this particular case, we use a different approach given that we have linear constraints as well.

Proposition 5.2. *For a given set of forward decisions, the solution to the spot-market equilibrium (definition 5.1) always exists.*

Proof. We begin by stating some elementary results concerning fixed-point theory and linear complementarity problems. Recall that the solution of a linear complementarity problem

LCP	$0 \leq z \perp Mz + q \geq 0$

may be stated as

$$\min(z, Mz + q) = 0.$$

Let $g(z) := \min(z, Mz + q)$. Then the solution of the zero-finding problem may be cast as a fixed-point problem $h(z) = z$, where $h(z) = z - g(z)$. Then, $h(z)$ is given by

$$h(z) = \max(0, (I - M)z - q).$$

The complementarity constraint in LCP_1 may be written as

$$0 \leq g_i^\omega \perp g_i^\omega + (\sum_j g_j^\omega - a_s^\omega) + f_i + c_i \geq 0, \qquad \forall i, \omega.$$

One possible solution to the under-determined system $a_f - \sum_j f_j = \mathbb{E}_\omega(a_s^\omega - \sum_j g_j^\omega)$ is given by

$$(a_s^\omega - \sum_j g_j^\omega) = a_f - \sum_j f_j, \quad \forall \omega \in \Omega.$$

Therefore, the complementarity system becomes

$$0 \leq g_i^\omega \perp g_i^\omega + (\sum_j f_j - a_f) + f_i + c_i \geq 0, \qquad \forall i, \omega.$$

But a solution to such a problem always exists and is given by

$$g_\omega^i = \max(0, -(\sum_j f_j - a_f) - f_i - c_i), \quad \forall \omega, i.$$

■

5.2.2 A Nash-Stackelberg Equilibrium (NSE)

The Stackelberg approach solves for an equilibrium in forward contracts subject to equilibrium in the spot-market (as specified by LCP_1). The ith firm then solves G_i in g and f_i while taking all

the other forward positions f^{-i} as inputs:

$\mathrm{G}_i(f^{-i})$	$\underset{g\geq 0, f_i}{\text{maximize}}$	$\mathbb{E}_\omega\{(p_s^\omega - c_i)g_i^\omega\}$
	subject to	$\mathrm{LCP}_1(g, f_i; f^{-i})$.

The problem G_i is an ill-posed nonlinearly constrained optimization problem: it contains a nonsmooth equilibrium or complementarity constraint. The resulting problem is often called a mathematical program with complementarity (or equilibrium) constraints or MPCC (or MPEC) [LPR96]. Apart from the complementarity constraint being nonconvex, it also lacks an interior, implying that the Mangasarian-Fromovitz constraint qualification does not hold at any feasible point. The resulting Nash equilibrium problem is defined as follows.

Definition 5.3. *($EPCC_1$): The Nash-Stackelberg equilibrium in f_i is an equilibrium in forward decisions $\{f_1^*, \ldots, f_n^*\}$, where f_i^* solves the Stackelberg problem $G_i(f^{-i,*})$:*

$$f_i^* = SOL(G_i(f^{-i,*})), \quad i = 1, \ldots, n. \tag{5.2}$$

Such a problem is a multi-leader multi-follower game and has been recently termed an equilibrium problem with equilibrium (complementarity) constraints or an EPEC (or EPCC). This leads us to the natural notions of *EPCC-feasibility* and *EPCC-optimality*.

Definition 5.4.

1. *The vector (f, g) is EPCC-feasible if (f, g) is a solution of LCP_1.*
2. *The vector (f, g) is EPCC-optimal if (f_i, g) is a strong-stationary[4] point of $G_i(f^{-i})$ for all i.*

These notions are important in helping us characterize the equilibria obtained by solving related games. There are no globally convergent algorithms for such problems. Other commonly used approaches include a Jacobi-iteration approach that solves each generator's problem and passes the solution to the next generator with the hope that the iterates converge to an equilibrium point. Scholtes [Sch01] shows that such approaches may result in cycling. However, we use this approach to solve EPCC_1 in §5.2.4.

5.2.3 A Simultaneous Stochastic Nash Equilibrium (SSNE)

The nonconvexity and ill-posedness of each generator's problem in the previous approach is problematic. It prevents the construction of globally convergent algorithms for finding an equilibrium. Moreover, existence and uniqueness answers are less easily provided.

An easier problem arises if we remove the restriction that the first-period decisions should be subject to a complementarity constraint. Instead, we require that agents make forward and recourse-based spot decisions simultaneously. The intention is to find an equilibrium point in forward and spot-decisions.

[4] We postpone defining *strong-stationarity* till section 5.2.5.

It is important to consider the effect on the structure of the problem. In the NSE framework, agents make decisions in two stages. In the SSNE framework, the decisions are made in a single period yet agents solve recourse problems, providing an implicit two-stage structure to the decision-making. More importantly, such a structure does not adversely impact obtaining an algorithm as we shall show. Moreover, the SSNE has a relationship with the original EPCC: it is an *EPCC-feasible* solution. Under some circumstances, it may even be an NSE or an *EPCC-optimal* solution.

We term this game a simultaneous stochastic Nash game and term the equilibrium a simultaneous stochastic Nash equilibrium or SSNE and define it as follows.

Definition 5.5. *Simultaneous Stochastic Nash Equilibrium (SSNE): Let $(g_i^\omega)^*$ for all $\omega \in \Omega$ be denoted by g_i^*. Then a simultaneous stochastic Nash equilibrium is defined as a set of $(g_i^*, f_i^*), i = 1, \ldots, n$, where firm i solves a spot-market problem P_i^ω for each $\omega \in \Omega$ and a forward-market problem P_i^f simultaneously. The two problems are defined as*

$$P_i^\omega \qquad \underset{g_i^\omega \geq 0}{\text{maximize}} \;\; (p_s^\omega(g_i^\omega - f_i) - c_i g_i^\omega)$$

and

$$P_i^f \qquad \underset{f_i}{\text{maximize}} \;\; (p^f - \mathbb{E}_\omega p_s^\omega) f_i.$$

However, if one imposes a risk-neutrality constraint $p_i^f = \mathbb{E}_\omega p_s^\omega$, it suffices to consider a game in which agent i solves

$$\mathrm{P}_i^\omega \qquad \begin{array}{ll} \underset{f_i, g_i^\omega \geq 0}{\text{maximize}} & (p_s^\omega(g_i^\omega - f_i) - c_i g_i^\omega) \\ \text{subject to} & p^f = \mathbb{E} p_s^\omega. \end{array}$$

The existence of an equilibrium to this game may be proved constructively.

Lemma 5.6. *The Nash equilibrium problem in which generator i simultaneously solves P_i^f and P_i^ω for each $\omega \in \Omega$, has a solution given by $f = \mathbb{E}_\omega(g^\omega - q_f/n)$, where g_ω is the unique solution of an LCP and $q_f := a_f - \mathbb{E}_\omega a_s^\omega$.*

Proof. We shall prove this constructively by providing a solution for LCP_1. The system

$$\mathbb{E} e^T g^\omega - e^T f + a_f - \mathbb{E}_\omega a_s^\omega = 0$$

is underdetermined and one solution to this system may be obtained by stating that

$$e^T f = \mathbb{E}_\omega e^T g^\omega + a_f - \mathbb{E}_\omega a_s^\omega = e^T \mathbb{E}_\omega(g^\omega - q_f/n),$$

where $q_f := a_f - \mathbb{E}_\omega a_s^\omega$. Therefore, a feasible solution to the equality constraints is $f = \mathbb{E}_\omega(g^\omega -$

q_f/n). It suffices to show that the resulting complementarity problem

$$0 \leq g^\omega \perp Mg^\omega + \mathbb{E}_\omega(g^\omega - q_f/n) + q^\omega \geq 0, \forall\omega \tag{5.3}$$

is solvable, where $M = I + ee^T$. This may be expanded as

$$0 \leq \begin{pmatrix} g^{\omega_1} \\ g^{\omega_2} \\ \vdots \\ g^{\omega_n} \end{pmatrix} \perp \begin{pmatrix} (M + p_{\omega_1} I) & p_{\omega_2} I & \dots & p_{\omega_n} I \\ p_{\omega_1} I & (M + p_{\omega_2} I) & \ddots & \vdots \\ \vdots & \ddots & \ddots & p_{\omega_n} I \\ p_{\omega_1} I & \dots & p_{\omega_{n-1}} I & (M + p_{\omega_n} I) \end{pmatrix} \begin{pmatrix} g^{\omega_1} \\ g^{\omega_2} \\ \vdots \\ g^{\omega_n} \end{pmatrix} + \begin{pmatrix} \bar{q}^{\omega_1} \\ \bar{q}^{\omega_2} \\ \vdots \\ \bar{q}^{\omega_n} \end{pmatrix} \geq 0$$

$$\text{or} \quad 0 \leq g \perp \mathbf{M}g + \mathbf{q} \geq 0, \tag{5.4}$$

where $\mathbf{M}$ is clearly a P matrix (a matrix with positive principal minors). This in turn allows us to claim that the resulting solution of (5.4) is unique. ■

In the EPCC approach, generator i may control f_i and g while being given f^{-i}. In this approach, however, one may obtain g and f by solving the mixed LCP given by LCP_1 directly. The following trivial result relates the two equilibria.

Theorem 5.7. *If (g^{LCP}, f^{LCP}) solves LCP_1, then the SSNE, (g^{LCP}, f^{LCP}), is an EPCC-feasible solution.*

Proof. This follows from noting that a solution to LCP_1 is feasible for all the generator's problems. ■

Essentially, the NSE, as one may expect, is a refinement of the SSNE.

5.2.4 From SSNEs to NSEs

An important question is when is an SSNE an NSE. Clearly, all NSEs are SSNEs. But the reverse characterization does not always hold and requires a deeper discussion of NSEs. Recall that an NSE requires that every agent solves a Stackelberg problem or a mathematical program with complementarity constraints. Consider such a problem denoted by MPCC:

MPCC	$\underset{x,y,w}{\text{minimize}}$	$f(x, y, w)$
	subject to	$c(x, y, w) = 0,$
		$0 \leq y \perp w \geq 0,$

in which f and c represent the objective function and a vector of constraints. Both are assumed to be twice continuously-differentiable over $x \in \Re^n$ and y and $w \in \Re^m$. The specification of necessary optimality conditions requires the point to subscribe to a regularity condition. One such condition is the linear independence constraint qualification (LICQ).

Since MPCCs do not satisfy the LICQ at any feasible point, a different tack is taken in specifying the stationarity conditions: one defines a relaxed problem denoted by RNLP. This relaxed problem uses the active inequality constraints in the complementarity system and represents them as equality constraints in the relaxed problem. In addition, the complementarity constraint is dropped. For example, if the solution to the complementarity constraint $0 \leq w_j \perp y_j \geq 0$ is given by $w_j^* > 0 = y_j^*$, then we use constraints $w_j^* \geq 0$ and $y_j^* = 0$ and drop $w_j^* y_j^* = 0$. We now define RNLP at $\bar{z}$ using a form adopted by [dMFNS05]:

RNLP	$\underset{x,y,w}{\text{minimize}}$	$f(x,y,w)$		
	subject to	$c(x,y,w) = 0$		
		$(y)_j = 0,$	$(w)_j \geq 0,$	$j \in \mathcal{I}^y(\bar{z})$
		$(y)_j \geq 0,$	$(w)_j = 0,$	$j \in \mathcal{I}^w(\bar{z})$
		$(y)_j \geq 0,$	$(w)_j \geq 0.$	$j \in \mathcal{I}^y(\bar{z}) \cap \mathcal{I}^w(\bar{z}).$

The index sets $\mathcal{I}^y(\bar{z})$ and $\mathcal{I}^w(\bar{z})$ are defined as $\{j : \bar{y}_j = 0\}$ and $\{j : \bar{w}_j = 0\}$, respectively. We reiterate that the crucial difference between RNLP and MPCC is in the treatment of complementarity constraints. When $\bar{y}_j = 0 < \bar{w}_j$, it replaces the constraint $w_j y_j = 0$ by $y_j = 0$ and $w_j \geq 0$. However, when $\bar{y}_j = 0 = \bar{w}_j$, $y_j w_j = 0$ is replaced by $y_j, w_j \geq 0$. In all these instances, the gradients of the active constraints are linearly independent. The term *relaxation* emerges from the fact that the feasible region of MPCC, namely $\mathcal{F}_{MPCC}$, is contained in the feasible region of $\mathcal{F}_{RNLP}$ (that is $\mathcal{F}_{MPCC} \subseteq \mathcal{F}_{RNLP}$). Furthermore, we may define an MPCC-LICQ (MFCQ) as follows.

Definition 5.8 (MPCC-LICQ (MFCQ))**.** *The MPCC satisfies an MPCC-LICQ (MFCQ) at a feasible point $\bar{z}$ if RNLP, the relaxed NLP, satisfies the LICQ (MFCQ) at $\bar{z}$.*

This allows us to define a strong stationary point.

Definition 5.9 (Strong Stationarity)**.** *A point $(x^*, y^*, w^*, \lambda^*, \pi^*)$ is said to be a strong stationary point of MPCC if it is a KKT point of RNLP. In particular, it satisfies the first-order conditions:*

$$\begin{pmatrix} \nabla_x f(z^*) \\ \nabla_w f(z^*) \\ \nabla_y f(z^*) \end{pmatrix} - \begin{pmatrix} \nabla_x c(z^*) \\ \nabla_w c(z^*) \\ \nabla_y c(z^*) \end{pmatrix}^T \pi^* - \begin{pmatrix} \\ \lambda_w^* \\ \lambda_y^* \end{pmatrix} = 0, \tag{5.5}$$

$$[\lambda_w^*]_j, [\lambda_y^*]_j \geq 0, \quad i \in \mathcal{I}^y(z^*) \cap \mathcal{I}^w(z^*) \tag{5.6}$$

$$c(z^*) = 0, \tag{5.7}$$

$$w_j^* [\lambda_w^*]_j = 0, \tag{5.8}$$

$$y_j^* [\lambda_y^*]_j = 0, \tag{5.9}$$

$$z^* \in \mathcal{F}_{MPCC}. \tag{5.10}$$

If z^* satisfies MPCC-LICQ, then (λ^*, π^*) is unique.

We may now refine the notion of Nash-Stackelberg game.

Definition 5.10. *(Strong-Stationary Equilibrium Point) A Nash-Stackelberg equilibrium in forward decisions is defined as a strong-stationary equilibrium point in forward decisions* $\{f_1^*, \ldots, f_n^*\}$*, where* $(f_i^*, g^*, \lambda_i^*)$ *is a strong-stationary point of* $G_i(f^{-i,*})$*. If* $G_i(f^{-i,*})$ *satisfies the MPCC-LICQ for* $i = 1, \ldots, n$*, then* $(\lambda_1^*, \ldots, \lambda_n^*)$ *is unique.*

The remainder of this section focuses on proving an important result: *the SSNE is indeed an NSE*. Proving such a result is not always possible and entails answering the following question: Given that we have a feasible point for each agent problem, is the point a strong-stationary point? This question requires the construction of a feasible set of multipliers for each agent problem. Our discussion is restricted to the deterministic case but is easily extended to account for stochasticity.

Recall that the SSNE requires the solution of the following mixed-complementarity problem:

SSNE	$0 \leq g \perp Mg + f + (b - a) \geq 0,$ $a - e^T g = a_f - e^T f.$

The NSE is given by an equilibrium problem with complementarity constraints (EPCC). In particular, an EPCC represents an equilibrium problem in forward decisions f_i in which agent i solves

MPCC(f^{-i})	$\underset{g, f_i}{\text{minimize}}$	$\frac{1}{2} g_i^T B_i g_i + c_i^T g_i$
	subject to	$0 \leq g \perp Mg + f + (b - a) \geq 0,$ $a - e^T g = a_f - e^T f,$

where $M = I + ee^T$ and B_i is defined in accordance with $g^T B_i g = \sum_j g_i g_j + g_i^2$. Similarly, c_i is defined by $[c_i]_i = b_i - a$ and $[c_i]_j = 0, j \neq i$. We assume without loss of generality that the price functions of forwards and generation are identical as specified by $a_f = a$.

Definition 5.11. *Let* $(\bar{g}, \bar{f})$ *be an SSNE. Then the index sets* $\mathcal{I}_M$ *and* $\mathcal{I}_g$ *are defined by*

$$\begin{aligned} \mathcal{I}_M &= \{i : [M\bar{g} + \bar{f} + q]_i = 0\} \\ \text{and } \mathcal{I}_g &= \{i : [\bar{g}]_i = 0\}. \end{aligned}$$

Then we define a strong-stationary point of the EPCC as (g^*, f^*)*, where* (g^*, f_i^*) *is a strong-stationary point of MPCC*$(f^{-i,*})$*, which along with* λ_i^* *is given by*

$$\begin{aligned} \begin{pmatrix} B_i g + c_i \\ 0 \end{pmatrix} - \begin{pmatrix} M^T \\ I_i^T \end{pmatrix} \lambda_M - \begin{pmatrix} I \\ 0 \end{pmatrix} \lambda_g - \begin{pmatrix} -e \\ e \end{pmatrix} \lambda_E &= 0 \\ \lambda_i^M, \lambda_i^g \geq 0, \quad i \in \mathcal{I}_M \cap \mathcal{I}_g \\ [M\bar{g} + \bar{f} + q]_i \lambda_i^M &= 0 \\ [\bar{g}]_i \lambda_g^i &= 0 \\ a - e^T \bar{g} &= a_f - e^T \bar{f}. \end{aligned}$$

Theorem 5.12. *Assume that the SSNE satisfies strict complementarity, implying that* $\mathcal{I}_M(\bar{g}, \bar{f}) \cap$

$\mathcal{I}_g(\bar{g}, \bar{f}) = \emptyset$, and that the agent problems satisfy MPCC-LICQ. Then the vector $(\bar{g}, \bar{f}_i)$ represents a strong-stationary equilibrium point of the EPCC($\bar{f}^{-i,*}$) with a feasible multiplier vector λ_i^*.

Proof. We must show that given a solution to the mixed-complementarity problem, we can construct a λ_i^* that is feasible with regard to the strong-stationarity conditions. Under the assumption of strict complementarity, we evade the problem of ensuring nonnegative multiplers for biactive indices (indices from the set $\mathcal{I}_M(\bar{g}, \bar{f}) \cap \mathcal{I}_g(\bar{g}, \bar{f})$).

We consider the three possible cases arising from different possibilities of index sets.

1. $\mathcal{I}_g = \emptyset$: This implies that $\lambda_g = 0$ by the complementarity condition. Then, we may obtain λ_M and λ_E by solving

$$\begin{pmatrix} M^T & -e \\ I_i^T & 1 \end{pmatrix} \begin{pmatrix} \lambda_M \\ \lambda_E \end{pmatrix} = \begin{pmatrix} B_i g + c_i \\ 0 \end{pmatrix}.$$

   By MPCC-LICQ, the active constraints are linearly independent, implying that the system is nonsingular.

2. $\mathcal{I}_M = \emptyset$: This follows in an identical fashion but the system to be solved is

$$\begin{pmatrix} I & -e \\ & 1 \end{pmatrix} \begin{pmatrix} \lambda_g \\ \lambda_E \end{pmatrix} = \begin{pmatrix} B_i g + c_i \\ 0 \end{pmatrix}$$

   and $\lambda_M = 0$.

3. $\mathcal{I}_M \cap \mathcal{I}_g = \emptyset$: We note that $[\lambda_M]_i = 0$ for $i \in \{1, \ldots, n\} \cap \mathcal{I}_M^c$ and $[\lambda_g]_i = 0$ for $i \in \{1, \ldots, n\} \cap \mathcal{I}_g^c$. The remaining multiplier vector may be obtained from the following system:

$$\begin{pmatrix} \bar{M} & & -e \\ & I & -e \\ I_i^T & & 1 \end{pmatrix} \begin{pmatrix} \lambda_M^c \\ \lambda_g^c \\ \lambda_E \end{pmatrix} = \begin{pmatrix} B_i g + c_i \\ 0 \end{pmatrix}.$$

   Again by invoking MPCC-LICQ, we may claim that the system is nonsingular.

The uniqueness of the multiplier set follows from the fact that the systems are square and nonsingular. ■

This is an important result because in general it is difficult to be certain of the existence of Nash-Stackelberg equilibria. However, in this context, under the assumption of MPCC-LICQ and strict complementarity for all the agent problems, we may claim the existence of a Nash-Stackelberg equilibrium point.

5.2.5 A Numerical Comparison

This subsection focuses on comparing an SSNE with an NSE. We consider an equilibrium problem with n players with the cost of generation b specified as zero or as a random vector. The price

TABLE 5.1
Computing SSNE and NSE with zero costs

n	$\|g_{ssne} - g^*\|$	$\|f_{ssne} - f^*\|$	$\|g_{rand} - g^*\|$	$\|f_{rand} - f^*\|$	t^{ssne}	t^{nse}_{ssne}	t^{nse}_{rand}
5	3.7e-09	3.7e-09	1.8e-2	1.8e-2	0.01	0.250	0.351
10	1.9e-09	1.9e-09	3.7e-2	3.7e-2	0.03	0.360	0.752
15	8.3e-10	8.3e-10	6.1e-2	6.1e-2	0.01	0.621	1.532
20	4.2e-10	4.2e-10	7.8e-2	7.8e-2	0.05	0.791	1.613
25	2.4e-10	2.4e-10	1.0e-1	1.0e-1	0.04	1.041	2.074
30	1.5e-10	1.5e-10	1.5e-1	1.5e-1	0.03	1.282	2.484
35	9.4e-11	9.4e-11	1.8e-1	1.8e-1	0.06	1.482	3.045
40	6.5e-11	6.5e-11	1.8e-1	1.8e-1	0.07	1.892	3.546

TABLE 5.2
Computing SSNE and NSE with random costs

n	$\|g_{ssne} - g^*\|$	$\|f_{ssne} - f^*\|$	$\|g_{rand} - g^*\|$	$\|f_{rand} - f^*\|$	t^{ssne}	t^{nse}_{ssne}	t^{nse}_{rand}
5	3.7e-09	3.7e-09	1.8e-02	1.8e-02	0.01	0.250	0.351
10	1.9e-09	1.9e-09	3.7e-02	3.7e-02	0.03	0.360	0.752
15	8.3e-10	8.3e-10	6.1e-02	6.1e-02	0.01	0.621	1.532
20	4.2e-10	4.2e-10	7.8e-02	7.8e-02	0.05	0.791	1.613
25	2.4e-10	2.4e-10	1.0e-01	1.0e-01	0.04	1.041	2.074
30	1.5e-10	1.5e-10	1.5e-01	1.5e-01	0.03	1.282	2.484
35	9.4e-11	9.4e-11	1.8e-01	1.8e-01	0.06	1.482	3.045
40	6.5e-11	6.5e-11	1.8e-01	1.8e-01	0.07	1.892	3.546

functions at the spot and forward market have the same intercept, $a = a_f$. For purposes of simplicity, we assume a single scenario, implying that $|\Omega| = 1$.

The SSNE is given by a mixed-LCP and is solvable by PATH [DF93]. The NSE is given by an EPCC and may be solved by a Jacobi iteration. This involves solving the MPCC for the first agent and passing its forward decision to the next agent and so on. This method may not always converge. We consider two different starting points. The first starting point is the SSNE. The second starting point is randomly chosen.

We show the results for the zero and random costs in Tables 5.1 and 5.2. We note that the SSNE is an NSE in that the agents do not deviate from it. However, from a random starting point, the algorithm may not always converge toward the neighborhood of the SSNE. This is expected because the Jacobi iteration has no global convergence theory and is merely a heuristic for solving the problem.

In terms of computational effort, there is an immense difference when one raises the number of agents from 5 to 40. We see that while the growth in computational cost is similar in ratio, it is nearly 30 times more expensive for a small deterministic problem. In a stochastic large-scale setting, the Jacobi method would be very computationally intensive.

5.3 Application: Electricity Forward Markets

§5.2 introduced a 2-node spot-forward market model in a simple setting with n producers and infinite capacities. In this section, we place the problem in the context of electricity markets.

Technically speaking, alternating-current (AC) power flows across an electrical path in a network and is comprised of real and reactive power. The former (measured in Watts) is consumed by the resistive appliances on the network. The latter (measured in Volt-Amps-Reactive or VARs) is required by inductive or capacitive loads. Such a model requires the specification of real and reactive power flows at each node as well as the voltage magnitudes and phase angles. Generally, only physical linkages between nodes or buses are considered and this node-linkage specification is denoted by the node-admittance matrix. The admittance characteristics of the linkages are articulated through the branch-admittance matrix. Our analysis is restricted to *high voltage* transmission systems, allowing us to assume that the voltage angles are small and the voltage magnitudes are constant. Moreover, the losses are considered to be negligible. The resulting power flow equations are often termed DC load flow equations. Further details may be found in [SCTB88]. Throughout our analysis, we use DC load flow analysis to specify flows.

5.3.1 Spot-Market Equilibrium

Consider an n-node network with a firm at each node. We assume that firm i has a generator at node i but may sell to all other nodes in the network (we assume a fully connected grid but this assumption is without loss of generality). The sales by firm i (housed at node i) to node j are denoted by s_{ij}. We collectively denote the sales decisions by firm i by $s_{i,\cdot} = (s^\omega_{i1}, \ldots, s^\omega_{in})$. In addition, s^ω_{-i} refers to the generation decisions of all agents excepting i, namely, $(s^\omega_{j,\cdot}, j \neq i)$. Suppose that the nodal demand function at node j under realization ω is given by

$$p^\omega_j(s_{\cdot,j}) := a^\omega_j - m^\omega_j \sum_i s^\omega_{ij}. \tag{5.11}$$

Suppose firm i generates g^ω_i units of power and sells s^ω_{ij} units of power to node j under realization ω. Also the capacity on sales and generation is denoted by C^ω_{ij} and G^ω_i. The capacity and conservation constraints are given by $g^\omega_i \leq G^\omega_i$ and $g^\omega_i = \sum_j s^\omega_{ij}$, respectively. The resulting firm problem is a stochastic quadratic program:

B^ω_i

$$\begin{array}{lll} \underset{s^\omega_{i,\cdot}, g^\omega_i}{\text{maximize}} & h(s^\omega_{i,\cdot}, g^\omega_i) & \\ \text{subject to} & g^\omega_{ij} - \sum_j s^\omega_{ij} = 0 & : \psi_i \\ & G^\omega_i - g^\omega_{ij} \geq 0 & : \theta_{ij} \\ & s^\omega_{ij} \geq 0 & : \gamma^\omega_{ij} \quad \forall j \\ & C^\omega_{ij} - s_{ij} \geq 0 & : \alpha^\omega_{ij}, \quad \forall j, \end{array}$$

where $h(s^\omega_{i,\cdot}, g^\omega_i)$ is defined as

$$h(s^\omega_{i,\cdot}, g^\omega_i) := p_i(s^\omega_{i,\cdot})(s^\omega_{ii} - f_{ii}) + \sum_{k\neq i}(p^\omega_k(s^\omega_{k,\cdot}))(s^\omega_{ik} - f_{ik}) - c^\omega_i g^\omega_i - \frac{1}{2}d^\omega_i(g^\omega_i)^2.$$

We may eliminate the generation variable g^ω_i by using the conservation constraints to obtain a reduced model:

$$\begin{array}{lll} \underset{s^\omega_{i,\cdot}}{\text{maximize}} & h(s^\omega_{i,\cdot}) & \\ & G^\omega_i - \sum_j s^\omega_{ij} \geq 0 & : \psi^\omega_i \\ \text{subject to} & s^\omega_{ij} \geq 0 & : \gamma^\omega_{ij} \quad \forall j \\ & C^\omega_{ij} - s^\omega_{ij} \geq 0 & : \alpha^\omega_{ij}, \quad \forall j, \end{array}$$

where $h(s^\omega_{i,\cdot})$ is defined as

$$h(s^\omega_{\cdot,i}) := p_i(s^\omega_{\cdot,i})(s_{ii} - f_{ii}) + \sum_{k\neq i}(p^\omega_k(s^\omega_{\cdot,k}))(s^\omega_{ik} - f_{ik}) - c^\omega_i \sum_j s^\omega_{ij} - \frac{1}{2}d^\omega_i \sum_j (s^\omega_{ij})^2.$$

In addition to the firm problems, the firms have to ensure that they satisfy risk-neutrality constraints. We may now define the spot-market Nash equilibrium.

Definition 5.13. *Let s^*_i denote $(s^\omega_i)^*$ for all $\omega \in \Omega$. Then, given forward decisions (f_i), the Nash equilibrium in sales decisions is given by $(s^*_1, \ldots, s^*_n)$, where $(s^\omega_i)^*$ is a solution of B^ω_i for all $\omega \in \Omega$ and*

$$a^f_j - m^f_j \sum_i f_{ij} = \mathbb{E}(a^\omega_i - m^\omega_j \sum_j (s^\omega_{ij})^*), \quad \forall j. \tag{5.12}$$

This equilibrium point may be obtained by solving the set of sufficient stationarity conditions associated with each problem:

$$0 \leq s^\omega_{ij} \perp 2m^\omega_j s^\omega_{ij} + \sum_{k\neq j} m^\omega_j s^\omega_{kj} + m^\omega_j f_{ij} + c^\omega_i - a^\omega_j = \gamma_{ij} \geq 0, \forall j, i$$
$$a^f_j - m^f_j \sum_i f_{ij} = \mathbb{E}(a^\omega_j - m^\omega_j \sum_i s^\omega_{ij}), \quad \forall j.$$

This set may be compactly written as the linear complementarity problem LCP_2:

$$\text{LCP}_2 \qquad 0 \leq \begin{pmatrix} s^\omega \\ \psi^\omega \\ \alpha^\omega \end{pmatrix} \perp \begin{pmatrix} \widehat{M}^\omega & \bar{E}^T & I \\ -\bar{E} & & \\ -I & & \end{pmatrix} \begin{pmatrix} s^\omega \\ \psi^\omega \\ \alpha^\omega \end{pmatrix} + N^\omega f + \begin{pmatrix} \ell^\omega \\ G^\omega \\ C^\omega \end{pmatrix} \geq 0$$
$$B^f f - \mathbb{E} D_\omega s^\omega + (\mathbb{E} a^\omega - a^f) = 0,$$

where

$$\ell^\omega_{ij} = c^\omega_i - a^\omega_j, \bar{E} = \begin{pmatrix} e^T & & \\ & \ddots & \\ & & e^T \end{pmatrix}, N^\omega = \begin{pmatrix} \text{diag}(m^\omega) \\ \\ \end{pmatrix}, B^f = \Big(\text{diag}(m^f) \ldots \text{diag}(m^f)\Big),$$

$$D^\omega = \Big(\text{diag}(m^\omega) \ldots \text{diag}(m^\omega)\Big) \text{ and } \widehat{M}^\omega = \bar{M}^\omega + \begin{pmatrix} d^\omega_1 I & & \\ & \ddots & \\ & & d^\omega_n I \end{pmatrix}.$$

A transmission provider is now introduced into the framework. He maximizes transmission revenue subject to meeting transmission constraints as shown in the transmission provider's problem $T(s)$, where s collectively refers to the sales decisions (see below). Let the price of transmitting a unit across link (ij) for realization ω be given by w^ω_{ij} with the corresponding flow being denoted by y^ω_{ij}. The linkage capacity during realization ω is given by t^ω_{ij}.

T^ω

$$\begin{array}{lll} \text{maximize} & \sum_{i,j} (w^\omega_{ij})^T y^\omega_{ij} & \\ \text{subject to} & t^\omega_{ij} - y^\omega_{ij} \geq 0 & : \lambda^\omega_{ij} \\ & t^\omega_{ij} + y^\omega_{ij} \geq 0 & : \lambda^\omega_{ji}, \quad \forall i, j. \end{array}$$

The i^{th} generator's problem, B^ω_i, is now modified to the following:

$$\begin{array}{lll} \underset{s^\omega_{i,\cdot}}{\text{maximize}} & h(s^\omega_{i,\cdot}) & \\ & \sum_j s^\omega_{ij} - G^\omega_i \leq 0 & : \theta^\omega_{ij} \\ \text{subject to} & s^\omega_{ij} \geq 0 & : \gamma^\omega_{ij} \quad \forall j \\ & C^\omega_{ij} - s_{ij} \geq 0 & : \alpha^\omega_{ij}, \quad \forall j, \end{array}$$

where $h(s_{i,\cdot})$ is defined by

$$h(s^\omega_{i,\cdot}) := p^\omega_i(s_{i,\cdot})(s^\omega_{ii} - f_{ii}) + \sum_{k \neq i} (p^\omega_k(s^\omega_{k,\cdot}) + w^\omega_{ik})(s^\omega_{ik} - f_{ik})$$
$$- c^\omega_i \sum_j s^\omega_{ij} - \frac{1}{2} d^\omega_i (\sum_j s^\omega_{ij})^2.$$

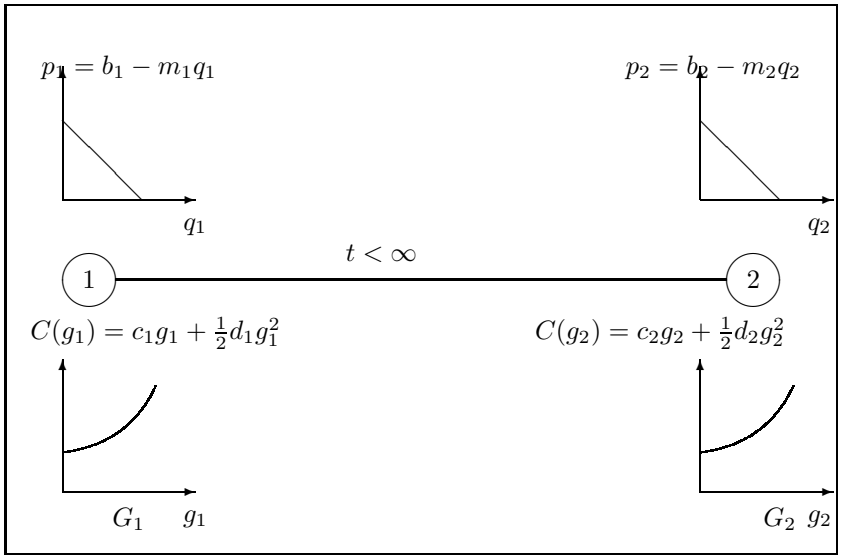

FIGURE 5.1 *Finite generation capacities and quadratic costs of generation*

The net flow across linkage (ij) during realization ω is given by

$$y_{ij}^{\omega} = s_{ij}^{\omega} - s_{ji}^{\omega}, \quad \forall j, i. \tag{5.13}$$

We may now provide a revised definition for the spot-market Nash equilibrium.

Definition 5.14. *The Nash equilibrium in sales, generation, and transmission decisions is given by the tuples* $(s_{1,\cdot}^*, \ldots, s_{n,\cdot}^*)$, $(g_1^*, \ldots, g_n^*)$, *and* $(y_{1,\cdot}^*, \ldots, y_{n,\cdot}^*)$, *where* $s_{i,\cdot}^*$ *is a solution of* B_i^{ω} *for all* $\omega \in \Omega$, $(y^{\omega})^*$ *solves* T^{ω} *and*

$$a_i^f - m_i^f \sum_j f_{ji} = \mathbb{E}(a_i^{\omega} - m_i^{\omega} \sum_j s_{ji}^{\omega}), \quad \forall i. \tag{5.14}$$

The equilibrium point may be obtained by solving a larger mixed-complementarity problem LCP_3:

$$\mathrm{LCP}_3 \qquad 0 \le \begin{pmatrix} s^{\omega} \\ \psi^{\omega} \\ \lambda^{\omega} \\ \alpha^{\omega} \end{pmatrix} \perp \begin{pmatrix} \widehat{M}^{\omega} & \bar{E}^T & \bar{F}^T & I \\ -\bar{E} & & & \\ -\bar{F} & & & \\ -I & & & \end{pmatrix} \begin{pmatrix} s^{\omega} \\ \psi^{\omega} \\ \lambda^{\omega} \\ \alpha^{\omega} \end{pmatrix} + \begin{pmatrix} I \\ \\ \\ \\ \end{pmatrix} f + \begin{pmatrix} \ell^{\omega} \\ G^{\omega} \\ t^{\omega} \\ C^{\omega} \end{pmatrix} \ge 0$$
$$Df - \mathbb{E} D_{\omega} s^{\omega} + (\mathbb{E} a^{\omega} - a^f) = 0.$$

Suppose that the kth row of $\bar{F}$ corresponds to link (i, j). Then $\bar{F}_{k,i*n+j} = -\bar{F}_{k,j*n+i} = 1$.

Lemma 5.15. *The matrix $\bar{M}_3$ is positive semidefinite, where*

$$\bar{M}_3 := \begin{pmatrix} \bar{M} & \bar{E}^T & \bar{F}^T & I \\ -\bar{E} & & & \\ -\bar{F} & & & \\ -I & & & \end{pmatrix}.$$

Proof. This follows from the definition of $\bar{M}_3$ and the positive definiteness of $\bar{M}$. ■

We conclude this section with an existence theorem for the spot-market equilibrium.

Lemma 5.16. *Given a set of forward decisions, the solution to the spot-market equilibrium as specified by definition (5.13) exists.*

Proof. The proof follows from lemma 5.2. ■

Having shown the existence of a spot-market equilibrium, we introduce the Nash-Stackelberg equilibrium in the next section.

5.3.2 A Nash-Stackelberg Equilibrium

The Nash-Stackelberg equilibrium in forward decisions requires each generator to maximize the expected profit from forward and spot positions subject to the following complementarity constraint and risk-neutrality constraint:

SCP_1	$0 \leq z^\omega \perp \mathcal{M}^\omega z^\omega + N^\omega f + q^\omega \geq 0 \quad \forall \omega$ $\mathbb{E}_\omega W^\omega z^\omega - B^f f + q^f = 0.$

Agent i's optimization problem is

$\text{D}_i(f^{-i})$	$\underset{f_i, z}{\text{maximize}}$	$\mathbb{E}(\frac{1}{2} z^{\omega,T} Q^\omega z^\omega + (r^\omega)^T z^\omega)$
	subject to	$0 \leq z^\omega \perp \mathcal{M}^\omega z^\omega + N^\omega f + q^\omega \geq 0 \quad \forall \omega$ $\mathbb{E}_\omega W^\omega z^\omega - B^f f + q^f = 0,$

where the cost function is defined by (5.15):

$$\frac{1}{2}(z^\omega)^T Q^\omega z^\omega + (r^\omega)^T z^\omega := p_i^\omega(s_{i,\cdot}) s_{ii}^\omega + \sum_{k \neq i} (p_k^\omega(s_{jk}^\omega) + w_{ik}^\omega) s_{ik}^\omega - c_i^\omega g_i^\omega - \frac{1}{2} d_i^\omega (g_i^\omega)^2. \tag{5.15}$$

We may now define the Nash-Stackelberg equilibrium as a solution to the equilibrium problem with complementarity constraints (EPCC_2).

Definition 5.17. *$EPCC_2$: The Nash-Stackelberg equilibrium in forward decisions is given by a tuple $(f_1^*, \ldots, f_n^*)$, where (f_i^*, z^*) is a solution of $D_i(f^{-i,*})$.*

In §5.2.2, we discussed how such equilibrium problems are often termed EPCCs or stochastic EPCCs to be precise. The Nash-Stackelberg equilibrium point is an *EPCC-optimal* point.

5.3.3 A Simultaneous Stochastic Nash Equilibrium

We begin by defining an SSNE in the context of a spot-forward electricity market. Such a definition needs an extension of the definition of the forward-market problem P_i^f for $i = 1, \ldots, n$. In the earlier section, we avoided having to define this problem by using the risk-neutrality constraint (which is a sufficient condition for optimality in the forward-market problem). The forward-market problem is given by

$$\mathrm{B}_i^f \qquad \underset{f_{i,\cdot}}{\text{maximize}} \;\; (p_i^f - \mathbb{E}_\omega p_i^\omega) f_i,$$

where $p_i^f = (a_i^f - m_i^f \sum_j f_{ji})$ and $p_i^\omega = (a_i^\omega - m_i^\omega \sum_j s_{ji}^\omega)$.

Definition 5.18. *An SSNE is defined as a set of (s, f, y), where generator i solves a spot-market problem B_i^ω for all $\omega \in \Omega$ and a forward market problem B_i^f, subject to a risk-neutrality constraint. Moreover, the transmission provider solves the problem T^ω for all $\omega \in \Omega$.*

If we insist that the expected price of spot-sales is equal to the price of forwards, then generator i only solves $B_i^s(f)$ with the equilibrium point being constrained by the risk-neutrality constraint.

Lemma 5.19. *Consider a game in which generator i solves B_i^ω, $\forall \omega \in \Omega$ and the transmission provider solves T^ω, $\forall \omega \in \Omega$. Then a Nash equilibrium exists.*

Proof. Follows from lemma 5.6. ■

5.3.4 Relaxing the Risk-neutrality Constraint

We now consider the problem that emerges from relaxing the risk-neutrality constraint. In effect, our complementarity problem SCP_1 is modified to be

$$\mathrm{SCP}_2 \qquad \begin{aligned} 0 \le z^\omega \perp \bar{M}^\omega z^\omega + N^\omega f + \bar{q}^\omega \ge 0 \quad \forall \omega \\ \mathbb{E} W^\omega z^\omega - B^f f + q^f = 0, \end{aligned}$$

where B^f is symmetric positive definite. The resulting forward positions may then be stated as

$$f = (B^f)^{-1} \sum_{j=1}^{K} p^{\omega_j} W^{\omega_j} z^{\omega_j} + (B^f)^{-1} q^f.$$

This expression may be substituted into the complementarity constraint, resulting in a set of complementarity constraints

$$\begin{aligned} 0 \le z^{\omega_i} \perp \bar{M}_i^\omega z^{\omega_i} + \bar{q}_i^\omega \ge 0, \qquad (5.16) \\ \text{where } \bar{M}_i^\omega = (M^{\omega_i} + p^{\omega_i} N^{\omega_i} (B^f)^{-1} W^{\omega_i}) \\ \text{and } \bar{q}_i^\omega = \sum_{j \ne i} p^{\omega_j} N^{\omega_j} (B^f)^{-1} W^{\omega_j} z^{\omega_j} + q^{\omega_i} + p^{\omega_i} (B^f)^{-1} q^f \end{aligned}$$

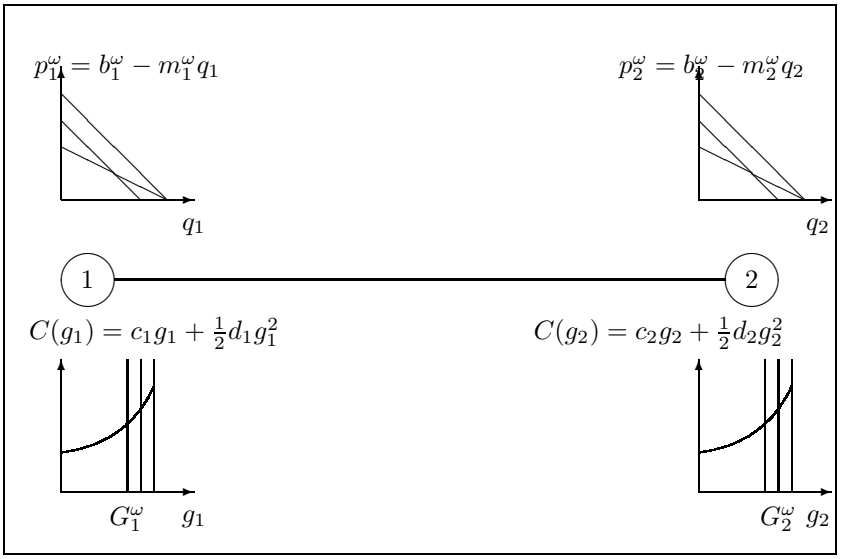

FIGURE 5.2 *Uncertainty in capacity and demand functions*

for $i = 1, \ldots, n$. In matrix form, this may be represented as $0 \leq z \perp \bar{M}z + \bar{q} \geq 0$, where $\bar{M}$ and $\bar{q}$ are given by

$$\begin{pmatrix} M^{\omega_1} + p^{\omega_1}\bar{N}^{\omega_i}W^{\omega_1} & p^{\omega_2}\bar{N}^{\omega_2}W^{\omega_2} & \ldots & p^{\omega_K}\bar{N}^{\omega_K}W^{\omega_K} \\ p^{\omega_1}\bar{N}^{\omega_1}W^{\omega_1} & M^{\omega_2} + p^{\omega_2}\bar{N}^{\omega_2}W^{\omega_2} & \ldots & p^{\omega_K}\bar{N}^{\omega_K}W^{\omega_K} \\ \vdots & & \ddots & \\ p^{\omega_1}\bar{N}^{\omega_1}W^{\omega_1} & p^{\omega_2}\bar{N}^{\omega_2}W^{\omega_2} & \ldots & M^{\omega_K} + p^{\omega_K}\bar{N}^{\omega_K}W^{\omega_K} \end{pmatrix}$$

and $\begin{pmatrix} q^{\omega_1} + (B^f)^{-1}q^f \\ \vdots \\ q^{\omega_K} + (B^f)^{-1}q^f \end{pmatrix}$, and where $\bar{N}^{\omega_i} = N^{\omega_i}(B^f)^{-1}$.

The following theorem provides a condition for solvability of the problem.

Proposition 5.20. *Suppose there exists a z such that $\bar{M}z + \bar{q} \geq 0$ and $z \geq 0$. Then if $\bar{M}$ is a $\mathbf{P}_0$ matrix, an equilibrium point exists.*

Proof. It suffices to show that $\bar{M}$ is a $\mathbf{P}_0$ matrix. The matrix $\bar{M}$ may be decomposed as follows:

$$\begin{aligned} \bar{M} &= \begin{pmatrix} M^{\omega_1} & & \\ & \ddots & \\ & & M^{\omega_K} \end{pmatrix} + \begin{pmatrix} \bar{m}^{\omega_1}W^{\omega_1} & \ldots & \bar{m}^{\omega_K}W^{\omega_K} \\ \vdots & \vdots & \vdots \\ \bar{m}^{\omega_1}W^{\omega_1} & \ldots & \bar{m}^{\omega_K}W^{\omega_K} \end{pmatrix} \\ &= \bar{M}_1 + \bar{M}_2. \end{aligned}$$

The matrix $\bar{M}_1$ is positive semidefinite because M_1^ω is positive semidefinite. It suffices to show that

$\bar{M}_2$ is a positive semidefinite matrix:

$$\begin{aligned} x^T \bar{M}_2 x &\geq \min_j\{\bar{m}^{\omega_j}\} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}^T \begin{pmatrix} W^{\omega_1} & \dots & W^{\omega_K} \\ \vdots & \vdots & \vdots \\ W^{\omega_1} & \dots & W^{\omega_K} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ &\geq (\min_j\{\bar{m}^{\omega_j}\})^2 \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}^T \begin{pmatrix} W & \dots & W \\ \vdots & \vdots & \vdots \\ W & \dots & W \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \\ &= (\min_j\{\bar{m}^{\omega_j}\})^2 (x_1 + \ldots + x_n)^T W (x_1 + \ldots + x_n) \\ &\geq 0. \end{aligned}$$

The last inequality follows from $\min_j\{\bar{m}^{\omega_j}\} > 0$ and $W \succeq 0$. Therefore $\bar{M}_1 + \bar{M}_2 \succeq 0$. ■

5.4 A Decomposition-based Splitting Algorithm

The past section has introduced two stochastic complementarity problems SCP_1 and SCP_2. The important distinction between the two problems lies in the singularity of B^f. In this section, we propose an algorithm for solving a mixed-complementarity problem with a singular B_f.

We present an algorithm based on solving the mixed-LCP through a sequence of LCPs. Each LCP is stochastic in nature and can be arbitrarily large. We employ a decomposition-based iterative method for solving this LCP. We refer to this method as the **D**ecomposition and **S**plitting (DS) method. In §5.4.1 and 5.4.2, we present the DS algorithm and prove its convergence. The computational burden may be lightened considerably by the use of sampling, and these ideas are discussed in section 5.4.3. Finally, in section 5.4.4 we provide a description of the performance of the DS method and compare it with solving the problem directly using PATH.

5.4.1 The DS Algorithm: Description and statement

We begin by stating the complementarity problem of interest:

SCP_1	$0 \leq z^\omega \perp \bar{M}^\omega z^\omega + N^\omega f + \bar{q}^\omega \geq 0 \quad \forall \omega$ $\mathbb{E} W^\omega z^\omega - B^f f + q^f = 0.$

We solve the stochastic optimization problem by decomposing the problem by scenarios. This ensures that when the number of scenarios grows, the problem may still be solved efficiently. However, problem SCP_1 is not immediately scenario-separable because

1. the equality constraints contain an expectation term;
2. the complementarity constraint contains the first-stage forward decisions f.

We propose an iterative method for the solution of such a problem and begin by restating the problem compactly. By denoting

$$z = \begin{pmatrix} z^{\omega_1} \\ \vdots \\ z^{\omega_K} \end{pmatrix}, \bar{M} = \begin{pmatrix} \bar{M}^{\omega_1} & & \\ & \ddots & \\ & & \bar{M}^{\omega_K} \end{pmatrix}, N = \begin{pmatrix} N^{\omega_1} \\ \vdots \\ N^{\omega_K} \end{pmatrix} \bar{q} = \begin{pmatrix} q^{\omega_1} \\ \vdots \\ \bar{q}^{\omega_K} \end{pmatrix}, W = \begin{pmatrix} p^{\omega_1}(W^{\omega_1})^T \\ \vdots \\ p^{\omega_K}(W^{\omega_K})^T \end{pmatrix}^T,$$

we may write SCP_1 as

SCP_1	$0 \le z \perp \bar{M}z + Nf + \bar{q} \ge 0$ $Wz - B^f f + q^f = 0.$

Our intention is to construct a sequence of (f_k, z_k) such that $(f_k, z_k) \to (f^*, z^*)$ as $k \to \infty$, where (f^*, z^*) is a solution to SCP_1. Since B^f is singular, we may not express f as an explicit function of z_k. However, we may rewrite the risk-neutrality constraint at the solution as

$$(B_f + I)f^* = Wz^* + f^*$$

Using the same idea at the kth iterate, we obtain

$$(B_f + \theta I)f_k = Wz_k + \theta f_{k-1}, \tag{5.17}$$

where $\theta > 1$. This allows us to write f_k as

$$\begin{aligned} f_k &= (B^f + \theta I)^{-1}(Wz_k + f_{k-1}) \\ &= (B^f + \theta I)^{-1}Wz_k + (B^f + \theta I)^{-2}Wz_{k-1} + (B^f + \theta I)^{-2}f_{k-2} \\ &= (B^f + \theta I)^{-1}Wz_k + \cdots + (B^f + \theta I)^{-k}Wz_1 + (B^f + \theta I)^{-k}f_0. \end{aligned} \tag{5.18}$$

We may now define SCP_1^k as

SCP_1^k	$0 \le z_k \perp \widehat{M}z_k + \bar{q}_k \ge 0,$

where[5]

$$\begin{aligned} \widehat{M} &= \bar{M} + N(B^f + \theta I)^{-1}W \\ \text{and } \bar{q}_k &= (B^f + \theta I)^{-2}Wz_{k-1} + \cdots + (B^f + \theta I)^{-k}Wz_1 + (B^f)^{-1}q^f + \bar{q}. \end{aligned}$$

The statement of the outer-loop of the **DS** algorithm follows.

1. *Initialize*: Let $k = 0$ and let $z^0 \ge 0$ be an arbitrary vector in $\mathbb{R}^n$ and $f_0 = 0$.
2. **Major**: Solve SCP_1^k to obtain z_k, and define f_k by (5.18).

[5]Note that if B^f is nonsingular, then we may write f as $(B^f)^{-1}(Wz + q^f)$ and the resulting matrices $\widehat{M}$ and $\bar{q}_k$ may be written as $\widehat{M} = \bar{M} + N(B^f)^{-1}W$ and $\bar{q}_k = (B^f)^{-1}q^f + \bar{q}$.

3. *Termination*: If $\|z^k - z^{k-1}\| < \epsilon$, terminate; else $k := k + 1$ and return to step 2.

This algorithm does not specify how one may solve SCP_1^k. Notice that SCP_1^k is a linear complementarity problem given by $\text{LCP}(\widehat{M}, q_k)$ in which the matrix $\widehat{M}$ may be written as $B + C$, where

$$B = \bar{M} + \delta I \quad \text{and} \quad C = \widehat{M} - B.$$

The specification of B in this fashion allows us to claim the following trivial result.

Lemma 5.21. *The matrix B is positive definite.*

Proof. Follows immediately by definition and by noting that B is positive semidefinite, where

$$B' = \begin{pmatrix} M^{\omega_1} & & \\ & \ddots & \\ & & M^{\omega_n} \end{pmatrix}$$

■

SCP_1^k may now be written as

SCP_1^k	$0 \leq y \perp By + Cy + c \geq 0.$

Such a decomposition of matrices is often called a *splitting* and is discussed extensively in [CPS92] in the context of LCPs. The idea is to solve this inner problem using an iterative method that solves a sequence of problems $\text{LCP}(B, c_k)$, where $c_k = Cy_{k-1}$. The benefit of solving the original subproblem in this fashion is that $\text{LCP}(B, c_k)$ is separable into scenario-based LCPs. This is ensured by an appropriate choice of B.

5.4.2 The DS Method: Convergence Theory

This section provides convergence theory for the proposed decomposition method. There are two parts to proving convergence of this method:

1. Prove that the constructed sequence (f_k, z_k) does indeed converge to (f^*, z^*).
2. Prove that an iterative-splitting method for solving the subproblem, viz. SCP_1^k, is convergent.

Before proceeding with the first part, we show that the sequence (z_k, f_k) stays bounded. The construction of this sequence requires the solution of the sequence $\text{SCP}_k^1(\widehat{M}, \bar{q}_k)$, which gives us the sequence $\{z_k\}$. We may then obtain f_k from (5.17), assuming that f_0 is given. In showing some properties of this sequence, we need to show some properties of the underlying matrix $\bar{M}$.

Lemma 5.22. *The matrix $\bar{M}$ in SCP_1^k is a copositive $\mathbf{R}_0$ matrix. The solution z_k of SCP_1^k is therefore bounded. Furthermore, problem SCP_1^k is always solvable.*

Proof.

1. The matrix $\bar{M}$ is $\mathbf{R}_0$ if $\mathrm{SOL}(0, \bar{M}) = \{0\}$. But this follows from noting that if $\bar{q} = 0$, then the generation and transmission capacities are zero. This implies that the sales, generations, and transmissions have to be zero. In other words, the only solution of $\mathrm{SOL}(0, \bar{M})$ is the zero vector and $\bar{M}$ is an $\mathbf{R}_0$ matrix. The boundedness of the solution set of the complementarity problem follows from proposition 3.9.23 in [CPS92].

2. The agent problems B_i and T are convex over a compact set for all specifications of f. Therefore, a first-order KKT point always exists and represents a solution to the complementarity system SCP_k^1.

■

Lemma 5.23. *Assume that $\|W\| \leq \bar{W}$, $|f_0| \leq \bar{f}$ and $\theta > 1$ from (5.17). Then the sequence $\{f_k\}$ always stays bounded. Furthermore, $\bar{q}_k \to \bar{q}^*$.*

Proof. From (5.17), we have

$$\begin{aligned}
\|f_k\| &= \|(B^f + \theta I)^{-1} W z_k + \ldots + (B^f + \theta I)^{-k} W z_1 + (B^f + \theta I)^{-k} f_0\| \\
&\leq \sum_{j=1}^{k} \|(B^f + \theta I)^{j-k-1} W z_j\| + \|(B^f + \theta I)^{-k} f_0\| \\
&\leq \sum_{j=1}^{k} \|(B^f + \theta I)^{j-k-1}\| \|W\| \|z_j\| + \|(B^f + \theta I)^{-k}\| \|f_0\| \\
&\leq \sum_{j=1}^{k} \|(B^f + \theta I)^{j-k-1}\| \bar{W} \bar{z} + \|(B^f + \theta I)^{-k}\| \bar{f}.
\end{aligned}$$

But $(B^f + \theta I)$ is the sum of a positive semidefinite singular matrix B^f and θI. Therefore, the minimum eigenvalue of $B^f + \theta I$ is θ. This implies that

$$\|(B^f + \theta I)^{-k}\| \leq \frac{1}{\lambda_{\min}(B^f + \theta I)^k} = \theta^{-k} < \infty.$$

Furthermore, as $k \to \infty$, we have $\|(B^f + \theta I)^{-k}\| \to 0$. Consequently, the sequence

$$\sum_{j=1}^{k} \|(B^f + \theta I)^{j-k-1} W z_j\| < \infty$$

as $k \to \infty$, implying that $\bar{q}_k \to \bar{q}^*$. ■

Theorem 5.24. *If f_k and z_k are defined by (5.17) and the solution of SCP_k^1, then the sequence (f_k, z_k) converges to (f_*, z_*), the solution of SCP_1.*

Proof. By Lemma 5.23, we have $\bar{q}_k \to \bar{q}_*$. Then, we have that $\mathrm{SCP}_k^1 \to \mathrm{SCP}_*^1$. Moreover, the solution $z_k \to z_*$. The sequence $\{z_k\}$ gives rise to a unique bounded f_k for each k. By definition of $\{f_k\}$ and $\theta > 1$, we have $\{f_k\} \to f_*$. ■

Our next step is to prove that the iterative-splitting method used to solve SCP_k^1 is a convergent method. This theory was developed for deterministic LCPs in Cottle et al. [CPS92]. We merely state the theorem from this reference. Note that the theorem applies to the solution of an LCP: $0 \leq x \perp (B+C)x + q \geq 0$.

Theorem 5.25. *Let M be positive semidefinite and B be symmetric positive definite. Then if LCP(q, M) is feasible, $\{x^k\}$ is uniquely defined and converges to x^*, a solution of LCP(q, M).*

Before we proceed, it should be noted that in our context M is positive semidefinite and B is positive definite by construction.

5.4.3 Introducing Sampling

In this section, we consider how one may further ease the computational burden by using a sample of the distribution. Specifically, at the k^{th} iteration of the DSM method, we solve

$$\hat{\mathrm{SCP}}_1^{\mathrm{k}} \qquad \begin{array}{l} 0 \leq z_k^\omega \perp \bar{M}^\omega z_k^\omega + N f_k + \bar{q}^\omega \geq 0 \quad \forall \omega \in \Omega_k \\ \frac{1}{|\Omega_k|} \sum_{\omega \in \Omega_k} W^\omega z_k^\omega - B^f f_k + q^f = 0. \end{array}$$

In effect, at the kth iteration, the problem size is proportional to n_k. If the sequence n_k increases fast enough, such a scheme is seen to converge in practice. The set of steps may then be summarized as follows:

0. Set $k = 1$ and initialize f_j^k and $z^{\omega_j,k}$ for all j. Let $n_k := N < s$.
1. Generate n_k scenarios from distribution $\mathbb{P}$ of ω.
2. For $j = 1, \ldots, n_k$, solve the scenario-based mixed-LCP to obtain f_j^k and $z^{\omega_j,k}$.
3. If $\|f^k - f^{k-1}\| < \epsilon$ then terminate; else $k := k+1$ and $n_k := \min(s, \gamma n_k)$ and return to step 1.

An important question is which distribution to use in the construction of the sample. We could use a Monte-Carlo strategy that chooses from the original distribution and refer to this technique as MC. However, we may use the current solution to construct a sample. For instance, given z_k^ω for each $\omega \in \Omega$. Note that for the unsampled scenarios, $z_k^\omega = z_k^0$ (the initial estimate). We construct residuals based on

$$r_k^\omega = (z_k^\omega)^T (\bar{M}^\omega z^\omega + \bar{q}_k^\omega).$$

The physical interpretation of this residual vector is that scenarios with large residuals are further away from the solution than those scenarios with smaller residuals.

We propose two sampling techniques. The first technique merely sorts the residual vector and chooses the largest n_k, and is denoted by Sort. The second technique biases the true distribution by the normalized residual vector. In effect, this raises the likelihood of choosing a scenario if the the residual associated with it is large. We refer to this strategy as Samp.

TABLE 5.3
Comparison of PATH with the DS method and Monte-Carlo-DS

n	s	PATH	$\|f_*^{DS} - f_*\|$	DS	$\|f_*^{MC} - f_*\|$	MC
5	30	1.08	2.7e-7	1.67	4.4e-7	1.23
5	35	1.20	3.4e-7	1.35	6.4e-7	1.05
5	40	1.44	3.2e-7	1.64	1.6e-3	0.40
5	45	1.67	2.8e-7	1.85	2.9e-7	1.65
5	50	2.04	2.8e-7	2.04	6.0e-7	1.61
5	55	2.26	2.8e-7	2.21	4.6e-7	1.73
5	60	2.63	2.9e-7	2.62	9.1e-7	2.23
5	65	2.94	2.7e-7	2.66	7.2e-7	2.84
5	70	3.72	2.6e-7	2.89	1.0e-6	1.73
5	75	3.99	2.6e-7	3.02	9.8e-7	2.50
5	80	4.52	2.7e-7	3.30	1.1e-6	2.76
5	85	5.11	2.8e-7	3.43	3.3e-7	3.03
5	90	5.69	3.0e-7	3.60	4.1e-7	2.98

5.4.4 Computational Experience

In this section, we provide two comparisons. Table 5.3 provides a comparison between PATH, the DS method and the MC sampling method. The implementation was tested on a 1.8GHz Pentium with 512 MB of RAM running Windows XP. Table 5.4 compares the behavior of the DS method and the three sampling methods discussed in section 5.4.3, viz. MC, Sort, and Samp. Our basis of comparison is a set of equilibrium problems based on an n-node network with s scenarios. The resulting deterministic problems are of the order of $n^2 s$. The termination criterion in the DS methods and its sampling variations is

$$\|f_k - f_{k-1}\| < 10^{-5}. \tag{5.19}$$

The initial values for the forward and spot-positions are zero. Moreover, the sampling extensions are started at $n_1 = K/4$ and are incremented by 1.7 at the end of each major iteration. When comparing PATH to the iterative methods, we use CPU time as a basis of comparison. Note that the CPU time only accounts for the calls to the solver and not for linear algebra operations. Moreover, all calls to PATH are with default options in terms of optimality criteria. However, when comparing the iterative methods, we use the number of LCPs solved. This is analogous to using the number of function and gradient evaluations for first-derivative optimization methods.

In Table 5.3, we compare the performance of PATH, the DS method and the Monte-Carlo sampling extension for problems with an n-node network with s scenarios. In each instance, we compare the obtained forward positions with that obtained from the base method, which in this case is PATH.

We observe from Table 5.3 that the growth in CPU time for PATH is nearly twice as much as for MC. Moreover, this ratio grows with the number of realizations and the size of the underlying network. We find that the real benefit of sampling methods only becomes evident for larger problems and is not clearly observable for the sizes considered in Table 5.3. Figure 5.3 shows the normalized computational effort across PATH, the DS method, and the Monte-Carlo sampling extension. The normalization is with respect to the effort to solve the problem for 30 scenarios. In Figure 5.4, we compare the DS method with the three sampling extensions, MC, Sort, and Samp. We make this comparison for a 3-node network with scenarios growing from 90 to 250. We find a clear linear

TABLE 5.4
Comparison of DS method with Monte-Carlo sampling, residual sorting and residual sampling

n	s	DS	$\|\Delta^{MC}\|$	MC	$\|\Delta^{Sort}\|$	Sort	$\|\Delta^{Samp}\|$	Samp
3	90	630	2.5e-6	398	1.3e-6	578	7.2e-7	488
3	95	665	1.3e-6	609	2.1e-6	514	9.8e-8	514
3	100	700	1.1e-7	541	5.6e-7	341	1.4e-6	641
3	105	735	2.2e-6	568	4.6e-7	463	7.4e-8	568
3	110	770	2.2e-7	595	1.4e-6	485	4.4e-7	595
3	115	805	1.1e-7	622	9.0e-8	622	1.4e-6	507
3	120	840	1.7e-6	648	2.3e-6	528	2.2e-7	528
3	125	875	1.9e-6	552	2.1e-7	677	2.8e-7	677
3	130	910	1.6e-6	703	1.9e-6	703	1.8e-7	573
3	135	945	1.4e-6	595	1.3e-6	865	2.8e-7	730
3	140	980	7.3e-7	757	7.2e-7	757	1.9e-7	757
3	145	1015	6.9e-7	349	1.5e-6	639	1.4e-6	784
3	150	1050	1.4e-6	961	5.9e-8	811	2.8e-7	361
3	155	1085	2.5e-6	682	1.8e-8	837	5.0e-7	527
3	160	1120	1.5e-6	864	4.2e-7	864	1.9e-6	864
3	165	1155	5.0e-7	893	4.2e-7	893	1.5e-6	728
3	170	1190	8.7e-9	919	3.9e-7	919	1.2e-7	919
3	175	1225	5.2e-7	946	1.0e-7	946	2.0e-7	946
3	180	1260	6.3e-7	973	5.3e-7	973	3.8e-7	973
3	185	1295	1.5e-6	815	1.6e-6	815	3.4e-7	1000
3	190	1330	2.5e-6	837	2.6e-7	1027	1.4e-6	1217
3	195	1365	3.0e-7	1053	4.8e-7	1053	1.6e-7	1053
3	200	1400	1.4e-7	1080	2.4e-6	880	2.3e-7	1080
3	205	1435	2.5e-7	1109	1.2e-7	1109	2.7e-7	1109
3	210	1470	3.9e-7	1135	1.0e-9	1135	1.9e-6	925
3	215	1505	3.6e-7	1162	1.2e-6	1377	5.0e-7	1162
3	220	1540	3.8e-7	1188	5.3e-7	1188	5.5e-7	968
3	225	1575	1.0e-8	1216	3.9e-7	1216	3.2e-7	1216
3	230	1610	3.1e-8	1243	1.3e-7	1243	4.4e-7	1243
3	235	1645	1.6e-6	1034	3.8e-7	1269	4.6e-7	1269
3	240	1680	8.8e-8	1296	1.2e-7	1296	3.3e-7	1296
3	245	1715	3.8e-7	1325	8.5e-8	1325	5.7e-7	1325
3	250	1750	1.9e-6	1101	4.4e-8	1351	5.2e-7	1351

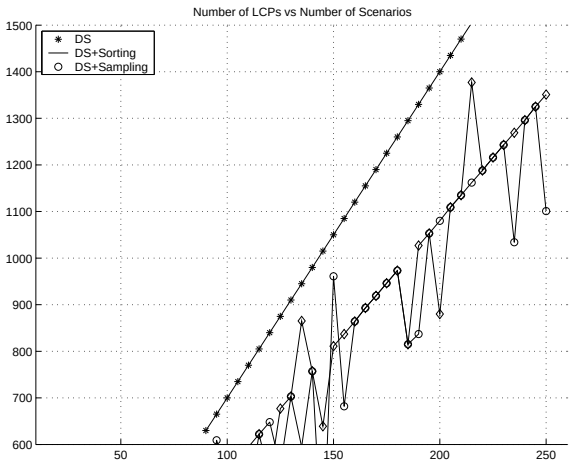

FIGURE 5.3 *Scalability: CPU time*

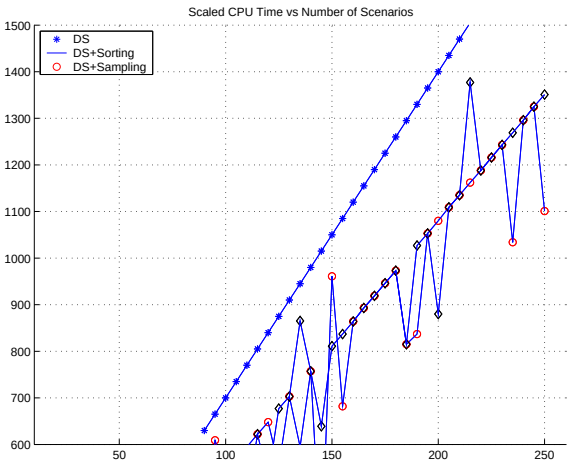

FIGURE 5.4 *Scalability: number of LCPs*

relationship between the number of LCPs solved and the number of realizations in the discrete distribution. We see that the sampling methods do significantly better than the DS algorithm. In fact, we construct a ratio between the number of LCPs solved by the sampling extension to that by the DS method. This ratio is denoted by r^{LCP}_{MC}, r^{LCP}_{Sort}, and r^{LCP}_{Samp}. We find that the mean ratios for the three methods are 0.733, 0.75, and 0.73. In general, we find that there is at least a 25% gain from the use of sampling.

To judge which sampling extension is superior, more testing is required. We find that for some problems, the Samp approach did significantly better (such as with $s = 150, 210, 220$)[6] and on other occasions both MC ($s = 145, 1043$) and Sort ($s = 200$) outperformed the rest. For some proportion of the test problems, all three methods performed the same.

5.5 An Electricity Market Model

In this section, we consider an n-node electrical network. We assume that each node houses an independent generator. Moreover, the node-arc incidence matrix of the network is given. Each firm is faced with specifying forward positions in the first period. Subject to these positions and the realization of the uncertainty, the firms then compete on a spot-market so as to meet the random demand at each node. It is assumed that there are s possible realizations that the randomness can assume.

In §5.5.1, we describe the model in greater detail and articulate the sources of uncertainty. We also discuss the four settings that we shall be using to analyze the model. Then §5.5.2–5.5.5 discuss the comparisons of the firm behavior in these settings based on expected profits, expected sales, and expected prices.

5.5.1 Description of the model

We restrict ourselves to a 6-node model with 20 scenarios in the second period ($n = 6, s = 20$). Forward sales are assumed to be endogenously priced using the function

$$p^f_i = a^f_i - m^f_i \sum_j f_{ji}, \quad i = 1, \ldots, 6. \tag{5.20}$$

Similarly spot-market prices are specified based on a random demand function

$$p^\omega_i = a^\omega_i - m^\omega_i \sum_j s^\omega_{ji}, \quad i = 1, \ldots, 6. \tag{5.21}$$

Risk-neutrality constraints are imposed along the lines of SCP_1. For all 4 settings, we assume that both forward and spot-market demand functions are given by $p_i := (40 - 10q_i)$. The cost of generation is given by $c_i g^\omega_i + \frac{1}{2} d_i (g^\omega_i)^2$, where $d_i = 0, \forall i$. Moreover, c_i is specified as in Table 5.5. Note that the cost of generator 3 changes in scenarios 2,3,4. Moreover, this cost is randomized for each of the s scenarios in a given setting, by adding a normal random variable. We assume that the

[6] s is the number of realizations in the distribution.

TABLE 5.5
Specification of cost functions

Scenario	c_i
S1	$c_i = 10$ for all i
S2	$c_3 = 150, c_{i\neq 3} = 10$ and $c_i = c_i + 0.1N(0,1)$
S3	$c_3 = 150, c_{i\neq 3} = 10$ and $c_i = c_i + 0.1N(0,1)$
S4	$c_3 = 15, c_{i\neq 3} = 10$ and $c_i = c_i + 0.1N(0,1)$

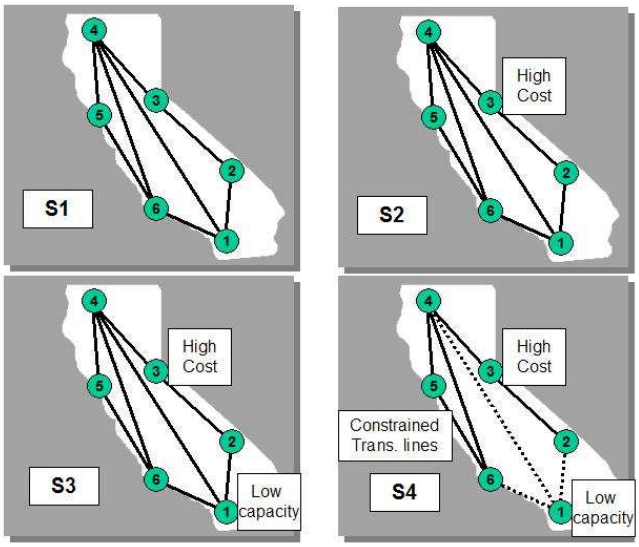

FIGURE 5.5 *Schematic of settings*

generation capacity of generator j may take value G_j^ω in realization ω. We also assume that the capacity of the single link between the nodes is given by t_{cap}^ω. Again, generation capacity is kept fixed for all scenarios but at levels specified in Table 5.6. The transmission capacity is normally distributed at a mean of 1 for the first three settings. In the last setting, the means for all linkages connecting node 1 are brought down to 0.01. Figure 5.5 provides a schematic summary of each setting. The resulting complementarity problems were solved by the PATH solver [DF93] with MATLAB 6.5 and a Windows XP operating system.

TABLE 5.6
Specification of cost functions

Scenario	G_i	t_{cap}
S1	$G_i = 1$ for all i	$t_i = 1 + 0.001N(0,1)$
S2	$G_i = 1$ for all i	$t_i = 1 + 0.001N(0,1)$
S3	$G_1 = 0.5, G_{i\neq 1} = 1$	$t_i = 1 + 0.001N(0,1)$
S4	$G_1 = 0.5, G_{i\neq 1} = 1$	$t_{12,14,16} = 0.01, t_{i\neq 12,14,16} = 1, t_i = t_i + 0.001N(0,1)$

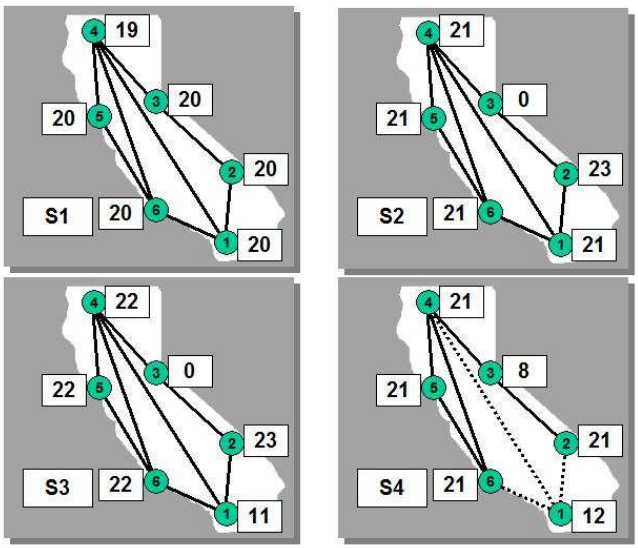

FIGURE 5.6 *Expected profits*

5.5.2 Expected Profits

Figure 5.6 shows the expected profits of each firm in each of the four settings. In S2, the main change is that the cost of generation is raised by a factor of 10 and it is impossible for the firm to make any profits on the spot-market. Note that the risk-neutrality constraints imply that the expected spot-price is equal to forward price, implying zero profits on the forward market. Also a prisoner's dilemma effect is observed in that the profits of the other firms rise in S2. In S3, generator 1 has a lower capacity, which directly impacts its spot profits. It may be noticed that the remaining firms see an increment in profit. Finally, in S4, the transmission linkages emanating from node 1 are congested. Firm 1's profits see a small upward change because the expected spot-price rises slightly (see Figure 5.8).

5.5.3 Expected Spot-market Sales

In Figure 5.7, it can be seen that all the generation constraints are binding in S1. In S2, this still holds except for generator 3, which does not generate any power. In S3 and S4, generator 1 supplies at 0.5 (the upper bound on its capacity in those two settings). Note that the cost of firm 3's generation is dropped in S4, resulting in firm 4 generating 0.48 units of power.

5.5.4 Forward Prices

Finally, Figure 5.8 shows the forward prices at each of the nodes in each setting. By the risk-neutrality constraint, the expected spot-price is equal to the forward price. It can be seen that steadily constraining the model results in higher prices (S2 and S3). In S4, the connectivity to node 1 is congested, implying that less of its demand can be met, implying that its spot-price goes up.

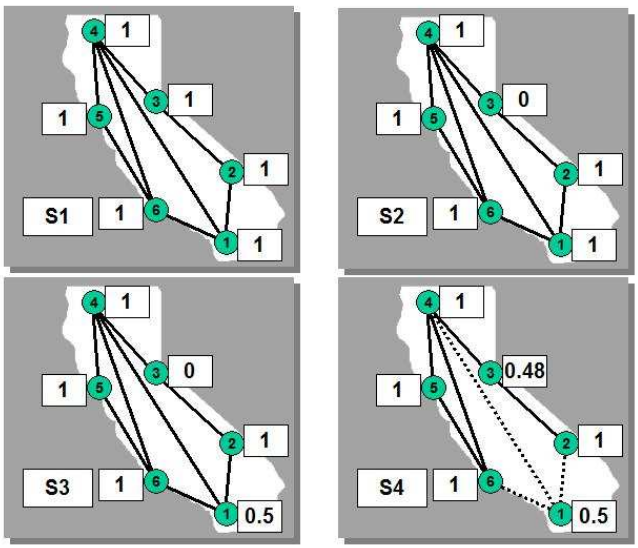

FIGURE 5.7 *Expected sales*

Also interesting is that despite congestion, profits of firm 1 increase from 11 to 12.

5.6 Contributions and Future Research

This chapter is motivated by the difficulties in obtaining a Nash-Stackelberg equilibrium in a multiperiod equilibrium problem under uncertainty. We introduce a simultaneous stochastic Nash equilibrium in the context of a forward contracting model. Existence of such an equilibrium is proved.

We compare the SSNE with the NSE and computationally show that the SSNE is indeed an NSE. Also, the effort to find an NSE is far less if one begins from an SSNE. We prove that the SSNE obtained in simpler settings is indeed an NSE. We also demonstrate computationally that this is indeed the case.

The SSNE may be obtained as a solution to a stochastic mixed-complementarity problem. We present a *scalable* algorithm for solving large-scale problems. The convergence of both algorithms to the equilibrium point is proved. Computational tests are used to compare the scalability of the algorithm relative to PATH. Also, sampled variants of the algorithms are shown to perform even better.

Finally, some insights are provided from a 6-node electricity market model with stochastic demand and capacity. The following intuitive results are obtained from the model:

- Under the possibilities of generator outage, agents participate in the forward market to a greater extent. This compensates for lost generator sales.
- If we introduce a high chance of network congestion in the linkages to a particular node, then the prices at that node increase. In fact, a firm may make more profit on average

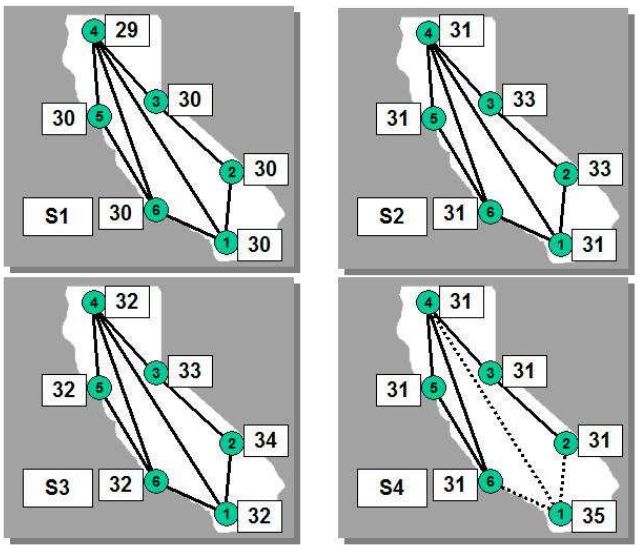

FIGURE 5.8 *Expected prices*

under congestion providing there is a somewhat perverse incentive in the presence of vertical integration.

Several questions emerge from this chapter and shall be tackled as future research:

- An important question that has been partially answered is the existence of Nash-Stackelberg equilibria. It seems that under somewhat strong conditions, one may show existence. Can one further weaken the assumptions required to claim existence of such equilibria?
- We present one of the first few algorithms for stochastic complementarity problems. One question is whether these ideas may be formalized to construct an algorithm for more general, albeit, monotone complementarity problems.
- We present a reasonably general electricity market model. It is necessary to add more realistic features to this model. One question that has received some attention has been the use of piecewise-linear demand functions. Also of relevance is the implicit dynamics in the system, so far ignored. In particular, one may need to introduce ramping constraints to constrain the change in output of a generator. This would require mixed-integer formulations.

Bibliography

[All92] B. Allaz. Oligopoly, uncertainty and strategic forward transactions. *Journal of Industrial Organization*, 10:297–308, 1992.

[Ani05] M. Anitescu. On solving mathematical programs with complementarity constraints as nonlinear programs. *SIAM J. Optim.*, 15(4):1203–1236, 2005.

[Bas04] F. Bastin. *Trust-Region Algorithms for Nonlinear Stochastic Programming and Mixed Logit Models*. PhD thesis, Department of Mathemtics, University of Namur, 2004.

[Ben62] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4:238–252, 1962.

[BL97] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming: Springer Series in Operations Research*. Springer, 1997.

[CPS92] R. W. Cottle, J-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, Inc., Boston, MA, 1992.

[CW00] X. Chen and R. S. Womersley. Random test problems and parallel methods for quadratic programs and quadratic stochastic programs. *Optim. Methods Softw.*, 13(4):275–306, 2000.

[Dan55] G. B. Dantzig. Linear programming under uncertainty. *Management Sci.*, 1:197–206, 1955.

[DF93] S. P. Dirkse and M. C. Ferris. The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems. *CS-TR-1993-1179*, Department of Computer Science, University of Wisconsin at Madision, 1993.

[DG89] G. B. Dantzig and P.W. Glynn. Parallel processors for planning under uncertainty. *Ann. Oper. Res.*, 22:1–21, 1989.

[DI95] G. B. Dantzig and G. Infanger. A probabilistic lower bound for two-stage stochastic programs. Technical Report *SOL 95-6*, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1995.

[DM02] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002.

[dMFNS05] A.V. de Miguel, M.P. Friedlander, F.J. Nogales, and S. Scholtes. A two-sided relaxation scheme for mathematical programs with equilibrium constraints. *SIAM J. Optim. (to appear)*, 2005.

[Dor61] W. S. Dorn. Duality in quadratic programming. *Quart. Appl. Math.*, 18:155–162, 1960/1961.

[Doy03] M. Doyle. *A Barrier Algorithm for Large Nonlinear Optimization Problems*. PhD thesis, Scientific Computing and Computational Mathematics (SCCM) Program, Stanford University, 2003.

[FGM95] A. Forsgren, P. E. Gill, and W. Murray. Computing modified Newton directions using a partial Cholesky factorization. *SIAM J. Sci. Comput.*, 16(1):139–150, 1995.

[FLRS02] R. Fletcher, S. Leyffer, D. Ralph, and S. Scholtes. Local convergence of SQP methods for mathematical programs with equilibrium constraints. Technical report, Department of Mathematics, University of Dundee, UK, Numerical Analysis Report NA/209., 2002.

[FP03] F. Facchinei and J-S. Pang. *Finite Dimensional Variational Inequalities and Complementarity Problems: Vols I and II*. Springer-Verlag, NY, Inc., 2003.

[GMS97] P. E. Gill, W. Murray, and M. A. Saunders. User's guide for SQOPT 5.3: A Fortan package for large-scale linear and quadratic programming. Technical report, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1997.

[GMS05] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47(1):99–131 (electronic), 2005.

[GMW81] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, Boston, MA, USA, 1981.

[GOR99] G. Gurkan, A.Y. Ozge, and S.M. Robinson. Sample-path solution of stochastic variational inequalities. *Mathematical Programming*, 84:313–333, 1999.

[HMP00] B. F. Hobbs, C. B. Metzler, and J-S. Pang. Strategic gaming analysis for electric power systems: An MPEC approach. *IEEE Transactions on Power Systems*, 15:638–645, 2000.

[Hol99] K. Holmström. The TOMLAB Optimization Environment in Matlab. *Advanced Modeling and Optimization*, 1(1):47–69, 1999.

[HR04] X. M. Hu and D. Ralph. Convergence of a penalty method for mathematical programming with complementarity constraints. *J. Optim. Theory Appl.*, 123(2):365–390, 2004.

[HS91] J. Higle and S. Sen. Stochastic decomposition: An algorithm for two stage linear programs with recourse. *Mathematics of Operations Research*, 16(3):650–669, 1991.

[HS96] J. Higle and S. Sen. *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*. Kluwer Academic Publishers, Boston, MA., 1996.

[Inf92] G. Infanger. Monte Carlo (importance) sampling within a Benders decomposition algorithms for stochastic linear programs. *Ann. Oper. Res.*, 39:41–67, 1992.

[Inf94] G. Infanger. *Planning Under Uncertainty*. Boyd and Fraser Publishing Co., 1994.

[JR99] H. Jiang and D. Ralph. QPECGEN, a Matlab generator for mathematical programs with quadratic objectives and affine variational inequality constraints, and its application to algorithms. *Computational Optimization and Applications*, 13:25–29, 1999.

[JR00] H. Jiang and D. Ralph. Smooth SQP methods for mathematical programs with nonlinear complementarity constraints. *SIAM Journal on Optimization*, 10(3):779–808, 2000.

[LCF03] G-H. Lin, X. Chen, and M. Fukushima. Smoothing implicit programming approaches for stochastic mathematical programs with linear complementarity constraints. *Technical Report, 2003-006, Department of Applied Mathematics and Physics, Kyoto University*, 2003.

[Ley00] S. Leyffer. macMPEC: AMPL collection of MPECs. Technical report, University of Dundee, 2000.

[LLCN05] S. Leyffer, G. Lopez-Calva, and J. Nocedal. Interior methods for mathematical programs with complementarity constraints. *SIAM J. Optim. (to appear)*, 2005.

[LPR96] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, Cambridge, 1996.

[LS04a] X. Liu and J. Sun. Generalized stationary points and an interior-point method for mathematical programs with equilibrium constraints. *Math. Program.*, 101(1, Ser. B):231–261, 2004.

[LS04b] X. Liu and J. Sun. A new decomposition technique in solving multistage stochastic linear programs by infeasible interior point methods. *Journal of Global Optimization*, 28(4):197–215, 2004.

[MP95a] W. Murray and F. J. Prieto. A second-derivative method for nonlinearly constrained optimization. Technical Report SOL, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, 1995.

[MP95b] W. Murray and F. J. Prieto. A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM J. Optim.*, 5(3):590–640, 1995.

[MP03] J. M. Moguerza and F. J. Prieto. An augmented Lagrangian interior-point method using directions of negative curvature. *Math. Program.*, 95(3, Ser. A):573–616, 2003.

[MS79] J. J. Moré and D. C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Math. Programming*, 16(1):1–20, 1979.

[Ng03] K-M. Ng. *A Continuation Approach for Solving Nonlinear Optimization Problems with Discrete Variables*. PhD thesis, Department of Management Science and Engineering, Stanford University, 2003.

[NW99] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[OKZ98] J. Outrata, M. Kočvara, and J. Zowe. *Nonsmooth Approach to Optimization Problems with Equilibrium Constraints*, volume 28 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, 1998. Theory, applications and numerical results.

[Pie01] H. Pieper. *Algorithms for Mathematical Programs with Equilibrium Constraints with Applications to Deregulated Electricity Markets*. PhD thesis, Department of Management Science and Engineering, Stanford University, 2001.

[RB05] A. U. Raghunathan and L. T. Biegler. An interior point method for mathematical programs with complementarity constraints (MPCCs). *SIAM J. Optim.*, 15(3):720–750 (electronic), 2005.

[RS93] R. Y. Rubinstein and A. Shapiro. *Discrete Event Systems*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Ltd., Chichester, 1993.

[RS03] A. Ruszczynski and A. Shapiro. Introduction. In *Handbook in Operations Research and Management Science*, volume 10, pages 1–64. Elsevier Science, Amsterdam, 2003.

[Rus03] A. Ruszczynski. Decomposition methods. In *Handbook in Operations Research and Management Science*, volume 10, pages 141–212. Elsevier Science, Amsterdam, 2003.

[Sch01] S. Scholtes. Convergence properties of a regularization scheme for mathematical programs with complementarity constraints. *SIAM J. Optim.*, 11(4):918–936 (electronic), 2001.

[SCTB88] F. C. Schweppe, M. C. Caramanis, R. D. Tabors, and R. E. Bohn. *Spot Pricing of Electricity*. Kluwer Academic Publishers, 1988.

[Sha03] A. Shapiro. Monte carlo sampling methods. In *Handbook in Operations Research and Management Science*, volume 10, pages 353–426. Elsevier Science, Amsterdam, 2003.

[Sta52] H. V. Stackelberg. *The Theory of Market Economy*. Oxford University Press, London, 1952.

[SX05] A. Shapiro and H. Xu. Stochastic mathematical programs with equilibrium constraints, modeling and sample average approximation. *Optimization-Online*, 2005.

[VS99] R. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.

[VSW69] R. M. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM J. Appl. Math.*, 17:638–663, 1969.

[YOA04] J. Yao, S. Oren, and I. Adler. Computing Cournot equilibria in two settlement electricity markets with transmission constraints. In *Proceeding of the 38th Hawaii International Conference on Systems Sciences (HICSS 38). Big Island, Hawaii*, 2004.

[ZPR00] G. Zakeri, A. B. Philpott, and D. M. Ryan. Inexact cuts in Benders decomposition. *SIAM J. Optim.*, 10(3):643–657, 2000.