Stochastic Optimization of Electricity Transmission: Dynamic Programming Algorithms under Uncertainties

by

Ishwar Krishnan Ashok Sivakumar

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2002

Author ....
Department of Electrical Engineering and Computer Science
May 18, 2002

Certified by ....
Marija D. Ilić
Senior Research Scientist
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Stochastic Optimization of Electricity Transmission: Dynamic Programming Algorithms under Uncertainties

by

Ishwar Krishnan Ashok Sivakumar

Submitted to the Department of Electrical Engineering and Computer Science on May 18, 2002, in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis is motivated by the need to optimize transmission capacity as electricity trades take place in an electric power system. The results obtained use actual data taken from New England electricity markets for a working week in February, 2002. The power demand in the New England system is correspondingly split into anticipated multilateral market demand and leftover spot demand. The problem is studied by posing the decision making for transmission service and pricing as a dynamic programming problem. The implementation handles stochastic inputs for a three-bus system. This thesis suggests that near-optimum objective values can be achieved even when generation and transmission of electricity are treated in an unbundled manner. The multilateral agreements are modeled and their effects on network congestion are simulated. The end users communicate and coordinate with each other, providing demand functions that reveal their internalized value of transmission. The spot demand is analyzed under different probabilistic models to estimate the inherent uncertainties.

A deterministic simulation is created to automate the multilateral trading process. It outputs the multilateral agreement profit as a function of forward market capacity allocation. The program is also run to simulate an entirely multilateral market structure, providing insights regarding total social welfare, end user quantities, and profits. Moving to a probabilistic regime, the profits gained from the deterministic multilateral setup and the statistical distributions of the noisy spot market are fed as inputs into a dynamic programming simulation. The case of a broken topology is then considered for both cases, where one of the non-congested links has such high impedance levels that flow along this link is severely reduced. The significance of such smart software for a transmission service provider is shown, together with possible new frameworks to further optimize the long-term transmission resource allocation.

Thesis Supervisor: Marija D. Ilić
Title: Senior Research Scientist

Acknowledgments

I am extremely indebted and thankful to my thesis supervisor Maria Ilić for guiding me along and helping me understand the beautiful results of research. Her patient advice and motivational attitude truly brought out some of my best work in this thesis, and I am extremely grateful to have had such an advisor.

In addition, much thanks goes to Özge Gözüm, Eric Hsieh, and Prasad Ramanan. Özge was the first to introduce me to this subject, and her mentoring throughout this year has been something I sincerely appreciate. She has been an exemplary role-model since my junior year and I cannot thank her enough for her impact on the work presented here. There is no better way to thank Eric and Prasad than to state that my thesis would not have been finished on time without their help. I am truly indebted to them for their relentless attitude towards understanding the given problem and implementing the deterministic simulation. I learned much more through working with them, and I hope they receive similar good fortune of timely help when pursuing their graduate studies next year.

MIT has taught me so many things in the past five years, but most importantly, it has helped me truly love learning and the pursuit of knowledge. While classes and books can teach fundamentals, I believe this feeling was brought on by my interactions with faculty and instructors. Specifically, I would like to thank Dr. Amar Bose and the late Prof. Gian-Carlo Rota for opening my eyes in ways I never thought imaginable. I would also like to thank Prof. George Verghese for being an excellent undergraduate advisor and a great friend. Furthermore, I have to extend my deepest gratitude to MAJ Tony Cho for his inspiration and positive encouragement in all that I attempted at MIT. I am sure that what I learned under his supervision as a cadet will not only benefit me during my time in the military, but will stay with me lifelong. Finally, I want to thank Dr. Ben Sachtler for supervising my undergraduate research, and teaching me to see the fun in experimental science.

Futhermore, I owe many thanks to Julius Kusuma and Sripriya Natarajan; their help with the figures in this thesis was an invaluable contribution. It was also great to have Julius as an officemate, and I hope to one day emulate his work ethic and discipline. Priya has been a great friend over the past five years, and I cannot say enough about her caring personality, and positive encouragement. There are too many other friends to list here, but I genuinely appreciate all of their efforts to ensure my well-being and happiness over the past five years.

Futhermore, I also must thank the members of the LIDS lab and E-Lab group for their helpful insights regarding the problem of interest. Specifically, my conversations with Hungjen Wang and Poonsaeng Visudhiphan were always englightening, and their willingness to help me understand key concepts is greatly appreciated. Many of the ideas and formulations presented here would not have been possible without their time and patience. I would also like to mention Aurelie Thiele, Shan-Yuan Ho, and Wolfgang Gatterbauer for their friendship and always cheering me up during the writing process.

In closing, I owe my deepest thanks to my family for their never-ending love and affection. I would like to dedicate this thesis to my parents, Ganga and Venkataraman Sivakumar, and my sister Anjana, as my small way of saying thanks for everything they have taught me in life.

The work in this thesis was funded by the Merrill Lynch project on electricity markets. This support is greatly appreciated.

Contents

1 Introduction **17**

1.1 Context 17

1.2 Problem Statement 18

1.2.1 Topology Changes 30

1.2.2 Decision Models 30

1.2.3 Market Structure Comparisons 31

1.3 Thesis Summary and Key Results 31

2 Background and Preliminaries **33**

2.1 Electricity and Transmission Markets 33

2.1.1 Economic Efficiency Arguments 35

2.2 Algorithms for Transmission Constrained Markets 37

2.2.1 Optimal Power Flow 37

2.2.2 Dynamic Programming 41

3 Bilateral and Multilateral Agreements **47**

3.1 Theory and Implementation of Multilateral Trades 52

3.2 Summary 54

4 Spot Market **55**

4.1 Characterization of Load Demand Uncertainty 55

4.2 Stochastic Models 56

4.2.1 Basic version of OPF 56

4.2.2 Bounding the Spot Market Loss . . . 57

5 Simulation of Multilateral Trading 61

5.1 Initial Conditions and Assumptions in Simulation Example . . . 61

5.1.1 Generator Cost Functions . . . 62

5.1.2 Load Utility Functions . . . 63

5.1.3 Congested Link Capacity . . . 65

5.1.4 Curtailment Strategy . . . 66

5.1.5 Distribution Factors and Broken Network . . . 66

5.2 Quantities of Individual End-Users . . . 66

5.2.1 Effects of the Capacity Constraint . . . 66

5.2.2 Impact of the Load Demand Level . . . 69

5.3 Participant Profits . . . 71

5.3.1 Effects of the Capacity Constraint . . . 71

5.3.2 Impact of the Load Demand Level . . . 73

5.4 TSP profits . . . 75

5.5 Total Social Welfare . . . 77

5.5.1 Effects of the Capacity Constraint . . . 77

5.5.2 Impact of the Load Demand Level . . . 79

5.6 Different Criteria Used to Choose Agreements . . . 81

5.6.1 Individual Participants Quantities . . . 81

5.6.2 Total Social Welfare . . . 83

5.7 Demand Curves for Transmission . . . 85

6 Multilateral Trading Over The Week 87

6.1 Individual End-User Quantities Over the week period . . . 88

6.2 TSP Profits or Sum of End-user Profits . . . 92

6.3 Total Social Welfare over the week . . . 94

7 TSP Profits using Dynamic Programming 99

7.1 Normal Network case . . . 101

7.2 Broken Network case 104

8 Conclusions and Future Research **107**

8.1 Future Research 108

8.1.1 Investment Protocols 109

8.1.2 Network Analysis 109

8.1.3 Hidden Markov Models 109

9 Source Code **111**

List of Figures

1-1 The basic TSP decision problem 20
1-2 TSP overall decision tree 21
1-3 TSP Multilateral market decision tree 24
1-4 Anticipated Demand for one week period 27
1-5 Anticipated Demand in Multilateral Market 27
1-6 Anticipated Demand in Spot Market 28
1-7 General 3-bus network 29
1-8 Uncertainties 30

3-1 General 3-bus Network 48
3-2 Optimally Efficient Trade 49
3-3 Flows after curtailment 49
3-4 New Trade 51

5-1 Candidate multilateral sections for combined market 62
5-2 Quantities for Night Demand, Line Capacity=100 67
5-3 Quantities for Night Demand, Line Capacity=700 67
5-4 Quantities for Night Demand, Line Capacity=1200 68
5-5 Quantities for Night Demand, Line Capacity=700 69
5-6 Quantities for Daytime Demand, Line Capacity=700 70
5-7 Quantities for Evening Demand, Line Capacity=700 70
5-8 Profits for Line Capacity=100, Night Demand 71
5-9 Profits for Line Capacity=700, Night Demand 72
5-10 Profits for Line Capacity=1200, Night Demand 72

5-11 Profits for Line Capacity=700, Night Demand 73
5-12 Profits for Line Capacity=700, Daytime Demand 74
5-13 Profits for Line Capacity=700, Evening Demand 74
5-14 TSP Profits for Night Demand of 11775 MW 75
5-15 TSP Profits for Daytime Demand of 17273 MW 76
5-16 TSP Profits for Evening Demand of 18594 MW 76
5-17 Total Social Welfare for Line Capacity=100, Night Demand 77
5-18 Total Social Welfare for Line Capacity=700, Night Demand 78
5-19 Total Social Welfare for Line Capacity=1200, Night Demand 78
5-20 Total Social Welfare for Line Capacity=700, Night Demand 79
5-21 Total Social Welfare for Line Capacity=700, Daytime Demand . . . 80
5-22 Total Social Welfare for Line Capacity=700, Evening Demand 80
5-23 End User driven case 81
5-24 Network Driven case 82
5-25 End User driven case 83
5-26 Network Driven case 84
5-27 Transmission Demand Curve, Anticipated demand = 11775 MW . . 85

6-1 Load Quantities, 100 MW 88
6-2 Load Quantities, 700 MW 89
6-3 Load Quantities, 1100 MW 89
6-4 Generator Quantities, 100 MW 90
6-5 Generator Quantities, 700 MW 90
6-6 Generator Quantities, 1100 MW 91
6-7 TSP Profits, 100 MW 92
6-8 TSP Profits, 700 MW 93
6-9 TSP Profits, 1100 MW 93
6-10 Total Social Welfare, 100 MW 94
6-11 Total Social Welfare, 700 MW 95
6-12 Total Social Welfare, 1100 MW 95

6-13 TSP Profits: Broken Network, 700 MW 96
6-14 Total Social Welfare: Broken Network, 700 MW 97

7-1 Anticipated Demand in Multilateral Market 100
7-2 Anticipated Demand in Spot Market 101
7-3 Continuous Time 102
7-4 Discrete Time 103
7-5 Continuous Time 104
7-6 Discrete Time 105

List of Tables

4.1 Statistics for Spot Demand 57

5.1 Generator Cost parameters 63

5.2 Load Utility parameters 64

5.3 Maximum Line Capacity 65

Chapter 1

Introduction

1.1 Context

As news regarding the California blackouts and the Enron scandal continue to plague the headlines, the issue of electricity trading and its delivery can no longer be taken lightly. Efficient mechanisms have been proposed in the past to deal with the generation and transmission of electricity, yet a universally accepted solution still remains to be found. In this analysis, it is suggested that electricity generation and transmission be treated as two independent separable problems. Furthermore, in the deregulated industry, increased competition has led to multilateral coordination by end users to maximize profits. The implementation of socially most beneficial transactions under transmission constraints has been a very difficult challenge. A transmission service provider (TSP) in charge of implementing transactions has not been able to facilitate the most valuable transactions, nor to relate those to its own business objectives. Instead, a TSP is forced to act as an arbiter among end users, solely in charge of ensuring reliability and security through any network transactions.

In this thesis, we consider the objectives of a transmission service provider and examine policies, strategies, and algorithms for meeting its clearly defined goals. These go beyond technical objectives and could be social welfare maximization and/or a TSP's own profits. Specifically, the TSP's main resource involves the capacity of the links connecting end users in the network, and its scarcity forces the TSP to optimize its allocation and price. However, the decisions by the TSP are based on the end users' specification of demand for transmission capacity and their willingness to pay for it.

This thesis takes the view of a centrally coordinated setup where a TSP is the final decision maker. Several cases are examined and analyzed for varying network topology, decision algorithms, and market structures. Heuristic measures and algorithms are introduced for a TSP to maximize his own revenue and/or total social welfare.

It is not common to find an analysis of the impact of market and physical uncertainties on end users, total social welfare, and transmission revenue. In this thesis, we suggest that these fundamental concepts differ as a function of market structure, initial conditions, and uncertainties. This thesis attempts to implement a number of the initial formulations to this problem introduced by Gözüm in [1], and to provide simulations to better understand the dynamics of transmission provision and pricing. Building upon this original work [1], the scope is enlarged to include multilateral agreements (MA), and increase the complexity of the system to a 3-bus example with demand and supply requirements at each bus. While we assume the existence of forward and spot energy markets, the transmission service allocation problem is investigated here as an independent problem. The strict technical constraints of the network and the numerous opportunities for arbitrage between contract flows and physical flows (as discussed in section 3.2) increase the complexity herein.

Finally, the contribution of this thesis is in conceptualizing algorithms and software implementations for transmission delivery, a fairly new concept when compared to the pure energy formulations that have been studied in the past.

1.2 Problem Statement

In each of the several cases, the basic problem remains the same, i.e. it concerns capacity allocation. The TSP obtains revenue from two sources - he can sell longer term forward contracts *ex ante* or transmission rights in a real-time setting *ex post.*

The total load demand for transmission capacity here can be thought of as consisting of two components: the anticipated load demand level that most bilateral and multilateral agreements would consume, and the leftover load demand for the spot market. Other inputs are generator specific, including cost of generation quantities and maximum generation capacities per hour. The TSP also has estimates for what percentage of the total load will be consumed at each bus, and can hence approximate the utility functions of each load. Finally, the TSP is aware of which links are most likely to be congested and what the capacity constraints of these links are.

This thesis assumes a centrally coordinated system where the TSP is given entire schedules and bids at the start of the week, as well as demand curves reflecting each end user's value of transmission. From Gözüm[1], we see that one version of this problem is to have users submit bilateral agreements, that are either accepted or rejected by the TSP at each hour. The alternative formulation this thesis takes has users submit transmission demand curves for varying levels of capacity on a specific link in the network. Along with such curves, the end users imply that at a set level of capacity, they will take part in trilateral agreements that maximize the user's profits. The TSP realizes that such agreements not only push the total social welfare closer to an optimum level, but also receives a fraction of these profits for implementing them.

Hence, the control decision at each time step in question, is what level of line capacity to reserve for multilateral agreements and hence the corresponding allocation for the spot market. This is shown in Figure 1-1. At each time step, the TSP's decision is to choose what allocation of flow along the congested link should come from the spot market, denoted here as $\kappa_{s,t}$. The inputs to the decision making problem are the "noise" in load data at each hour, ω_t and the expected profits from the multilateral trading process for a set demand level and known thermal capacity of congested link, $\Pi_{D,K_{i \to j}^{max}}$. The TSP must use these inputs to optimally choose $\kappa_{s,t}$.

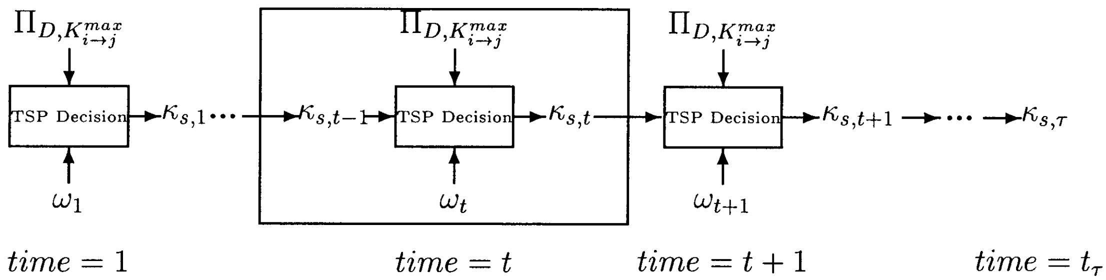

Figure 1-1: The basic TSP decision problem

The important constraint regarding this control decision is the fact that such multilateral agreements cannot be rescinded once approved. This thesis does not allow for penalty functions; once point to point agreements are permitted, they are enforced for the entire length of the agreement.[1]

This idea is best represented in a decision tree for the TSP. Such a tree in which each leaf node represents the TSP's total profits is shown in Fig. 1-2.

[1]As described by Joskow and Tirole in [19] and [20], there is a distinction between financial rights and physical rights. The TSP here is guaranteeing the physical right along the congested link; the financial rights are out of the scope of this thesis.

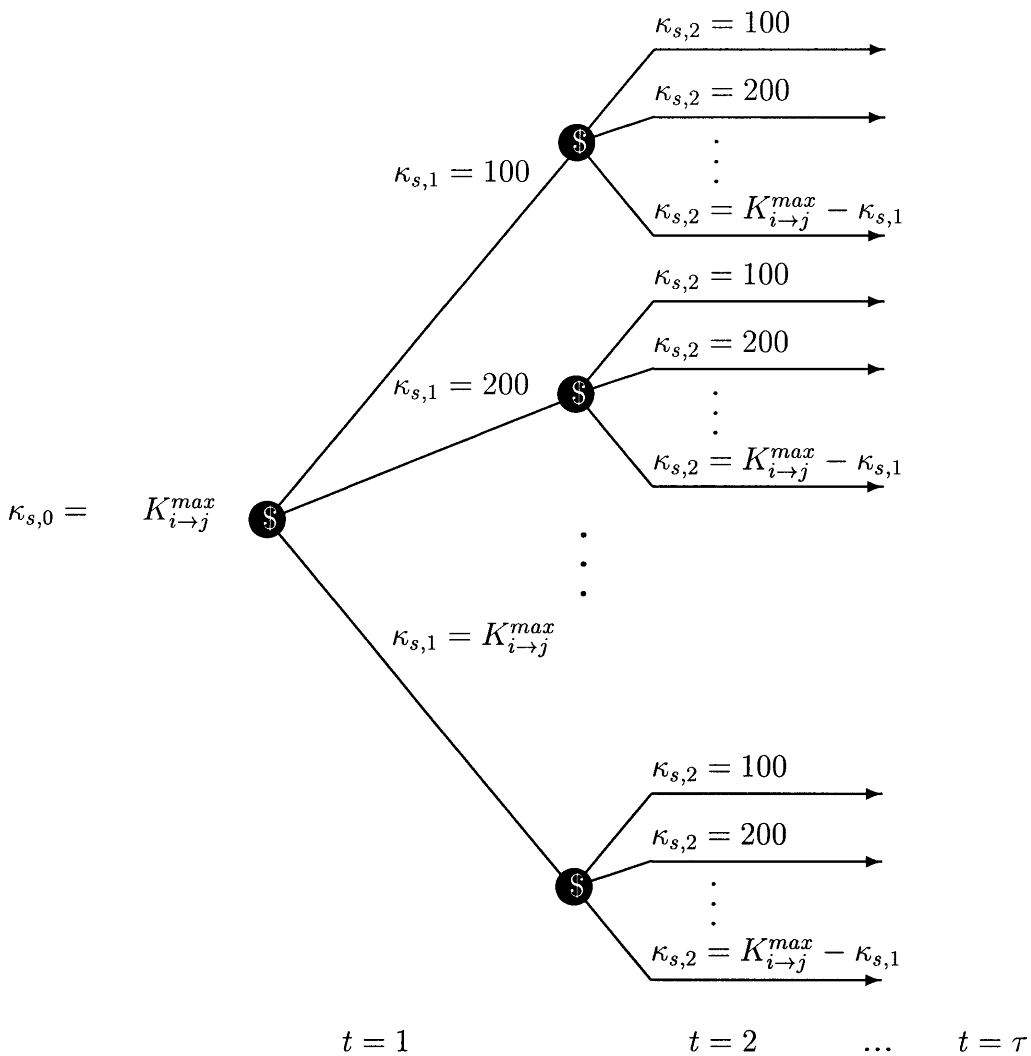

Figure 1-2: TSP overall decision tree

This decision tree shows precisely that certain choices for spot allocation at hour (t+1), may be limited depending upon the choice of the previous hour. This is a result of our assumption that multilateral contracts cannot be retracted once approved, and hence the next hours allocation for the multilateral market cannot decrease. This implies that $\kappa_{s,t}$ cannot increase either. This decision tree also shows our assumption that the TSP must make his decision in discrete increments of 100 MW. This was chosen for computational convenience, and an extension to a smaller increment size (and hence more choices for the TSP) easily follow.

In this setup, the objective of the TSP is to maximize his revenues from the forward and real-time markets for a finite time-horizon. The tradeoff comes from realizing that forward agreements, while more risk-averse may consume much of the line's capacity for a length of time, whereas the spot market is more flexible to changes in real-time demand, but it comes with a more volatile price and risk level. Furthermore, the TSP must meet strict flow constraint of links , set by the technical parameters of the network. We can formulate the objective function as:

$$\max_{\kappa_s,\kappa_{ma}} \sum_{t=1}^{\mathcal{T}} \underbrace{\Pi_{\kappa_s,t}}_{\text{Spot Profit}} + \underbrace{\Pi_{\kappa_{ma},t}}_{\text{Multilateral Profit}} \tag{1.1}$$

Noting that the spot profit is simply the product of the quantity (i.e. transmission capacity) sold times the spot price of transmission, we can re-write equation (1.1). Thus, the entire formulation is as follows.

Objective:

$$\max_{\kappa_s,\kappa_{ma}} \sum_{t=1}^{\mathcal{T}} \widehat{P}_{spot,t}\widehat{Q}_{spot,t}(\kappa_s,\omega_t) \quad + \quad \Pi_{\kappa_{ma},t} \tag{1.2}$$

subject to:

$$\kappa_{ma} + \kappa_s \leq K_{i\to j}^{max} \tag{1.3}$$

$$\kappa_{s,t+1} \leq \kappa_{s,t} \tag{1.4}$$

where,

$K_{i\to j}^{max}$ = Physical Capacity of Line[2]

κ_s = Amount of flow on congested link due to spot market

κ_{ma} = Amount of flow on congested link due to multilateral market

t = Discrete time index

$$
\begin{aligned}
\tau &= \text{Time Horizon for TSP} \\
\Pi_{\kappa_{ma},t} &= \text{Profit determined from multilateral market} \\
\widehat{P}_{spot,t} &= \text{Expected spot price of transmission on link } 1 \rightarrow 2 \\
\widehat{Q}_{spot,t} &= \text{function to determine expected spot quantity on link } 1 \rightarrow 2 \\
\omega_t &= \text{noise following known statistical distribution}
\end{aligned}
$$

As seen above, the TSP's profit maximization problem assumes the multilateral market trading settles, and end users profits from implementing these trilateral agreements will be transferred to the TSP. Thus, it remains to be shown how this exact profit, κ_{ma} is derived.

We assume that the TSP treats all end-users equally and hence, the objective function for implementing one multilateral agreement over another should be fair to all users. The best heuristic would then be to choose the multilateral agreement that maximizes the total social welfare of all users at the current time period. As shown in[25], multilateral trading opportunities for profit exist even after the first multilateral agreement has been implemented. In other words, the TSP must attempt to optimally choose a sequential set of multilateral agreements with this objective function in mind for a series of iterations, as shown in Figure 1-3.

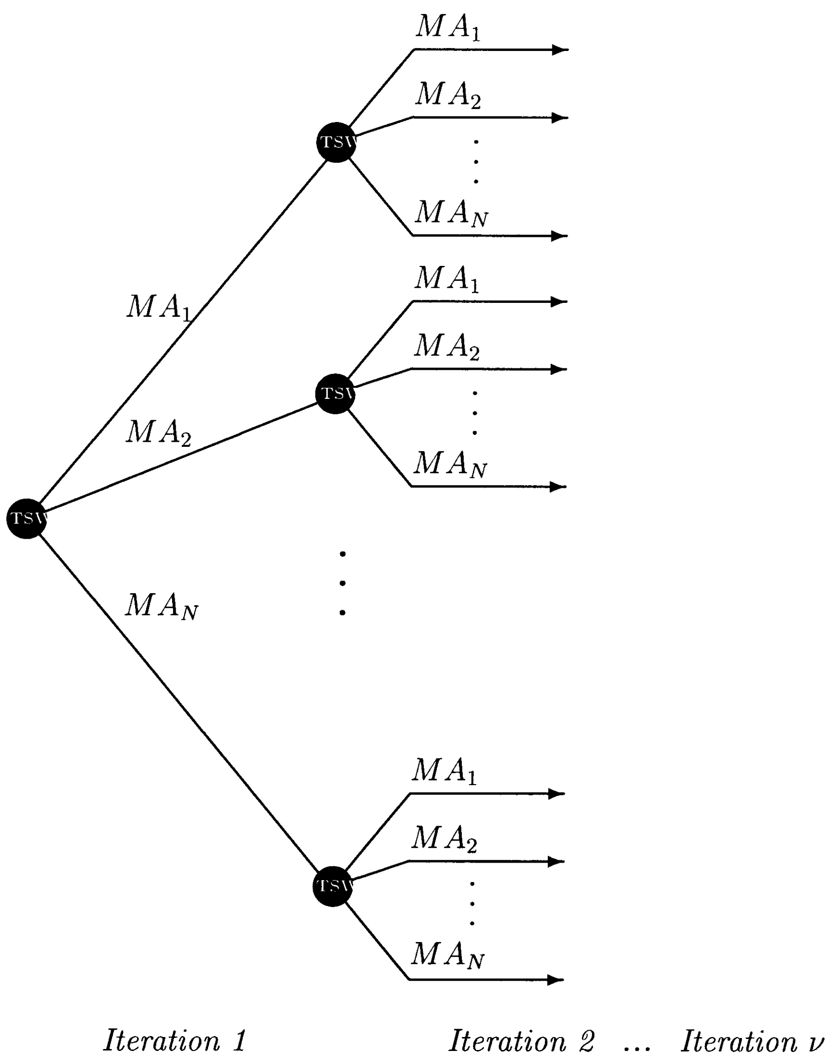

Figure 1-3: TSP Multilateral market decision tree

We formally present this problem as the following:

Objective:

$$\sum_{v=1}^{\nu} \max_{(y_{j,v}, z_{i,v})} \left[\sum_{i=1}^{N_L} U_i(Q_{L_i}) z_{i,v} \; - \; \sum_{j=1}^{N_G} C_j(Q_{G_j}) y_{j,v} \right] \tag{1.5}$$

such that,

$$
\begin{aligned}
\sum_{k=1}^{N} (y_k Q_{G_k} - z_k Q_{L_k}) DF_k^{link} &\leq \kappa_{ma} && (1.6)\\
Q_{G_i} - Q_{L_i} &= 0 && (1.7)\\
y_{j,v} &\in \{0,1\} && (1.8)\\
z_{i,v} &\in \{0,1\} && (1.9)\\
1 \leq \sum_j y_{j,v} &\leq 3 && (1.10)\\
0 \leq \sum_i z_{i,v} &\leq 1 && (1.11)\\
\sum y_{j,v} + \sum z_{i,v} &= 3 && (1.12)\\
0 \leq Q_{G_j} &\leq Q_{G_j}^{MAX_{ma}} && (1.13)\\
0 \leq Q_{L_i} &\leq Q_{L_i}^{MAX_{ma}}
\end{aligned}
$$

where,

$$
\begin{aligned}
v &= \text{iteration index}\\
i, j, k &= \text{Indices referring to nodes}\\
Q_{L_i} &= \text{the net injections of load } L_i\\
Q_{G_i} &= \text{the net injections of generator } G_i\\
DF_{bus}^{link} &= \text{Distribution factor for effect of bus on link}\\
N_L, N_G, N &= \text{Number of loads, generators, and total nodes respectively}\\
y_j &= \text{Binary variable at iteration v to accept or reject generator } j\\
z_i &= \text{Binary variable at iteration v to accept or reject load } i\\
\kappa_{ma} &= \text{allocation of flow on congested link from MA market}
\end{aligned}
$$

By solving this subproblem, we can solve for the optimal trilateral set of loads and generators to maximize total social welfare. The TSP can also solve for the exact profits made by these end-users in the electricity markets, after all iterations are complete. Assuming he collects this total amount or a fraction of the end-users profits, the TSP can solve for $\Pi_{\kappa_{ma},t}$ in order to use as the input into his stochastic optimization problem.

This thesis applies the general formulation given above to a smaller, more illustrative example. We note that a priori information include:

1. τ in hours
2. Anticipated Demand for $t = 1 \dots \tau$
3. N_L, N_G, and N
4. DF_{bus}^{link}
5. $K_{i \to j}^{max}$
6. Statistical Distribution of ω_t

To demonstrate our results with significance, yet maintain a computationally feasible problem, the TSP's problem is examined for one working week. Thus $\tau = 120$. For number 2 above, we take real anticipated demand data for New England, from Feb 4-8 of 2002, as shown in Fig. 1-4. When the probabilistic algorithm is introduced, this graph is split into a plausible multilateral portion and the leftover noise or spot market portion. These are graphed in Figures 1-5 and 1-6 respectively. In other words, Fig: 1-4 can be described as the sum of Figures 1-5 and 1-6.

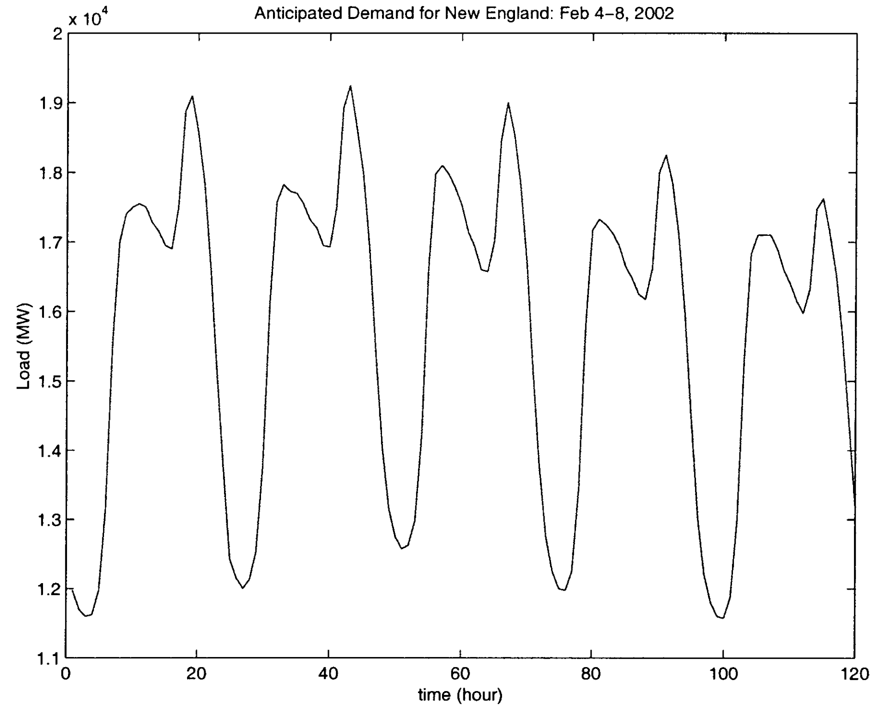

Figure 1-4: Anticipated Demand for one week period

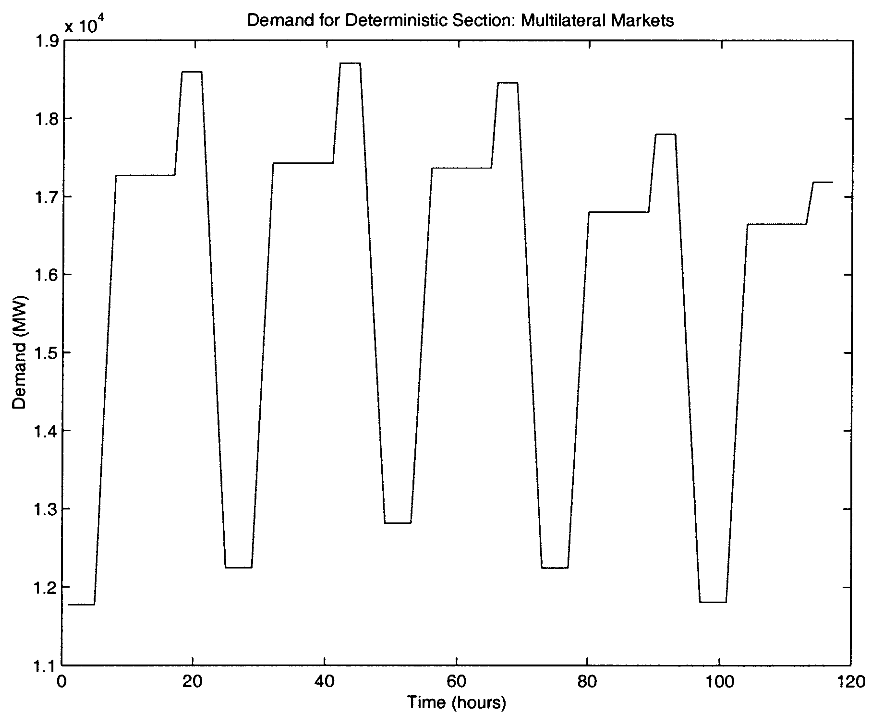

Figure 1-5: Anticipated Demand in Multilateral Market

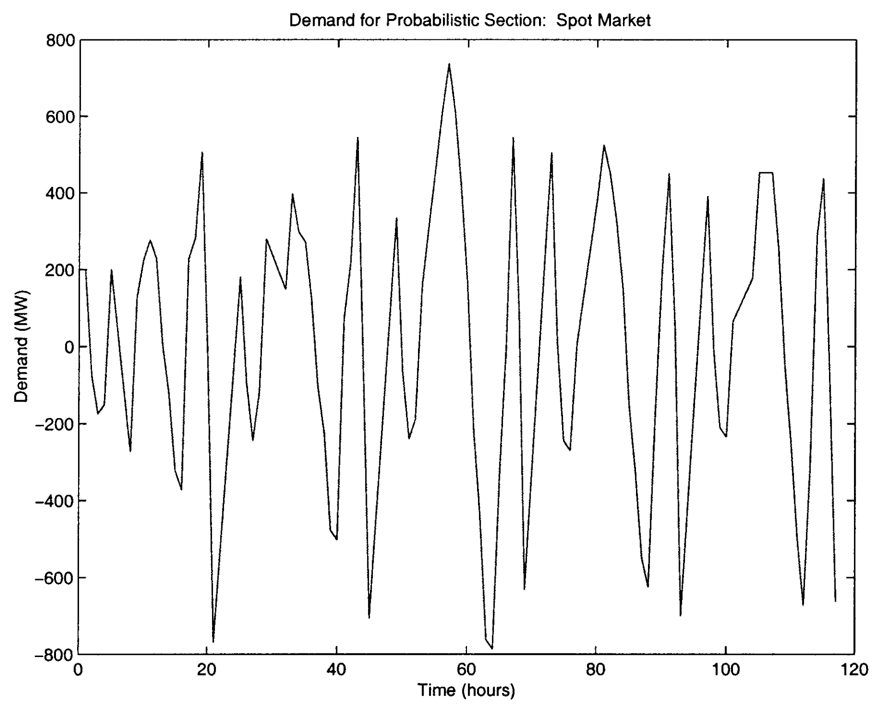

Figure 1-6: Anticipated Demand in Spot Market

To solve for the final constants in our example, we assume the following network structure shown in Fig. 1-7: Throughout the thesis, the lines are assumed lossless. As detailed in Chapter 5, the Distribution Factors change according to the topology being simulated. Also shown in chapter 5, $K_{i \to j}^{max}$ and ω_t differ depending on the hour of the week.

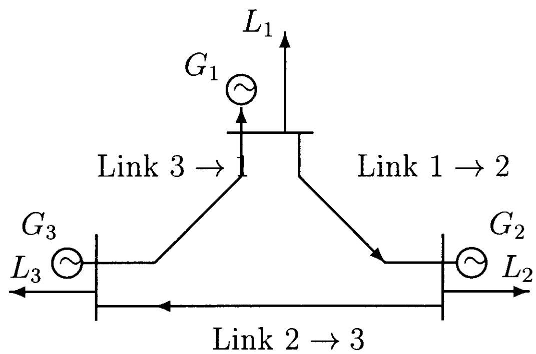

Figure 1-7: General 3-bus network

With this representation, we now re-visit our introductory questions regarding how profits and total social welfare concepts change, as a function of uncertainties, network topology and market structure. The above structure can be adapted to model several cases as summarized in Fig: 1-8.

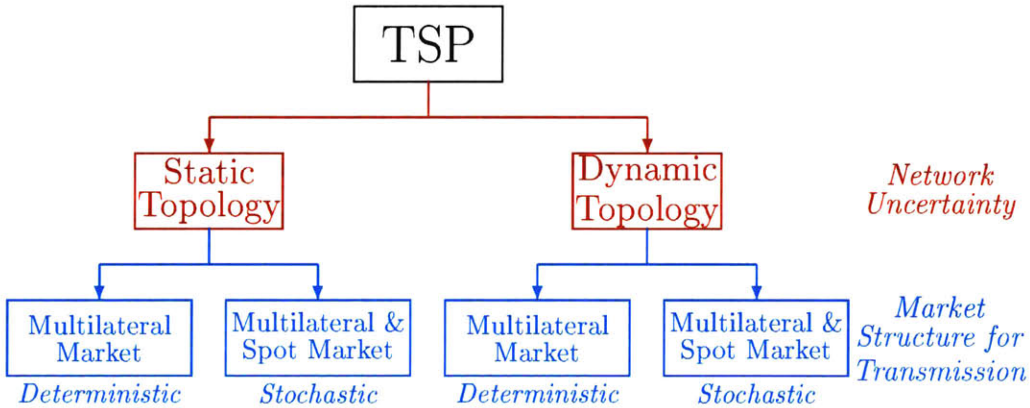

Figure 1-8: Uncertainties

We now briefly describe these varying parameters.

1.2.1 Topology Changes

As described in [13], network changes can occur at any time leading to adverse effects on reliability. We simulate the effects of a line outage by increasing the impedance of the affected link to a much higher level. Due to newly calculated distribution factors (see section 3.2), very little flow now travels along this link so it closely approximates an open circuit. The line chosen to be "down" is different than the limited capacity link. Results are shown to compare this dynamic topology to the static version.

1.2.2 Decision Models

To evaluate the dynamic programming algorithm, we compare results to a static optimization problem or greedy method at each time step. The static optimization implies the TSP is not forward-looking and is only interested in maximizing his gains for the current time period. Efficiency and profit values are contrasted for the two methods.

1.2.3 Market Structure Comparisons

Finally, scenarios are examined for three different market structures. Different possibilities are extensions of Allen et. al. [5] and can be summarized as:

- a bundled market using current techniques of economic dispatch
- an entirely bilateral/multilateral trading market, with TSP only providing transmission services
- a market with deterministic bilateral agreements and where spot is assumed to be probabilistic
- a market where end-users simply bid transmission demand curves, while TSP responds with complementary supply curves - an iterative auction procedure takes place till market clears for transmission.

Case 1 is introduced and briefly reviewed. This thesis examines cases 2 and 3 in depth. For each of these cases, the profits to the end-users and TSP are compared. Also, key results are found regarding the variations in total social welfare's change under the different market structures and decision environments. Case 4 is treated as a future research question.

1.3 Thesis Summary and Key Results

The motivation for this thesis is two-fold. As pointed out by Hogan [18], congestion contracts for flowgates need to be seriously considered for efficient use of a decentralized market structure. Moreover, there is a lack of tools and software for the TSP to efficiently and optimally process the numerous information requests in real-time. The contributions of this thesis lie in conceptual demonstration of the importance of such software, and the frameworks that can be used for longer-term resource allocation.

Chapter 2 provides the background information necessary regarding terms, notation and algorithm descriptions. Specifically economic efficiency arguments are presented to provide perspective from the end users, and the optimization algorithms are presented for the basic problem of interest. Chapter 3 then provides a complete summary of multilateral agreements, including an illustrative example and the theory behind their efficiency. The motivations behind using such multilateral agreements and relevance to this problem are also included. Chapter 4 gives an analogous description of the spot market as well as various formulations for estimating the uncertainties therein. Comments are made regarding each estimation method and the advantages and disadvantages behind such concepts.

Chapter 5 re-introduces the problem statement in question, and describes the proposed simulation used to implement various multilateral agreement by a TSP. Computations are done for a single snapshot in time for the anticipated load level and possible line constraints along the congested link. The fundamental concepts such as bus injections, profits, and total social welfare are shown as a function of the number of iterations used to implement the requests for transmission; remarks are made in regards to the effects of varying the load level or line capacity on these values. Chapter 6 extends the multilateral market of Chapter 5 for the entire week. Results are computed over this time, as if a spot market did not exist, and the corresponding effects on total social welfare and profits when applied to the anticipated demand curve from Fig. 1-4. Finally, Chapter 7, puts together the concepts of chapters 3 and 4 to achieve a market with deterministic forward requests for transmission and stochastically changing spot requests for transmission. Optimizations using the algorithms from Chapter 2 are shown here and compared to the benchmark greedy approach. Results are also compared between chapters 6 and 7, so as to better understand the impact of the stochastic spot market on the social welfare, TSP profits, and end user profits. Finally, some remarks are made regarding future work, and extensions of the simulations produced. The thesis concludes with the fundamental results and an appendix with source code for implementing the multilateral agreements and dynamic programming formulations.

Chapter 2

Background and Preliminaries

This chapter provides definitions and concepts relevant to the transmission service provider problem. We begin by analyzing how electricity markets are modeled and proceed into more specific formulations regarding the bilateral and spot markets. Basic economic efficiency arguments are also given to understand the interplay between end-users. Finally, the algorithms applied for the example simulation are provided and some remarks are made regarding their use.

2.1 Electricity and Transmission Markets

Our approach follows the earlier formulations by Gözüm [1] and does not consider the role of intermediary players and secondary or ancillary market effects. The key players in the network are the loads and generators. The loads are traditionally power consumers, while generators are power suppliers. However, this is by no means a constraint as various multilateral agreements may exist where a load ends up selling electricity while a generator buys electricity. There exist two mechanisms by which end users can coordinate to either supply or demand electricity. These are [4]:

1. *Long-term bilateral contracts for anticipated demand that exist in a forward market setup* These are point-to-point contracts from specific users for the transfer of a set amount of power. These contracts are comprised of three key components: quantity, price, and duration. Unlike [1], this thesis does not consider

the case where such contracts are rescinded after being accepted. Thus, there is no need for penalty functions; instead, the risk of not being implemented is internalized in the bid curves for delivery themselves. The other assumption is that requests for transmission are made simultaneously by all players, to preclude any particular player from gaining an advantage through additional information.

2. *Short-term spot market for on-line adjustments to demand uncertainties* This market ensures that hourly load variations from the long-term anticipated demand are adequately supplied at the market clearing price. This price is determined from the hourly characterization of end users' supply and demand curves. The resulting spot price of electricity depends upon the characteristics of the network, the cost and utility functions of the generators and loads respectively, as well as on the stochastic demand level. Selling and purchasing electricity at the spot price is usually more risky due to the increased number of uncertainties, in particular the actual load level and the network capacity status.

Each end user must carefully choose the amount to sell or buy in each market setup to maximize his own profit and/or benefits. It is assumed that information regarding the cost of generation or utility of a load is private information that is not shared among end users at any time. The TSP may have access to such information, but may also be provided such information with the caveat that specific parameters may change at short notice. In this thesis, we differentiate between the TSP having knowledge of supply and demand functions for electricity, on one side, from the demand functions by the end users for transmission delivery. We also show the inter-relation between the two characterizations.

Thus, the market structure has enough complexity that it is not a trivial problem to balance tradeoffs between:

1. the system-wide optimum efficiency

2. the profits for individual end users

3. the profits of the TSP

This is especially true when line constraints exist on the network constraining the amount of flow that may be required at a certain node. The next section makes some simplifying assumptions to better understand how economic arguments can be used to relate total social welfare and participants' profits.

2.1.1 Economic Efficiency Arguments

We adopt a similar line of arguments as Allen et. al. [5] in analyzing the electricity market economics. As a first step, we consider a general electric power network that has infinite line capacities for all links. Thus, transmission congestion is not an issue here. For this case, we analyze the network as seen by individual competitive market participants. We first define the following variables:

$$\begin{aligned}
P &= \text{the current market clearing or spot price of electricity} \\
Q_{G_i} &= \text{the quantity of real power produced by the generators} \\
Q_{L_i} &= \text{the quantity of real power withdrawn by the load at bus} \\
MC_i(Q_{Gi}) &= \text{the marginal cost of supplying } Q_{G_i} \\
MU_i(Q_{Li}) &= \text{the marginal utility of demanding } Q_{L_i} \\
\pi_{Gi}(Q_{Gi}) &= \text{the profit of supplying } Q_{G_i} \\
\pi_{Li}(Q_{Li}) &= \text{the profit of utilizing } Q_{L_i}
\end{aligned}$$

If we assume quadratic functions for cost and utility functions, for generators and loads respectively, then:

$$C_i(Q_{G_i}) = a_{G_i}Q_{G_i}^2 + b_{G_i}Q_{G_i} + c_{G_i} \tag{2.1}$$

$$U_i(Q_{L_i}) = -a_{L_i}Q_{L_i}^2 + b_{L_i}Q_{L_i} + c_{L_i} \tag{2.2}$$

$$MC_i(Q_{Gi}) = \frac{dC_i(Q_{G_i})}{dQ_{G_i}} = 2a_{G_i}Q_{G_i} + b_{G_i} \tag{2.3}$$

$$\begin{aligned}
\pi_{G_i}(Q_{G_i}) &= PQ_{G_i} - C_i(Q_{G_i}) && (2.4)\\
MU_i(Q_{L_i}) &= \frac{dU_i(Q_{L_i})}{dQ_{L_i}} = -2a_{L_i}Q_{L_i} + b_{L_i} && (2.5)\\
\pi_{L_i}(Q_{L_i}) &= U_i(Q_{L_i}) - PQ_{L_i} && (2.6)
\end{aligned}$$

Also, we can establish that each generator will continue producing until the marginal cost is equal to the current market price [37]:

The maximum profit for generators and loads is found by setting the derivative equal to zero:

$$\frac{d\pi_{Gi}(Q_{Gi})}{dQ_{Gi}} = P - MC_i(Q_{Gi}) = 0 \qquad (2.7)$$

$$\frac{d\pi_{Li}(Q_{Li})}{dQ_{Li}} = MU_i(Q_{Li}) - P = 0 \qquad (2.8)$$

Thus, we can now write supply function of generator i as:

$$S_i(P) = Q_{Gi} = \frac{P - b_{Gi}}{2a_{Gi}} \qquad (2.9)$$

Analogously, the demand function for load i is

$$D_i(P) = Q_{Li} = \frac{b_{Li} - P}{2a_{Li}} \qquad (2.10)$$

Taking the aggregate curve as the sum of the individual curves for generators and loads implies:

$$S(P) = \alpha_S P - \beta_S \qquad (2.11)$$

$$D(P) = \beta_D - \alpha_D P \qquad (2.12)$$

where

$$\alpha_S = \sum_{i=1}^{N_G} \frac{1}{2a_{Gi}}$$

$$\beta_S = \sum_{i=1}^{N_G} \frac{b_{Gi}}{2a_{Gi}}$$
$$\alpha_D = \sum_{i=1}^{N_L} \frac{1}{2a_{Li}}$$
$$\beta_D = \sum_{i=1}^{N_L} \frac{b_{Li}}{2a_{Li}}$$

Furthermore it is clear that the economic equilibrium price is the intersection of the supply and demand curves.

$$P_\lambda = \frac{\beta_D + \beta_S}{\alpha_D + \alpha_S} \tag{2.13}$$

If we assume a marketplace that is relatively competitive and stable, it follows that the price will converge to the equilibrium price. Finally, we note that total social welfare is defined as the total utility minus the total cost of all participants [22]:

$$TSW = \sum_{i=1}^{N_L} U_i(Q_{Li}) - \sum_{i=1}^{N_G} C_i(Q_{Gi}) \tag{2.14}$$

2.2 Algorithms for Transmission Constrained Markets

The theoretical algorithms used in the simulation are described here. They include the process of Lagrange multipliers, optimal power flow, and dynamic programming.

2.2.1 Optimal Power Flow

The most commonly used tool to solve the transmission problem used in the market today is the optimal power flow (OPF) or economic dispatch solution [23]. The main objective of this tool is to maximize the total social welfare with respect to power quantities generated or withdrawn subject to the network constraints. The framework assumes a central dispatch setup and the constraints are both operational and security related. We follow the arguments of Raikar and Ilic [9] to more thoroughly explain the optimization of the dispatcher. The main problem can be posed as follows:

$$\min_{Q_{G_i}} \left(\sum_{i=1}^{N_G} C_{G_i}(Q_{G_i}) - \sum_{i=1}^{N_L} U_{L_i}(Q_{L_i}) \right) \tag{2.15}$$

subject to

$$\sum_{i=1}^{N_G} Q_{G_i} = \sum_{i=1}^{N_L} Q_{L_i} \tag{2.16}$$

$$|Q_{i\to j}| \le Q_{i\to j}^{max} \tag{2.17}$$

$$0 \le Q_{G_i} \le Q_{G_i}^{max} \tag{2.18}$$

$$0 \le Q_{L_i} \le Q_{L_i}^{max} \tag{2.19}$$

where,

$$\begin{aligned}
C_{G_i}(Q_{G_i}) &= \text{Cost function of Generator } i \\
U_{L_i}(Q_{L_i}) &= \text{Utility function of Load } i \\
Q_{G_i} &= \text{Power produced by Generator } i \\
Q_{L_i} &= \text{Power consumed by Node } i \\
Q_i &= \text{Power flow on link } i \to j \\
N &= \text{Number of nodes in network} \\
N_G &= \text{Number of Generators} \\
N_L &= \text{Number of Loads} \\
Q_{G_i}^{max} &= \text{Maximum generation capacity of Generator } i \\
Q_{G_i}^{max} &= \text{Maximum power consumption by Load } i \\
Q_i^{max} &= \text{Maximum power flow limit of transmission link from } i \to j
\end{aligned}$$

The above constrained OPF is for the simpler D.C. power flow model[1]. In words, the objective function above represents the maximization of total social welfare. Constraint (2.16) states that power supplied must equal power demand. Constraint (2.17) is the transmission congestion constraint, while constraints (2.18) and (2.19) are capacity constraints for the end users.

Based on the Lagrangian technique for nonlinear optimization [3], we can gain further insight into the above problem. Let

$$\begin{aligned}
\lambda &= \text{multiplier for constraint (2.16)} && (2.20)\\
\mu_{i\to j} &= \text{multiplier for constraint (2.17)} && (2.21)\\
\eta_{G_i} &= \text{multiplier for constraint (2.18)} && (2.22)\\
\nu_{L_i} &= \text{multiplier for constraint (2.19)} && (2.23)
\end{aligned}$$

we can define the Lagrangian as

$$\begin{aligned}
L &= \sum_{i=1}^{N_G} C_{G_i}(Q_{G_i}) - \sum_{i=1}^{N_L} U_{L_i}(Q_{L_i})\\
&\quad + \lambda\Big(\sum_{i=1}^{N_G} Q_{G_i} - \sum_{i=1}^{N_L} Q_{L_i}\Big)\\
&\quad + \mu_{i\to j}\Big(Q_{i\to j} - Q_{i\to j}^{max}\Big)\\
&\quad + \eta_{G_i}\Big(Q_{G_i} - Q_{G_i}^{max}\Big)\\
&\quad + \nu_{L_i}\Big(Q_{L_i} - Q_{L_i}^{max}\Big)
\end{aligned}$$

Solving the partial derivatives,

[1] A comparison of using the A.C. model and effects upon the network can be found in [?]

$$\eta_{G_i} + \frac{\delta\ C_{G_i}(Q_{G_i})}{\delta\ Q_{G_i}} = \lambda + \mu_{i\to j}\Big(Q_{i\to j} - Q_{i\to j}^{max}\Big) \tag{2.24}$$

If the spot price is ρ_i, then

$$\rho_i = \lambda + \mu_{i\to j}\Big(Q_{i\to j} - Q_{i\to j}^{max}\Big) \tag{2.25}$$

Furthermore, Wu and Varaiya [25] show that the Merchandising Surplus (MS) collected by the TSP is

$$MS = \sum_i \rho_i Q_i = \frac{1}{2}\sum_i \sum_j (\rho_j - \rho_i) Q_{i,j} \tag{2.26}$$

For a transmission service provider, the line constraint equation (2.17) and capacity limits (2.18) and (2.19) are the most important points of interest. If there is no congestion, then by complementary slackness condition, we infer that

$$\mu_{i\to j} = 0 \tag{2.27}$$

$$\rho_i = \frac{\delta\ C_{G_i}(Q_{G_i})}{\delta\ Q_{G_i}} + \eta_{G_i} = \lambda \tag{2.28}$$

However, if the line is operating at capacity, then

$$\mu_{i\to j} \neq 0 \tag{2.29}$$

$$|Q_{i\to j}| = Q_{i\to j}^{max} \tag{2.30}$$

Since we know $\mu_{i\to j}$ gives the shadow price of (2.17), the transmission congestion rent is given by

$$MS = \sum_i \sum_j \left(\mu_{i\to j} Q_{i\to j}^{max}\right) \tag{2.31}$$

Thus, we can use this formulation to realize transmission rent and the effective spot price of transmission at a given instant of time. This will be further detailed in (Section 4.1).

2.2.2 Dynamic Programming

As introduced earlier, the TSP's objective function in equation (1.1) is over a multi-stage period. Since any decision at one time affects the decisions at future time periods, a dynamic programming approach is well suited to the problem [1]. DP's main advantage of over other algorithms is its ability to capture additive costs over multiple stages in the recursive objective function. The basic theory is as follows and is adapted from Bertsekas [6]:

The basic model has a dynamic system that has a constantly evolving cost function that is additive over time. More specifically, we assume a discrete-time dynamic system with variables:

$$\begin{aligned} k &= \text{Discrete time index} \\ x_k &= \text{State of system which includes past information at time } k \\ u_k &= \text{Decision variable to be selected at time } k \\ w_k &= \text{Random disturbance or Noise at time } k \end{aligned}$$

The evolution function is

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, \ldots, N-1 \tag{2.32}$$

We now consider a class of policies that consist of the sequence of functions

$$\pi = \{\mu_0, \mu_1, \ldots, \mu_{N-1}\} \tag{2.33}$$

that map states x_k into controls $\mu_k = \mu_k(x_k)$. Given such a policy and an initial state x_0, the states x_k and noise w_k are defined by

$$x_{k+1} = f_k(x_k, u_k(x_k), w_k), \quad k = 0, 1, \ldots, N-1 \tag{2.34}$$

It thus follows that for given profit functions g_k, $k = 0, 1, \ldots, N$, the expected profit of π starting at x_0 is

$$J_\pi(x_0) = E\Big[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k(x_k), w_k)\Big] \tag{2.35}$$

Then, it is usually possible to find a policy π^* that is optimal for all initial states denoted by

$$J^*(x_0) = \max_\pi J_\pi(x_0) \tag{2.36}$$

If these controls are made at every hour, and we assume that a given state x_t occurs at time t with positive probability, then this problem setup [35] intrinsically contains the principle of optimality. If we consider the subproblem of being at time t in state x_i and wish to maximize our future profits or "profits-to-go" from time $t \ldots N$, the we are trying to maximize:

$$E\Big[g_N(x_N) + \sum_{k=t}^{N-1} g_k(x_k, u_k(x_k), w_k)\Big] \tag{2.37}$$

Furthermore, the truncated policy $\{\mu_t^*, \mu_{t+1}^*, \ldots, \mu_{N-1}^*\}$

$$E\Big[g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k(x_k), w_k)\Big] \tag{2.38}$$

Thus, if we recursively use the above principle starting backwards in time, we arrive at the formal DP algorithm as follows. For every initial state x_0, the optimal profit $J^*(x_0)$ is equal to $J_o(x_0)$, given by the last step in the algorithm. The algorithm uses backward induction and calculates the profits as follows from period $N-1$ to period 0:

$$J_N(x_k) \;\leq\; g_N(x_N),$$

$$J_k(x_k) = \max_{u_k} E\Big[g_k(x_k, u_k, w_k) + J_{k+1}\Big(f_k(x_k, u_k, w_k)\Big)\Big], \quad (2.39)$$
$$k = 0, 1, \ldots, N-1.$$

The expected value is taken with respect to the probability distribution of w_k, which is dependent on x_k and u_k. While the form above looks extremely closed and compact, it should be noted that the "profit-to-go" function J is extremely hard to find in such a form. In most cases, including this thesis, enumeration and brute-force simulation techniques are required to solve equation (2.39). This equation can be understood best by viewing each decision as comprising a single branch in a decision tree. The stage before the last one or before the leaves of the tree are reached have a cost denoted by $g_N(x_N)$. Thus, equation (2.39) simply ensures that at any time, k, the decision is made to maximize the current profit *as well as expected future profits.* Thus, the equation represents building all the possible paths from root to leaf in the decision tree, and then performing backwards induction to reveal the most optimal path.

Unfortunately, the massive computational effort to itemize all possible actions when building the dynamic programming tree turns the problem infeasible very quickly. As discussed in [1] it is this fact that motivates various approximations and heuristic techniques to prune the tree to a manageable size, and reduce the number of choices considered at every node. While this leads us into various suboptimal techniques, it can be shown that the bounds on such techniques are justifiable for the problem this thesis attempts to address.

Finally, this thesis uses the above formulations in a discrete-time manner with the control being applied at each hour over a week period. As given in Figures 1-1 and 1-2, the algorithm described here is well suited for our problem of interest. Indeed, the decision tree in 1-2 is a clear candidate for such optimization, and as t grows large, it follows such suboptimal techniques will be necessary to prune the tree and achieve a solution. The long-time horizon on which to optimize over is the week period, while each time increment (and hence decision values) is every hour.

Suboptimal Control - Limited Lookahead Strategies

In numerous cases of dynamic programming, the number of choices grows exponentially as the number of stages increases, producing a computationally infeasible problem. Referred to this as the "curse of dimensionality" [32], it can make even a simple problem intractable. Various workarounds to this problem have recently been introduced by Bertsekas, in [8] and [7]. The applications of these specific techniques to the problem at hand are also shown in Gözüm [1].

The simplest possibility for reducing the dimensionality is simply to truncate the time horizon, and use the profit-to-go function of only a small number of stages. Thus, a two-step lookahead applies the following heuristic for the profit-to-go:

$$\tilde{J}_{k+1}(x_{k+1}) = \max_{u_k} E\Big[g_{k+1}(x_{k+1}, u_{k+1}, w_{k+1}) + \tilde{J}_{k+2}\big(f_{k+1}(x_{k+1}, u_{k+1}, w_{k+1})\big)\Big] \quad (2.40)$$

Here, $\tilde{J}$ represents an approximation to J given in equation (2.39). Clearly, with the above modification, a limited-lookahead is only satisfactory if the approximate profit-to-go function is chosen wisely. As shown in [6], it is important that the profit-to-go differentials are approximated well: For an m-step lookahead,

$$\tilde{J}_{k+m}(x) - \tilde{J}_{k+m}(x') \approx J_{k+m}(x) - J_{k+m}(x') \quad (2.41)$$

In this equation x and x' represent two representative states in the problem. Numerous choices exist for the profit-to-go function based on this characteristic. Some of the most promising include:

- Problem Approximation: In this case, we try to find the optimal profit-to-go with some actual cost from a simpler problem. A good candidate is to use the profit-to-go of the the smaller-horizon subproblem currently being examined.

- Heuristic Cost-to-Go: This is an example of approximating the profit-to-go by a set of parameters, that vary according to some heuristic strategy. This case also makes a good example when learning strategies can be applied from one subproblem to the next.

- Rollout Approach: In this case, a simulation based on a suboptimal policy is created and used to approximate the profit-to-go.

In summary, dynamic programming is a powerful approach when solving multi-stage problem. It should be remembered that neuro-dynamic techniques must be used for computationally vast decision spaces [7]. Furthermore, it is rare to find closed form solutions or neuro-dynamic techniques that work for every problem; instead, each problem must be individually examined, and heuristics should be derived for tendencies intrinsic to the problem.

Chapter 3

Bilateral and Multilateral Agreements

Bilateral contracts for electric power comprise the demand to the TSP's forward market; the end users are constantly striking deals between each other ahead of time for power supply or demand and request transmission capacity to deliver this power. A bilateral contract is defined as the right to inject a certain quantity of power at node i and withdraw it at node j at a specified electricity price P_{ij}. We notice that these transactions are point-to-point in nature, and along with the price, have a set duration of time.

These contracts for transmission will be seen by a TSP only when the end user's marginal cost is smaller than the nodal price of the selling bus. Mathematically, we follow [27] to deduce that a bilateral contract from i to j is viable only if $p_i \leq p_j$.[1] However, it should be noted that a severe problem with this approach is that in electrical networks, the contract flows usually do not equal the physical flows. As Hogan first pointed out [24], Kirchoff's laws on the network dictate the flow in accordance with the impedance and susceptance values of each link. Therefore, some bilateral contracts may cause congestion on parallel-flow links and therefore not be allowed for implementation.

[1]This is obvious in a single two bus network and one should be careful in generalizing to larger networks. See [25] for further details.

Before delving deeper into multilateral trading concepts, it is instructive to examine how net injections at the buses are mapped into flows along lines. These are calculated from the power transfer Distribution Factors (DF) as follows [17].

Let us again consider the following simple electric network with three interconnected nodes as shown in Fig. 3-1.

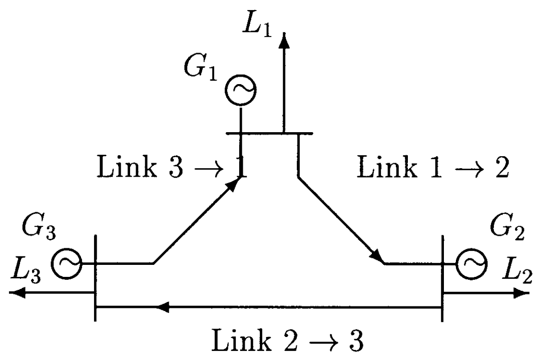

Figure 3-1: General 3-bus Network

Thus, depending on the impedance levels of the interconnecting links (and resistive levels if losses are considered), the flows on the line vary with the net injections at the buses, as described by the detailed formulation in [17] . For simplicity, we assume equal impedance along the three links and that the lines are lossless. In Fig: 3-1, bus 3 is the slack bus and the corresponding distribution factors are applied to the net injections which equal the difference of total generation and total load.

It is also important to realize some subtleties of the above example that are not immediately straightforward. When using distribution factors as above, the injections at each node are the net injections are total generation - total load. It is also important to reserve one bus as the slack bus. In the network, it is at this bus where the regulation is completed to maintain stability of the voltage level [2], and therefore, the net injections at this bus have no effect on line flows. Without loss of generality and the fact that the sum of net injections must sum to 0, we can pick any bus as the slack bus when solving for line flows.

It thus follows that an efficient equilibrium may not be reached if only viable Bilateral Agreements (BA) are considered. This is best understood by an example. The numbers and arguments shown here follow directly with Wu and Varaiya's example in [26].

Assume initial trades in a 3-bus network are as follows (these can be thought of as the unconstrained optimal solution of generation quantities described in chapter 2). The values are shown in Fig. 3-2.

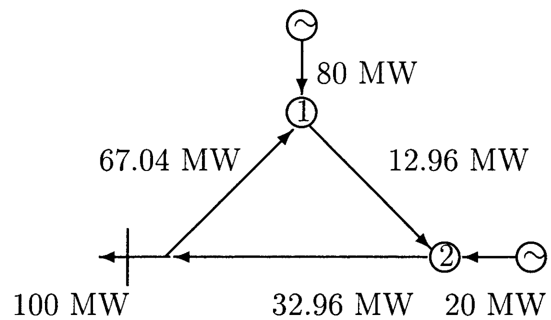

Figure 3-2: Optimally Efficient Trade

However, in the more realistic setting, links have capacity constraints that a transmission service provider must overcome. As shown in [22], a curtailment strategy must be adopted to bring the flow down on congested links. If we assume that link $1 \rightarrow 2$ has a maximum capacity of 5 MW, then the following curtailment produces flows shown in Figure 3-3,

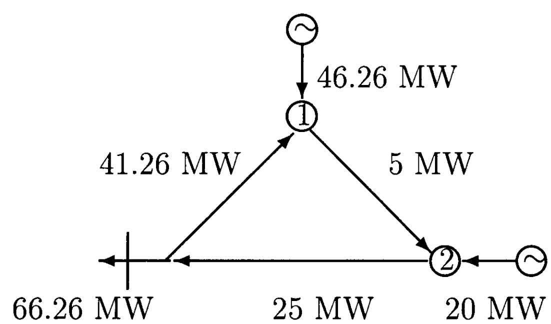

Figure 3-3: Flows after curtailment

However, such a reassignment of net injections at all three nodes has caused for $MC_1 < MC_3$. However G_1 and L_3 cannot strike a bilateral agreement to take advantage of this situation because any flow from 1 to 3 will increase the flow on the congested link beyond its capacity. Thus, these agreements will not be approved by a TSP.

In order to circumvent this problem of sub-optimality with bilateral agreements and still obtain an efficient solution, [25] suggests 2 alternative methods. The first proposition involves having and independent system operator or TSP impose a "transmission charge" $= p_i - p_j$ to transfer power from node j to i. This alternative has the disadvantage of requiring the TSP to solve the economic dispatch or OPF problem in order to be knowledgeable of the nodal prices. As mentioned before, this may pose problems if all the information regarding cost and utility functions are not known beforehand. Moreover, the forward market necessitates knowledge of transmission charges *a priori* for an end user to maximize his gains, which may not be available or accurate ahead of time. If a clear adjustment bid or mechanism has not been revealed for how transmission charges will converge to the optimal values, the end-user may not have enough information to make transactions in the bilateral markets.

The second method is to introduce trilateral contracts or multilateral agreements such that counterflow is produced to prevent congestion along the constrained link. We re-examine the above example to better understand how multilateral (trilateral in this case) agreements can provide profitable opportunities that cannot be attained by bilateral contracts alone.

We take Fig. 3-3, and include the trilateral agreement of having node inject 15 more MW (relieving congestion on the line), allowing node 1 to inject 18.74 more MW, which maintains the flow on link $2 \rightarrow 3$ at the maximum 5 MW. This is shown in figure 3-4.

Thus, by involving a third party, a more efficient solution is achieved in accordance with the line constraints as well as having all parties receive nonnegative profits. Further analysis of multilateral trades reveals that optimal total social welfare can be achieved through numerous of these trades taking place, no matter what the curtailment strategy is. This is demonstrated theoretically by Wu and Varaiya as Theorem 1 in [26]. Our simulation also confirms this result and further insight is gained when analyzing the actual data.

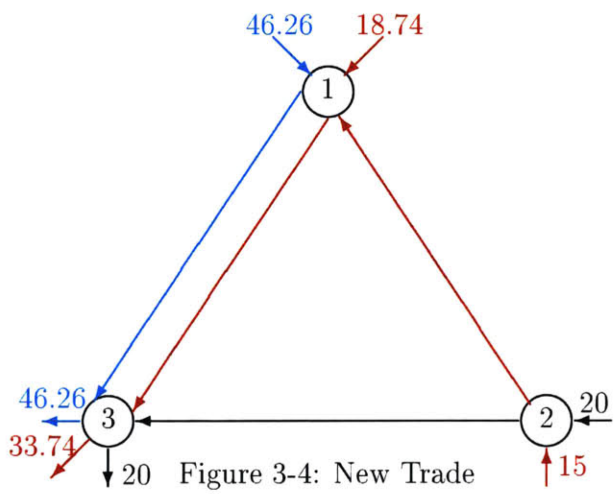

Figure 3-4: New Trade

3.1 Theory and Implementation of Multilateral Trades

When looking at the network, it is apparent that certain lines have capacity limits due to maximum physical levels. Properties intrinsic to the line constrain the directional flow amount. Under such conditions, we now investigate multilateral trades, and determine their effect on the system transmission capacity and the social welfare. From Theorem 1 in [26], we can see that an operating paradigm that includes such multilateral trades indeed achieves efficiency. This proof is repeated here to show that the multilateral trading process converges to an optimal solution under the assumption that all participants are rational. More precisely, we denote trades $\Delta\ \mathbf{q}$ with profit less than ϵ as ϵ-unworthy

$$[c(\mathbf{q}) - c(\mathbf{q} + \Delta\ \mathbf{q})] < \epsilon \tag{3.1}$$

and trades with

$$[c(\mathbf{q}) - c(\mathbf{q} + \Delta\ \mathbf{q})] \geq \epsilon \tag{3.2}$$

as ϵ-worthy. We also assume that for $\epsilon > 0$

1. ϵ-unworthy trades are not accepted
2. ϵ-worthy trades will eventually be implemented
3. participants must carry out trade if accepted

Then, we can show that multilateral trading converges to optimal set of quantities $\mathbf{q}^*$. The first claim is that the process must terminate at point $\mathbf{q}^N$ which is less than or equal to ϵ away from the optimal solution:

$$[c(\mathbf{q}^N) - c(\mathbf{q}^*)] \leq \epsilon \tag{3.3}$$

Then, since assumption 1 is valid $\forall\ \epsilon$, let $\epsilon \to 0$, so $\mathbf{q}^N \to \mathbf{q}^*$.

The sketch of the proof is as follows. We are guaranteed that the process will terminate. At some point then,

$$\left[c(\mathbf{q}^N) - c(\mathbf{q}^N + \Delta\ \mathbf{q}^N)\right] \leq \epsilon, \tag{3.4}$$

$\forall\ \Delta\mathbf{q}^N$ satisfying $\langle \mathbf{n}_k, \Delta\mathbf{q}^N \rangle \leq 0$.

From the fact that the set

$$X := [\mathbf{q} : (\mathbf{q} = \mathbf{q}^N + \Delta\mathbf{q}^N), \langle \mathbf{n}_k, \Delta\mathbf{q}^N \rangle \leq 0]$$

is convex, we are guaranteed that the convex cost function c achieves some minimum. If we asssume that this occurs at some m then,

$$c(\mathbf{q}^m) \geq c(\mathbf{q}^N) - \epsilon \tag{3.6}$$

Finally, the feasible set has more constraints than X, so it follows $S \subseteq X$. Clearly, $\mathbf{q}^*$ is in the feasible set, so it too satisfies $c(^m) \leq c(\mathbf{q}^*)$. It follows that $c(\mathbf{q}^*) \geq [c(\mathbf{q}^N)\ -\ \epsilon]$, so $c(\mathbf{q}^N)$ is ϵ-away from the optimum. The first claim is then proved. Thus, since $\mathbf{q}^N \to \mathbf{q}^*$ as $\epsilon \to 0$, the proof is complete [26].

Keeping this result in mind, this thesis analyzes sequences of coordinated private multilateral trades as long as:

1. All participants in the trade receive non-negative profit; and

2. It leads to efficient operations while optimizing the objective at each iteration.

Note that the trades are in a sequential format of iterations, and are not implemented simultaneously. Instead some criteria is used either by the TSP or a social welfare maximizer to choose the multilateral agreement increasing the objective function by the maximum amount. Once this trade is implemented, the quantities and values for all end users are updated. All participants then talk amongst themselves to supply the next set of multilateral agreements for the next iteration, and the process continues. A fixed number of iterations is predetermined for each hour by the TSP.

The model is also beneficial because it achieves the advantages of a centralized pool system, while allowing the transmission and energy markets to be unbundled.

In effect, the TSP has no influence on the electric power trading decisions behind the proposed multilateral agreements. In terms of feasibility it has already been demonstrated in [26] and [5] that the existing communication and control infrastructure is sufficient to support such trades.

3.2 Summary

The issues presented above are not new, yet much debate still revolves around how end-users and independent market players can hedge their risks of congestion. Chao et. al. in [16] indicate that transmission assumes new strategic importance in supporting market trading between individual buyers and sellers. Furthermore, individual end-users must now take into consideration the effects of power flows that diverge from the contract path.

In addition, there is a clear need for well-defined transmission rights that enable external effects associated with transactions to be incorporated into decision making process. It is imperative that current technology is used to its maximum capabilities, for a transmission service provider to compute an optimally efficient set of transmission rights. Thus, this thesis uses the theory presented above along with software algorithms, for a TSP to ensure flow-based rights are computed correctly.

Moreover, the solutions presented herein may finally allow the concept of distribution factor-based insurance to become a reality. Since both network and market topologies are examined, a TSP can be well-prepared to maximize his gains over a set time period.

Finally, it is apparent that trilateral agreements are only a one-step extension. It can be deduced that coordination among an even higher number of end-users may lead to achieving the system optimum even faster. Tradeoffs must be made between competition and system efficiency when such concepts are extended to a more general case.

Chapter 4

Spot Market

Unfortunately, the anticipated demand values are not perfect and hence the deterministic scheme given in chapter 3 does not account for the actual values of demand. Hence, the need for a spot market is apparent, and transmission costs and contracts must be examined for such a market.

Furthermore, for the TSP problem in question, the prediction of the spot market demand for transmission is vital for the TSP to make optimal decision at each time interval. The more accurately the TSP can predict such spot demand, the better it can allocate a sufficient amount (but not excessively large quantity) to be available on congested links in order to strike its own profit on the spot. However, if his predictions are inaccurate, then the TSP risks rejecting more profitable bilateral agreements, or risks not being able to buy larger quantities on the spot if line allocation existed.

4.1 Characterization of Load Demand Uncertainty

As shown in the problem statement, one can think of the spot values as additive noise to the set BA levels. Furthermore, one can begin to think about how these current noise parameters are related to the similar time periods in the past. Indeed, weather conditions, forecasts, and historic data are key inputs when computing the anticipated load demand. Thus, to correctly account for such errors in these load "guesses", it follows that probabilistic and stochastic models can be applied to historic data, to make necessary adjustments for the current time period.

4.2 Stochastic Models

Two models are presented and formulated below to approximate the value of the spot quantities to a TSP. In each case, the TSP is attempting to derive expected profits on the spot market for all possible allocations of the congested link. Hence, he is attempting to guess with a very high probability the exact profit that could be made on the spot market for every possible allocation. In this way, he can best split the capacity of the congested link between the forward market and spot market. From this point forward and for notational convenience, we will refer to the allocation of the congested link for the spot market as κ_s as given in 1.2.

4.2.1 Basic version of OPF

A simple model to account for spot market profit is as follows. The TSP adopts the methodology that $Revenue = Price * Quantity$, and attempts to solve for these values individually. First, the spot price of transmission along the congested link is clearly related to the optimal power flow algorithm described in section 2.2.1. [24]. Indeed, the price of transmission is seen to be simply the shadow price of the line constraint equation in the optimal power flow model. It describes how 1 MW along the congested link beyond the capacity affects the overall objective function. Thus, if the TSP has access to the OPF data, he can solve the economic dispatch solution and derive the spot price of transmission.

To solve for the quantity of spot flow that will actually occur along the congested link, the TSP can use the stochastic distributions of the previous year's data to arrive at some preliminary estimates. We note from Fig.1-4, there exists three main time periods during each day for electricity consumption: early morning (previous day's night), daytime, and evening. We see each day has a similar a pattern and assume that:

	Mean	Variance	Standard Deviation
Night	11844.77	234195.02	483.94
Daytime	16492.07	640857.60	800.54
Evening	17488.86	735329.53	857.51

Table 4.1: Statistics for Spot Demand

- Night was between the hours of 1 am - 5 am
- Daytime was between the hours of 8 am - 5 pm
- Evening was between the hours of 6 pm - 9 pm

From Fig. 1-4, hours of 6 am, 7 am, 10 pm, 11 pm, and 12 am, can be seen to be more as transition times and are not included in the statistical modeling. Moreover, we see that the distribution is relatively constant between days and prior years. Thus, we can derive approximate values for the mean and variance of a probabilistic distribution during these three times of interest. Doing such analysis on New England data for the years 2000 and 2001, we see the results presented in Table: 4.1. These values were taken from data for anticipated load demand for the first working week in February, to correspond to the data we were estimating from 2002.

Now for each possible value of κ_s, and given the period in question, the TSP can model an approximate value of power flows along the congested link. The TSP can then compute the disturbance in exact quantity as the value of a random variable created from the above process.

Using these expected quantities and shadow prices of transmission, the TSP can thus arrive at the expected profit values for each feasible κ_s value, which is exactly what is desired.

4.2.2 Bounding the Spot Market Loss

Building upon the previous section, we use next the same given statistical information to bound the spot market loss rather than compute the spot market profit directly. For each possible value of κ_s, it is clear that only one of two cases exist:

1. κ_s was chosen sufficient to handle flows requested on the spot market

2. κ_s is not large enough to accommodate all of the spot flows

In case 1, it is clear there is no loss of profits on the spot market since the entire flow did not cause any congestion. However, in case 2, it follows that a loss of profits occurred on the spot market due to the excess flow requests that were impossible to accommodate. Thus, our formulation here minimizes the expected spot market loss and solves for this loss for each possible value of κ_s. In particular, we pose the following problem:

$$
\begin{aligned}
x_i &= \text{the actual demand level at bus } i \\
\widehat{x_i} &= \text{the expected multilateral amount of demand at bus } i \\
K_{i \to j}^{max} &= \text{the capacity of congested link} \\
\kappa_s &= \text{the amount of flow on congested link from spot market} \\
\kappa_{ms} &= \text{the amount of flow on congested link from bilateral market} \\
DF_i &= \text{the distribution factor of bus } i \text{ on congested link} \\
\epsilon_i &= \text{tolerance error of } x_i - \hat{x}_i \\
\rho_i &= \text{the expected spot price for injecting 1 MW at node } i
\end{aligned}
$$

It follows that $\kappa_{ms} + \kappa_s = K_{i \to j}^{max}$ at optimality, and $x_i - \hat{x}_i$ is the spot demand at bus i. Therefore, from the Chebyshev inequality, we can bound our error on this spot market demand as:

$$
\Phi_i(\epsilon_i) = Pr\Big\{(x_i - \widehat{x_i}) \geq \epsilon_i \Big\} \leq \frac{E\Big[(x_i - \widehat{x_i})^2\Big]}{\epsilon_i^2} \tag{4.1}
$$

We note that the right hand side of (4.1) only requires the first and second moments of x_i which we can compute from our stochastic data above. Then, if N represents the number of nodes in our network, we want to minimize our losses on the spot market so we can solve the following quadratic program:

$$\min_{\widehat{x_i}} \sum_{i=1}^{N} \rho_i[\Phi_i(\epsilon_i)]\epsilon_i \tag{4.2}$$

such that

$$\begin{aligned}
\sum_{i=1}^{N} DF_i\epsilon_i &\leq \kappa_s && (4.3)\\
\sum_{i=1}^{N} DF_i\widehat{x}_i &= \kappa_{ms} && (4.4)\\
\kappa_{ms} + \kappa_s &\leq K_{i\to j}^{max} && (4.5)\\
\widehat{\mathbf{x}} &\geq 0 && (4.6)
\end{aligned}$$

Solving this minimization problem gives the optimal $\widehat{x_i}\forall i$, and then it is straightforward to find κ_s using the constraints above. Thus, this is a second method by which the TSP can optimally solve for the quantity to allocate for the spot market at each demand level.

Chapter 5

Simulation of Multilateral Trading

Our simulation was created to iterate through the multilateral trading process detailed in chapter 3. *A priori* information includes the anticipated load demand, cost of generation curves, the maximum flow capacity of congested links (here link $2 \rightarrow 3$), and the number of iterations to process the agreements over. It should also be remembered that the iterations proceed sequentially, and thus iteration 2 takes into account that iteration 1 has been approved and implemented. Therefore, all transmission rights implemented are physical rights.

We next examine this scenario, as a function of the number of iterations.

5.1 Initial Conditions and Assumptions in Simulation Example

We use the demand data given by Fig. 1-4 and simulate the multilateral trading process for a given demand level and link capacity constraint. Either parameter can be varied for each simulation run, and thus numerous results can be computed for different demand levels and different capacity constraints. We show again here the possible locations for multilateral agreements and the corresponding spot demand at such times. As noted on Fig. 5-1, there are fifteen different demand levels for the simulation to run, if we split the market as multilateral and spot (see chapter 7).

Alternatively, if we run only a multilateral agreement market as in chapter 6, the simulation is run for all 120 demand levels of Fig. 1-4.

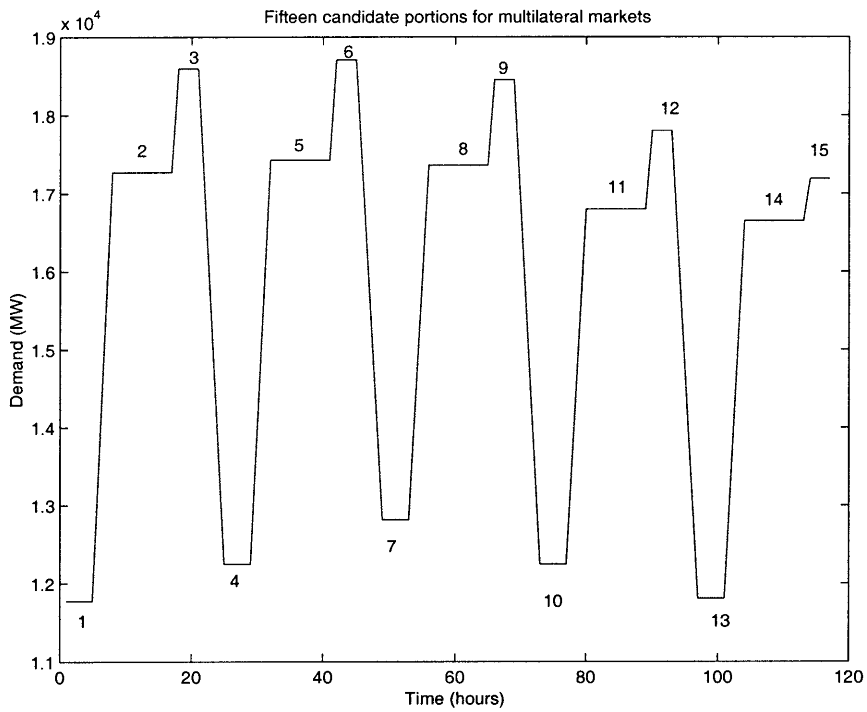

Figure 5-1: Candidate multilateral sections for combined market

$_{max} = 700$

In this chapter, we take a static snapshot of one hour, and hence one anticipated load demand level. To compare, we look at the results from a representative sample of demands during the night, daytime, and evening, as defined in chapter 4.

Before the results are shown, the following initial conditions were assumed throughout the simulation program.

5.1.1 Generator Cost Functions

The generator cost functions were assumed to be quadratic as follows

$C(Q_{G_i}) = a_i Q_{G_i}^2 + b_i Q_{G_i} + c_i$

The coefficients for each cost function is summarized in Table 5.1.

Generator	a	b	c
G1	0.11	5	150
G2	0.085	1.2	600
G3	0.1225	1	335

Table 5.1: Generator Cost parameters

5.1.2 Load Utility Functions

The load utility functions were also quadratic in nature:

$U(Q_{L_i}) = -a_i Q_{L_i}^2 - b_i Q_{L_i}$

It was assumed that before curtailment, the loads proportionally consume the total demand, L_T as follows:

$$
\begin{aligned}
L1 &= 0.2 * L_T \quad (5.1) \\
L2 &= 0.3 * L_T \quad (5.2) \\
L3 &= 0.5 * L_T
\end{aligned}
$$

These values were then used as the coefficient of the most significant term in the utility function Thus, for each demand level, $a_1 = 0.2$, $a_2 = 0.3$, and $a_3 = 0.5$.

The b values in Table 5.2 were approximated by setting the marginal utilities equal to the static equilibrium price. (See section 2.1) Therefore, these utility functions depend on the expected demand level. Since there are fifteen multilateral agreements over the week for the simulation done in chapter 7, load utility functions parameters are presented for each demand level in Table 5.2.

Similar b_i values were found for all 120 demand levels when simulating the entirely multilateral market structure of chapter 6.

Demand (MW)	b_1	b_2	b_3
11775	-1755.9	-2933.4	-6701.4
17273	-2574.9	-4302.4	-9830.4
18594	-2769.0	-4626.0	-10570.0
12245	-1789.4	-2989.4	-6829.4
17428	-2582.4	-4314.9	-9858.9
18706	-2780.0	-4645.0	-10613.0
12815	-1875.0	-3132.5	-7156.5
17363	-2556.3	-4271.3	-9759.3
18456	-2765.0	-4620.0	-10556.0
12245	-1789.4	-2989.4	-6829.4
16800	-2481.9	-4146.9	-9474.9
17800	-2661.0	-4446.0	-10158.0
11810	-1729.8	-2889.8	-6601.8
16648	-2474.4	-4133.4	-9446.4
17188	-2552.6	-4265.1	-9745.1

Table 5.2: Load Utility parameters

Demand	$K_{2\to3}^{max}$
11775	1200
17273	1800
18594	1800
12245	1300
17428	1800
18706	1800
12815	1300
17363	1800
18456	1800
12245	1300
16800	1800
17800	1800
11810	1200
16648	1700
17188	1800

Table 5.3: Maximum Line Capacity

5.1.3 Congested Link Capacity

For the three multilateral contract periods per day, we note the demand levels are quite different. Indeed, it follows that the capacity of link $2 \to 3$ may indeed vary for the three periods. This is not too much of a stretch, since numerous real cases have been shown where independent system operators "switch on and off" links between the networks. Thus, it is plausible that the capacity of the congested link varies with the demand level as a means to save on operational and fixed costs. The corresponding $K_{2\to3}^{max}$ for each agreement is summarized in Table 5.3.

5.1.4 Curtailment Strategy

With this knowledge given beforehand, the simulation follows the methodology outlined by Allen et. al. in [5]. First, using the economic efficient arguments presented in section 2.1, the optimally efficient quantities of generation are computed. Then, if the line constraint is met, the simulation calculates the respective profits and total social welfare for this solution. However, if the flow along the congested link is larger than $K^{max}_{2\to3}$, then the end-users are curtailed. The curtailment strategy uses a minimax procedure to keep all end-users as close as possible to their optimal operating points. After curtailment sets the flow along link $2 \to 3 = K^{max}_{2\to3}$, the simulation begins. The trilateral groupings are as mentioned in the original problem statement, with possibilities being all three generators, or two generators and one load. The simulation solves the subproblem given in section 1.2. In the next few sections, we examine the results of this simulation.

5.1.5 Distribution Factors and Broken Network

In the standard operating case, the impedances of all the lines are equal at 1. To simulate a broken network, or line outage of link $1 \to 2$, we increase the impedance of link $1 \to 2$ to be 1000, while keeping the impedance on the other two links at 1. Thus, very little power flows through this link, effectively modeling the broken network topology.

5.2 Quantities of Individual End-Users

Our first point of interest is to examine how the quantities traded are different for a sample selection of demand levels, and different capacity link constraints.

5.2.1 Effects of the Capacity Constraint

We take the three cases of maximum amount of flow on link $2 \to 3 = 100, 700, 1200$ MW, when the anticipated demand is fixed at 11775 MW.

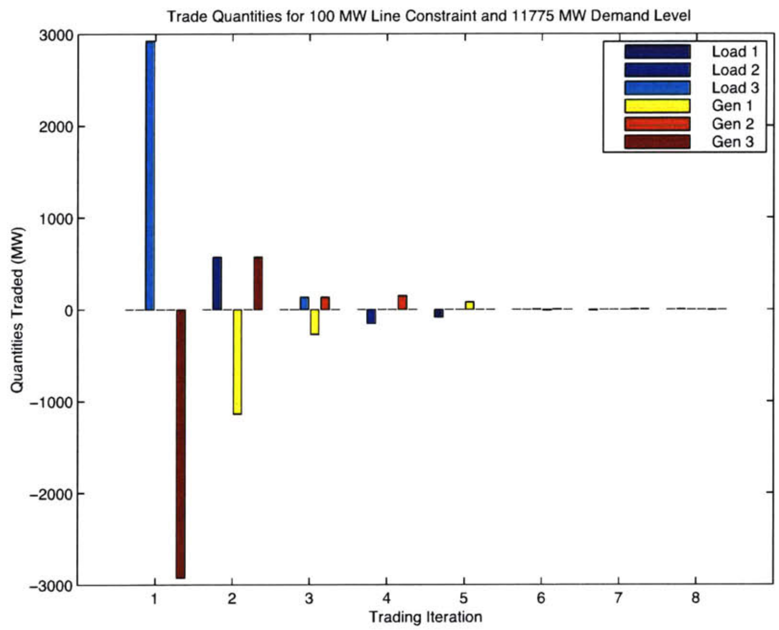

Figure 5-2: Quantities for Night Demand, Line Capacity=100

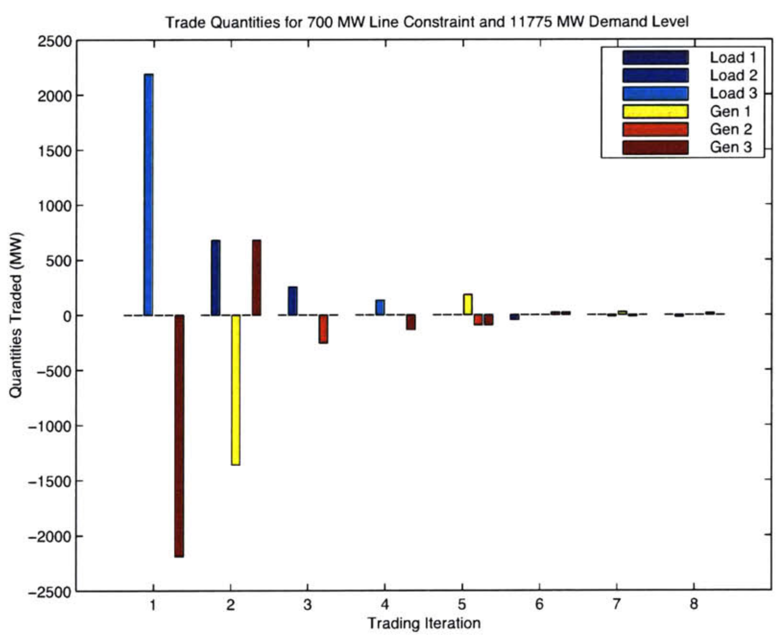

Figure 5-3: Quantities for Night Demand, Line Capacity=700

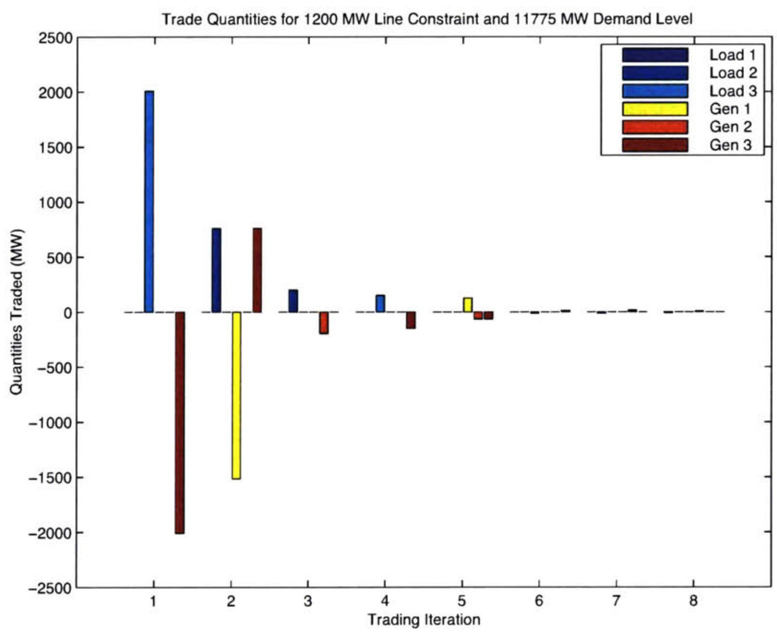

Figure 5-4: Quantities for Night Demand, Line Capacity=1200

From these results, it is evident that the quantities exponentially decrease as the iterations proceed. Another interesting result is that the same-end users are first to act regardless of the capacity level. However, the users do change after the first iteration (arguably the most important one), is complete and implemented. Finally, the magnitude of the quantities traded on the first iteration clearly decreases as the line capacity increases, but the subsequent iterations are much more comparable in nature.

5.2.2 Impact of the Load Demand Level

We now fix the capacity constraint, $\kappa_{ma} = 700MW$, and examine these results for varying demand levels:

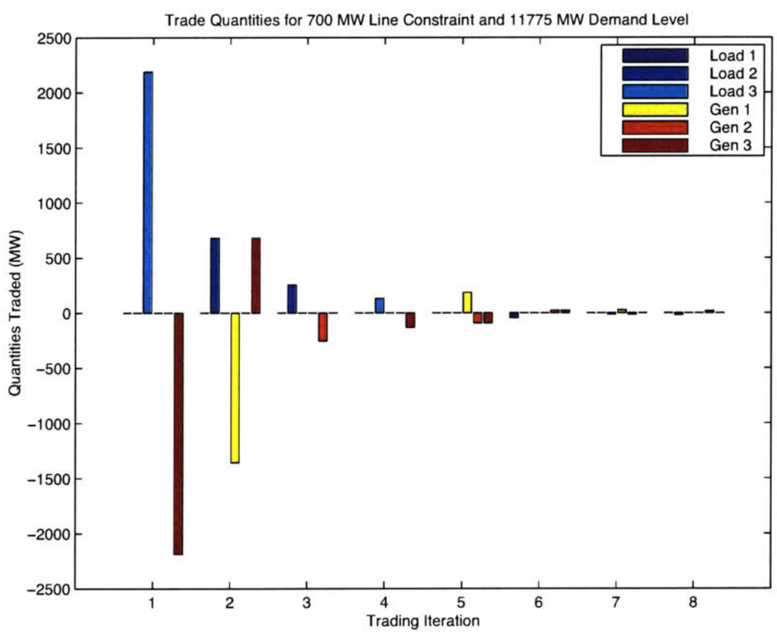

Figure 5-5: Quantities for Night Demand, Line Capacity=700

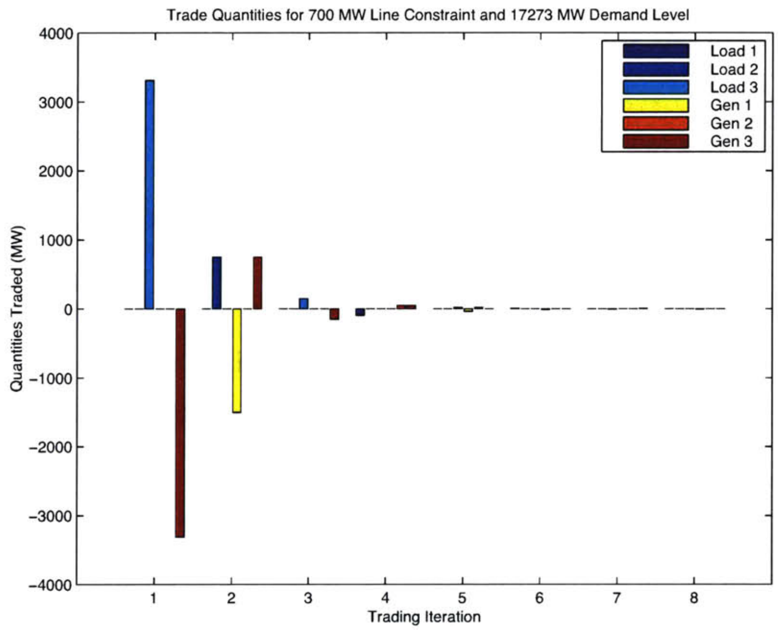

Figure 5-6: Quantities for Daytime Demand, Line Capacity=700

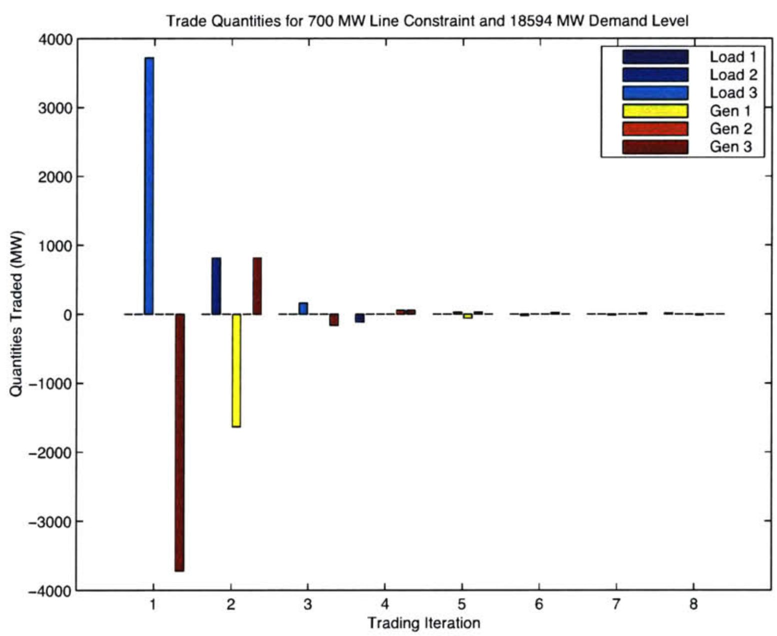

Figure 5-7: Quantities for Evening Demand, Line Capacity=700

As can be seen, the quantities traded decay faster as the demand level increases. In this case as well, depending on the iteration number, the end-users involved in the trading are usually the same.

5.3 Participant Profits

We observe the results now for the profit of end users. These are the profits end-users receive in the electricity market for having their trilateral bids accepted and implemented by the TSP.

5.3.1 Effects of the Capacity Constraint

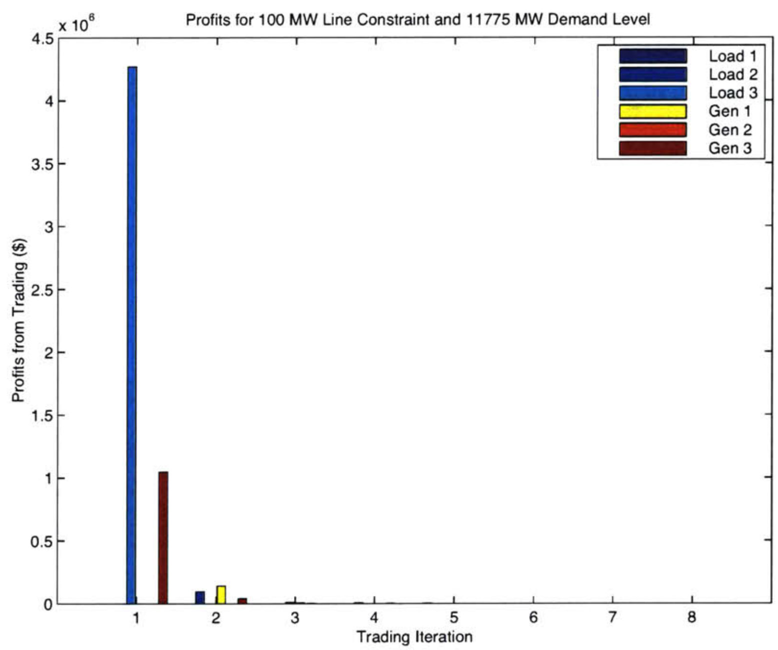

Figure 5-8: Profits for Line Capacity=100, Night Demand

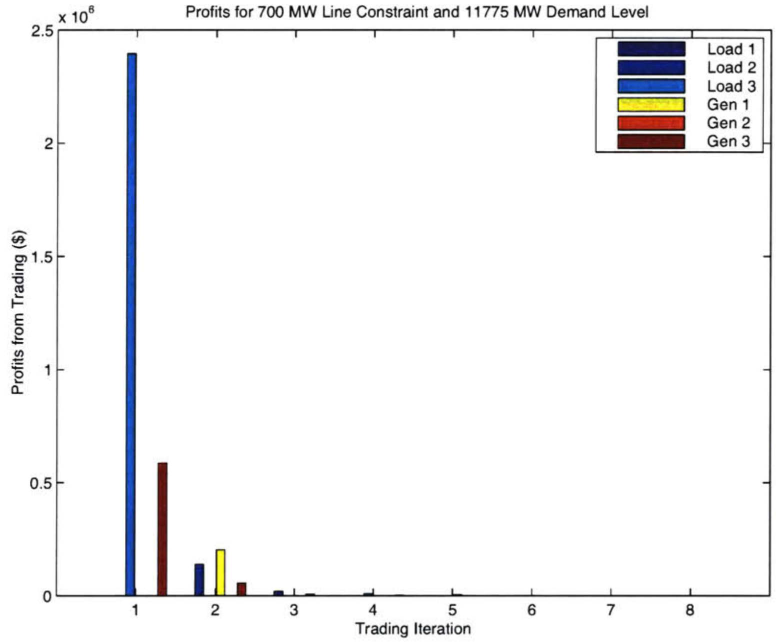

Figure 5-9: Profits for Line Capacity=700, Night Demand

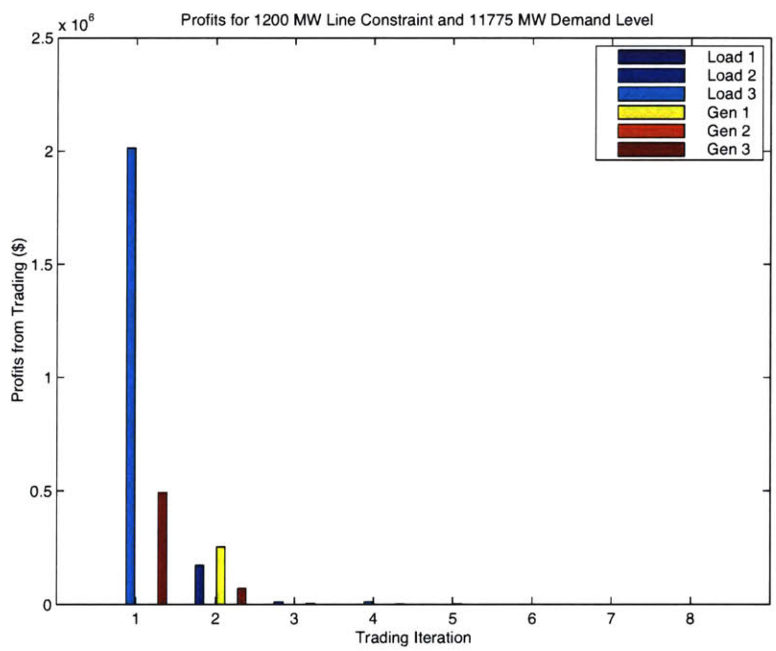

Figure 5-10: Profits for Line Capacity=1200, Night Demand

The results here indicate that profit is inversely proportional to the value of κ_{ma}. As the constraint is tightened, much more profit is made in the first few iterations. Moreover, the amount of iterations required before equilibrium is reached decreases as as κ_{ma} decreases.

5.3.2 Impact of the Load Demand Level

Again, we take 700 MW along congested link and now compare for different time periods.

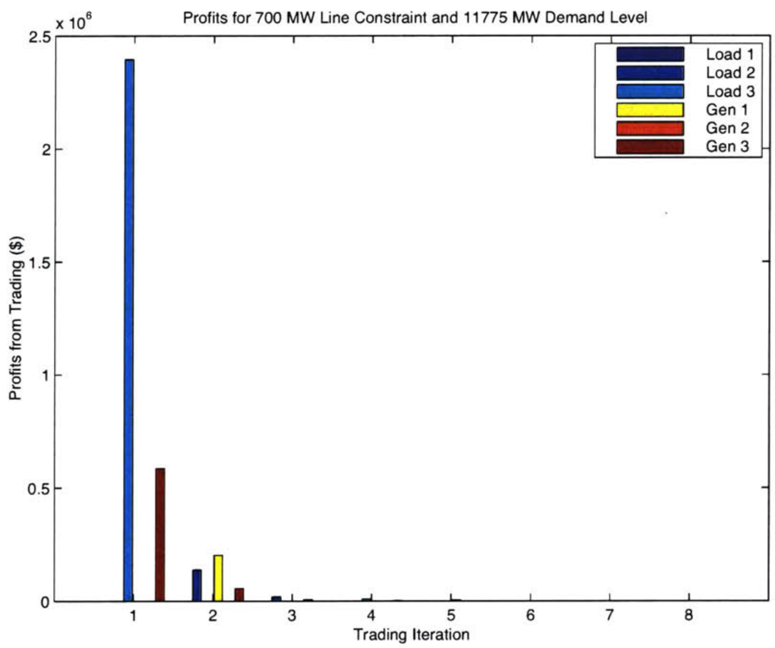

Figure 5-11: Profits for Line Capacity=700, Night Demand

From these graphs, it is clear the profit increases as the demand increases. This can be explained by the fact that increased quantities are being traded to meet the increased demand. Moreover, it is seen that regardless of the demand level, it is shown that usually the same trilateral group of users act at each iteration. Thus, end users can be somewhat confident that if their coordination maximizes total social welfare at one demand level, there is a high probability that similar coordination will be optimal at different demand levels at the same iteration.

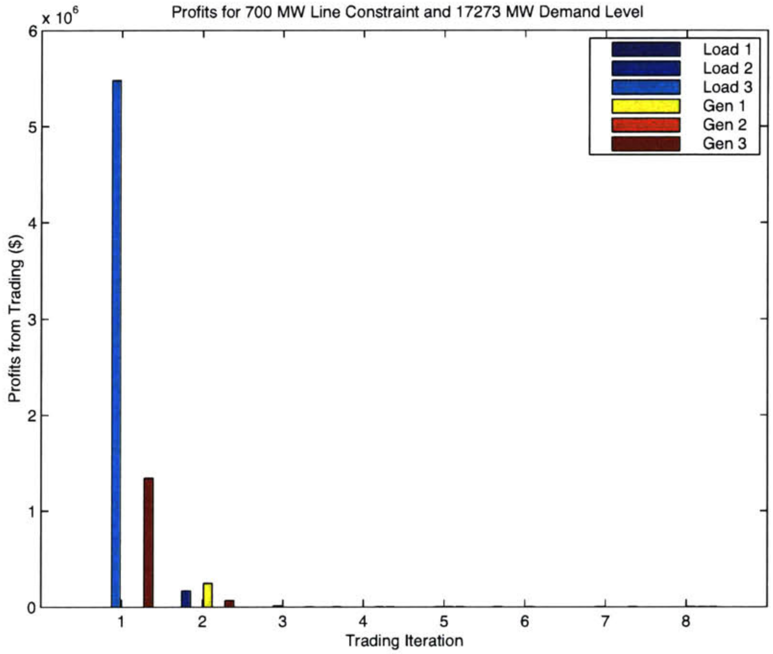

Figure 5-12: Profits for Line Capacity=700, Daytime Demand

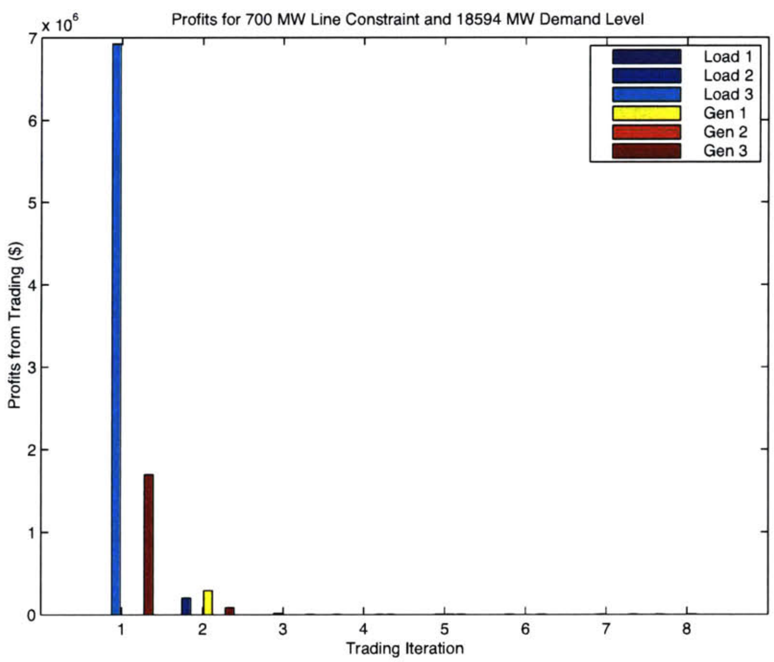

Figure 5-13: Profits for Line Capacity=700, Evening Demand

5.4 TSP profits

In this section, we assume the transmission service provider will receive some scaled portion of the total profits of those end-users involved in accepted multilateral agreements. We show here the sum of such profits as a function of κ_{ma}, for different demand levels.

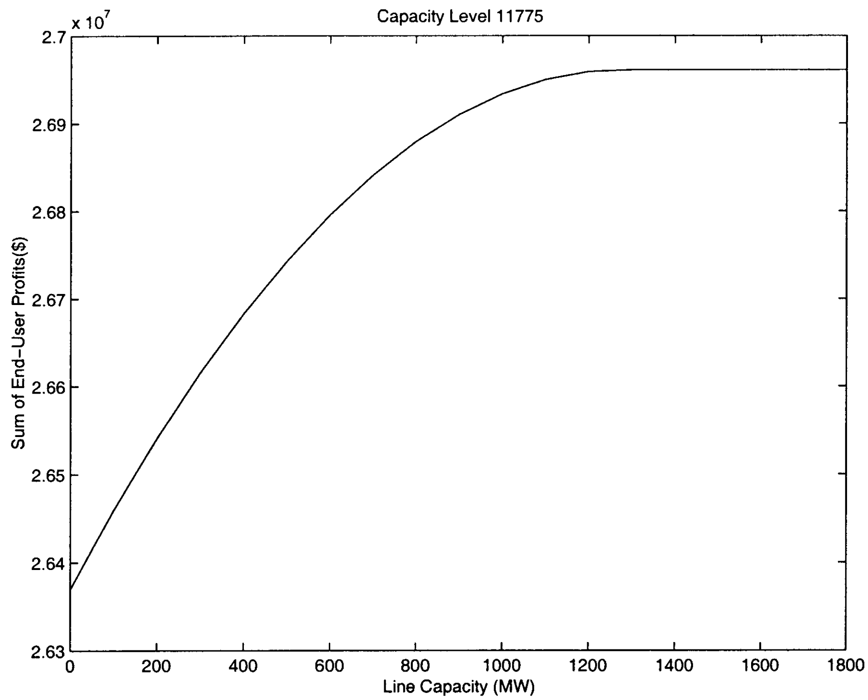

Figure 5-14: TSP Profits for Night Demand of 11775 MW

It is interesting to see that the TSP's profits will generally increase as more allocation is give for multilateral trading.

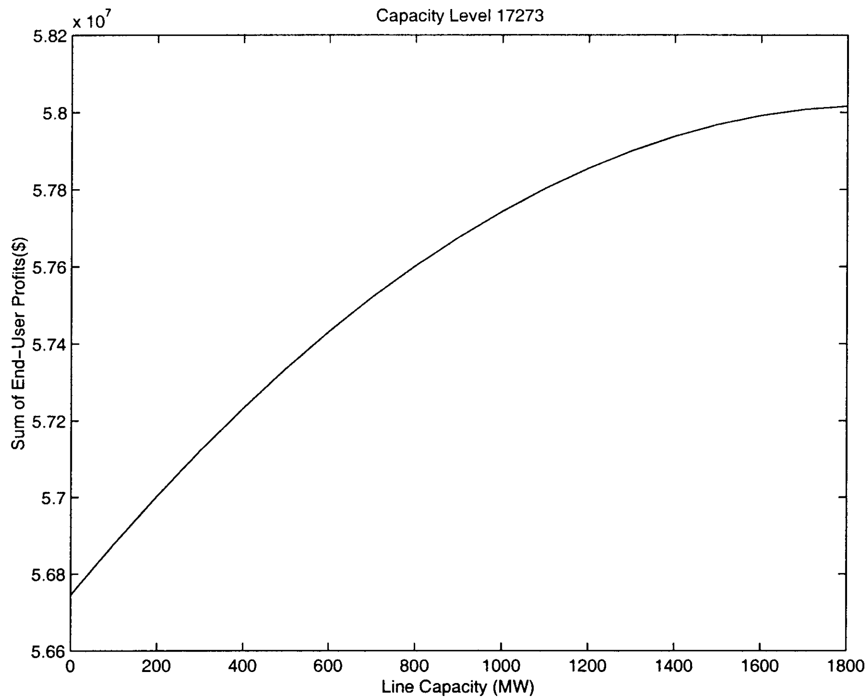

Figure 5-15: TSP Profits for Daytime Demand of 17273 MW

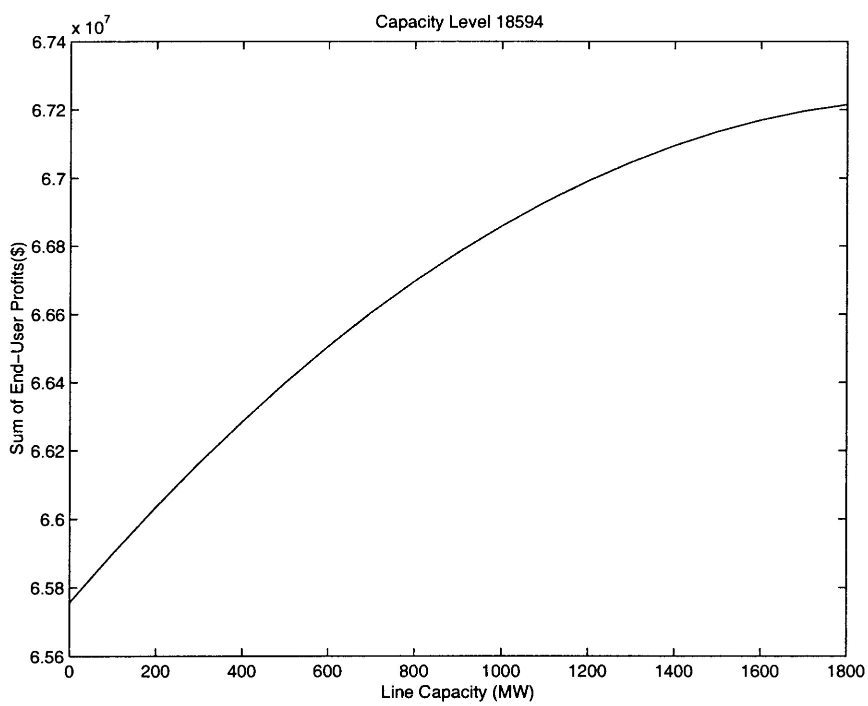

Figure 5-16: TSP Profits for Evening Demand of 18594 MW

5.5 Total Social Welfare

Our next point of interest is to examine how total social welfare change as the number of iterations of multilateral trading increases. For this purpose, we set the demand level and show representative plot of different line capacities under each of the given scenarios, and vice versa.

5.5.1 Effects of the Capacity Constraint

We take the three cases of maximum amount of flow on link $2 \rightarrow 3 = 100, 700, 1200$ MW, when the anticipated demand is fixed at 11775 MW.

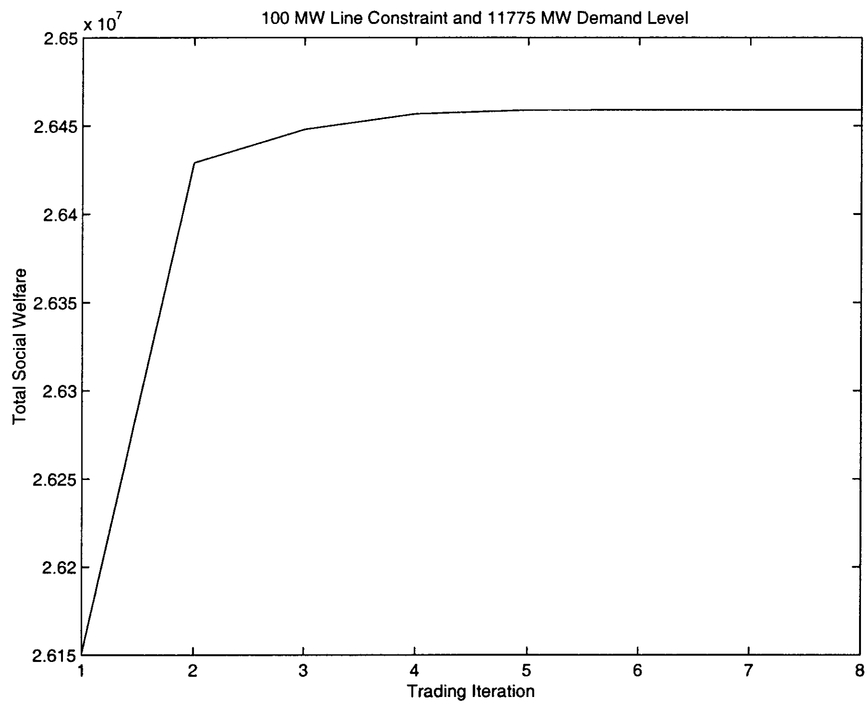

Figure 5-17: Total Social Welfare for Line Capacity=100, Night Demand

These results show that total social welfare is indeed approaching the optimal value. There is also some indication here that this optimal value can be reached faster at higher levels of K_{ma}. Nevertheless, the final value reached is the same independent of the line constraint, if the number of iterations is large enough.

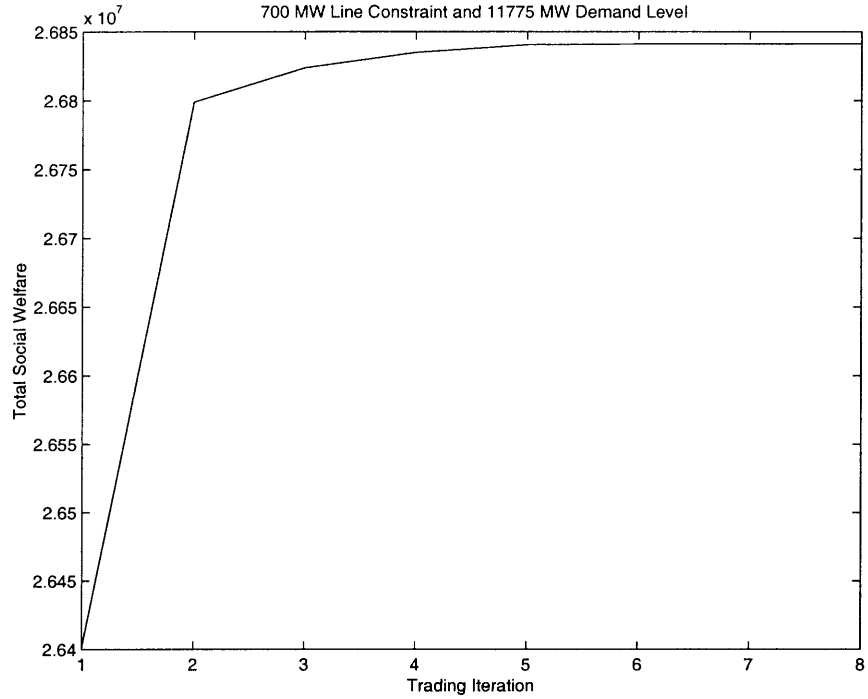

Figure 5-18: Total Social Welfare for Line Capacity=700, Night Demand

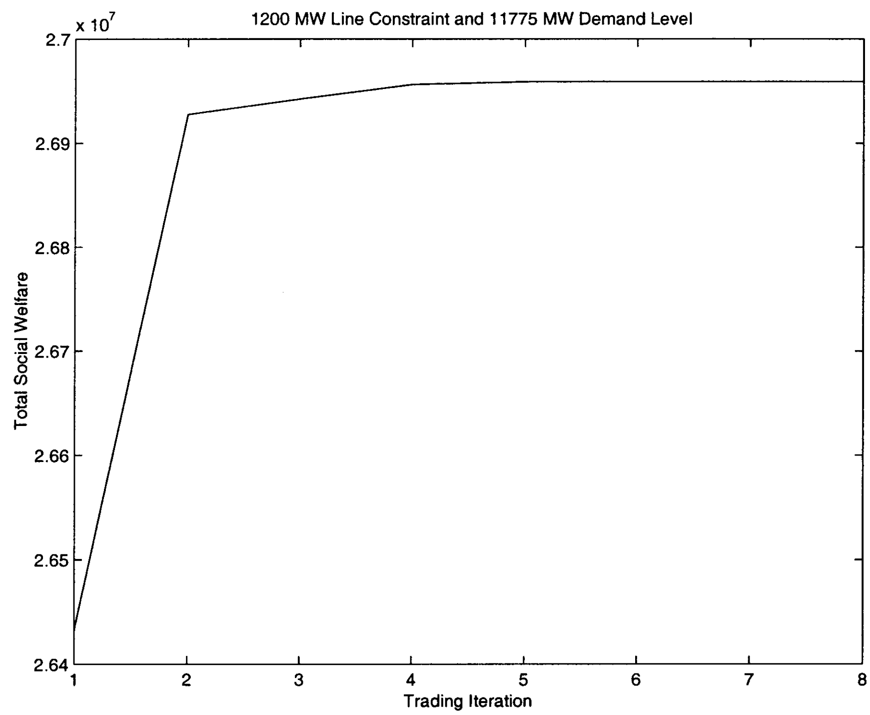

Figure 5-19: Total Social Welfare for Line Capacity=1200, Night Demand

5.5.2 Impact of the Load Demand Level

We assume $K_{ma} = 700$ MW along link $2 \rightarrow 3$ and now compare for different time periods.

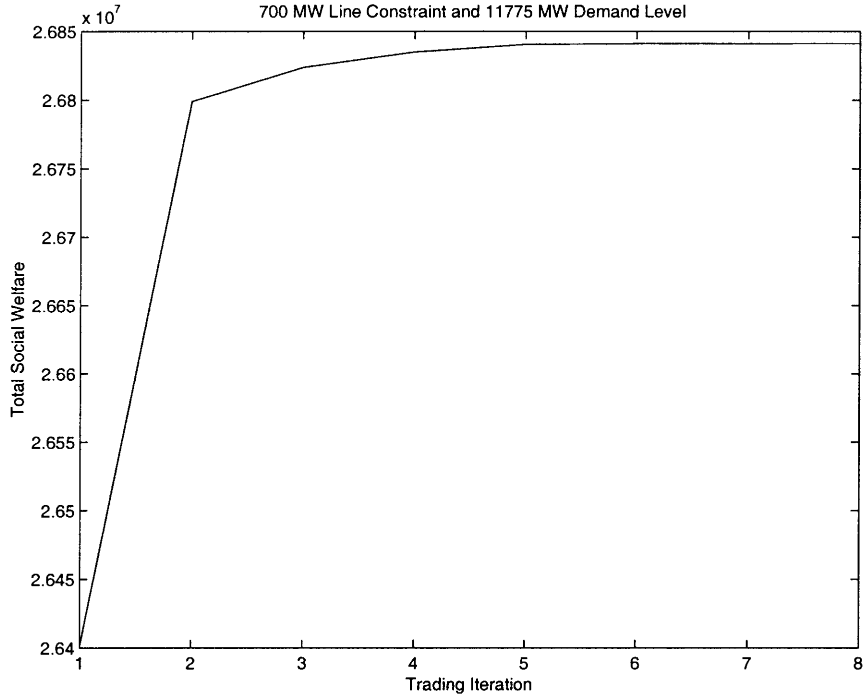

Figure 5-20: Total Social Welfare for Line Capacity=700, Night Demand

The results indicate that the magnitude of the total social welfare reached is proportional to the anticipated demand level.

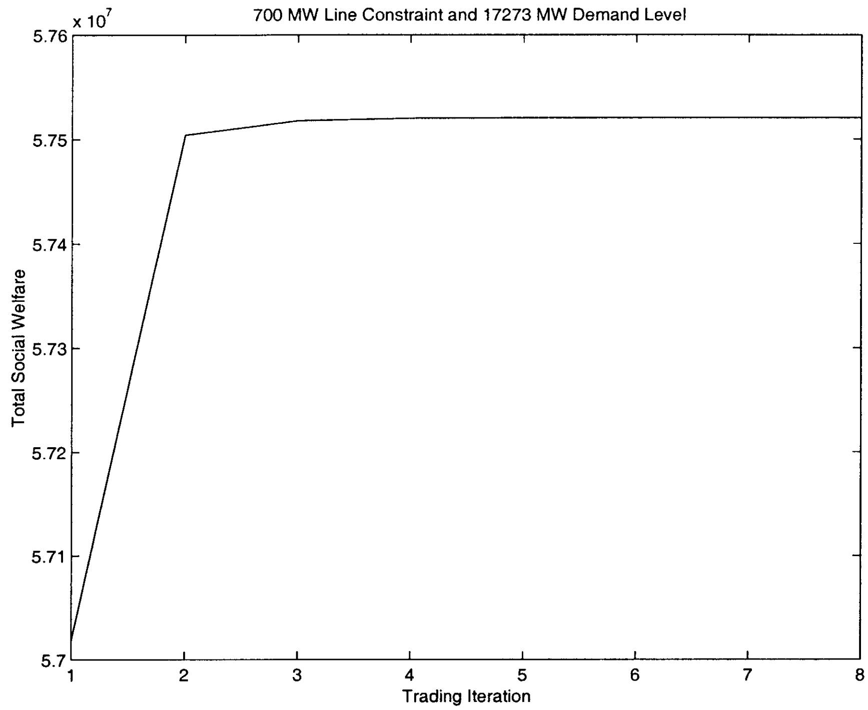

Figure 5-21: Total Social Welfare for Line Capacity=700, Daytime Demand

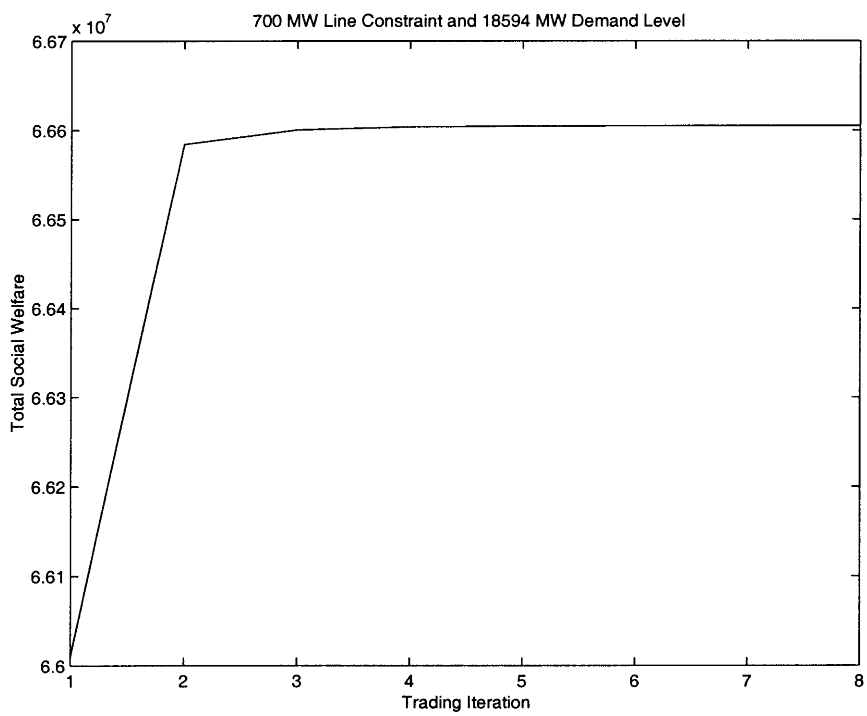

Figure 5-22: Total Social Welfare for Line Capacity=700, Evening Demand

5.6 Different Criteria Used to Choose Agreements

Since the simulation created has end users submitting their "best" agreement for each iteration, we can use different heuristics to optimize over and show how the items of interest change. For example, let us define a user-driven market as one that computes trilateral market bids which maximize profits for all end users involved. In contrast, let us call a network-driven market as one that computes trilateral market bids which maximize flows along all the links of the network. Therefore, this market does not allow local trading, i.e. between users at the same bus. These market structures thus produce different trilateral bid possibilities. However, the mechanism to choose and implement bids is the same as before, namely pick those bids that maximize total social welfare.

5.6.1 Individual Participants Quantities

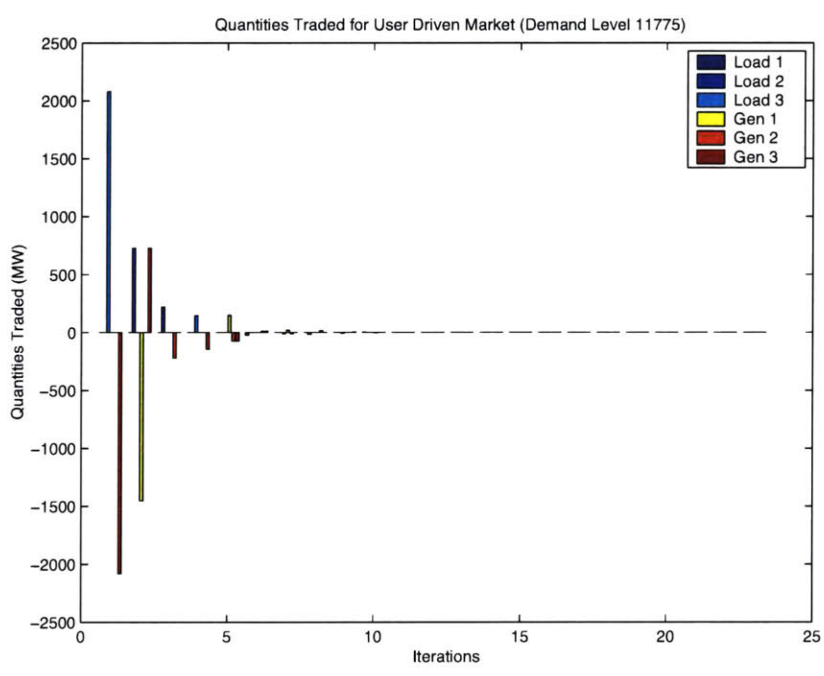

Figure 5-23: End User driven case

The results here show that the network driven case takes nearly four times the

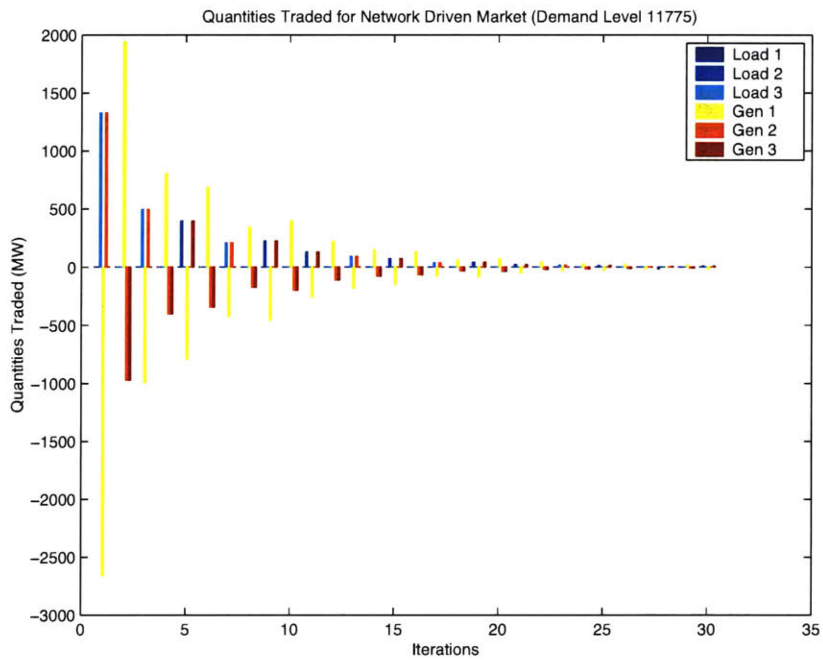

Figure 5-24: Network Driven case

number of iterations as the user driven case before the quantities traded decay to 0. Thus, the impact of local trades on the number of iterations required to reach equilibrium, is quite significant.

5.6.2 Total Social Welfare

We next examine how these different criteria affect total social welfare.

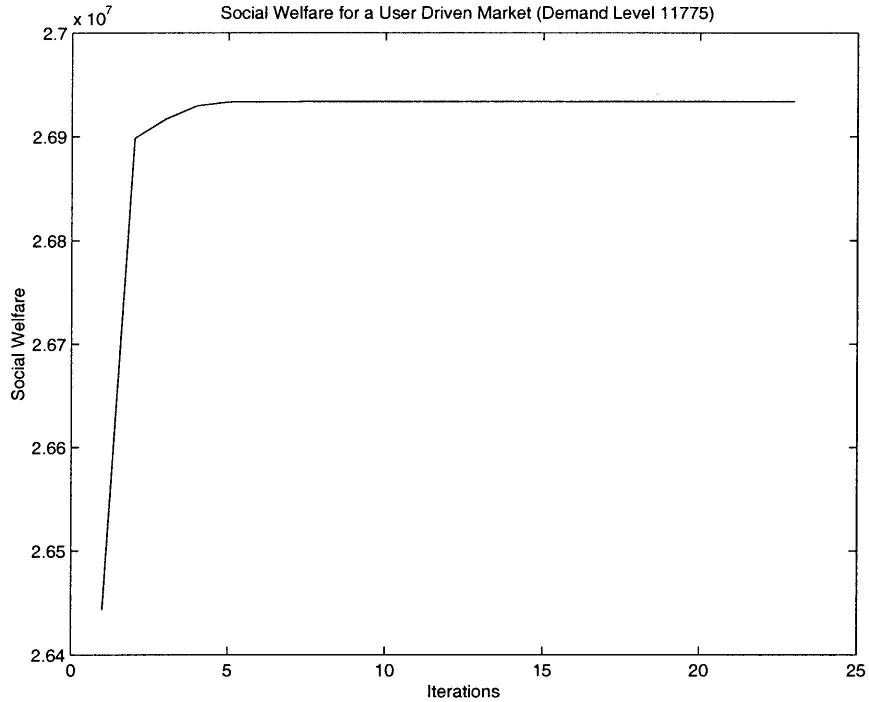

Figure 5-25: End User driven case

These simulations greatly elucidate insights of why and how it takes longer to reach to optimal total social welfare, when dealing with a network-driven case as compared to the user-driven case.

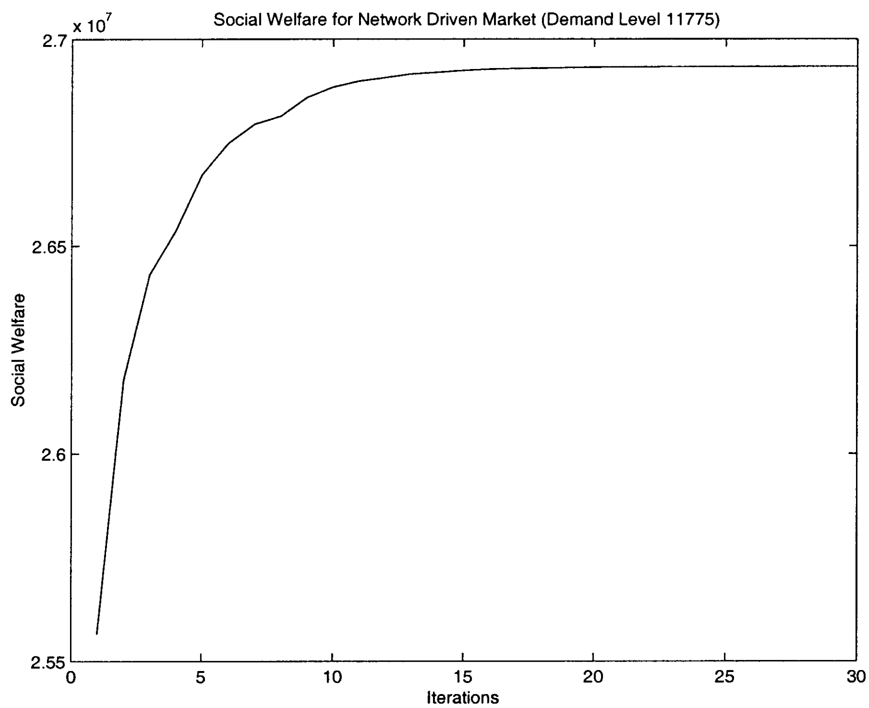

Figure 5-26: Network Driven case

5.7 Demand Curves for Transmission

Finally, if now we have the original bids as before, i.e. a user-driven market which allows for local trades, we can derive the total quantities and profits end-users make for any given capacity of the congested line. We exploit this fact to answer the most interesting question:

- What is the total demand function for transmission for all end-users?

Clearly the demand for transmission should imply the price the set of end-users are willing to pay for the total amount of flow along the congested link (i.e. which is equal to the capacity constraint). The results for a sample set of demand values is as follows:

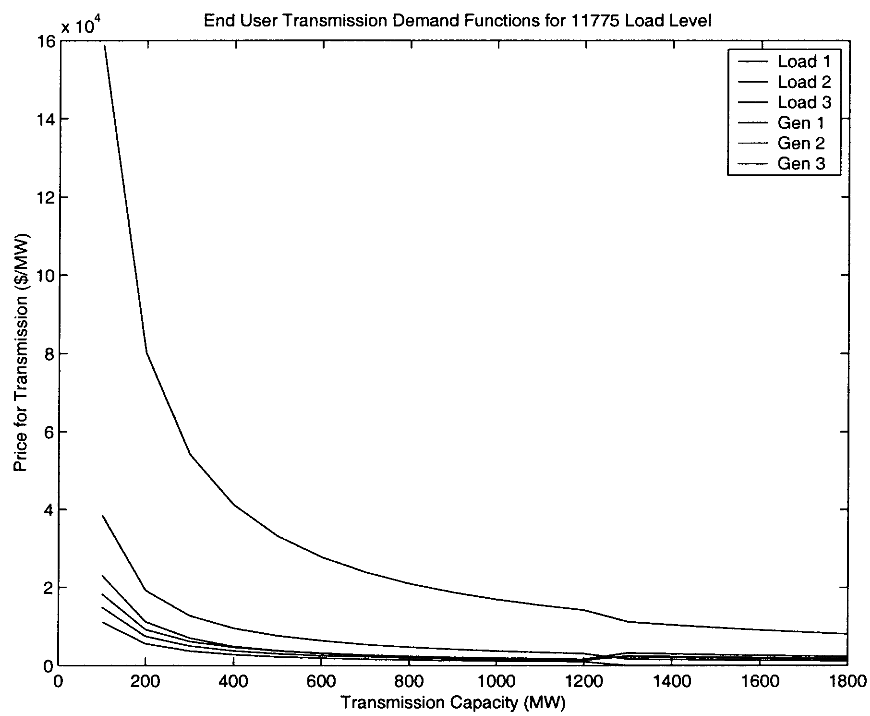

Figure 5-27: Transmission Demand Curve, Anticipated demand = 11775 MW

This graph was derived by examining each end-users net profits after all iterations were complete for the different levels of κ_{ma}. These profits were then divided by the corresponding κ_{ma} value to obtain price or willingness to pay, as a function of κ_{ma}. Thus, it follows that these demand curves represent each end users' willingness to pay for transmission, for different values of flow along the congested link. Thus, by providing such information to the TSP, it is possible for the TSP to optimize his decision making regarding allocation of flows without knowing cost or utility function for electricity of the end users. Thus, it is demonstrated that it is possible for a scheme to exist that unbundles the two markets of electricity generation and transmission.

This chapter presented details of the simulation for a static instant in time. This is relaxed in the next two chapters, so full dynamics of two different market structures can be compared for the full week of interest.

Chapter 6

Multilateral Trading Over The Week

We now use the concepts of chapter 5, but extend them to a dynamic setup for the week setup, instead of the analysis for a single static snapshot. We again look at the key items of interest, graphed for each hourly period as follows. This chapter limits the number of iterations of multilateral agreements described in chapter 5 to three. Finally, it should be noted that since the market structure is entirely multilateral, $\kappa_{i \to j}^{max}$ is allotted solely for multilateral agreements. Thus, in this chapter $\kappa_{ma} = \kappa_{i \to j}^{max}$.

This chapter is based on the problem statement presented in chapter 1, specifically for the TSP objective in (1.1), with profits coming from the spot market being equal to 0. Since this chapter involves only a multilateral market structure, the simulation examined in chapter 5, is now run for the anticipated load demand levels given by Fig. 1-4. Representative plots are shown for the three time periods of each day, as well as for a low, medium, and high capacity constraint.

6.1 Individual End-User Quantities Over the week period

The end-user quantities are taken individually and graphed over the week period. Plots are shown for κ_{ma} values of 100 MW, 700 MW and 1100 MW.

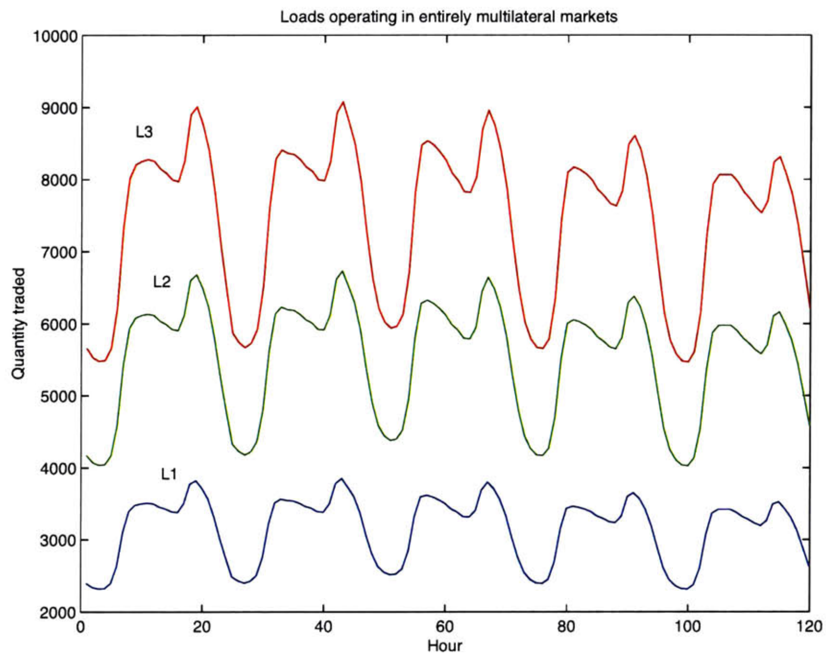

Figure 6-1: Load Quantities, 100 MW

We note the difference in values for the three loads is most probably a consequence of the differing utility functions. More specifically, the values here are similar in proportion to the a_i coefficients specified for the loads in chapter 5. Similarly, the different generator quantities are most probably due to their unique cost functions. Furthermore, these results indicate that as $\kappa_{i \to j}^{max}$ increases, more generation is required from G_1 and G_2 than G_3.

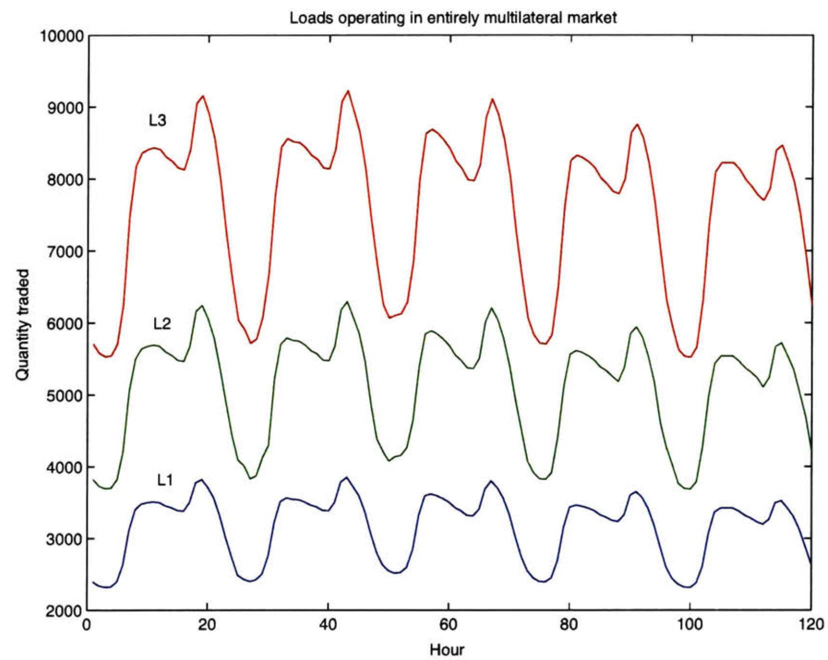

Figure 6-2: Load Quantities, 700 MW

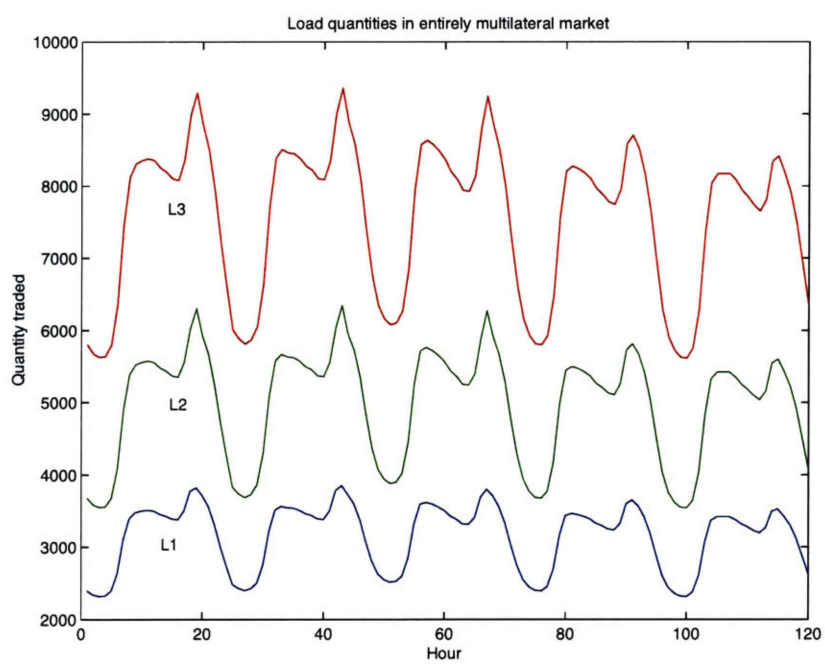

Figure 6-3: Load Quantities, 1100 MW

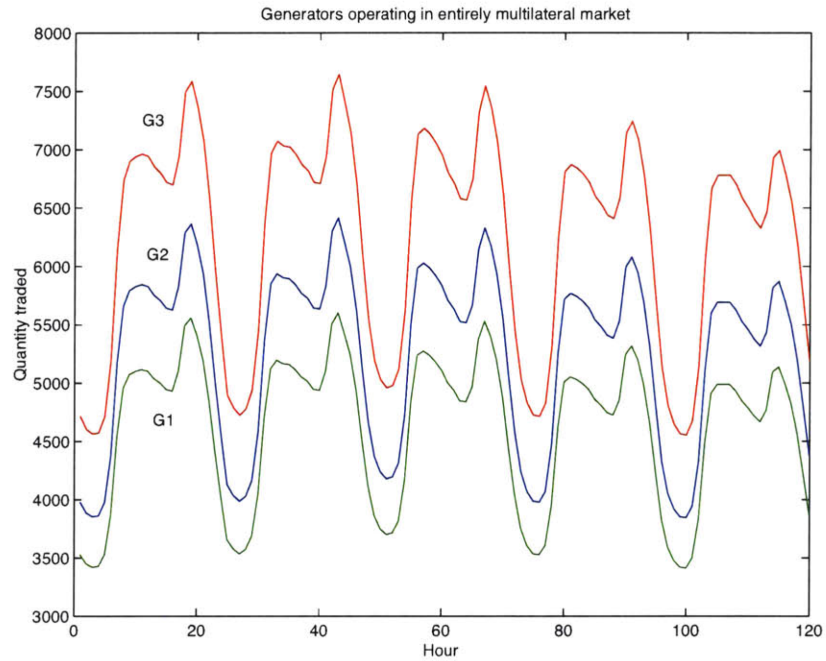

Figure 6-4: Generator Quantities, 100 MW

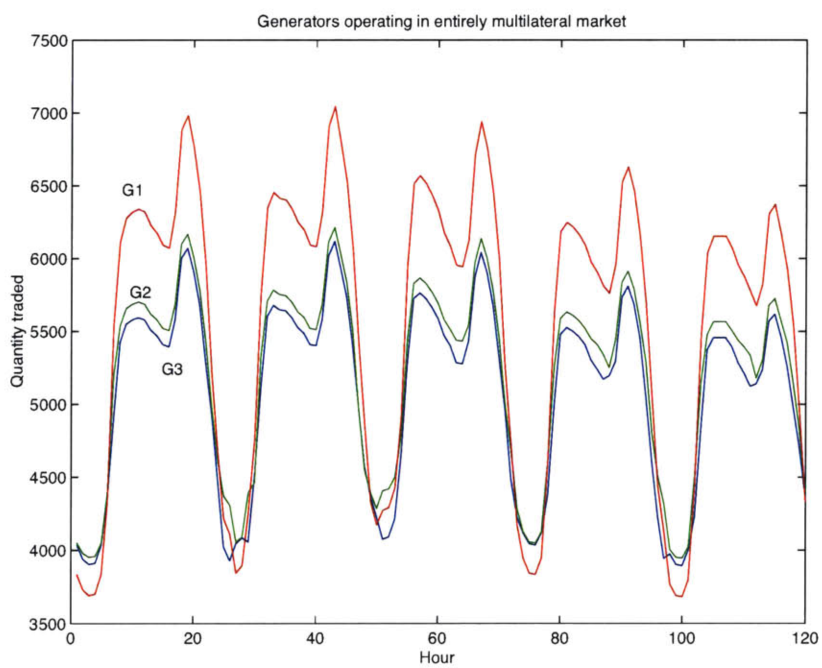

Figure 6-5: Generator Quantities, 700 MW

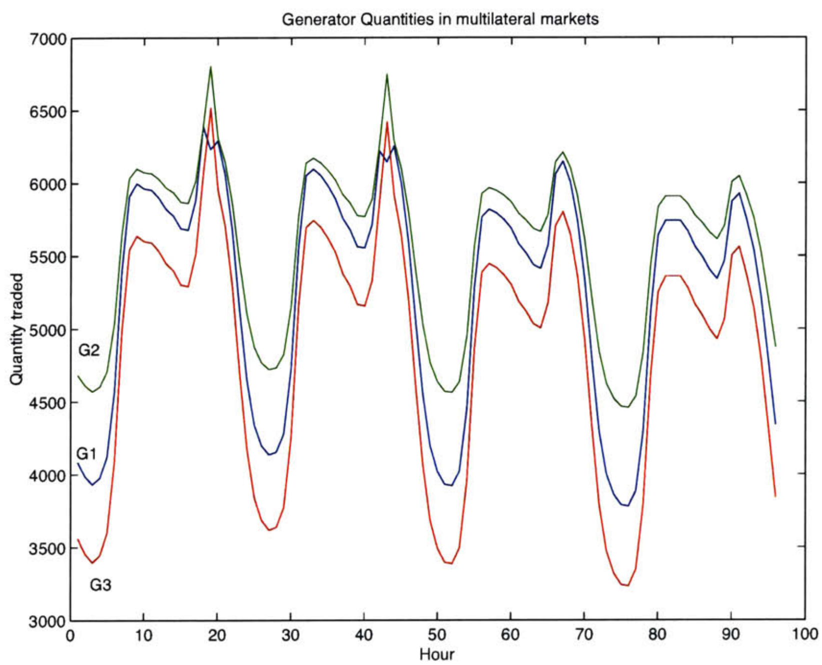

Figure 6-6: Generator Quantities, 1100 MW

6.2 TSP Profits or Sum of End-user Profits

We now look at how profit changes occur over time, depending on the line capacity constraint.

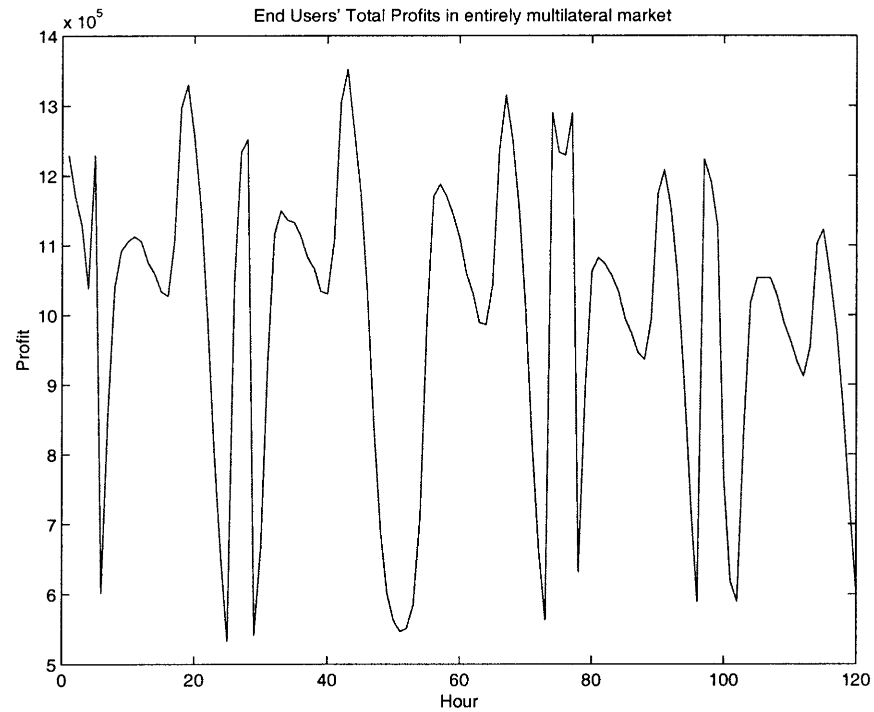

Figure 6-7: TSP Profits, 100 MW

As expected, the pattern of profits follows the anticipated load graph given in Fig: 1-4. We assume here that the TSP is paid the full amount of profits the end-users make in the electricity market, when such trilateral deals are implemented. As expected, the pattern of profits follows the anticipated load graph given in Fig: 1-4. Specifically, the profits more closely resemble this shape when $\kappa_{i \to j}^{max} = 1100MW$. However as $\kappa_{i \to j}^{max}$ decreases, the profits to the TSP are much more erratic and volatile. It is clear this constraint affects which trades are physically feasible at certain hours. In addition, the TSP's net profits increase as $\kappa_{i \to j}^{max}$ increase.

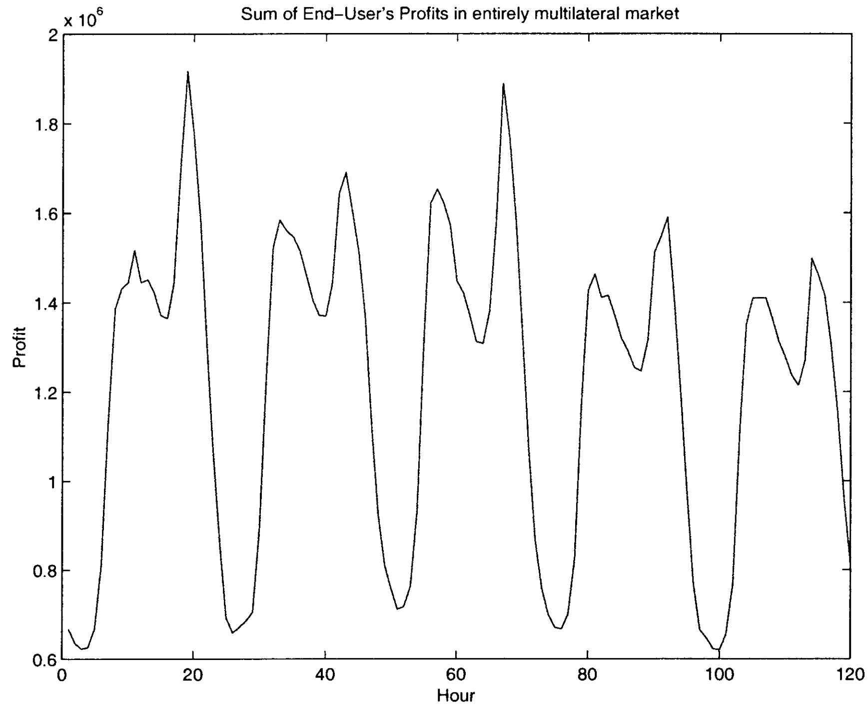

Figure 6-8: TSP Profits, 700 MW

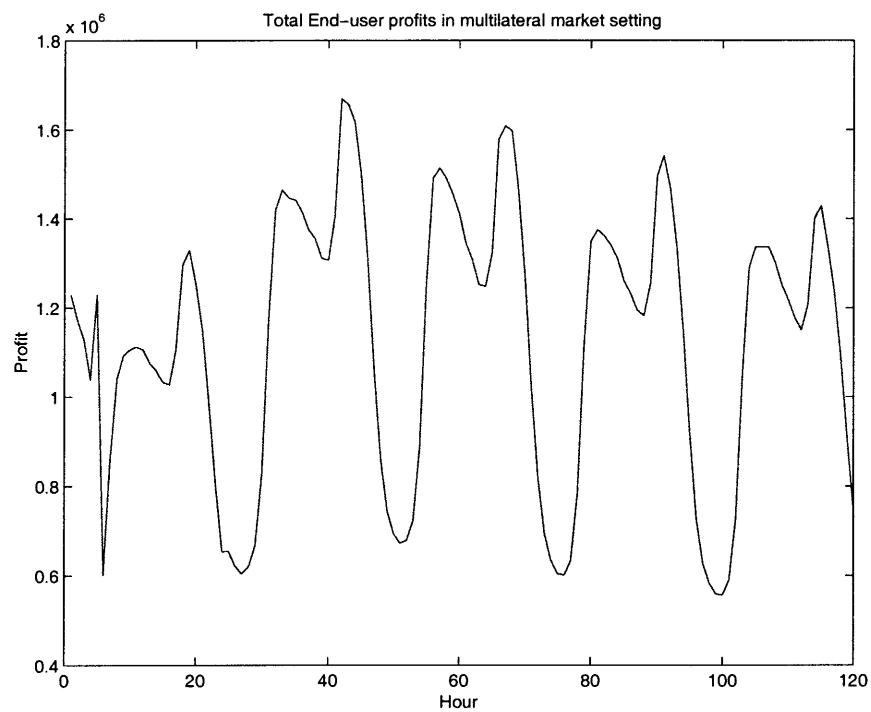

Figure 6-9: TSP Profits, 1100 MW

6.3 Total Social Welfare over the week

Finally, we examine total social welfare in this entirely multilateral market.

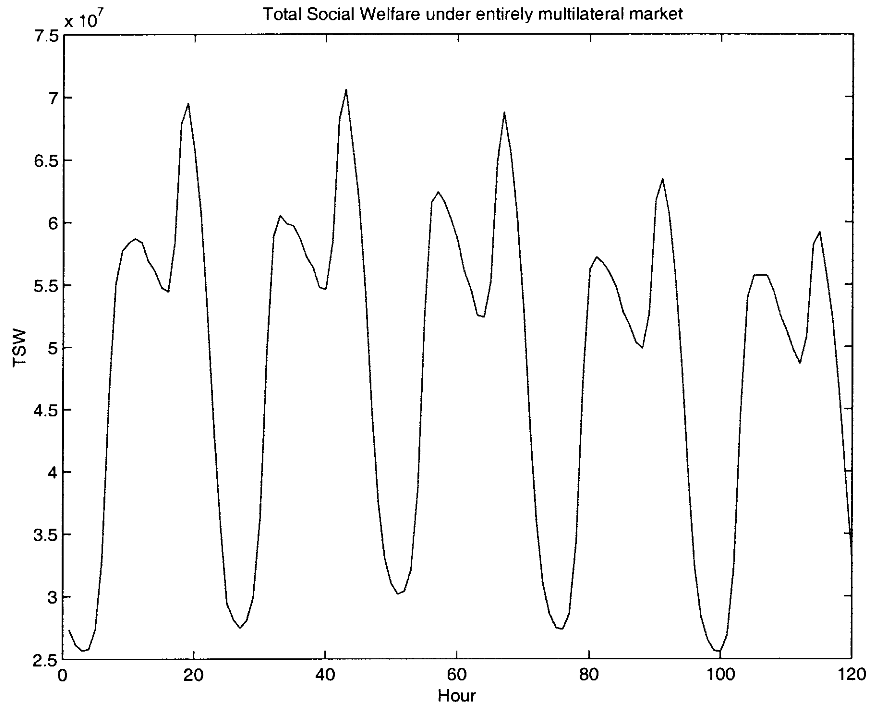

Figure 6-10: Total Social Welfare, 100 MW

We conclude from these plots that total social welfare reached is independent of the line constraint, $\kappa_{i \to j}^{max}$. These observations confirm those found in Section 5.5.1.

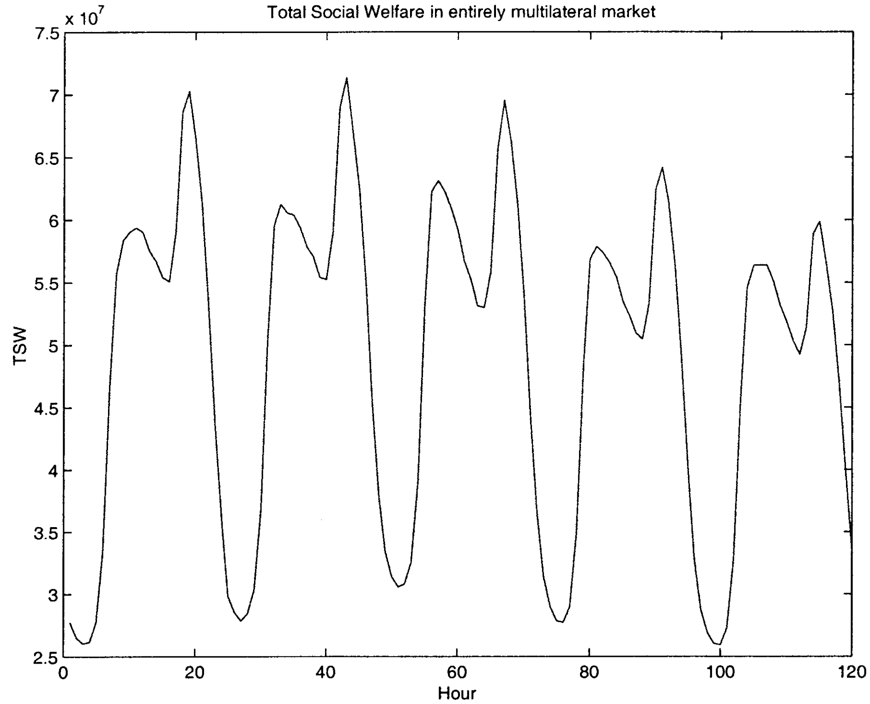

Figure 6-11: Total Social Welfare, 700 MW

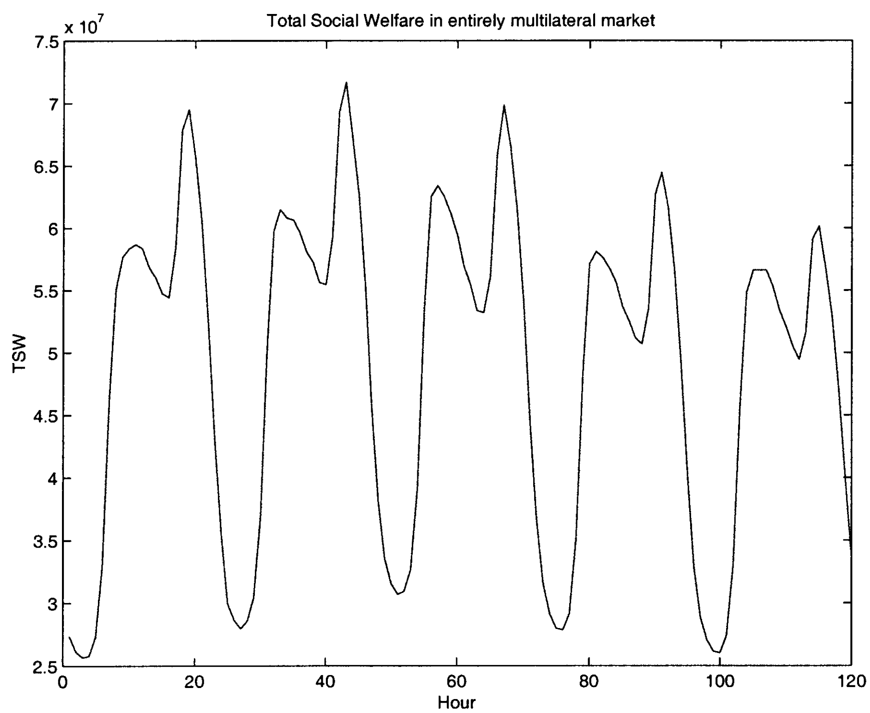

Figure 6-12: Total Social Welfare, 1100 MW

For the broken network case where link $1 \rightarrow 2$ carries very little flow, the results for TSP profits and total social welfare are as follows. The capacity of the line for this case was 700 MW.

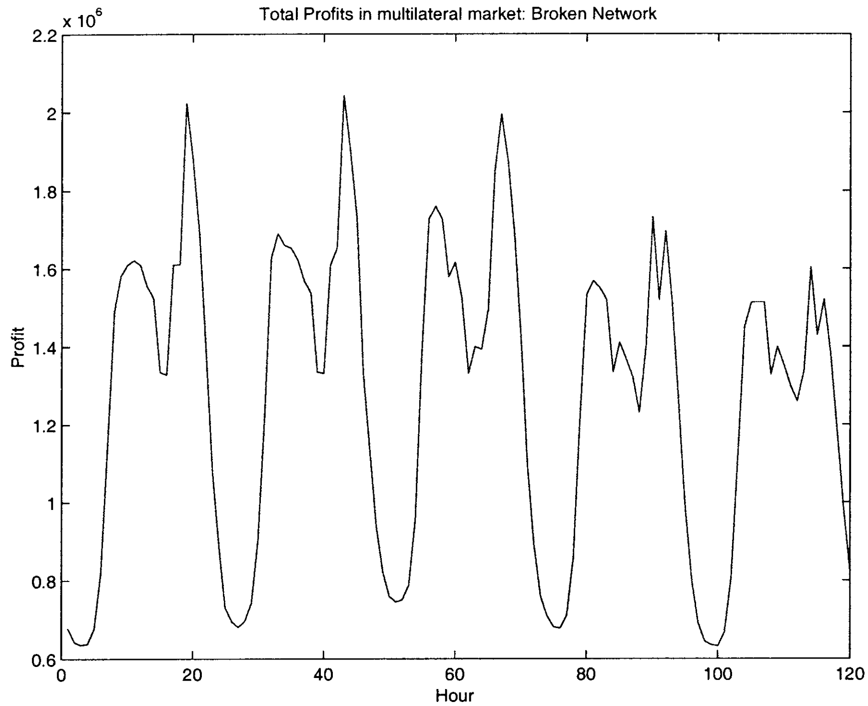

Figure 6-13: TSP Profits: Broken Network, 700 MW

We note that Fig: 6-13 is extremely similar to Fig: 6-8. Similarly, Fig: 6-14 is extremely similar to Fig: 6-11. Hence, we conclude that the broken network has little to no effect on the TSP's profits, or total social welfare.

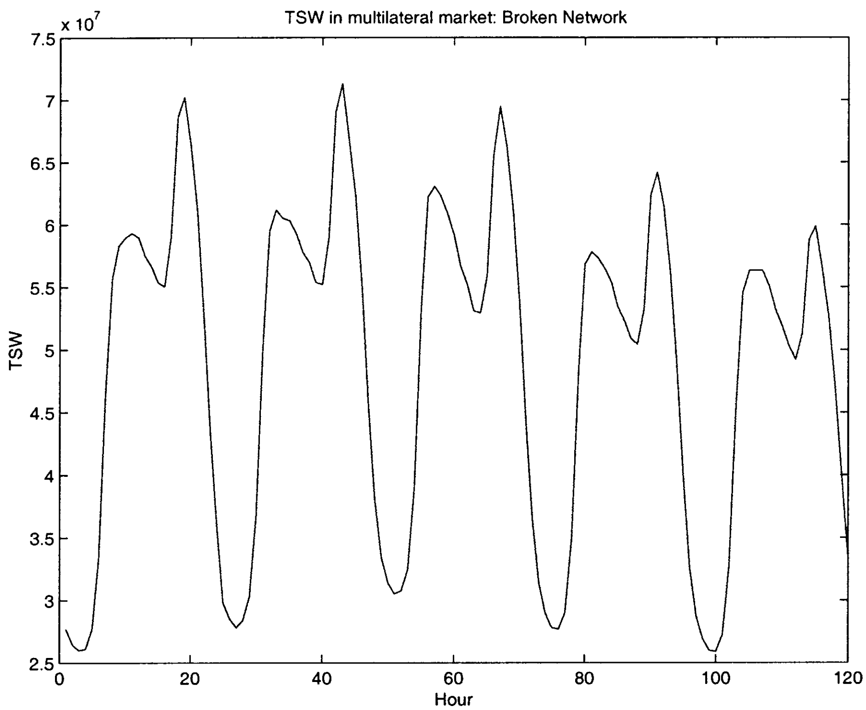

Figure 6-14: Total Social Welfare: Broken Network, 700 MW

Chapter 7

TSP Profits using Dynamic Programming

This chapter is a combination of the concepts in the previous chapters to finally introduce a market structure consisting of multilateral agreements and a varying spot market for the original problem statement. Examining the anticipated load graph in Figure 1-5, and the analysis of chapter 6 and section 2.2, we see that this formulation is well suited for a dynamic programming limited lookahead approach. We again show here in Fig. 7-1 the expected quantities of demand for the multilateral market. More specifically, we see that we can split the computationally infeasible problem of optimizing over 120 time periods, into 15 smaller subproblems of smaller horizons each, as shown in Fig. 5-1. Thus, we exploit the fact that these "near-flat" areas where we assumed bilateral agreements will exist, are the same periods to run the smaller DP over.[1] Furthermore, we know from Chapter 6 the exact deterministic profits that can be made in the trilateral marketplace for a variety of line constraints.

This approach is also well suited for the analysis done in Chapter 4, where the quantities on the spot market are approximated by stochastic distributions, and hence

[1]Note: The transition areas between night and day, and evening and night are not well suited for this BA+spot type splitting. As described in chapter 4, and from Fig: 1-4, we see no optimization exists for hours 6,7,22,23 and 24. This explains why the graphs have a maximum of 95 on the x-axis, as these five times per day were not included. If exact results are required as to what profits were made during these time periods, one should use the same values given in Chapter 6.

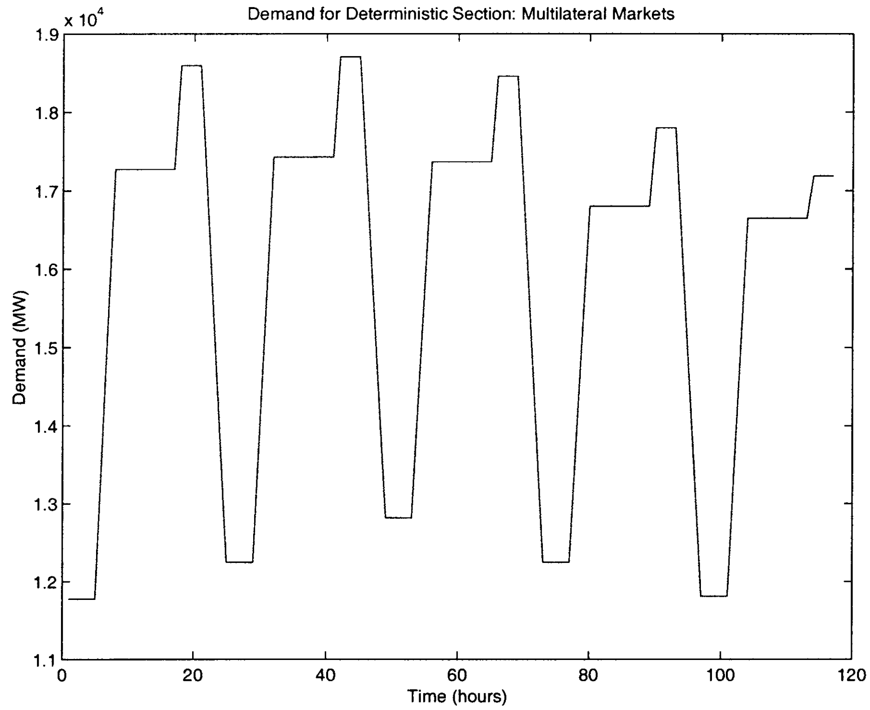

Figure 7-1: Anticipated Demand in Multilateral Market

the spot market profits can be adequately solved, by the elementary method described in section 4.2. The spot demand is shown again in Fig. 7-2. The distribution is independent of the day and the expected profits can be calculated depending if the specific agreement is during the night, day, or evening. These spot market profits are also calculated as a function of line capacity for the specific agreement in question.

As for the multilateral agreements, we use a similar approach as chapter 6, and calculate the total profits, $\Pi_{D,\kappa_{ma}}$ for each (demand, κ_{ma}) pair. Finally, the noise variable ω_t has the appropriate distribution as discussed earlier. Combining the above terms, the TSP has all the information necessary to predict the expected total profits starting at any feasible $\kappa_{s,t}$ and completing a feasible action resulting in $\kappa_{s,t+1}$.

We also assume that the spot market quantity allocation, i.e. $K_{i\to j}^{max} \geq \kappa_s > 0$, that is some spot market must exist. Also, due to the large magnitude of numbers involved, (capacity of congested link can be as high as 1800 for evening demand levels), the simulation also assumes that decisions above must be in increments of 100 MW. In this manner the states are described by $\kappa_{s,t}$, and can range from $100 \ldots K_{i\to j}^{max}$. The

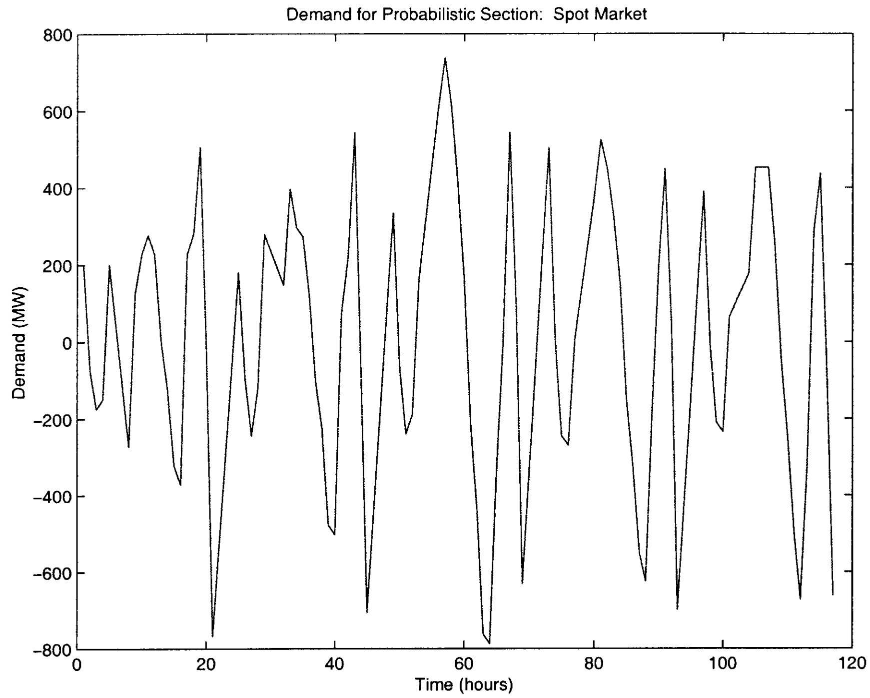

Figure 7-2: Anticipated Demand in Spot Market

action space is the quantity to decrement $\kappa_{s,t}$ by and ranges from $0 \ldots (K_{i \to j}^{max} - 100)$. The feasible actions are those that ensure $\kappa_{s,t+1} \leq \kappa_{s,t}$.

Thus, the fifteen smaller sub-problems are solved one at a time using the dynamic programming and greedy algorithms. The results are compared to a greedy approach where decisions are made for only the current time period in mind. In essence, this is equivalent to a 0-step lookahead algorithm at each period. The greedy algorithm does not take into time into account and simply performs a static optimization with the choices available at the current time. Continuous and discrete graph are shown in Figures 7-3 and 7-4.

7.1 Normal Network case

TOTAL CUMULATIVE PROFIT GAIN BY DP: 3.36 x 10^8

The value of the dynamic programming technique is thus significant.

We also want to compare this market structure to the entirely multilateral setup

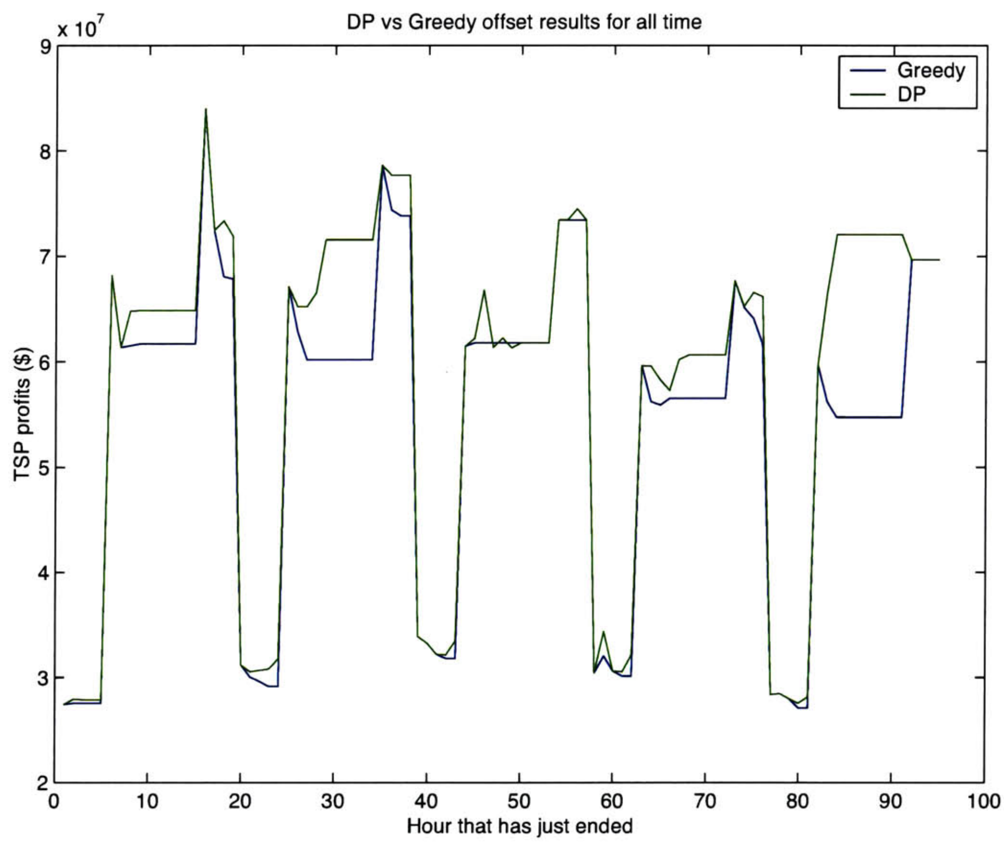

Figure 7-3: Continuous Time

in chapter 6. Reviewing Figure 6-9, we see this algorithm has nearly an order of magnitude greater profits at each time period. Thus, this provides some evidence for the need of a transmission spot market, and that a TSP can indeed profit from performing such optimizations.

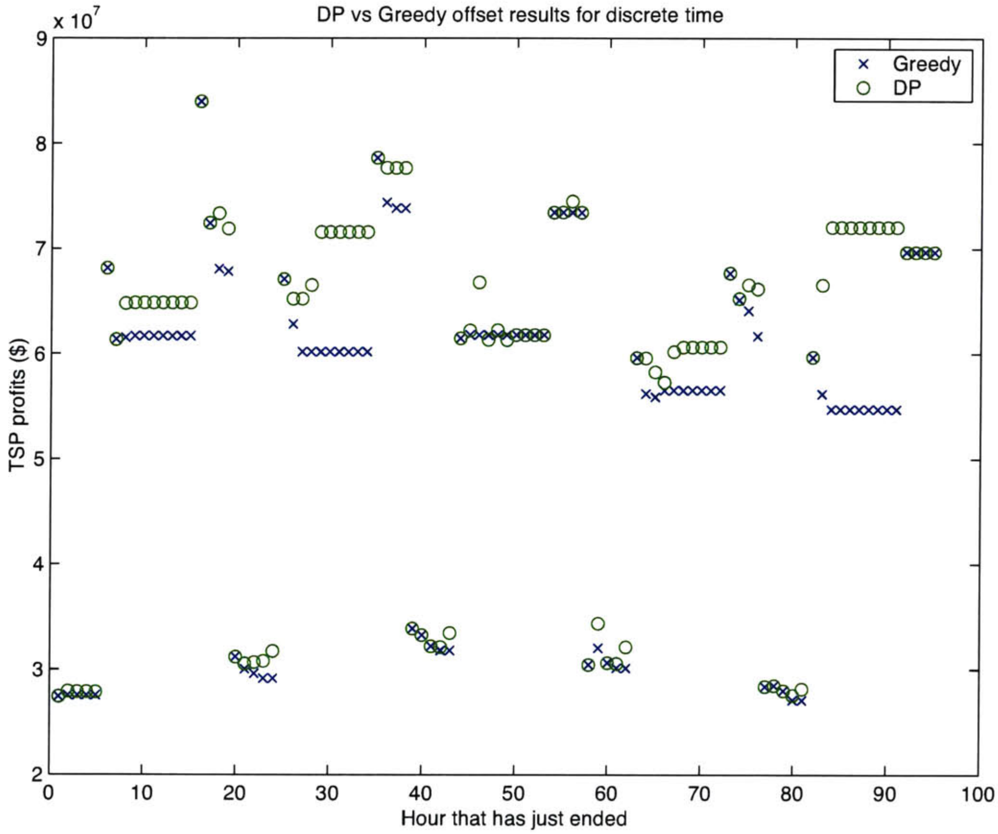

Figure 7-4: Discrete Time

7.2 Broken Network case

In this case, the same algorithm is performed for the broken network topology. The formulations are the same and the continuous time and discrete time results are presented below:

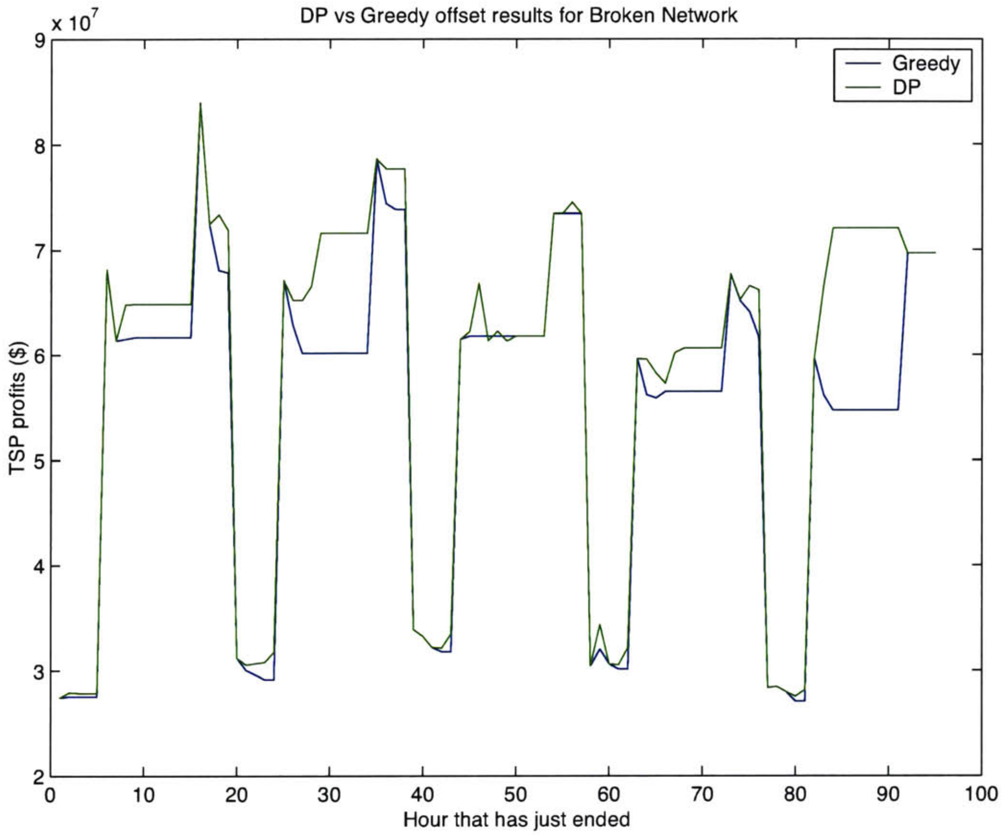

Figure 7-5: Continuous Time

TOTAL CUMULATIVE PROFIT GAIN BY DP: 1.51 x 10^8

To confirm our results in section 7.1, we again compare these results to the broken network in the entirely multilateral setup of Chapter 6. As expected, the profits to the TSP are nearly ten times higher in the market structure that includes a spot market than one that does not.

We also note that the greedy algorithm here does much better than the case of a static network topology. Indeed the total profit gained by the dynamic programming

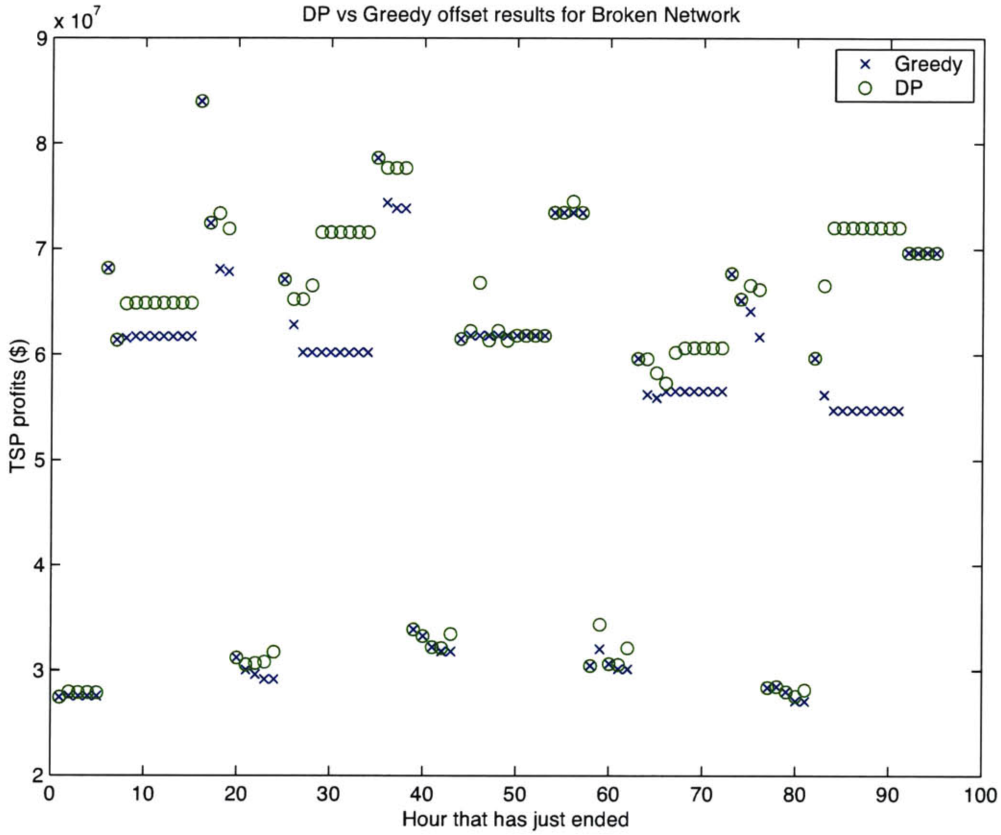

Figure 7-6: Discrete Time

algorithm is less than half of what was seen in section 7.1. This may indicate that the dynamic programming algorithm loses some value as the reliability of the network decreases.

Chapter 8

Conclusions and Future Research

This thesis has examined the resource allocation problem encountered by a transmission service provider under uncertainties. It is attempted to unbundle the electricity and transmission markets in regard to the quantities end users produce or consume. The analysis assumes the existence of multilateral forward markets as well as the real-time spot market. The TSP is able to earn revenue from both services and is attempting to maximize his revenues over a week period, when a single link in a three bus network is congested.

Real load data is taken for a Monday-Friday period in New England. Observation of the data suggests that 3 time periods exist per day (early morning, daytime and evening), where the total demand is relatively constant with low variance around the mean. Hence, the total load is split into fifteen portions of anticipated multilateral levels and expected spot values.

Spot market formulations are presented to best approximate the quantities that will be traded in real-time and determine the expected profit amount as a function of spot market capacity allocation.

Analogously, a deterministic simulation is created to automate the multilateral trading process and output the deterministic multilateral profit as a function of bilateral market capacity allocation. First, the simulation is run for a single load demand level. and results for total social welfare, end-user quantities, end-user profits, and TSP profits are shown as a function of the number of iterations.

Second, the number of iterations is fixed and the program is now run over the anticipated values of the week to simulate an entirely multilateral market structure. Results regarding total social welfare, end-user quantities, and TSP profits are computed for the week, while varying capacity level on congested link.

Third, the case of a broken topology is considered, where one of the non-congested links has such high impedance levels that flow along the link is severely reduced. We examine how an entirely multilateral trading market copes with such a case, and compare the results of total social welfare and TSP profits to those acquired previously under standard operating conditions.

Moving to a probabilistic regime, the profits gained from the deterministic multilateral setup along with statistical distributions regarding the noisy spot market are fed as inputs into a stochastic simulation. This program runs a limited lookahead dynamic programming algorithm over each period of each day for the week. These results are compared to a greedy algorithm. Total Profits to the TSP are also compared under this market structure with the entirely multilateral setup discussed earlier.

Finally, the broken topology case is examined here as well, and the resulting TSP profits are compared to those achieved when the network has all three links in operation.

8.1 Future Research

In light of the above summary, it follows that the research presented here can be extended in numerous ways. Some of the main ideas the author would have liked to develop further are presented below. This is by no means an exhaustive list since the theory, concepts, and simulations shown here can be applied to a variety of open problems in the current transmission market.

8.1.1 Investment Protocols

One next step would be to introduce the idea of time into the deterministic simulation described in chapter 5. Currently, the concept of multilateral trading does not have the "duration" the term bilateral trading generally implies. Indeed, if the exact multilateral trades accepted in one time period were necessarily also implemented in the next, the question of investment can be throughly investigated. Applying this to asset allocation markets could also be a dual benefit from such research.

8.1.2 Network Analysis

We note that the network itself presents a study for optimization of transmission. It may be appropriate to think of algorithms such as min-cost flow with the appropriate weights to enhance our notion of which links are most valuable. Furthermore, using network algorithms, it may be possible for the TSP to develop a model that appropriately weights the cost of transmission with the series of multilateral bids being requested. This can help the TSP optimize for both total social welfare and profits simultaneously.

8.1.3 Hidden Markov Models

Finally, we note that this problem may be also viewed as a hidden Markov model. The anticipated demand curve may provide more information regarding the "states" various end users are currently operating in. It is not clear whether the process is indeed Markov, but estimation techniques on the anticipated demand may provide some insight into how such states evolve. In addition, the TSP may be able to group or cluster some states together through the process of state augmentation. This could allow the TSP to computationally examine more choices for transmission, and hence choose a more optimal set of controls.

Chapter 9

Source Code

This is the matlab code used to run the simulations in chapters 5-7. First, the code is presented for the multilateral market structure or the TSP's sub-problem. This is followed by the dynamic programming code used to solve the main optimization problem. Note, this code uses a MATLAB toolbox referenced with [41]. The toolbox can be found at: http://www4.ncsu.edu/unity/users/p/pfackler/www/compecon/toolbox.html

```
The files are separated by horizontal dashed lines.  The order of file names are:

find_ggg_market_prices.m
find_lgg_market_prices.m
make_ggg_market.m
make_lgg_market.m
new_eval_markets.m
pmetric.m
tg.m
CongestedLine.m
CurtailAll.m
GeneratorInjections.m
LoadDraws.m
```

LoadUtilities.m

Price.m

flows.m

----

setup.m

dpalgo.m

```
function qp = find_ggg_market_prices(dfg,df0,df1,cfg,cf0,cf1,qgg,qg0,qg1)
% previous to calling this function, the market reached an
% unfeasible equilibrium, and the ISO curtailed loads and
% generators by the necessary amount
% this function calculates a feasible post-curtailment market
% equilibrium; the market is formed in this case by geng selling a
% total amount qggT to generators gen0 and gen1 (the proportions
% that geng sells to gen0 and gen1 are determined by the
% distribution factors on the congested line)
%
% inputs :
% dfg - distribution factor of geng on congested line
% df0 - distribution factor of gen0 on congested line
% df1 - distribution factor of gen1 on congested line
% cfg - cost function for geng : cfg(1)*q^2 + cfg(2)*q + cfg(3)
% cf0 - cost function for gen0 : cf0(1)*q^2 + cf0(2)*q + cf0(3)
% cf1 - cost function for gen1 : cf1(1)*q^2 + cf1(2)*q + cf1(3)
% qgg - quantity generated by geng post curtailment
% qg0 - quantity generated by gen0 post curtailment
% qg1 - quantity generated by gen1 post curtailment
%
%
% output : struct where
% qp.quantities = [qggT, -1*qg0T, -1*qg1T];
% qp.prices = [0. price0, price1];
% qp.profits = [eval(eval(pi_gg)), eval(eval(pi_g0)), ...
%      eval(eval(pi_g1))];
% qp.costs = [polyval(cfg,qgg+qggT), polyval(cf0, qg0+qg0T), polyval(cf1, qg1+qg1T)];

% syms for price in g-0 and g-1 markets
syms pg0 pg1;

% syms for quantities qggT, qg0T, qg1T
syms qggT qg0T qg1T;

% the trading rules
ratios = make_ggg_market(dfg,df0,df1);
r0 = ratios(1);
r1 = ratios(2);
```

```
agg = cfg(1);
bgg = cfg(2);
cgg = cfg(3);

ag0 = cf0(1);
bg0 = cf0(2);
cg0 = cf0(3);

ag1 = cf1(1);
bg1 = cf1(2);
cg1 = cf1(3);

% profit for geng
pi_gg = 'pg0*qggT*r0 + pg1*qggT*r1 + agg*qgg^2 + bgg*qgg + cgg - agg*(qgg+qggT)^2 - bgg*(qgg+qggT)-cgg';


% profit for gen0
pi_g0 = '-pg0*qg0T + ag0*qg0^2 + bg0*qg0 + cg0 - ag0*(qg0-qg0T)^2 - bg0*(qg0-qg0T)-cg0';

% profit for gen1
pi_g1 = '-pg1*qg1T + ag1*qg1^2 + bg1*qg1 + cg1 - ag1*(qg1-qg1T)^2 - bg1*(qg1-qg1T)-cg1';

% diff pi_gg for marginal profit
mpg = diff(pi_gg, 'qggT');

% solve for qggT to get supply/demand for geng
geng_sd = solve(mpg, 'qggT');

% diff gen0 for marginal profit
mp0 = diff(pi_g0, 'qg0T');

% solve mp0 for qg0T to get supply/demand for gen0
gen0_sd = solve(mp0, 'qg0T');

% solve mp0 for pg0
price0 = solve(mp0, 'pg0');
qg0T = r0*qggT;
price0 = eval(price0);

% diff gen1 for marginal profit
mp1 = diff(pi_g1, 'qg1T');

% solve mp1 for qg1T to get supply/demand for gen1
gen1_sd = solve(mp1, 'qg1T');

% solve mp1 for pg1
price1 = solve(mp1, 'pg1');
qg1T = r1*qggT;
price1 = eval(price1);

% plug price0 = pg0, price1 = pg1 into mpg, solve for qggT
pg0 = price0;
pg1 = price1;

mpg = eval(eval(mpg));
qggT = eval(solve(mpg, qggT));
qg0T = eval(qg0T);
qg1T = eval(qg1T);

price0 = eval(price0);
```

```
price1 = eval(price1);

qp.quantities = [qggT, -1*qg0T, -1*qg1T];
qp.prices = [0. price0, price1];
qp.profits = [eval(eval(pi_gg)), eval(eval(pi_g0)), ...
      eval(eval(pi_g1))];
qp.costs = [polyval(cfg,qgg+qggT), polyval(cf0, qg0+qg0T), polyval(cf1, qg1+qg1T)];


% the profit functions for this market are internal to this function,
% so we return the profit functions for each agent evaluated
% around a set of points so that we can examine it outside this function
mag = log10(abs(qggT));
points_qggT = linspace(qggT-2*10^mag, qggT+2*10^mag,25);
points_qg0T = linspace(qg0T-2*10^mag, qg0T+2*10^mag,25);
points_qg1T = linspace(qg1T-2*10^mag, qg1T+2*10^mag,25);

syms qggTT qg0TT qg1TT;

pi_gg = strrep(pi_gg, 'qggT', 'qggTT');
pi_g0 = strrep(pi_g0, 'qg0T', 'qg0TT');
pi_g1 = strrep(pi_g1, 'qg1T', 'qg1TT');

pi_gg = eval(pi_gg);
pi_g0 = eval(pi_g0);
pi_g1 = eval(pi_g1);

%qp.ppf = pi_gg;
%qp.p0f = pi_g0;
%qp.p1f = pi_g1;

for i=1:length(points_qggT)
  qggTT = points_qggT(i);
  qp.pp(i) = eval(pi_gg);
end
qp.points_pp = points_qggT;

for i=1:length(points_qg0T)
  qg0TT = points_qg0T(i);
  qp.p0(i) = eval(pi_g0);
end
qp.points_p0 = points_qg0T;

for i=1:length(points_qg1T)
  qg1TT = points_qg1T(i);
  qp.p1(i) = eval(pi_g1);
end
qp.points_p1 = points_qg1T;

------------------------------------------------------

function qp = find_lgg_market_prices(dfL,df0,df1,cfL,cf0,cf1,qL,qg0,qg1)
% previous to calling this function, the market reached an
% unfeasible equilibrium, and the ISO curtailed loads and
% generators by the necessary amount
% this function calculates a feasible post-curtailment market
% equilibrium; the market is formed in this case by load buying a
% total amount qLT from generators gen0 and gen1 (the proportions
% that the load buys from gen0 and gen1 are determined by the
% distribution factors)
%
```

```
% inputs :
% dfL - distribution factor of load on congested line
% df0 - distribution factor of gen0 on congested line
% df1 - distribution factor of gen1 on congested line
% cfL - utility function for load : cfL(1)*q^2 + cfL(2)*q + cfL(3)
% cf0 - cost function for gen0 : cf0(1)*q^2 + cf0(2)*q + cf0(3)
% cf1 - cost function for gen1 : cf1(1)*q^2 + cf1(2)*q + cf1(3)
% qL - quantity consumed by load post curtailment
% qg0 - quantity generated by gen0 post curtailment
% qg1 - quantity generated by gen1 post curtailment
%
%
% output :  struct qp where
% qp.quantities = [qLT, qg0T, qg1T];
% qp.prices = [0,price0, price1];
% qp.profits = [eval(eval(pi_L)), eval(eval(pi_g0)), ...
%       eval(eval(pi_g1))];
% qp.costs = [polyval(cfL,qL+qLT), polyval(cf0, qg0+qg0T), ...
%      polyval(cf1, qg1+qg1T)];

% Data for the 3 bus example in the E.Allen et. all paper
% cfL = [-10, 214.16667,0]
% cf0 = [1, 1, 1/2]
% cf1 = [2, 1/2, 1]
% qL = 8.9446
% qg0 = 6.0556
% qg1 = 2.8890

% syms for price in g-0 and g-1 markets
syms pg0 pg1;

% syms for quantities qLT, qg0T, qg1T
% (the T subscript is for tilda : quantities bought after curtailment)
syms qLT qg0T qg1T;

% the trading rules
ratios = make_lgg_market(dfL,df0,df1);
r0 = ratios(1);
r1 = ratios(2);

aL = cfL(1);
bL = cfL(2);
cL = cfL(3);

ag0 = cf0(1);
bg0 = cf0(2);
cg0 = cf0(3);

ag1 = cf1(1);
bg1 = cf1(2);
cg1 = cf1(3);

% profit for load
pi_L = ' - pg0*qLT*r0 - pg1*qLT*r1 - aL*qL^2 - bL*qL - cL + aL*(qL+qLT)^2 + bL*(qL+qLT)+cL';

% profit for gen0
pi_g0 = 'pg0*qg0T + ag0*qg0^2 + bg0*qg0 + cg0 - ag0*(qg0+qg0T)^2 - bg0*(qg0+qg0T)-cg0';

% profit for gen1
pi_g1 = 'pg1*qg1T + ag1*qg1^2 + bg1*qg1 + cg1 - ag1*(qg1+qg1T)^2 - bg1*(qg1+qg1T)-cg1';
```

```
% diff load for marginal profit
mpL = diff(pi_L, 'qLT');

% solve for qgL to get supply/demand for load
load_sd = solve(mpL, 'qLT');

% diff gen0 for marginal profit
mp0 = diff(pi_g0, 'qg0T');

% solve mp0 for qg0T to get supply/demand for gen0
gen0_sd = solve(mp0, 'qg0T');

% solve mp0 for pg0
price0 = solve(mp0, 'pg0');
qg0T = r0*qLT;
price0 = eval(price0);

% diff gen1 for marginal profit
mp1 = diff(pi_g1, 'qg1T');

% solve mp1 for qg1T to get supply/demand for gen1
gen1_sd = solve(mp1, 'qg1T');

% solve mp1 for pg1
price1 = solve(mp1, 'pg1');
qg1T = r1*qLT;
price1 = eval(price1);

% plug price0 = pg0, price1 = pg1 into mpg, solve for qLT
pg0 = price0;
pg1 = price1;

mpL = eval(eval(mpL));

qLT = eval( solve(mpL, 'qLT') );
qg0T = eval(qg0T);
qg1T = eval(qg1T);

price0 = eval(price0);
price1 = eval(price1);

qp.quantities = [qLT, qg0T, qg1T];
qp.prices = [0,price0, price1];
qp.profits = [eval(eval(pi_L)), eval(eval(pi_g0)), ...
      eval(eval(pi_g1))];
qp.costs = [polyval(cfL,qL+qLT), polyval(cf0, qg0+qg0T), ...
      polyval(cf1, qg1+qg1T)];

% the profit functions for this market are internal to this function,
% so we return the profit functions for each agent evaluated
% around a set of points so that we can examine it outside this function
mag = log10(abs(qLT));
points_qLT = linspace(qLT-3*10^mag, qLT+3*10^mag,25);
points_qg0T = linspace(qg0T-3*10^mag, qg0T+3*10^mag,25);
points_qg1T = linspace(qg1T-3*10^mag, qg1T+3*10^mag,25);

syms qLTT qg0TT qg1TT;

pi_L = strrep(pi_L, 'qLT', 'qLTT');
pi_g0 = strrep(pi_g0, 'qg0T', 'qg0TT');
pi_g1 = strrep(pi_g1, 'qg1T', 'qg1TT');
```

```
pi_L = eval(pi_L);
pi_g0 = eval(pi_g0);
pi_g1 = eval(pi_g1);

for i=1:length(points_qLT)
  qLTT = points_qLT(i);
  qp.pp(i) = eval(pi_L);
end
qp.points_pp = points_qLT;

for i=1:length(points_qg0T)
  qg0TT = points_qg0T(i);
  qp.p0(i) = eval(pi_g0);
end
qp.points_p0 = points_qg0T;

for i=1:length(points_qg1T)
  qg1TT = points_qg1T(i);
  qp.p1(i) = eval(pi_g1);
end
qp.points_p1 = points_qg1T;
```

```
function ratios = make_ggg_market(dfg, df0, df1)
% denote by geng, gen0, and gen1 the generators
% that have distribution factors dfg, df0, and df1
% respectively on the congested line
% geng wants to sell a total of one unit :
% this function returns the ratios [r0, r1]
% that geng must sell to gen0 and gen1 in order
% to keep the flow on the congested line the same
% (this function ensures r0 + r1 = 1)
% (ri < 0 means geng buys from geni)

% we are just solving AX = B:
A = [1,1;df0-dfg, df1-dfg];
B = [1;0];
ratios = A \ B;
```

```
function ratios = make_lgg_market(dfl, df0, df1)
% denote by load, gen0, and gen1 the load and generators
% that have distribution factors dfl, df0, and df1
% respectively on the congested line
% load wants to buy a total of one unit :
% this function returns the ratios [r0, r1]
% that load must buy from gen0 and gen1 in order
% to keep the flow on the congested line the same
% (this function ensures r0 + r1 = 1)
% (ri < 0 means load sells to geni)

% we are just solving AX = B:
A = [1,1;df0+dfl, df1+dfl];
B = [1;0];
```

```
ratios = A \ B;

---------------------------------------------


function best = new_eval_markets(cl, dfg, dfL, Cfg, Ufl, Qgc, Qlc, Gc)
% evaluates all of the possible bilateral exchanges and
% returns the 'most' best. the congested line is cl (:= 1 | 2 | 3)
% ('most' is an integer local variable defined inside this
% function)
%
% positive flow is defined in the counterclockwise direction
% line 1 : g3 -> g1
% line 2 : g1 -> g2
% line 3 : g2 -> g3

% distribution factors for generators
% dfg(3*3) : dfg(i,j) is effect on line i by generator j's production

% distribution factors for loads :
% dfL(3*3) : dfL(i,j) is effect on line i by load j's consumption

% cost function for each generator : cfgi is a 2nd degree
% polynomial with decreasing coefficients : cfgi = [a_cfgi, b_cfgi, c_cfgi]

% utility function for each load : ufli is a 2nd degree
% polynomial with decreasing coefficients : ufli = [a_cfgi, b_cfgi, c_cfgi]


% post curtailment quantities for generators (Qgc) and loads (Qlc)

index = 1;
most = 3;

total_profits = zeros(most,1);
tsf_before = total_social_welfare(Cfg, Qgc, Ufl, Qlc);

% evaluate the possible (gen) <-> (gen, gen) exchanges :
% g1 <-> g2, g3
% g2 <-> g1, g3
% g3 <-> g1, g2
for i = 1:3
  if(i+1 > 3)
    j = i - 1;
  else
    j = i + 1;
  end
  k = 6 - (i+j);
  market_name = ggg_id(i,j, k);
  struct = find_ggg_market_prices(dfg(cl,i), dfg(cl,j), dfg(cl,k), ...
      Cfg(i,:), Cfg(j,:), Cfg(k,:), Qgc(i), Qgc(j), Qgc(k));


  % check to see if the line constraints, generator constraints are
  % satisfied.
  % only keep this agreement if it is one of the 'most' best, as
  % defined by pmetric

  pm = pmetric(struct.profits);

  tmpg(i) = struct.quantities(1);
  tmpg(j) = struct.quantities(2);
```

```
  tmpg(k) = struct.quantities(3);
  tmp_gq = Qgc + tmpg';
  check = tmp_gq(tmp_gq > Gc);
  if( length(check) > 0)
    okay = 0;
  else
    okay = 1;
  end

  tsf = total_social_welfare(Cfg, tmp_gq, Ufl, Qlc);
  tsf_before;

  % just in case, make sure that total social welfare is increasing
  better_tsf = tsf >= tsf_before;

  if(pm > min(total_profits) & okay & better_tsf)
    % keep it
    [m,minIndex] = min(total_profits);

    total_profits(minIndex) = pm;

    tsf_before;
    all_tsf(minIndex) = tsf;

    l(minIndex,1) = 0;
    l(minIndex,2) = 0;
    l(minIndex,3) = 0;
    g(minIndex,i) = struct.quantities(1);
    g(minIndex,j) = struct.quantities(2);
    g(minIndex,k) = struct.quantities(3);
    names(minIndex, :) = market_name;

    all_profits(minIndex, 1) = 0;
    all_profits(minIndex, 2) = 0;
    all_profits(minIndex, 3) = 0;
    all_profits(minIndex, 3+i) = struct.profits(1);
    all_profits(minIndex, 3+j) = struct.profits(2);
    all_profits(minIndex, 3+k) = struct.profits(3);

    pp(minIndex,:) = struct.pp;
    p0(minIndex,:) = struct.p0;
    p1(minIndex,:) = struct.p1;

    points_pp(minIndex,:) = struct.points_pp;
    points_p0(minIndex,:) = struct.points_p0;
    points_p1(minIndex,:) = struct.points_p1;

    all = [l g];

    mapping(minIndex,1) = 3 + i;
    mapping(minIndex,2) = 3 + j;
    mapping(minIndex,3) = 3 + k;

    prices(minIndex,:) = struct.prices;

    genq(minIndex,:) = Qgc' + g(minIndex, :);
    loadq(minIndex,:) = Qlc' + l(minIndex, :);

  end
  index = index + 1;
end
```

```
% evaluate the possible (load) <-> (gen, gen) exchanges :
% l1 <-> g1, g2
% l1 <-> g1, g3
% l1 <-> g2, g3
% l2 <-> g1, g2
% l2 <-> g1, g3
% l2 <-> g2, g3
% l3 <-> g1, g2
% l3 <-> g1, g3
% l3 <-> g2, g3
for i = 1:3
  for j = 1:2
    for k = j+1:3
      market_name = lgg_id(i,j,k);
      struct = find_lgg_market_prices(dfL(cl,i), dfg(cl,j), ...
  dfg(cl,k), Ufl(i,:), Cfg(j,:), Cfg(k,:), Qlc(i), Qgc(j), ...
  Qgc(k));

      % check to see if the line constraints, generator constraints are
      % satisfied.
      % only keep this agreement if it is one of the 'most' best, as
      % defined by pmetric

      pm = pmetric(struct.profits);

      tmpl(1) = 0;
      tmpl(2) = 0;
      tmpl(3) = 0;
      tmpg(1) = 0;
      tmpg(2) = 0;
      tmpg(3) = 0;
      tmpl(i) = struct.quantities(1);
      tmpg(j) = struct.quantities(2);
      tmpg(k) = struct.quantities(3);

      tmp_gq = Qgc + tmpg';
      tmp_lq = Qlc + tmpl';

      check = tmp_gq(tmp_gq > Gc);
      if( length(check) > 0)
okay = 0;
      else
okay = 1;
      end

      tsf = total_social_welfare(Cfg, tmp_gq, Ufl, tmp_lq);
      tsf_before;

      % just in case, make sure total social welfare is better than
      % it was before
      better_tsf = tsf >= tsf_before;

      if(pm > min(total_profits) & okay & (tsf >= tsf_before))

% keep it

[m,minIndex] = min(total_profits);

tsf_before;
all_tsf(minIndex) = tsf;
```

```
total_profits(minIndex) = pm;

l(minIndex,1) = 0;
l(minIndex,2) = 0;
l(minIndex,3) = 0;
l(minIndex,i) = struct.quantities(1);
g(minIndex,1) = 0;
g(minIndex,2) = 0;
g(minIndex,3) = 0;
g(minIndex,j) = struct.quantities(2);
g(minIndex,k) = struct.quantities(3);

all = [l g];

names(minIndex, :) = market_name;

pp(minIndex,:) = struct.pp;
p0(minIndex,:) = struct.p0;
p1(minIndex,:) = struct.p1;

points_pp(minIndex,:) = struct.points_pp;
points_p0(minIndex,:) = struct.points_p0;
points_p1(minIndex,:) = struct.points_p1;

all_profits(minIndex, 1) = 0;
all_profits(minIndex, 2) = 0;
all_profits(minIndex, 3) = 0;
all_profits(minIndex, i) = struct.profits(1);

all_profits(minIndex, 4) = 0;
all_profits(minIndex, 5) = 0;
all_profits(minIndex, 6) = 0;
all_profits(minIndex, 3+j) = struct.profits(2);
all_profits(minIndex, 3+k) = struct.profits(3);

mapping(minIndex,1) = i;
mapping(minIndex,2) = 3 + j;
mapping(minIndex,3) = 3 + k;

prices(minIndex,:) = struct.prices;

genq(minIndex,:) = Qgc' + g(minIndex, :);
loadq(minIndex,:) = Qlc' + l(minIndex, :);

index = index + 1;
      end

    end
  end
end


% in each of these fields, the ith row corresponds to the ith agreement

best.number = most; % the number of agreements we have saved
best.total_profits = total_profits; % the profit metric for each agreement (see pmetric.m)
best.generators = g; % additional generation by generators in each
                     % agreement (one row is the quantity for
                     % generators in an agreement)
best.loads = l; % additional consumption by loads in each agreement
```

```
                % (one row is the additional consumption for loads
                % in an agreement)
best.pp = pp; % profit as a function of quantity for the 'pivot'
              % agent in the agreement
best.p0 = p0; % profit as a function of quantity for the 'g0' agent
              % in the agreement
best.p1 = p1; % profit as a function of quantity for the 'g1' agent
              % in the agreement
best.points_pp = points_pp; % the quantities at which the profit
                            % function for the pivot is evaluated
best.points_p0 = points_p0; % the quantities at which the profit
                            % function for the 'g0' agent is evaluated
best.points_p1 = points_p1; % the quantities at which the profit
                            % for the 'g1' agent is evaluated
best.names = names; % string description of the agreement, eg : (l2<->(g1,g3))

% all_profits[i, 1:3] : profits for loads in agreement i
% all_profits[i, 4:6] : profits for generators in agreement i
best.all_profits = all_profits;

% all(i, 1:3) := additional consumption by loads 1:3 in agreement 'i'
% all(i, 4:6) := additional generation by generators 1:3 in agreement 'i'
best.all = all;

best.Qgc = genq;
best.Qlc = loadq;

best.prices = prices;

best.tsf = all_tsf;

% mapping(i, 1) is the index in 'all' where we can find the quantity
% for the pivot in the ith agreement. for example, if we are looking
% at the second agreement, and mapping(2, 1) = 3,
% then load 3 is the pivot in the 2nd agreement and all(2, 3) is
% the additional quantity consumed by load 3 in this
% agreement. (this is the same as the quantity in loads(2,3)
% similarly, mapping(i, 2) is the index in 'all' where we can find
% the quantity for the 'g0' agent in the 'ith' agreement.  if
% mapping(2,2) = 4, then generator 1 is the 'g0' agent, and
% all(2,4) is the additional quantity generated by generator 1 in
% the 2nd agreement (this is the same as the quantity
% generators(2,1))
% mapping(i,3) <-> index in 'all' where we can find the quantity
% for the 'g1' agent in the 'ith' agreement
best.mapping = mapping;

-----------------------------------------------------------------


function sumprofit = pmetric(agents)
% takes three profits (numerically)
% returns a profit metric based on
% the 'method' variable below.
% the different profit metrics are described
% in the three constants defined in this function :
% SUM_PROFITS_ALLOW_NEGATIVE = 0;
% SUM_PROFITS_CONSTRAIN_POSITIVE = 1;
% AVERAGE_PROFIT = 2;
```

```
SUM_PROFITS_ALLOW_NEGATIVE = 0;
SUM_PROFITS_CONSTRAIN_POSITIVE = 1;
AVERAGE_PROFIT = 2;


method = 2;

if(method == SUM_PROFITS_ALLOW_NEGATIVE)
  sumprofit = sum(agents);
end
if(method == SUM_PROFITS_CONSTRAIN_POSITIVE)
  agent1 = agents(1);
  agent2 = agents(2);
  agent3 = agents(3);

  negative = 0;
  ptemp = 0;

  if (agent1 < 0)
    negative = 1;
  end;

  if (agent2 < 0)
    negative = 1;
  end;

  if (agent3 < 0)
    negative = 1;
  end;

  if (negative == 1)
    ptemp = -1;
  else
    ptemp = agent1 + agent2 + agent3;
  end;

  sumprofit = ptemp;
end
if(method == AVERAGE_PROFIT)
  sumprofit = mean(agents);
end



----------------------------------------------



% begin calculations for one time period
% following two lines should be commented out for execution
  M10=[11775];
  i=1;

  % Set up Generator Cost Functions for the three generators
  CG = [.11 5 150; .085 1.2 600; .1225 1 335]

  % Set up generator constrains (Each line is min, max)
  Gk = [10, 12000; 10, 12000; 10, 12000]

  % Calculate quantities generated by each G
```

```
Qg = GeneratorInjections(M10(i), CG, Gk)

% Find the price of electricity for this time period
Pl = Price(Qg, CG)

% Set up load ratios for the three loads
Lr = [.2 .3 .5]

% Make a guess as to the loads' utility functions' 'b' coefficient
GUL = [Lr(1) 0 0; Lr(2) 0 0; Lr(3) 0 0]

% Calculate the 'a' coefficient based on above guess, and
% return a complete cost function
UL = LoadUtilities(M10(i), Pl, GUL, Lr)

% Calculate the amount drawn out by each load based on the load ratios
% (But LoadDraws should be changed to use the utility functions
Ql = LoadDraws(M10(i), Pl, Lr)

% Define the distribution factors (separate matrices for
% generators and loads

% DF's below use bus 3 as reference (4/28)
DFg = [1/3 -1/3 0; 1/3 2/3 0; -2/3 -1/3 0]; %dfg([1,1,1],3);
DFl = -1*DFg;

% Calculate the flows on each line based on generator injections,
% load draws, and distribution factors
F = flows(Qg, Ql, DFg, DFl)

% save F for future use
FF = F;
j = 0;
% pz is profit for a given capacity constraint
for j=0:10

  F = FF;

  % Define line capacity constraints
  K = [99999; 1000 - j*100; 99999];

  % Curtail injections/draws if constraints exceeded
  line = CongestedLine(F, K)
  if(line == 0)
    continue;
  end

  [Qgc, Qlc] = CurtailAll(K(line), DFg(line,:), DFl(line,:), Qg, Ql)
  Qgc_before = Qgc;
  Qlc_before = Qlc;
  Fc = flows(Qgc, Qlc, DFg, DFl)

  % loop through all combinations and test profitability
  % pick the best deal through three iterations
  numDeals=15;

  % make our polynomial notation consistent
  ULn = UL.*[-1 1 1;-1 1 1;-1 1 1];
  tsfBefore = total_social_welfare(CG, Qgc, ULn, Qlc)
  gen_constraints = Gk(:,2)
```

```
    for i=1:numDeals
      % get the 'most' best markets
      bestmkt= new_eval_markets(line, DFg, DFl, CG, ULn, Qgc, Qlc, gen_constraints);

      % keep only the best one
      [max, max_index] = max(bestmkt.total_profits);


      profits(i,:) = bestmkt.all_profits(max_index,:);
      comment = strcat('this is the ', num2str(i), 'th iteration')
      nms(i,:) =  bestmkt.names(max_index,:)
      gens(i,:) = bestmkt.generators(max_index,:)
      loads(i,:) = bestmkt.loads(max_index,:)

      % implement this market by updating the state of each node
      Qgc = Qgc + gens(i,:)';
      Qlc = Qlc + loads(i,:)';


      pz(i) = bestmkt.total_profits(max_index);
      prices(i,:) = bestmkt.prices(max_index,:);
      tsf(i) = total_social_welfare(CG, Qgc, ULn, Qlc)
    end

    for i = 1:6
      j_profits(j+1,i) = sum(profits(:,i));
    end



    clear profits;
    clear nms;
    clear gens;
    clear loads;
    clear pz;
    clear prices;
    clear tsf;
  end
```

```
function L = CongestedLine(F, K)
% given two nx1 matrices for flows (F) and constraints (K)
% checks to see if all constraints are met
% if flows exceed constraints, L = line with greatest excess
% otherwise L = 0

tempex = 0;
maxtempex = 0;
templine = 0;

for i=1:3
   % need to change for bounds to a matrix dimension parameter
   tempex = (F(i)) - K(i);
   if (tempex > 0)
```

```
      if (tempex > maxtempex)
         maxtempex = tempex;
 templine = i;
      end;
   end;
end;

L = templine;
```

```
function [Qgc, Qlc] = CurtailAll(K, DFg, DFl, Qg, Ql)
% Given a line constraint K, generation constraint matrix
% Kg, distribution factors for generators (DFg) loads (DFl), and
% generation (Qg) and load (Ql) quantities,
% returns new curtailed quantities such that the new flow on
% the line equals the constraint K

% note: the argument lists are unwieldly because matlab
% has restrictions on the arguments for objective functions
% 'QuantityDifferences' can only take a single argument 'x',
% but we also need to pass in initial values for all Qli's and
% Qgi's. So to work around the 1-arg limitation, use x as a
% matrix, with the first half of the arguments as variables,
% and the second half


% repeat the Qg and Ql to pass the pre-curtailment values
% to the objective function
x0 = [Qg Ql Qg Ql];
numBuses=max(size(Qg));
numAgents=numBuses*2;

% General identity matrix for function use
iden = eye(numAgents*2);

% Set up inequality constraints
% might be redundant with upper and lower bounds, below
A = iden(1:numAgents,:) * -1;
B = zeros(1,numAgents);

% Number of buses
Aeq0 = zeros(1,numAgents);

% Equality Constraint #1:
% Total Generation - Total Consumption = 0
Aeqtc = [ones(1,numBuses) ones(1,numBuses)*-1];

% Equality constraint #2
% Quantities * Distribution Factors = Line Constraint K
AeqT1 = [DFg DFl];

% Workaround Constraint
% the second set of Qg's and Ql's must remain the same
Aeqiden = iden(numAgents+1:numAgents*2,:);

% Concatenate all constraint into one giant A[]*x()=B[] matrix
% Pad where necessary with Aeq0's
Aeq = [Aeqtc Aeq0; AeqT1 Aeq0; Aeqiden];
```

```
Beq = [0 K Qg' Ql'];

% additional constraint: no agent produces/consumes MORE
% in post curtailment than in pre-curtailment
% n.b. some feasible solutions include having some agents produce
% or consume more than is economically legal. non-augmenting curtailing is
% always a 'safe' (i.e. everyone is still guaranteed to make
% a profit given rational decisions) strategy
ub = [Qg Ql];

% make sure no one goes below 0 either
lb = zeros(1,numAgents*2);

[x, fval] = fminimax('QuantityDifference', x0, A, B, Aeq, Beq, lb, ub);
Qgc=x(1:numBuses)';
Qlc=x(numBuses+1:numAgents)';
```

```
function Qg = GeneratorInjections (Qt, CG, Qk)
% given an overall quantity Q_t and a generator cost matrix CQ,
% and generator capacities Qk(min, max)
% this function returns the amount generated by each generator
% Q_gi.

% the generator cost functions a_gi, b_gi and c_gi can be
% modified below

ag1 = CG(1, 1);
bg1 = CG(1, 2);
cg1 = CG(1, 3);

ag2 = CG(2, 1);
bg2 = CG(2, 2);
cg2 = CG(2, 3);

ag3 = CG(3, 1);
bg3 = CG(3, 2);
cg3 = CG(3, 3);

syms q1 q2 q3;
mc(1,:) = '2 * ag1 * q1 + bg1';
mc(3,:) = '2 * ag3 * q3 + bg3';
mc(2,:) = '2 * ag2 * q2 + bg2';
totalq = 'q1 + q2 + q3 = Qt';
S = solve(eval(mc(1,:))-eval(mc(2,:)), eval(mc(2,:))-eval(mc(3,:)), totalq);

Qg = [eval(S.q1); eval(S.q2); eval(S.q3)];

% To create a constrained solution:
% use Qgimax and Qgimin

Qgmax(1)=Qk(1,2);
Qgmax(2)=Qk(2,2);
Qgmax(3)=Qk(3,2);
Qgmin(1)=Qk(1,1);
```

```
Qgmin(2)=Qk(2,1);
Qgmin(3)=Qk(3,1);

while ((Qgmax(1) < Qg(1))|(Qgmax(2) < Qg(2))|(Qgmax(3) < ...
      Qg(3))|(Qgmin(1) > Qg(1))|(Qgmin(2) > Qg(2))|(Qgmin(3) > Qg(3)))
  % loop because satisfying the first constraint can
  % violate others

  % counter for the number of constrained generators
  numGconstr = 0;

  % flag array for constrained generators
  % reset to 0 on every iteration
  for g=1:3
    Qgconstr(g) = 0;
  end;

  % determine the number of constrained generators
  for g=1:3
    if (Qg(g) >= Qgmax(g))
      numGconstr = numGconstr + 1;
      Qg(g) = Qgmax(g);
      Qgconstr(g)=1;
    end;
    if (Qg(g) < Qgmin(g))
      numGconstr = numGconstr + 1;
      Qg(g) = Qgmin(g);
      Qgconstr(g)=1;
    end;
  end;

  switch numGconstr
    case 3
      % all 3 generators at capacity
      error('Qt exceeds total production capacity');
    case 2
      % two generators at capacity,
      % third generator has market power

      % Qgremain = total constrained capacity
      % (sum of all generators who are producing at max capacity)

      Qgremain=0;

      % use side effect of true = 1
      % multiply Q(g) by the flag variable
      for g=1:3
Qgremain = Qgremain + Qg(g) * Qgconstr(g);
      end;
      for g=1:3
if (Qgconstr(g) == 0)
  Qg(g)=Qt-Qgremain;
end;
      end;
    case 1
      % one generator (gen a) exceeds capacity
      % competitive market between gens b and c (mcb = mcc)
      for g=1:3
if(Qgconstr(g) == 1)
  % Create 3rd equation: qx-qxmax=0 or qx-qxmin=0
  eqrestr = strcat('q',int2str(g),'-',int2str(Qg(g)));
```

```
end;
      end;
      mccounter = 1;
      for g=1:3
% grab the non-constrained MC's
if (Qgconstr(g) == 0)
  mct(mccounter,:) = mc(g,:);
  mccounter = mccounter + 1;
end;
      end;

      S = solve(eval(mct(1,:))-eval(mct(2,:)), totalq, eqrestr);
      Qg = [eval(S.q1); eval(S.q2); eval(S.q3)];
  end;
end;
```

```
function Ql = LoadDraws(Qt, Pl, Lr)
% Given a total quantity Qt and price Pl, and a matrix of Load Ratios Lr
% returns a matrix with loads
% to match the parameters

% parameters modified occasionally:
% ratio of the demands, the 'b' coefficient
r1 = Lr(1);
r2 = Lr(2);
r3 = Lr(3);

% calculations
Ql1 = r1 * Qt;
Ql2 = r2 * Qt;
Ql3 = r3 * Qt;
Ql = [Ql1; Ql2; Ql3]
```

```
function UL = LoadUtilities(Qt, Pl, GUL, Lr)
% Given a total quantity Qt, price Pl, a guess GUL as to the
% utility matrix of the loads, and a load ratio
% returns a matrix with load Utility Functions constrained
% to match the parameters

% parameters modified occasionally:
% ratio of the demands, the 'b' coefficient
r1 = Lr(1);
r2 = Lr(2);
r3 = Lr(3);

al1 = GUL(1,1);
al2 = GUL(2,1);
al3 = GUL(3,1);

cl1 = GUL(1,3);
cl2 = GUL(2,3);
cl3 = GUL(3,3);
```

```
% calculations
Ql1 = r1 * Qt;
bl1 = Pl + 2*al1 * Ql1;

Ql2 = r2 * Qt;
bl2 = Pl + 2*al2*Ql2;

Ql3 = r3 * Qt;
bl3 = Pl + 2*al3*Ql3;

UL = [al1 bl1 cl1; al2 bl2 cl2; al3 bl3 cl3];

% check notation for a coefficients
```

```
function Pl = Price(Qg, CG)
% given a 1x3 matrix for quantity produced and a 3x3 matrix for
% generator cost functions, this function returns the maximum
% marginal cost for any of the three generators

% normally, mc1=mc2=mc3, but if one generator is constrained, then
% set mc=max(1,2,3)

% if one generator is offline, ignore that generator

for i = 1:3
   if (Qg(i) == 0)
      for j = 1:3
         CG(i, j) = 0;
      end;
   end;
end;

ag1 = CG(1, 1)
bg1 = CG(1, 2)
cg1 = CG(1, 3)

ag2 = CG(2, 1)
bg2 = CG(2, 2)
cg2 = CG(2, 3)

ag3 = CG(3, 1)
bg3 = CG(3, 2)
cg3 = CG(3, 3)

ptemp = max((2 * ag1 * Qg(1) + bg1), (2 * ag2 * Qg(2) + bg2))
ptemp = max(ptemp,(2 * ag3 * Qg(3) + bg3))

Pl = ptemp
```

```
function F = flows(Qg, Ql, DFg, DFl)
% Given distribution factors for Generators and Loads (DFg, DFl)
% and Quantities (in nx1 matrix form for generators and loads)
% Calculates the flows on all lines
```

```
F = DFg * Qg + DFl * Ql;

----------------------------------------------------------

%% First we do all the multilateral stuff - Profits obtained from running multilateral simulation %%

%% for normal network case:

%MAProfits1 = 1e7* [2.6950 2.6934 2.6910 2.6879 2.6841 2.6796 2.6743 2.6683 2.6616 2.6541 2.6459 2.6370];
%MAProfits2 = 1e7* [5.8007 5.7992 5.7968 5.7938 5.7900 5.7855 5.8016 5.8007 5.7992 5.7968 5.7938 5.7900 5.7855 5.7232 5.
%MAProfits3 = 1e7* [6.7196 6.7170 6.7136 6.7095 6.7047 6.6992 6.6929 6.6859 6.6782 6.6697 6.6605 6.6506 6.6399 6.6285 6.
%MAProfits4 = 1e7* [2.9151 2.9138 2.9118 2.9091 2.9056 2.9014 2.8965 2.8909 2.8845 2.8774 2.8695 2.8610 2.8517]
%MAProfits5 = 1e7* [5.9051 5.9034 5.9010 5.8978 5.8939 5.8893 5.8839 5.8778 5.8710 5.8635 5.8552 5.8462 5.8365 5.8260
%MAProfits6 = 1e7* [6.8006 6.7979 6.7944 6.7902 6.7853 6.7797 6.7733 6.7662 6.7584 6.7499 6.7406 6.7306 6.7198 6.7083
%MAProfits7 = 1e7* [3.1922 3.1904 3.1880 3.1848 3.1809 3.1763 3.1709 3.1648 3.1580 3.1504 3.1421 3.1331 3.1234]
%MAProfits8 = 1e7* [5.8612 5.8596 5.8572 5.8541 5.8502 5.8457 5.8404 5.8343 5.8276 5.8201 5.8118 5.8029 5.7932 5.7828 5.
%MAProfits9 = 1e7* [6.6205 6.6180 6.6148 6.6108 6.6061 6.6006 6.5945 6.5876 6.5799 6.5716 6.5625 6.5527 6.5421 6.5308 6.51
%MAProfits10 = 1e7* [2.9151 2.9138 2.9118 2.9091 2.9056 2.9014 2.8965 2.8909 2.8845 2.8774 2.8695 2.8610 2.8517]
%MAProfits11 = 1e7* [5.4879 5.4866 5.4847 5.4820 5.4786 5.4745 5.4696 5.4641 5.4577 5.4507 5.4429 5.4344 5.4251 5.4152 5.40
%MAProfits12 = 1e7* [6.1594 6.1574 6.1546 6.1512 6.1470 6.1421 6.1364 6.1300 6.1229 6.1151 6.1065 6.0972 6.0872 6.0764 6.06
%MAProfits13 = 1e7* [2.7110 2.7094 2.7070 2.7039 2.7000 2.6954 2.6901 2.6841 2.6773 2.6698 2.6616 2.6527]
%MAProfits14 = 1e7* [5.3880 5.3862 5.3837 5.3804 5.3764 5.3716 5.3662 5.3600 5.3530 5.3454 5.3370 5.3278 5.3180 5.3074 5.29
%%MAProfits15 = 1e7* [5.7439 5.7424 5.7401 5.7371 5.7334 5.7290 5.7238 5.7179 5.7113 5.7040 5.6959 5.6870 5.6775 5.6672 5.6

%%for broken network case:

MAProfits1 = 1e7* [2.6959 2.6954 2.6948 2.6939 2.6929 2.6916 2.6901 2.6883 2.6864 2.6843 2.6819 2.6793]
MAProfits2 = 1e7* [5.8016 5.8012 5.8006 5.7998 5.7987 5.7975 5.796 5.7944 5.7925 5.7904 5.788 5.7855 5.7827 5.7798 5.7766 5.7732 5.7696 5.7657]
MAProfits3 = 1e7* [6.7223 6.7217 6.7208 6.7196 6.7183 6.7168 6.715 6.713 6.7108 6.7084 6.7058 6.703 6.6999 6.6966 6.6931 6.6894 6.6855 6.6814]
MAProfits4 = 1e7* [2.9155 2.9152 2.9147 2.9139 2.913 2.9118 2.9104 2.9088 2.9069 2.9049 2.9026 2.9002 2.8975]
MAProfits5 = 1e7* [5.9061 5.9057 5.9051 5.9042 5.9031 5.9019 5.9004 5.8987 5.8967 5.8946 5.8922 5.8897 5.8869 5.8839 5.8806 5.8772 5.8735
5.8697]
MAProfits6 = 1e7* [6.8035 6.8028 6.8019 6.8007 6.7994 6.7978 6.796 6.794 6.7918 6.7894 6.7867 6.7838 6.7808 6.7775 6.774 6.7702 6.7663 6.7621]
MAProfits7 = 1e7* [3.1931 3.1927 3.192 3.1911 3.19 3.1887 3.1872 3.1855 3.1835 3.1813 3.179 3.1764 3.1735]
MAProfits8 = 1e7* [5.8622   5.8618 5.8611 5.8603 5.8592 5.858 5.8565 5.8548 5.8529 5.8508 5.8484 5.8459 5.8431 5.8401 5.8369 5.8335 5.8298
5.826]
MAProfits9 = 1e7* [6.623 6.6223 6.6215 6.6204 6.6191 6.6176 6.6158 6.6139 6.6117 6.6094 6.6068 6.604 6.6009 6.5977 6.5942 6.5906 6.5867 6.5826]
MAProfits10 = 1e7* [2.9155 2.9152 2.9147 2.9139 2.913 2.9118 2.9104 2.9088 2.9069 2.9049 2.9026 2.9002 2.8975]
MAProfits11 = 1e7* [5.4883 5.488 5.4875 5.4868 5.4859 5.4847 5.4834 5.4818 5.48 5.478 5.4758 5.4734 5.4707 5.4679 5.4648 5.4615 5.458 5.4543]
MAProfits12 = 1e7* [6.1608 6.1603 6.1596 6.1586 6.1575 6.1561 6.1546 6.1528 6.1508 6.1485 6.1461 6.1434 6.1405 6.1375 6.1342 6.1306 6.1269
6.1229]
MAProfits13 = 1e7* [2.7119 2.7115 2.7108 2.7099 2.7089 2.7076 2.7061 2.7043 2.7024 2.7002 2.6979 2.6953]
MAProfits14 = 1e7* [5.3892 5.3887 5.3881 5.3872 5.3861 5.3847 5.3832 5.3815 5.3795 5.3773 5.3749 5.3723 5.3695 5.3664 5.3632 5.3597 5.356]
MAProfits15 = 1e7* [5.7446 5.7443 5.7437 5.7429 5.7419 5.7406 5.7392 5.7376 5.7357 5.7336 5.7313 5.7288 5.726 5.7231 5.7199 5.7165 5.7129
5.7091]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% Then we need to do all calculations to get Spot Profits %%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%DFbus1Line23 = 1/3;
%DFbus2Line23 = 2/3;



%% for broken network case
DFbus1Line23 = 1/1002;
DFbus2Line23 = 1001/1002;



LaGrange2= [(-2*.2) 0 0 0 0 0 1 DFbus1Line23;

            0 (-2*.3) 0 0 0 0 1 DFbus2Line23;

            0 0 (-2*.5) 0 0 0 1 0;

            0 0 0   -.22 0 0 -1 -DFbus1Line23;

            0 0 0      0 -.17 0 -1 -DFbus2Line23;

            0 0 0      0 0 -.245 -1 0;

            1 1 1     -1 -1 -1 0 0;

            DFbus1Line23 DFbus2Line23 0      -DFbus1Line23 -DFbus2Line23 0 0 0];




LGinverse2 = inv(LaGrange2);



valuesMon1 = zeros(8, (LineMax(1)/100 + 1));
valuesMon2 = zeros(8, (LineMax(2)/100 + 1));
valuesMon3 = zeros(8, (LineMax(3)/100 + 1));

valuesTues1 = zeros(8, (LineMax(4)/100 + 1));
valuesTues2 = zeros(8, (LineMax(5)/100 + 1));
valuesTues3 = zeros(8, (LineMax(6)/100 + 1));


valuesWed1 = zeros(8, (LineMax(7)/100 + 1));
valuesWed2 = zeros(8, (LineMax(8)/100 + 1));
valuesWed3 = zeros(8, (LineMax(9)/100 + 1));


valuesThurs1 = zeros(8, (LineMax(10)/100 + 1));
valuesThurs2 = zeros(8, (LineMax(11)/100 + 1));
valuesThurs3 = zeros(8, (LineMax(12)/100 + 1));

valuesFri1 = zeros(8, (LineMax(13)/100 + 1));
valuesFri2 = zeros(8, (LineMax(14)/100 + 1));
valuesFri3 = zeros(8, (LineMax(15)/100 + 1));
```

```
Anticipated=[
11775

17273

18594

12245

17428

18706

12815

17363

18456

12245

16800

17800

11810

16648

17188
]

LineMax=[
1200
1800
1800
1300
1800
1800
1300
1800
1800
1300
1800
1800
1200
1700
1800
]

%%%%%%%%%% Need loop to get Spot Prices for each Bilateral Agreement  %%%%%%%%%%%%%%
%% holding agreement, calculate by iterating over capacities %%%%%%%%%%%%%%

%%%%%% mon

for lineCapacity=0:100:LineMax(1)
    valuesMon1(:, (lineCapacity/100+1))= [-1.7559e3; -2.9334e3; -6.7014e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues1 = LGinverse2*valuesMon1;
end
spotprices1=opfvalues1(8,:)
```

```
for lineCapacity=0:100:LineMax(2)
    valuesMon2(:, (lineCapacity/100+1))= [-2.5749e3; -4.3024e3; -9.8304e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues2 = LGinverse2*valuesMon2;
end
spotprices2=opfvalues2(8,:)

for lineCapacity=0:100:LineMax(3)
    valuesMon3(:, (lineCapacity/100+1))= [-.2769e4; -.4626e4; -1.0570e4; 5; 1.2; 1; 0; lineCapacity];
    opfvalues3 = LGinverse2*valuesMon3;
end
spotprices3=opfvalues3(8,:)

%%%%%%%%%% tues

for lineCapacity=0:100:LineMax(4)
    valuesTues1(:, (lineCapacity/100+1))= [-1.7894e3; -2.9894e3; -6.8294e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues4 = LGinverse2*valuesTues1;
end
spotprices4=opfvalues4(8,:)

for lineCapacity=0:100:LineMax(5)
    valuesTues2(:, (lineCapacity/100+1))= [-2.5824e3; -4.3149e3; -9.8589e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues5 = LGinverse2*valuesTues2;
end
spotprices5=opfvalues5(8,:)

for lineCapacity=0:100:LineMax(6)
    valuesTues3(:, (lineCapacity/100+1))= [-.2780e4; -.4645e4; -1.0613e4; 5; 1.2; 1; 0; lineCapacity];
    opfvalues6 = LGinverse2*valuesTues3;
end
spotprices6=opfvalues6(8,:)

%%%%%%%%%% wed


for lineCapacity=0:100:LineMax(7)
    valuesWed1(:, (lineCapacity/100+1))= [-1.8750e3; -3.1325e3; -7.1565e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues7 = LGinverse2*valuesWed1;
end
spotprices7=opfvalues7(8,:)


for lineCapacity=0:100:LineMax(8)
    valuesWed2(:, (lineCapacity/100+1))= [-2.5563e3; -4.2713e3; -9.7593e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues8 = LGinverse2*valuesWed2;
end
spotprices8=opfvalues8(8,:)

for lineCapacity=0:100:LineMax(9)
    valuesWed3(:, (lineCapacity/100+1))= [-.2765e4; -.4620e4; -1.0556e4; 5; 1.2; 1; 0; lineCapacity];
    opfvalues9 = LGinverse2*valuesWed3;
end
spotprices9=opfvalues9(8,:)


%%% thurs


for lineCapacity=0:100:LineMax(10)
    valuesThurs1(:, (lineCapacity/100+1))= [-1.7894e3; -2.9894e3; -6.8294e3; 5; 1.2; 1; 0; lineCapacity];
```

```
    opfvalues10 = LGinverse2*valuesThurs1;
end
spotprices10=opfvalues10(8,:)

for lineCapacity=0:100:LineMax(11)
    valuesThurs2(:, (lineCapacity/100+1))= [-2.4819e3; -4.1469e3; -9.4749e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues11 = LGinverse2*valuesThurs2;
end
spotprices11=opfvalues11(8,:)

for lineCapacity=0:100:LineMax(12)
    valuesThurs3(:, (lineCapacity/100+1))= [-.2661e4; -.4446e4; -1.0158e4; 5; 1.2; 1; 0; lineCapacity];
    opfvalues12 = LGinverse2*valuesThurs3;
end
spotprices12=opfvalues12(8,:)


%%%%%%%%% fri

for lineCapacity=0:100:LineMax(13)
    valuesFri1(:, (lineCapacity/100+1))= [-1.7298e3; -2.8898e3; -6.6018e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues13 = LGinverse2*valuesFri1;
end
spotprices13=opfvalues13(8,:)

for lineCapacity=0:100:LineMax(14)
    valuesFri2(:, (lineCapacity/100+1))= [-2.4744e3; -4.1334e3; -9.4464e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues14 = LGinverse2*valuesFri2;
end
spotprices14=opfvalues14(8,:)

for lineCapacity=0:100:LineMax(15)
    valuesFri3(:, (lineCapacity/100+1))= [-2.5526e3; -4.2651e3; -9.7451e3; 5; 1.2; 1; 0; lineCapacity];
    opfvalues15 = LGinverse2*valuesFri3;
end
spotprices15=opfvalues15(8,:)


%% don't consider event of spot = 0:

realisticSpotPrices1=spotprices1(2:(LineMax(1)/100+1));
realisticSpotPrices2=spotprices2(2:(LineMax(2)/100+1));
realisticSpotPrices3=spotprices3(2:(LineMax(3)/100+1));
realisticSpotPrices4=spotprices4(2:(LineMax(4)/100+1));
realisticSpotPrices5=spotprices5(2:(LineMax(5)/100+1));
realisticSpotPrices6=spotprices6(2:(LineMax(6)/100+1));
realisticSpotPrices7=spotprices7(2:(LineMax(7)/100+1));
realisticSpotPrices8=spotprices8(2:(LineMax(8)/100+1));
realisticSpotPrices9=spotprices9(2:(LineMax(9)/100+1));
realisticSpotPrices10=spotprices10(2:(LineMax(10)/100+1));
realisticSpotPrices11=spotprices11(2:(LineMax(11)/100+1));
realisticSpotPrices12=spotprices12(2:(LineMax(12)/100+1));
realisticSpotPrices13=spotprices13(2:(LineMax(13)/100+1));
realisticSpotPrices14=spotprices14(2:(LineMax(14)/100+1));
realisticSpotPrices15=spotprices15(2:(LineMax(15)/100+1));



%%%%% Now need Spot Quantities
```

```
%% Mon %%%%%%%%%%%%%%%%%%%%%%%%%%
%% Distribution changes depending on time period, M= morning, D=daytime, E=evening

  rM=random('Normal', 11844.766/100, 483.937/100,1,1);
  rD=random('Normal', 16492.071/100, 800.535/100,1,1);
  rE=random('Normal', 17488.861/100, 857.513/100,1,1);



for index=100:100:LineMax(1)
   realisticSpotQuantities1(index/100)=index;
       realisticNoisySpotQuantities1(index/100)=rM*index/100;
end



for index=100:100:LineMax(2)
    realisticSpotQuantities2(index/100)=index;

    realisticNoisySpotQuantities2(index/100)=rD*index/100;
end



for index=100:100:LineMax(3)
    realisticSpotQuantities3(index/100)=index;

    realisticNoisySpotQuantities3(index/100)=rE*index/100;
end



%% Tues %%%%%%%%%%%%%%%%%%%%%%%%%%
rM=random('Normal', 11844.766/100, 483.937/100,1,1);
  rD=random('Normal', 16492.071/100, 800.535/100,1,1);
  rE=random('Normal', 17488.861/100, 857.513/100,1,1);


for index=100:100:LineMax(4)
    realisticSpotQuantities4(index/100)=index;


    realisticNoisySpotQuantities4(index/100)=rM*index/100;
end




for index=100:100:LineMax(5)
    realisticSpotQuantities5(index/100)=index;

    realisticNoisySpotQuantities5(index/100)=rD*index/100;
end



for index=100:100:LineMax(6)
    realisticSpotQuantities6(index/100)=index;

    realisticNoisySpotQuantities6(index/100)=rE*index/100;
```

```
end

%%  Wed %%%%%%%%%%%%%%%%%%%%%%%%%%
rM=random('Normal', 11844.766/100, 483.937/100,1,1);
  rD=random('Normal', 16492.071/100, 800.535/100,1,1);
  rE=random('Normal', 17488.861/100, 857.513/100,1,1);

for index=100:100:LineMax(7)
    realisticSpotQuantities7(index/100)=index;


    realisticNoisySpotQuantities7(index/100)=rM*index/100;
end




for index=100:100:LineMax(8)
    realisticSpotQuantities8(index/100)=index;


    realisticNoisySpotQuantities8(index/100)=rD*index/100;
end




for index=100:100:LineMax(9)
    realisticSpotQuantities9(index/100)=index;

    realisticNoisySpotQuantities9(index/100)=rE*index/100;
end



%%  Thurs %%%%%%%%%%%%%%%%%%%%%%%%%%
rM=random('Normal', 11844.766/100, 483.937/100,1,1);
  rD=random('Normal', 16492.071/100, 800.535/100,1,1);
  rE=random('Normal', 17488.861/100, 857.513/100,1,1);


for index=100:100:LineMax(10)
    realisticSpotQuantities10(index/100)=index;

    realisticNoisySpotQuantities10(index/100)=rM*index/100;
end



for index=100:100:LineMax(11)
    realisticSpotQuantities11(index/100)=index;

    realisticNoisySpotQuantities11(index/100)=rD*index/100;
end




for index=100:100:LineMax(12)
    realisticSpotQuantities12(index/100)=index;
```

```
    realisticNoisySpotQuantities12(index/100)=rE*index/100;
end


%%  Fri %%%%%%%%%%%%%%%%%%%%%%%%%
rM=random('Normal', 11844.766/100, 483.937/100,1,1);
  rD=random('Normal', 16492.071/100, 800.535/100,1,1);
  rE=random('Normal', 17488.861/100, 857.513/100,1,1);


for index=100:100:LineMax(13)
    realisticSpotQuantities13(index/100)=index;

    realisticNoisySpotQuantities13(index/100)=rM*index/100;
end



for index=100:100:LineMax(14)
    realisticSpotQuantities14(index/100)=index;

    realisticNoisySpotQuantities14(index/100)=rD*index/100;
end



for index=100:100:LineMax(15)
    realisticSpotQuantities15(index/100)=index;

    realisticNoisySpotQuantities15(index/100)=rE*index/100;
end





%% Now we must calculate SpotProfits as Prices*Quantites for each agreement:

%%%%%%%%%%%%



realisticSpotProfits1=diag(realisticNoisySpotQuantities1'*realisticSpotPrices1);
realisticSpotProfits2=diag(realisticNoisySpotQuantities2'*realisticSpotPrices2);
realisticSpotProfits3=diag(realisticNoisySpotQuantities3'*realisticSpotPrices3);
realisticSpotProfits4=diag(realisticNoisySpotQuantities4'*realisticSpotPrices4);
realisticSpotProfits5=diag(realisticNoisySpotQuantities5'*realisticSpotPrices5);
realisticSpotProfits6=diag(realisticNoisySpotQuantities6'*realisticSpotPrices6);
realisticSpotProfits7=diag(realisticNoisySpotQuantities7'*realisticSpotPrices7);
realisticSpotProfits8=diag(realisticNoisySpotQuantities8'*realisticSpotPrices8);
realisticSpotProfits9=diag(realisticNoisySpotQuantities9'*realisticSpotPrices9);
realisticSpotProfits10=diag(realisticNoisySpotQuantities10'*realisticSpotPrices10);
realisticSpotProfits11=diag(realisticNoisySpotQuantities11'*realisticSpotPrices11);
realisticSpotProfits12=diag(realisticNoisySpotQuantities12'*realisticSpotPrices12);
realisticSpotProfits13=diag(realisticNoisySpotQuantities13'*realisticSpotPrices13);
realisticSpotProfits14=diag(realisticNoisySpotQuantities14'*realisticSpotPrices14);
realisticSpotProfits15=diag(realisticNoisySpotQuantities15'*realisticSpotPrices15);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% DONE WITH SPOT PROFITS

%%%%%%%%%%%%%%%%%%



LineMax=[
1200
1800
1800
1300
1800
1800
1300
1800
1800
1300
1800
1800
1200
1700
1800
]


%%%%%%%%%  DP STUFF %%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



S=(100:100:LineMax(15))';
X=(0:100:LineMax(15)-100)';
n=length(S);
m=length(X);

%% low          horizon = 5
%% medium       horizon = 10
%% high         horizon = 4


N=4;  %% horizon

%% reward matrix

f = zeros(n,m);
for i=1:n           %% states (how much is currently left for spot)
    for k=1:m       %% actions (how much to decrease spot amount - implement more BA's)
        if X(k)<S(i)
```

```
            f(i,k) = (MAProfits15((S(i)- X(k))/100)/1)+(realisticSpotProfits15((S(i)- X(k))/100)*(randn));
        else
        f(i,k) = -inf;
        end
    end
end

g = zeros(n,m);
for i=1:n
    for k=1:m
        snext = S(i)-X(k);
        g(i,k) = getindex(snext,S);
    end
end




model.reward = f;
model.transfunc = g;
model.discount = 1.0;
model.horizon=N;

[v,x,pstar] = ddpsolve(model);


-----------------------------------------------------------------------
```

Bibliography

[1] Ozge Nadia Gozum, *Decision Tools for Electricity Transmission Service and Pricing: A Dynamic Programming Approach,* MIT M.Eng. Thesis for Electrical Engineering and Computer Science, 2001.

[2] Marija Ilic and John Zaborsky, *Dynamics and Control of Large Electric Power Systems,* New York, NY: Wiley, 2000.

[3] Richard Bronson and Govindasami Naadimuthu, *Schaum's Outline to Operations Research,* New York, NY: McGraw-Hill, 1997.

[4] Marija Ilic, Francisco Galiana and Lester Fink (editors), *Power Systems Restructuring,* Boston, MA: Kluwer, 1998.

[5] Eric Allen, Marija Ilic and Ziad Younes, *Providing for Transmission in times of scarcity: An ISO cannot do it all,* Electrical Power and Energy Systems no. 21, pp. 147-163, 1999.

[6] Dimitri Bertsekas, *Dynamic Programming and Optimal Control, Volume I,* Belmont, MA: Athena Scientific, 2000.

[7] Dimitri Bertsekas and John Tsitsiklis, *Neuro-Dynamic Programming,* Belmont, MA: Athena Scientific, 1996.

[8] Dimitri Bertsekas and David Castanon, *Rollout Algorithms for Stochastic Scheduling Problems,* Journal of Heuristics, no. 5, pp. 89-108, 1999.

[9] Santosh Raikar and Marija Ilic, *Assessment of Transmission Congestion for Major Electricity Markets in the US,* MIT Energy Laboratory, Publication MIT EL 01-009WP Feb 2001.

[10] Steven Stoft, *How Financial Transmission Rights Curb Market Power,* June 1997.

[11] James Bushnell, *Transmission Rights and Market Power,* Discussion Draft 1998.

[12] Stephen Rassenti, Vernon Smith and Bart Wilson *Demand-Side Bidding Will control Market Power and Decrease the Level and Volatility of Prices,* Economic Science Laboratory, Feb 2001.

[13] Marija Ilic, Jose Arce, Yong Yoon, and Elena Fumagalli, *Assessing Reliability as the Electric Power Industry Restructures,* The Electricity Journal pp. 55-67, March 2001.

[14] Richard O'Neill, Benjamin Hobbs, William Stewart, and Michael Rothkopf, *The Joint Energy and Transmission Rights Auction,* Working Paper July 2001.

[15] Santosh Raikar and Marija Ilic, *Interruptible Physical Transmission: Contracts for Congestion Management,* MIT Energy Laboratory, Publication MIT EL 01-010WP Feb 2001.

[16] Hung-po Chao, Stephen Peck, Shmuel Oren, and Robert Wilson *Flow-Based Transmission Rights and Congestion Management,* The Electricity Journal pp. 38-58, Oct 2000.

[17] Hung-po Chao and Stephen Peck, *A Market Mechanism for Electric Power Transmmission,* Journal of Regulatory Economics, vol 10, pp. 25-59, 1996.

[18] William Hogan, *Contract Networks for Electric Power Transmission,* Journal of Regulatory Economics, vol 4, no.3, pp. 211-242, 1992.

[19] P. Joskow and J. Tirole *Transmission Rights and Market Power on Electric Power Networks. I: Physical Rights,* Mimeo, MIT and EDEI 1998.

[20] P. Joskow and J. Tirole *Transmission Rights and Market Power on Electric Power Networks. II: Physical Rights,* Mimeo, MIT and EDEI 1998.

[21] Shmuel Oren, *Transmission Pricing and Congestion Management: Efficiency , Simplicity and Open Access,* Working Paper 1998.

[22] Pierre-Francois Quet, Jose Cruz, and Ali Keyhani, *Analysis of Coordinated Multilateral Trades,* 33rd International Conference on System Sciences, Hawaii, Proceedings, IEEE, 2000.

[23] D. Sun, K. Demaree and B. Brewer, *Application and Adaptation of Newton for Optimal Power Flow,* IEEE, vol 5, no. 90, pp. 1200-1214, 1990.

[24] William Hogan, *Flowgate Rights and Wrongs,* 2000.

[25] Felix Wu and Pravin Varaiya, Pablo Spiller and Shmuel Oren, Folk theorems on transmission access: proofs and counterexamples POWER report, PWR-023, University of California Energy Institute Oct 1994.

[26] Felix Wu and Pravin Varaiya, *Coordinated multilateral trades for electric power networks: theory and implementation* POWER report, PWR-031, University of California Energy Institute, June 1995.

[27] Steven Stoft, *Congestion Pricing with Fewer Prices than Zones,* Electricity Journal, pp. 23-31, May 1998.

[28] V. Petrov, C. Richter, and G. Sheble, *Gaming Strategies on Electric Power Markets Driven by Agents,* 1999.

[29] Paul Fackler and Mario Miranda, *Hybrid Methods for Continuous State Dynamic Programming,* 1999.

[30] Alan Oppenheim, Alan Willsky and Hamid Nawab, *Signals and Systems,* Upper Saddle River, NJ: Prentice Hall, 1997.

[31] Monson Hayes, *Statistical Digital Signal Processing and Modeling,* New York, NY: Wiley, 1996.

[32] Sheldon Ross, *Introduction to Stochastic Dynamic Programming,* New York, NY: Academic Press, 1983.

[33] D. White, *Finite Dynamic Programming,* New York, NY: Wiley, 1978.

[34] A. Balakrishnan, *Introduction to Random Processes in Engineering,* New York, NY: Wiley, 1995.

[35] John Bather, *Decision Theory,* New York, NY: Wiley, 2000.

[36] Ravindra Ahuja, Thomas Magnanti, and James Orlin, *Network Flows* Upper Saddle River, NJ: Prentice Hall, 1993.

[37] Earl Grinols, *Microeconomics,* Boston, MA: Houghton Mifflin, 1994.

[38] Sheldon Ross, *Stochastic Processes,* New York, NY: Wiley, 1996.

[39] Petter Skantze and Marija Ilic, *Valuation, Hedging and Speculation in Competitive Electricity Markets,* Boston, MA: Kluwer, 2001.

[40] Sidney Resnick, *Adventures in Stochastic Processes,* Boston, MA: Birkhauser, 1994.

[41] Mario Miranda and Paul Fackler, *Applied Computational Economics and Finance,* MIT Press, 2002.