ALTERNATIVE IMPLEMENTATIONS OF

EXPANDING ALGORITHM FOR

MULTI-COMMODITY SPATIAL PRICE EQUILIBRIUM

THESIS

Yu-Shen Ke, B.S.

Major, ROCAF, Taiwan

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/97M-12

ALTERNATIVE IMPLEMENTATIONS OF
EXPANDING ALGORITHM FOR
MULTI-COMMODITY SPATIAL PRICE EQUILIBRIUM

THESIS

Yu-Shen Ke, B.S.

Major, ROCAF, Taiwan

AFIT/GOR/ENS/97M-12

THESIS APPROVAL

Student: Yu-Shen Ke, Maj. ROCAF   Class: GOR-97M

Title: Alternative Implementations of Expanding Algorithm for Multi-Commodity Spatial Price Equilibrium

Defense Date: 4 March 1997

Committee:	Name/Title/Department	Signature
Advisor	YuPo Chan Professor Department of Operational Sciences	
Reader	M. Pachter Professor Department of Electrical and Computer Engineering	

AFIT/GOR/ENS/97M-12

ALTERNATIVE IMPLEMENTATIONS OF EXPANDING ALGORITHM FOR MULTI-COMMODITY SPATIAL PRICE EQUILIBRIUM

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operation Research

Yu-Shen Ke, B.S.

Major, ROCAF, Taiwan

March 1997

Preface

The objective of my research was to develop an algorithm to solve the general multi-commodity spatial price equilibrium problem for three different models; perfect competition, monopoly, and oligopoly. To reach the goal of this thesis effort, mathematical software, MATHCAD, and operational research software, GINO, were used.

I owe many thanks to my thesis advisor, Dr.YuPo Chan, not only for his patient supervising of this research,but also for his consideration. Without his help, I could not finish my research. I am very lucky to have the best committee in AFIT; Dr. Pachter, and Professor Reynolds. They not only provided several significant suggestions and comments, but also gave me almost complete freedom over the research and confidence.

To my father, words cannot express how much I miss you. I would be nothing without you. You are always in my mind and I miss you so badly. To my mother, all I want to say is thanks for everything you have given me. I only hope that I can love half as much as you and take care of you forever.

To my wife Lin, Hui-Hung (Julia), as one road ends and another begins, I thank God that I have had and will have you to walk them with. I thank my best friend,Sheu,Ding-Yuan (Steven) for everything. I thank God for giving me the strength and endurance to complete my thesis and bringing me most kind sisters and brothers. Finally, I want to extend my gratitude to the Hwang family for their support, for making me relax when I needed it most. I want to thank all my dear friends in AFIT, especially my American mom, Ms. Robb. I also want to thank all

my friends in Taiwan for their prayers and concern,especially my dear teacher Mr. Ru.

Ke, Yu-Shen

Table of Contents

List of figures

Figure Page

List of tables

AFIT/GOR/ENS/97M-12

Abstract

This thesis presents an algorithm based on the expanding algorithm (Jones[44]) to solve spatial price equilibrium problems for three different models: perfect competition, monopoly, and oligopoly. The expanding algorithm is used to solve the linear single commodity spatial price equilibrium (LSSE) problem for perfectly competitive markets. In order to reach the goal of this thesis effort, the mathematical software MATHCAD and operational research software GINO were used. As we mentioned above, the expanding algorithm is used to solve LSSE problems, i.e., the supply function, the demand function are linear, and the shipping cost per unit is constant. In this thesis, we also consider the general multi-commodity spatial price equilibrium (GMSPE) problems with all nonlinear functions, and variable shipping cost. We also show that more commodities in total are shipped, and there is more congestion, especially in oligopoly model. That means, transportation costs have much more impact in an oligopoly than in the other two models.

ALTERNATIVE IMPLEMENTATIONS OF EXPANDING ALGORITHM FOR MULTI-COMMODITY SPATIAL PRICE EQUILIBRIUM

I. Introduction

Equilibrium is defined as a state of balance due to the equal action of opposing forces. Equilibrium problems, in contrast to optimization problems, involve competition among agents for scarce resources. For example, in general economic equilibrium problems, the agents are producers and consumers, who trade commodities so as to maximize their utility, subject to their initial endowments and a production technology, until prices are established that clear the market. In the case of congested urban transportation systems, users of a transportation network seek to determine their cost-minimizing routes of travel, until their respective path costs cannot be reduced by unilateral action.

The development of activity analysis models by Koopmans [27] and Dantzig [12] opened up a new approach to the spatial pricing and allocation problem. Samuelson [44] pointed out that there exists an objective function whose maximization guarantees fulfillment of the conditions of perfectly competitive equilibrium among spatially separated markets. Later, Takayama and Judge [50] presented two versions of the spatial pricing and allocation models : a perfectly competitive market model and a monopoly model. However, markets of most primary commodities and manufactured goods lie somewhere between these two extremes, taking on some form of oligopoly. Therefore, neither version of the Takayama-Judge model is able to provide appropriate solutions for the equilibrium conditions in the actual markets of most commodities.

1.1 Background

Since the paper by Samuelson[44] and advanced by Takayama and Judge[49, 50] , the concept of a spatial price equilibrium has found many applications. Tobin and Friesz [15] showed that the spatial price equilibrium problem on a network with transshipment may be formulated and solved without difficulty as a convex mathematical programming problem, provided all functions employed are separable (or have symmetric Jacobian matrices). Their formulation follows the tradition, beginning with Samuelson [44] and extending through Takayama and Judge [50] to Rowse [43], of expressing such problems as extremal problems.

An alternate school, typified by MacKinnon[28], has sought to treat various special cases of the spatial price equilibrium problem as a fixed point or complementary problem rather than employing an extremal formulation. Following this tradition Friesz et al. [52] shows that the spatial equilibrium problem on a general network with nonseparable functions (or functions without symmetric Jacobians) may be readily handled as a nonlinear complementary problem and that the iterative use of a linear complementary algorithm provides an efficient and practical solution. For the single commodity linear case, Glassey[7] , Pang and Lee[21], and Jones, Saigal and Schneider[41] have presented algorithms which exploit the network structure of the problem. Pang[23] developed a network based algorithm for linear, multi-commodity problems. For the case of nonlinear excess demand functions, Ahn and Seong[3] have developed a parametric network-based method. All of the network-based methods listed above require constant transportation costs that satisfy the triangle inequality. Schneider[41] and Udomkesmalee[57] discuss nonlinear transportation costs which allow us to remove the triangle inequality constraint.

Diagonalization methods that were originally developed to deal with nonseparable Wardropian network equilibrium problems (Abdulaal and LeBlanc [1]) are now recognized as able to apply to any variational inequality (see, e.g., Dafermos [8]). Florian and Los [14] formulated the general spatial price equilibrium problem as a variational inequality, a finding which allows general results, such as those due to Dafermos [8], regarding the global convergence of diagonalization algorithms to be applied in order to develop convergence criteria specific to the spatial price equilibrium problem.

These general formulations include multi-commodity equilibrium models in which there are interactions among commodities, equilibrium models with transportation networks in which there are interactions among the markets in addition to the interactions due to the transportation of commodities. General methods for solving variational inequalities and nonlinear complementary problems can be applied to solve these formulations. For examples of these methods, see Dafermos[8] and Pang and Chan[20].

Theise and Jones [47] discuss issues related to microcomputer implementation of the import equilibration algorithm working directly on the equilibrium conditions. It was found that the expanding equilibrium algorithm was superior to the equilibration algorithm in all areas - solution time, memory requirements, numerical accuracy - except ease of implementation. It has been shown that it can be modified to efficiently process nonlinear, single commodity problems and linear, multiple commodity problems (Theise and Jones [48]) having similar structure. However, there appears to be no simple way of modifying the algorithm so that it can solve congestion problems: those having non-constant per-unit shipping costs.

Tobin [55] has proposed a variable dimension solution approach for the general spatial price equilibrium problem. In its most general form, spatial price equilibrium may have nonlinear demand and supply functions, nonlinear shipping cost functions, inter-commodity congestion effects; his algorithm is capable of solving models that include these complicated relationships. While numerical examples provided by Tobin [55] are five region problems trading in two commodities, and CPU times are not given. There are definite trade-offs between general-purpose and special-purpose algorithms: Tobin's algorithm can be used to solve models very rich in detail, but the number of regions and/or commodities in any one model is limited due to large computational demands for CPU time and data storage. The results presented by Theise and Jones [47] suggested that some large-scale nonlinear, multiple commodity spatial price equilibrium problems could be solved using the expanding equilibrium algorithm. But such problems cannot, at present, include certain modeling features that Tobin's algorithm can: nonlinear shipping cost functions.

1.2 Problem statement:

The problem of finding regional prices and interregional trade flows at the equilibrium of an n region economy trading in m homogenous commodities is known as the multiple commodity spatial price equilibrium (MSPE) problem. In this thesis I consider solution techniques for MSPE where the excess demand function in each region is nonlinear, and the cost of shipping a unit of commodity from one region to another region is dependent on the quantity shipped. Mathematically, the problem is to find values for θ_j^r and V_{ij}^r which satisfy the following equilibrium conditions(Theise[48]) :

$$S_l^r - D_l^r = 0 \quad \forall \text{ l, and r} \tag{1.1}$$

$$S_l^r - D_l^r + \sum_{i \in I} \sum_{(i,l) \in A} V_{il}^r - \sum_{j \in I} \sum_{(l,j) \in A} V_{lj}^r = 0 \qquad \forall \text{ r} \tag{1.2}$$

$$\sum_{r=1}^{m} \sum_{l \in I} S_l^r - \sum_{r=1}^{m} \sum_{l \in I} D_l^r = 0 \tag{1.3}$$

$$\psi_i^r + c_{ij}^r \geq \theta_j^r \quad \forall \text{ r} \tag{1.4}$$

$$V_{ij}^r (\psi_i^r + c_{ij}^r - \theta_j^r) = 0 \qquad \forall \text{ r} \tag{1.5}$$

$$\psi_i^r, \theta_j^r, V_{ij}^r \geq 0 \qquad \forall \text{ i,j, and r} \tag{1.6}$$

where

I : the full set of regions of the network.

A: the full set of flows of the network, each flow in A represents an origin - destination pair (O-D) connecting two regions.

m : the number of commodities in the economy ; r indexes these commodities,

n : the number of economic regions; i,j, and l index these regions,

D_l^r: the demand quantity of commodity r at region l,

θ_l^r: the demand price per unit of commodity r at region l,

S_l^r: the supply quantity of commodity r at region l,

$C_l^r(S_l^r)$: the total cost of producing S_l^r at region l,

ψ_l^r: the supply price per unit of commodity r at region l,

V_{ij}^r : the units of commodity r shipped from region i to region j, and

c_{ij}^r : the cost of shipping a unit of commodity r from region i to region j.

The first condition ensures that total demand equal total supply. This condition is redundant , since summing (1.2) over all $l \in I$ will yield the same result. However, Eq.(1.1) will play an important role in the network flow programming solution algorithms. The fourth condition guarantees that there is no further incentive to trade.

The fifth condition guarantees either that no profits will be made when trade between two regions exists or that no trade will occur between two regions if a loss will be taken.

1.3 Research Objectives

As we mentioned before, Samuelson [44] pointed out that there exists an objective whose maximization guarantees fulfillment of the conditions of perfectly competitive equilibrium among spatially separated markets. Takayama and Judge [50] presented another version of spatial pricing model: the monopoly model. However, markets of most primary commodities and manufactured goods lie somewhere between these two extremes, taking on some form of oligopoly. It is able to provide appropriate solutions for the equilibrium conditions in the actual markets of most commodities. In this thesis, we show how to formulate the equilibrium conditions for these three different models and how to solve the MSPE problem by using the same algorithm.

We also wish to indicate how transportation influences these three markets, since transportation not only affects the availability of goods, but it also has a major impact on the prices of goods sold on the market.

1.4 Assumption

In the expanding algorithm Theise[47], it shows that if the shipping costs obey the triangle inequality ($c_{ij}^r + c_{jk}^r \geq c_{ik}^r$, $\forall$ i, j, k, and r), an equilibrium solution, if one exists, can always be found whose trade flows form a forest with alternating arcs, that is , no market will simultaneously be both an exporter and importer. Therefore, for n markets in equilibrium there will be at most k - 1 arcs carrying trade. In Theise [47], they use the Manhattan distances between regions as shipping costs. In the real world, there are lots of routes that do not obey the triangle inequality. For this reason, we remove this constraint. That means, we consider the shipping cost is variable.

The existence and uniqueness of a solution to SPE is assured by assuming that

(i) the feasible set formed from (1.1) - (1.6) is nonempty,

(ii) θ_I^r is a strictly decreasing function,

(iii) $C_I^r(S_I^r)$ is a strictly convex and nondecreasing function (or that ψ_I^r is strictly increasing function), and (iv) $c_{ij}^r(V_{ij}^r)$ is a strictly increasing function, and that all functions are continuously differentiable.

The assumption that $c_{ij}^r(V_{ij}^r)$ is a strictly increasing function is somewhat troublesome in that freight systems tend to exhibit economies of density, and thus, average costs are U-shaped. However, c_{ij}^r represents economic price (rate plus level of service) and this factor tends to be less U-shaped than the rate alone. Furthermore, motor carriers tend to exhibit little or no economies of density, whereas the railroad industry does indeed exhibit these economies (Harker [18]). Thus , strictly increasing economic prices of transportation may indeed be the case in the motor carrier industry, and elsewhere this assumption may not be a bad approximation.

1.5 Approach

The basic approach to this thesis effort consists of the following steps :

Step 1 : An overview of market structure; perfect competition, monopoly, and oligopoly. We also provide formulation of all three of these models. These are presented in Chapter II.

Step 2 : An overview of the expanding equilibrium algorithm for linear single spatial price equilibrium. This is presented in Chapter III.

Step 3 : We present the alternate algorithm for multi-commodity spatial price equilibrium (MSPE). We also want to prove the existence and uniqueness of the solution. This is represented in Chapter IV

Step 4 : Computation experience, including numerical example. This is represented in Chapter IV.

II. Market structure and formulation

2.1 Introduction

Market structure describes the competitive environment in the market for any good or service. A market consists of all firms and individuals who are willing and able to buy or sell a particular product. This includes firms and individuals currently engaged in buying and selling a particular product, as well as potential entrants.

Market structure is typically characterized on the basis of four important industry characteristics: the number and size distribution of active buyers and sellers and potential entrants, the degree of product differentiation, the amount and cost of information about product price and quality, and the conditions of entry and exit. In the following sections, we will discuss three major market models: perfect competition, monopoly, and oligopoly. Before discussing these three models, we would like to discuss equilibrium conditions. The basic equilibrium condition of the spatial price equilibrium problem is conservation of flow in every region and it is formulated as Eq.(2.1).

$$S_I^r - D_I^r + \sum_{i \in I} \sum_{(i,I) \in A} V_{iI}^r - \sum_{j \in I} \sum_{(I,j) \in A} V_{Ij}^r = 0 \qquad \forall \text{ r} \tag{2.1}$$

Similarly, the market clearing condition (i.e., total demand equal total supply) is formulated as Eq.(2.2).

$$\sum_{I \in I} D_I^r - \sum_{I \in I} S_I^r = 0 \qquad \forall \text{ r} \tag{2.2}$$

As mentioned before, this condition is redundant, since summing Eq.(2.1) over all I will yield the same result.

However, Eq.(2.2) will play an important role in the network-flow-programming-solution-algorithm.

2.2 Perfect competition

Perfect competition is a market structure characterized by a large number of buyers and sellers of essentially the same product, where each market participant's transactions are so small that they have no influence on the market price of the product. Therefore, individual buyers and sellers are price takers. This means that firms take market prices as a given and devise their production strategies accordingly. Free and complete demand and supply information is available in a perfectly competitive market, and there are no meaningful barriers to entry and exit. As a result, vigorous price competition prevails, and only a normal rate of return on investment is possible in the long run. Economic profits are possible only in periods of short-run disequilibrium before rivals mount effective competitive responses. The The Classic Spatial Price Equilibrium (CSPE) is formulated as follows:

Objective function :

MAX

$$\sum_{r=1}^{m}\sum_{I\in I}\int_{0}^{D_I^r}\theta_I^r(y)\,dy-\sum_{r=1}^{m}\sum_{I\in I}C_I^r(S_I^r)+\sum_{r=1}^{m}\sum_{(i,j)\in A}\int_{0}^{V_{ij}^r}c_{ij}^r(y)\,dy \qquad (2.3)$$

Subject to

$$S_I^r-D_I^r+\sum_{i\in I}\sum_{(i,I)\in A}V_{iI}^r-\sum_{j\in I}\sum_{(I,j)\in A}V_{Ij}^r=0 \quad \forall\ \mathrm{r} \qquad (2.4)$$

$$\sum_{I\in I}D_I^r-\sum_{I\in I}S_I^r=0 \qquad \forall\ \mathrm{r} \qquad (2.5)$$

$$\sum_{r=1}^{m}\sum_{l\in I} S_l^r - \sum_{r=1}^{m}\sum_{l\in I} D_l^r = 0 \qquad (2.6)$$

$S_l^r, D_l^r, V_{ij}^r \geq 0 \quad \forall\ (i,j) \in A,\ l \in I$, and r (2.7)

If

(i) the revenue $\int_0^{D_l^r} \theta_l^r(y)\,dy$ is concave and nondecreasing for all l, and r,

(ii) the market price $\theta_l^r(D_l^r)$ is strictly decreasing and continuously differentiable,

(iii) the total cost of production $C_l^r(S_l^r)$ is convex , nondecreasing, and continuously differentiable,

(iv) the total transportation cost is convex and nondecreasing,

(v) the cost of transportation is strictly increasing and continuously differentiable,

(vi) no interaction between commodities,

then the Karash-Kuhn-Tucker (KKT) conditions of this problem are necessary and sufficient for a solution.

Let π_l^r : denote the dual variable of constraint (2.4)

KKT conditions :

$$(\theta_l^r - \pi_l^r)D_l^r = 0$$

$$\theta_l^r - \pi_l^r \leq 0, D_l^r \geq 0 \quad \forall\ l \in I, r \qquad (2.8)$$

Condition (2.8) states that if there is demand in region l, the shadow price π_l^r will equal the average revenue θ_l^r in region l. Similarly, if there is supply in region l,

$$(-C_l^{r'} + \pi_l^r)S_l^r = 0$$

$$-C_l^{r'} + \pi_l^r \leq 0, S_l^r \geq 0 \quad \forall\ l \in I, r \qquad (2.9)$$

Condition (2.9) states that the shadow price π_I^r equals the average cost of production

$$C_I^{r'}$$

$$(-c_{ij}^r + \pi_j^r - \pi_i^r)V_{ij}^r = 0$$

$$-c_{ij}^r + \pi_j^r - \pi_i^r \le 0, V_{ij}^r \ge 0 \quad \forall \text{ (i,j)} \in \text{A, r} \tag{2.10}$$

Condition (2.10) states that if there is flow between regions i and j, the average economic cost of transportation, c_{ij}^r, plus the average production cost $C_i^{r'}$ equals the average revenue, θ_j^r.

or

(a) if $V_{ij}^r \ge 0$

then $C_i^{r'}(S_i^r) + c_{ij}^r(V_{ij}^r) = \theta_j^r(D_j^r)$ $\quad \forall$ (i,j) $\in$ A, and r

(b) if $C_i^{r'}(S_i^r) + c_{ij}^r(V_{ij}^r) > \theta_j^r(D_j^r)$

then $V_{ij}^r = 0$ $\quad \forall$ (i,j) $\in$ A, and r

Then, the original problem for pure competition can be rewritten as :

MAX

$$\sum_{r=1}^{m}\sum_{I\in I}\int_0^{D_I^r}\theta_I^r(y)dy - \sum_{r=1}^{m}\sum_{I\in I}C_I^r(S_I^r) + \sum_{r=1}^{m}\sum_{(i,j)\in A}\int_0^{V_{ij}^r}c_{ij}^r(y)dy$$

Subject to

$$S_I^r - D_I^r + \sum_{i\in I}\sum_{(i,I)\in A}V_{iI}^r - \sum_{j\in I}\sum_{(I,j)\in A}V_{Ij}^r = 0$$

$$\sum_{I\in I}D_I^r - \sum_{I\in I}S_I^r = 0$$

$$\sum_{r=1}^{m}\sum_{l\in I} S_l^r - \sum_{r=1}^{m}\sum_{l\in I} D_l^r = 0$$

$$(C_i^{r'} + c_{ij}^r - \theta_j^r) V_{ij}^r = 0$$

$S_l^r, D_l^r, V_{ij}^r \geq 0 \qquad \forall\ (i,j) \in A,\ l \in I$, and r

2.3 Monopoly

Monopoly is a market structure characterized by a single seller of a highly differentiated product. Because a monopolist is the sole provider of a desired commodity and it has perfect information concerning the demand behavior in each region and fully controls the transportation system, the monopolist is the industry. Therefore, the monopolist can simultaneously determine price and output for itself. It is a price maker.

In this case, the firm's profit-maximization problem is formulated as follows:

MAX

$$\sum_{r=1}^{m}\sum_{l\in I} \theta_l^r (D_l^r) D_l^r - \sum_{r=1}^{m}\sum_{l\in I} C_l^r (S_l^r) - \sum_{r=1}^{m}\sum_{(i,j)\in A} c_{ij}^r (V_{ij}^r) V_{ij}^r \tag{2.11}$$

subject to

$$S_l^r - D_l^r + \sum_{i\in I}\sum_{(i,l)\in A} V_{il}^r - \sum_{j\in I}\sum_{(l,j)\in A} V_{lj}^r = 0 \qquad \forall\ r \tag{2.12}$$

$$\sum_{l\in I} D_l^r - \sum_{l\in I} S_l^r = 0 \qquad \forall\ r \tag{2.13}$$

$$\sum_{r=1}^{m}\sum_{l\in I} S_l^r - \sum_{r=1}^{m}\sum_{l\in I} D_l^r = 0 \tag{2.14}$$

$S_l^r, D_l^r, V_{ij}^r \geq 0 \qquad \forall\ (i,j) \in A,\ l \in I$, and r (2.15)

If

(i) the revenue $\theta_l^r(D_l^r)D_l^r$ is concave and nondecreasing for all l, r,

(ii) the market price $\theta_l^r(D_l^r)$ is strictly decreasing and continuously differentiable,

(iii) the total cost of production $C_l^r(S_l^r)$ is convex, nondecreasing, and continuously differentiable,

(iv) the total transportation cost is convex and nondecreasing,

(v) the cost of transportation is strictly increasing and continuously differentiable,

(vi) no interaction between commodities,

then the Karash-Kuhn-Tucker (KKT) conditions of this problem are necessary and sufficient for a solution.

Let π_l^r : denote the dual variable of constraint (2.12)

KKT conditions :

$$(\theta_l^r + D_l^r\theta_l^{r'} - \pi_l^r)D_l^r = 0$$

$$\theta_l^r + D_l^r\theta_l^{r'} - \pi_l^r \leq 0,\ D_l^r \geq 0 \quad \forall\, l \in I, r \tag{2.16}$$

Condition (2.16) states that if there is demand in region l, the shadow price π_l^r will equal the marginal revenue in region l , $\theta_l^r + D_l^r\theta_l^{r'}$. Similarly, if there is supply in region l,

$$(-C_l^{r'} + \pi_l^r)S_l^r = 0$$
$$-C_l^{r'} + \pi_l^r \leq 0, S_l^r \geq 0 \quad \forall\, l \in I, r \tag{2.17}$$

Condition (2.17) states that the shadow price π_l^r equals the marginal cost of production $C_l^{r'}$

$$(-c_{ij}^r - V_{ij}^r c_{ij}^{r'} + \pi_j^r - \pi_i^r) = 0$$

$$-c_{ij}^r - V_{ij}^r c_{ij}^{r'} + \pi_j^r - \pi_i^r \leq 0, V_{ij}^r \geq 0 \quad \forall \text{ (i,j)} \in \text{A, r} \qquad (2.18)$$

Condition (2.18) states that if there is flow between regions i and j, the marginal economic cost of transportation MTC_{ij}^r, $c_{ij}^r + V_{ij}^r c_{ij}^{r'}$, plus the marginal production cost $C_i^{r'}$ equals the marginal revenue $MR_j^r, \theta_j^r + D_j^r \theta_j^{r'}$.

or

(c) if $V_{ij}^r \geq 0$ then $C_i^{r'} + MTC_{ij}^r = MR_j^r$

(d) if $C_i^{r'} + MTC_{ij}^r > MR_j^r$ then $V_{ij}^r = 0$

The equilibrium conditions (c) and (d) are very similar to the CSPE conditions (a) and (b) except that the average transportation-costs and the average revenue are replaced by their marginal values.

Then, the original problem for monopoly can be rewritten as :

MAX

$$\sum_{r=1}^{m}\sum_{l\in I}\theta_l^r(D_l^r)D_l^r - \sum_{r=1}^{m}\sum_{l\in I}C_l^r(S_l^r) - \sum_{r=1}^{m}\sum_{(i,j)\in A}c_{ij}^r(V_{ij}^r)V_{ij}^r$$

subject to

$$S_l^r - D_l^r + \sum_{i\in I}\sum_{(i,l)\in A}V_{il}^r - \sum_{j\in I}\sum_{(l,j)\in A}V_{lj}^r = 0$$

$$\sum_{l\in I}D_l^r - \sum_{l\in I}S_l^r = 0$$

$$\sum_{r=1}^{m}\sum_{l\in I} S_l^r - \sum_{r=1}^{m}\sum_{l\in I} D_l^r = 0$$

$$\left[C_i^{r'} + c_{ij}^r + V_{ij}^r c_{ij}^{r'} - (\theta_j^r + D_j^r \theta_j^{r'}) \right] V_{ij}^r = 0$$

$S_l^r, D_l^r, V_{ij}^r \geq 0 \qquad\qquad \forall\ (i,j) \in A,\ l \in I$, and r

2.4 Oligopoly

In between perfect competition and monopoly is a model consisting of a few firms operating in spatially separated markets, which is often the more realistic case for discrete facility- location. With few competitors, economic incentives often exist for firms to devise illegal agreements to limit competition, fix prices, or otherwise divide markets. Under oligopoly, the price and output decisions of firms are interrelated in the sense that direct reactions from leading rivals can be expected. As a result, decisions of individual firms are based in part on the likely responses of competitors.

We assume that at most one firm operates in each region, and that each firm has knowledge of the demand behavior in each region and is neither a monopolist nor controls the transportation system as in the monopoly model. Instead, it takes the economic price of transportation service as given, resulting in the average economic price of transportation being used rather than the marginal values as in the monopoly model. Finally, let us assume that the producing firms behave in a Cournot-Nash manner in which each firm takes the other firm's production decisions as fixed when deciding upon its own supply/distribution strategy.

Let

Q : denote the set of firms operation in the market,

J_q :denote the set of production sites or regions under firm q's control,

D_{lq}^r: the amount of commodity r supplied by firm q to region l,

$\tilde{D}_{lq}^r$: the amount of commodity r supplied by all other firms to region l,

$$\tilde{D}_{lq}^r = \sum_{\substack{j \in Q \\ j \neq q}} D_{lj}^r \text{, and}$$

$$D_l^r = \tilde{D}_{lq}^r + D_{lq}^r.$$

The optimal strategy vector of firm q consisting of the total-amount supplied locally to region $l \in J_q$, the amount supplied by firm q to consumers elsewhere in the network and the specific shipment between production site i and consumer site j,can be written as

$$y_q^{r^T} = \left[(S_l^r \middle| l \in J_q), (D_{lq}^r \middle| l \in I), (V_{ij}^r \middle| i \in J_q, j \in I, (i,j) \in A \right] \text{, and}$$

firm q's profit-maximization can be written as:

MAX

$$\sum_{r=1}^{m} \sum_{l \in I} \theta_l^r (D_{lq}^r + \tilde{D}_{lq}^r) D_{lq}^r - \sum_{r=1}^{m} \sum_{l \in J_q} C_l^r (S_l^r) - \sum_{r=1}^{m} \sum_{\substack{l \in J_q \\ (l,j) \in A}} c_{lj}^r (V_{lj}^r) V_{lj}^r \qquad (2.19)$$

The set of constraints

$$\Omega_q^r = \left\{ y_q^r \right\}$$

it faces are :

$$S_I^r - D_I^r + \sum_{i \in I} \sum_{(i,I) \in A} V_{iI}^r - \sum_{j \in I} \sum_{(I,j) \in A} V_{Ij}^r = 0 \qquad \forall \text{ r} \tag{2.20}$$

$$\sum_{I \in I} D_I^r - \sum_{I \in I} S_I^r = 0 \qquad \forall \text{ r} \tag{2.21}$$

$$\sum_{r=1}^{m} \sum_{I \in I} S_I^r - \sum_{r=1}^{m} \sum_{I \in I} D_I^r = 0 \tag{2.22}$$

$$\tilde{S}_I^r \le S_I^r \le \overline{S}_I^r$$

$$\tilde{D}_I^r \le D_I^r \le \overline{D}_I^r$$

$$\tilde{V}_{ij}^r \le V_{ij}^r \le \overline{V}_{ij}^r$$

Notice that firm q takes $\tilde{D}_{Iq}^r$ and c_{ij}^r are fixed according to a Cournot - Nash assumption and is a price taker in the market.

If

(i) the total revenue $\theta_I^r(D_I^r)D_I^r$ is a strictly concave and nondecreasing function,

(ii) the market price $\theta_I^r(D_I^r)$ is a strictly decreasing function,

(iii) the total cost of production $C_I^r(S_I^r)$ is a convex and nondecreasing function,

(iv) no interaction between commodities,

(v) the feasible region Ω_q^r is nonempty,

then problem (2.19) is completely equivalent to the following variational inequality problem :

Find the optimal vector of commodity r for firm q

$$y_q^{r*T} = \left[(S_I^{r*} \middle| I \in J_q), (D_{Iq}^{r*} \middle| I \in I), (V_{ij}^{r*} \middle| i \in J_q, j \in I, (i,j) \in A \right]$$

such that

$$F_q^{r^T}(y_q^r)(z_q^r - y_q^r) =$$

$$\left[\sum_{l\in J_q} C_l^r(S_l^{r^*})(S_l^r - S_l^{r^*}) - \sum_{l\in I} MR_{lq}^r(D_{lq}^{r^*})(D_{lq}^r - D_{lq}^{r^*}) + \sum_{l\in J_q}\sum_{(l,j)\in A} c_{lj}^r(V_{lj}^{r^*})(V_{lj}^r - V_{lj}^{r^*})\right] \geq 0$$

MR_{lq}^r: the marginal revenue of commodity r in region l for firm q

$$MR_{lq}^r(D_l^r) = \frac{\partial}{\partial D_{lq}^r}\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r)D_{lq}^r = \theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r) + D_{lq}^r\frac{\partial\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r)}{\partial D_{lq}^r}$$

c_{lj}^r and $\tilde{D}_{lq}^r$ were taken as fixed when calculation the gradient of Eq.(2.19). It can be shown (Chapter IV) that a unique equilibrium exists in this model when these conditions are satisfied :

(i) $C_l^r(S_l^r)$ is a strictly convex, continuously differentiable function for all l

(ii) $c_{ij}^r(V_{ij}^r)$ is a monotone (nondecreasing), continuous function for all (i,j)

(iii) $-MR(D^r) = (\ldots, -MR_{lq}^r(D^r), \ldots)^T$ is a strictly monotone, continuous function

.

then the Karash-Kuhn-Tucker (KKT) conditions of this problem are necessary and sufficient for a solution.

Let: π_{lq}^r denote the dual variable of constraint (2.20)

KKT conditions :

$$(\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r) + D_{lq}^r\frac{\partial}{\partial D_{lq}^r}\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r) - \pi_{lq}^r)D_{lq}^r = 0$$

$$\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r) + D_{lq}^r\frac{\partial}{\partial D_{lq}^r}\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r) - \pi_{lq}^r \leq 0,\ D_{lq}^r \geq 0 \quad \forall l\in I,\ q\in J_q,\ r \quad (2.23)$$

Condition (2.23) states that if there is demand in region l, the shadow price π_{lq}^r will equal the average revenue in region l , $\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r) + D_{lq}^r \frac{\partial}{\partial D_{lq}^r}\theta_l^r(\tilde{D}_{lq}^r + D_{lq}^r)$.

Similarly, if there is supply in region l,

$$(-C_l^{r'} + \pi_l^r) S_l^r = 0$$

$$-C_l^{r'} + \pi_l^r \leq 0, S_l^r \geq 0 \quad \forall\, l \in I, r \qquad (2.24)$$

Condition (2.24) states that the shadow price π_l^r equals the marginal cost of production $C_l^{r'}$

$$(-c_{ij}^r - V_{ij}^r c_{ij}^{r'} + \pi_j^r - \pi_i^r) = 0$$

$$-c_{ij}^r - V_{ij}^r c_{ij}^{r'} + \pi_j^r - \pi_i^r \leq 0, V_{ij}^r \geq 0 \quad \forall\, (i,j) \in A, r \qquad (2.25)$$

Condition (2.25) states that if there is flow between regions i and j, the average economic cost of transportation, $c_{ij}^r + V_{ij}^r c_{ij}^{r'}$, plus the marginal cost of production $C_i^{r'}$ equals the marginal revenue $MR_{ji}^r, \theta_j^r(\tilde{D}_{ji}^r + D_{ji}^r) + D_{ji}^r \frac{\partial}{\partial D_{ji}^r}\theta_j^r(\tilde{D}_{ji}^r + D_{ji}^r)$.

or

(e) if $V_{ij}^r \geq 0$ then $C_i^{r'} + (c_{ij}^r + V_{ij}^r c_{ij}^{r'}) = MR_{ji}^r$

(f) if $C_i^{r'} + (c_{ij}^r + V_{ij}^r c_{ij}^{r'}) > MR_{ji}^r$ then $V_{ij}^r = 0$

Then, the original problem for oligopoly can be rewritten as:

MAX

$$\sum_{r=1}^{m}\sum_{l\in I}\theta_l^r(D_{lq}^r+\tilde{D}_{lq}^r)D_{lq}^r-\sum_{r=1}^{m}\sum_{l\in J_q}C_l^r(S_l^r)-\sum_{r=1}^{m}\sum_{\substack{l\in J_q\\(l,j)\in A}}c_{lj}^r(V_{lj}^r)V_{lj}^r$$

The set of constraints

$$\Omega_q^r=\left\{y_q^r\right\}$$

it faces are :

$$S_l^r-D_l^r+\sum_{i\in I}\sum_{(i,l)\in A}V_{il}^r-\sum_{j\in I}\sum_{(l,j)\in A}V_{lj}^r=0$$

$$\sum_{l\in I}D_l^r-\sum_{l\in I}S_l^r=0$$

$$\sum_{r=1}^{m}\sum_{l\in I}S_l^r-\sum_{r=1}^{m}\sum_{l\in I}D_l^r=0$$

$$\left[C_i^{r'}+(c_{ij}^r+V_{ij}^r c_{ij}^{r'})-(\theta_j^r(\tilde{D}_{ji}^r+D_{ji}^r)+D_{ji}^r\frac{\partial}{\partial D_{ji}^r}\theta_j^r(\tilde{D}_{ji}^r+D_{ji}^r))\right]=0$$

$$\tilde{S}_l^r\le S_l^r\le\overline{S}_l^r$$

$$\tilde{D}_l^r\le D_l^r\le\overline{D}_l^r$$

$$\tilde{V}_{ij}^r\le V_{ij}^r\le\overline{V}_{ij}^r\qquad\text{for (i,j)} \in A,\ l\in I,\ q\in J_q,\ \text{and } r$$

III. The Expanding Algorithm

3.1 Introduction

In this section we provide an overview and illustrate the use of the expanding equilibrium algorithm for single commodity spatial price equilibria. Theoretical justification of the algorithm may be found in Jones, Saigal and Schneider [41] and Schneider[30]. This algorithm follows an intuitively appealing path to solving LSSPE (Linear Single Spatial Price Equilibrium) and, in fact, its heritage may be traced to the first published article on spatial price equilibrium (Enke[46]).

The algorithm begins by determining the equilibrium prices and trade flow between two of the n regions in the economy. During each subsequent iteration of the main loop a new region, k , is brought into equilibrium by modifying the existing (k-1) region equilibrium. The algorithm derives its name from this region-at-a-time expansion of the economy.

Samuelson [44] made the observation that when shipping costs obey the triangle inequality ($c_{ij} + c_{jk} \geq c_{ik}, \forall$ i, j, and k), an equilibrium solution, if one exists, may always be found whose trade flows form a forest with alternating arcs. Glassey [7] formalized this property much later. An n region economy needs to have at most (n-1) arcs carrying flow at equilibrium. This allows SPE problems to be represented economically by using network data structures. In perfect competition, it is very useful. But, it is not going to work with the other two models. Mathematically, the problem solved by expanding algorithm is to find values of p_i and V_{ij}, for all $1 \in I$, $(i,j) \in A$, which satisfy the following equilibrium conditions :

$$b_i - d_i p_i + \sum_{j \neq i} V_{ij} - \sum_{j \neq i} V_{ji} \leq 0 \qquad (3.1)$$

$$p_i (b_i - d_i p_i + \sum_{j \neq i} V_{ij} - \sum_{j \neq i} V_{ji}) = 0 \qquad (3.2)$$

$$p_i + c_{ij} - p_j \geq 0 \qquad (3.3)$$

$$V_{ij} (p_i + c_{ij} - p_j) \geq 0 \qquad (3.4)$$

$$p_i, V_{ij} \geq 0 \qquad (3.5)$$

where

n : the number of regions; i and j index these regions,

p_i : the commodity price at region i,

V_{ij} : the quantity of commodity r shipped from region i to j,

$b_i - d_i p_i$: the linear excess demand function of commodity r at region i, and

c_{ij} : the cost of shipping a unit of commodity from region i to j .

3.2 Network data structure

The expanding equilibrium algorithm is inextricably bound to the network data structures used to represent the problem. Solutions are represented by rooted trees. Any node in a tree may be designated the root, although in this application the region currently being brought into equilibrium is always made the root. Once a root is chosen, all nodes and arcs in the tree have a specific orientation in relation to the root.

We define the following labels for each node:

1. thread of node i: $thread_i$ points to the next node in a circular list that passes through every node in the network.

2. predecessor of node i: $pred_i$ is the first node encountered on the path beginning at node i and ending at the root; $pred_{root} = 0$

3. number of successors of node i: nos_i .

4. slopes of successors of node I: $sos_i = \sum_{j \in succc_i} d_j$

5. trade status of node i : $ts_i = \begin{cases} +1 & \text{if region i imports} \\ 0 & \text{if region i does not trade} \\ -1 & \text{if region i exports.} \end{cases}$

6. flow of node I: $flow_i$ is the flow between node I and $pred_i$; $flow_{root} = 0$; $flow_i$ has the same sign as ts_i and the following relationship with V_{ij}:

   if $flow_i < 0$, then $x_{i, pred_i} = - flow_i$

   otherwise if $flow_i > 0$, then $x_{i, pred_i} = flow_i$

The first three labels are standard network data structures; see Appendix B of Kennington and Helgason [25] for a concentrated introduction to these structures in the context of the network simplex algorithm. The slopes of successors label is carried in order to reduce the amount of work performed in the critical step of the algorithm. The flow label measures the quantity of commodity shipped along the unique arc between every nonroot node and its predecessor.

The following additional notation is needed for this algorithm :

Let T_k be the tree containing node k, T_j be the tree containing any node $j \notin T_k$,

and S_i be the subtree consisting of the successors of node I . The algorithm follows.

3.3 Expanding Algorithm for LSSPE (Linear Single Spatial Price Equilibrium)

Given an ordering of regions from 1 to n and parameters b_i, d_i, and c_{ij}, solve for the values of p_i and $flow_i$ at equilibrium.

EE1. [Initialize.] (Isolated prices : the equilibrium prices in the absence of trade)

Set $p_i \leftarrow b_i / d_i$, $ts_i \leftarrow 0$ for $1 \leq i \leq n$. Set $k \leftarrow 2$.

EE2.[Main loop] (Determine which, if any, region currently in equilibrium trades with region k. If region k trades, set its price equal to the price at which trade could begin, and root T_k at k.)

Set $L \leftarrow \max_{j<k;ts_j=1}(p_j - c_{kj})$

Set $U \leftarrow \min_{j<k;ts_j=-1}(p_j - c_{jk})$

If $p_k < L$, set $p_k \leftarrow L$, $ts_k \leftarrow -1$, update T_k, and go to Step EE3a.

Otherwise. If $p_k > U$, set $p_k \leftarrow U$, $ts_k \leftarrow 1$, update T_k, and go to Step EE3b . Otherwise, set $k \leftarrow k+1$ and go to Step EE2.

EE3a. [Ratio test for exporters.] (Determine feasible price decrease.)

Set $d1 \leftarrow (b_k - d_k p_k) / sos_k$

For ($i \in T_k$: $ts_i = -1$)

Set d2 $\leftarrow$ $\max_i$ ($flow_i$ / sos_i)

Set d3 $\leftarrow \max_{i,j \notin T_k} (p_j - c_{ij} - p_i)$

Set δ $\leftarrow$ max (d1,d2,d3) and go to Step EE4 .

EE3b. [Ratio test for importers .] (Determine feasible price increase.)

Set d1 $\leftarrow$ (b_k - d_k p_k) / sos_k

For ($i \in T_k$: $ts_i = 1$)

Set d2 $\leftarrow$ $\min_i$ ($flow_i$ / sos_i)

Set d3 $\leftarrow \min_{i,j \notin T_k} (p_j + c_{ji} - p_i)$

Set δ $\leftarrow$ min (d1,d2,d3) and go to Step EE4 .

EE4. [Price and flow update]

Set $p_i \leftarrow p_i + \delta$ and update $flow_i$ for $i \in T_k$.

If $\delta = d2$, split S_i from T_k where $flow_i$ has become zero, root S_i at i, and return to that version of Step EE3 from whence you came.

Otherwise, if $\delta = d3$, root T_j at j, splice T_j into T_k , and return to that version of Step EE3 from whence you came.

Otherwise. if k = n , stop with a solution to LSSPE. Otherwise, set $k \leftarrow k + 1$ and go to Step EE2.

The algorithm is initialized by setting regional prices equal to the isolated prices : the equilibrium prices in the absence of trade. Step EE2 uses equilibrium condition (3) to determine if incentive to trade with region k exists in the current (k-1) region economy. If an incentive does not exist, the expansion moves on to the next region. If it does, p_k is set to the price at which region k and its trade partner would be indifferent to trade. Not satisfying the market clearing conditions (1), p_k must then be adjusted until this condition is met.

The purpose of steps EE3a and EE3b is to determine the feasible change that may be made to p_k . The value of d1 represents the price change necessary for market clearing within T_k . Adjusting p_k by this value may result in infeasibility of two types. The value of d2 represents the price change that would result in flow on a basic arc within T_k hitting zero. requiring a change (pivot) in the forest structure of the solution. Note the similarity with d1; d2 is found by measuring the market clearing price change over certain subtrees $S_i \subset T_k$. The value of d3 represents the price change that would result in a nonzero flow on a currently nonbasic arc from some node $j \notin T_k$ to a node $i \in T_k$. This would require a change (pivot) in the forest structure of the solution. Calculating values for d1,d2, and d3 is straightforward, but time - consuming;this is where the algorithm spends the majority of its time. Naturally, p_k may be changed by as much as the most restrictive value allows. Step EE4 is where updates of prices, flows, and forest structure take place. After a finite number of ratio tests and price and flow updates, the market clearing price will be reached and the expansion may proceed to the next region.

IV. Implementation

4.1 Introduction

In this section, we are going to prove the uniqueness of the solution of spatial price equilibrium problem for an oligopoly model. The proof builds upon Harker[18] for single commodity spatial price equilibrium problem. First of all, we assume that the feasible set in nonempty, then, we prove the existence and uniqueness of the solution to spatial price equilibrium problems. Before we prove this, there are some characters below that we should point out.

As is typical, we all know that a reduction in price increases the quantity demanded and, conversely, an increase in price decreases the quantity demanded. So ,we are very sure that $\theta_l^r(D_l^r)$ is a strictly decreasing (nonincreasing) function. It also shows us that $\theta_l^r(D_l^r)D_{lq}^r$ is a strictly concave function. It is well known that the supply curve is always strictly increasing. That is , $\psi_l^r(S_l^r)$ is a strictly nondecreasing function,it also means that $C_l^r(S_l^r)$ is a strictly convex function. The shipping cost, $c_{ij}^r(V_{ij}^r)$, is monotone (nondecreasing) with no doubt. In order to make sure that there is nonempty feasible set, existence, and uniqueness to solution. Again, we assume that there is no interaction between commodities and all the firms behavior in Cournot-Nash manner.

4.2 Existence of solution

The following theorem presents the conditions under which a solution to this model will exist and will be unique.

Theorem 1.(Harker [18])

If Ω is nonempty and all decision variables are bounded away from infinity for all $q \in Q, l \in J_q, (i,j) \in A$, and

(i) There is no interaction between commodities,

(ii) $C_l^r(S_l^r)$ is a strictly convex, nondecreasing, and continuously differentiable function for all $l \in I$,

(iii) $c_{ij}^r(V_{ij}^r)$ is a monotone (nondecreasing), continuous function for all $(i,j) \in A$, and

(iv) $- MR^r(D_l^r) = (\ldots, - MR_{lq}^r(D_l^r), \ldots)^t$ is a strictly monotone, continuous function, then a solution to (4.1) exists and is unique.

$$\sum_{q \in Q} F_q^r(x_q^{r^*})(x_q^r - x_q^{r^*}) \qquad \text{for all } x^r \in \Omega^r, r \tag{4.1}$$

Proof. As is well known (Kinderlehrer and Stampachia), if a variational inequality is defined over a nonempty, compact, convex set and if the function $F^r(x^r) = (F_q^{r^t}(x_q^r))^t$ is continuous, then a solution exists. By assumption, Ω^r is nonempty and by the assumption of finite bounds, it is bounded, Furthermore F^r is assumed to be contiguous, and by inspection, Ω^r is affine (and hence convex) and closed. Thus, a solution must exist.

4.3 Uniqeness of solution

For the uniqueness of the solution, it is well known that F^r being a strictly monotone function will suffice. Relabeling x^r and F^r, we can rewrite F^r as

$$F^r(x^r) = \begin{bmatrix} \nabla C^r(S^r) \\ -MR^r(D^r) \\ c^r(V^r) \end{bmatrix}$$

where $\nabla C^r(S^r) = (\ldots, C_l^{r'}(S_l^r), \ldots)^t$.

and $c^r(V^r) = (\ldots, c_{ij}^r(V_{ij}^r), \ldots)^t$.

As is well known, if $\nabla F^r(x^r)$ is positive definite, then F^r is a strictly monotone function. Writing $\nabla F^r(x^r)$, we have

$$\nabla F^r(x^r) = \begin{bmatrix} diag(C_l^{r'}(S_l^r)) & 0 & 0 \\ 0 & -\nabla MR^r(D^r) & 0 \\ 0 & 0 & diag(c_{ij}^{r'}(V_{ij}^r)) \end{bmatrix}.$$

Thus,we have

$$x^{r^t} \nabla F^r(x^r) x^r = \sum_{l \in I} S_l^{r^2} C_l^{r''}(S_l^r) + \sum_{(ij) \in A} V_{ij}^{r^2} c_{ij}^r(V_{ij}^r) - D^{r^t} \nabla MR^r(D^r) D^r.$$

By assumptions

(i) there is no interaction between commodities,

(ii) we have $C_l^{r''}(S_l^r) > 0$ for all $l \in I$,

(iii) $c_{ij}^r(V_{ij}^r) \geq 0$ for all $(i,j) \in A$, and

(iv) we have $-MR^r(D^r)$ is strictly monotone, which implies

$D^{r'} \nabla MR^r(D^r) D^r < 0$ for all $D^r \neq 0$.

Thus, $x^{r^T} \nabla F^r(x^r) x^r > 0$ for $x^r \neq 0$, and for all r. Therefore, $F^r(x^r)$ and

$\sum_{r=1}^{m} F^r(x^r)$ are strictly monotone increasing, and the solution is unique.

In order to interpret condition (iv), we shall state the following set of sufficient conditions:

Corollary 1.

If all conditions of Theorem 1 hold except that (iv) is replaced by, $\theta_l^r(D_l^r)$ is a continuous, concave, strictly monotone decreasing function, for all $l \in I$, then the conclusion of Theorem 1 holds.

Proof. For $MR^r(D^r)$ to be strictly monotone decreasing, it is sufficient that

$D^{r'} \nabla MR^r(D^r) D^r < 0$ for all $D^r \neq 0$. Writing $\nabla MR^r(D^r) = MR^r_{lq,il}$, where

$$MR^r_{lq,il} = \begin{cases} \frac{\partial}{\partial D^r_{lq}}\left[\theta^r_l(D^r_l) + D^r_{lq}\theta^{r'}_l(D^r_l)\right] = 2\theta^{r'}_l(D^r_l) + D^r_{lq}\theta^{r''}_l(D^r_l) = MR^r_{lq,lq} & \text{for } l=i, q=j \quad (4.2) \\ \frac{\partial}{\partial D^r_{ij}}\left[\theta^r_l(D^r_l) + D^r_{lq}\theta^{r'}_l(D^r_l)\right] = \theta^{r'}_l(D^r_l) + D^r_{lq}\theta^{r''}_l(D^r_l) = MR^r_{lq,-} & \text{otherwise} \quad (4.3) \end{cases}$$

Calculating $D^{r'} \nabla MR^r(D^r) D^r$, we have

$$D^{r'} \nabla MR^r(D^r) D^r = \sum_{l,q} \left\{ D^{r^2}_{lq}(MR^r_{lq,lq}) + \left(\sum_{\substack{i \neq l \\ j \neq q}} D^r_{ij}\right)^2 MR^r_{lq,-} \right\}. \quad (4.4)$$

By the assumption that $\theta_l^r(D_l^r)$ is strictly monotone decreasing we have

$\theta^{r'}_l(D^r_l) \leq 0$, and by the assumption that $\theta_l^r(D_l^r)$ is concave we have $\theta^{r''}_l(D^r_l) \leq 0$ for all $l \in I$. Thus, by Equations (4.2) and (4.3), $MR^r_{lq,ij} < 0$ for all l, q, I, j, and r and by

(4.4) we must have $D^{r'} \nabla MR^r(D^r) D^r < 0$ for all $D^r \neq 0$. therefore, $MR^r(D^r)$ is strictly monotone decreasing and the conclusion of Theorem 1 holds.

Note that $\theta_l^r(D_l^r)$ can be a linear function since it is a concave function. Furthermore, the proof of the preceding corollary illustrates that $\theta_l^r(D_l^r)$ can be convex as long as certain relationships between $\theta_l^{r'}(D_l^r)$ and $\theta_l^{r''}(D_l^r)$ hold. For example, by replacing D_{lq}^r by D_l^r in (4.3), we derive the following sufficient condition:

Corollary 2 .

If $\theta_l^r(D_l^r)$ is strictly monotone decreasing and strictly convex for all $l \in I$, then $MR^r(D^r)$ will be a strictly monotone decreasing function if

$$\theta_l^{r'}(D_l^r) + D_l^r \theta_l^{r''} < 0 \quad for \quad l \in I \tag{4.5}$$

Proof. Since $\theta_l^r(D_l^r)$ is strictly monotone decreasing $\theta_l^{r'}(D_l^r) < 0$ and since $\theta_l^r(D_l^r)$ is strictly convex,. $\theta_l^{r''}(D_l^r) > 0$ If (4.5) holds, it must be the case that :

$\theta_l^{r'}(D_l^r) + D_{lq}^r \theta_l^{r''}(D_l^r) < 0$ for all $q \in Q$ since $D_l^r = \sum_{q \in Q} D_{lq}^r \geq 0$ for all $q \in Q$.

Thus, we must have $MR_{lq,-}^r < 0$ for all l,q. Furthermore, if (4.5) holds, it must be the case that

$$2\theta_l^{r'}(D_l^r) + D_l^r \theta_l^{r''}(D_l^r) < 0 \quad for \quad l \in I$$

since $\theta_l^{r'}(D_l^r) < 0$. Thus $MR_{lq,lq}^r < 0$ for all l,q and $D^{r'} \nabla MR^r(D^r) D^r$ as defined by (4.4) must be less than zero and hence $MR^r(D^r)$ must be strictly monotone decreasing.

4.4 Algorithm

In this section, we provided an algorithm based on the expanding algorithm to solve these three different models;perfectly competitive, monopoly, and oligopoly. The solution of the original nonlinear programming problem is found by solving a series of linear system problems. Each linear system problem is generated by approximating the nonlinear constraint functions using first-order Taylor series expansions about the current point (vector), $\mathbf{X}_i$. The resulting linear system problem is solved using the expanding algorithm to find the new vector $\mathbf{X}_{i+1}$. If $\mathbf{X}_{i+1}$ does not satisfy the stated convergence criteria and meet all the constraints, the linear system problem is relinearized about the point $\mathbf{X}_{i+1}$ and the procedure is continued until the optimum solution $\mathbf{X}^*$ is found.

Original Problem : MIN $f(\mathbf{X})$

subject to $g_j(\mathbf{X}) \leq 0 \qquad j = 1,2,\ldots,m$

$h_k(\mathbf{X}) = 0 \qquad k = 1,2,\ldots,p$

Algorithm.

Step 1. Start with an arbitrarily initial vector $\mathbf{X}_1$ and set the iteration number as $i = 1$. The vector $\mathbf{X}_1$ need not be feasible.

Step 2. Linearize the objective and constraint functions about the vector $\mathbf{X}_i$ an as

$$f(\mathbf{X}) \cong f(\mathbf{X}_i) + \nabla f(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i)$$

$$g_j(\mathbf{X}) \cong g_j(\mathbf{X}_i) + \nabla g_j(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i)$$

$$h_k(\mathbf{X}) \cong h_k(\mathbf{X}_i) + \nabla h_k(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i)$$

Step 3. Formulate the approximating linear programming problem as

$$\text{Min} \quad f(\mathbf{X}_i) + \nabla f(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i)$$

subject to

$$g_j(\mathbf{X}_i) + \nabla g_j(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i) \le 0 \qquad j = 1,2,\ldots,m$$

$$h_k(\mathbf{X}_i) + \nabla h_k(\mathbf{X}_i)^T(\mathbf{X} - \mathbf{X}_i) = 0 \qquad k = 1,2,\ldots,p$$

Step 4. [Solve the approximating linear system problem using Expanding Algorithm to obtain the solution vector $\mathbf{X}_{i+1}$]

Step 4.a. Set k = 2

Step 4.b. [Determine which market (if any) trades with market k, and form the approximation linear system]

$$S_l^r - D_l^r + \sum_{i,l\le k} V_{il}^r - \sum_{l,j\le k} V_{lj}^r = 0 \quad \forall\, r$$

$$\sum_{r=1}^{m}(S_l^r - D_l^r + \sum_{i,l\le k} V_{il}^r - \sum_{l,j\le k} V_{lj}^r) = 0$$

$$(C_i^{r'} + c_{ik}^r) = \theta_k^r \quad \text{for some } i < k\text{, and all r (for perfect competition)}$$

$$(C_k^{r'} + c_{kj}^r) = \theta_j^r \quad \text{for some } j < k\text{, and all r (for perfect competition)}$$

$$(C_i^{r'} + c_{ik}^r + V_{ik}^r c_{ik}^{r'}) = \theta_k^r + D_k^r \theta_k^{r'} \quad \text{for some } i < k\text{, and all r (for monopoly)}$$

$$(C_k^{r'} + c_{kj}^r + V_{kj}^r c_{kj}^{r'}) = \theta_j^r + D_j^r \theta_j^{r'} \quad \text{for some } j < k\text{, and all r (for monopoly)}$$

$$(C_i^{r'} + c_{ik}^r + V_{ik}^r c_{ik}^{r'}) = \theta_k^r + D_{ki}^r \frac{\partial \theta_k^r}{\partial D_{ki}^r} \quad \text{for some } i< k\text{, and all r (for oligopoly)}$$

$$(C_k^{r'} + c_{kj}^r + V_{kj}^r c_{kj}^{r'}) = \theta_j^r + D_{jk}^r \frac{\partial \theta_j^r}{\partial D_{jk}^r} \quad \text{for some } j< k\text{, and all r (for oligopoly)}$$

Step 4.c. [Solve the approximating linear system above, and go to Step 4.b, if $k < n$ (number of regions),

go to Step 5 , otherwise.

Step 5. Evaluate the original constraints at $\mathbf{X}_{i+1}$; that is, find

$g_j(\mathbf{X}_{i+1}), \quad j = 1,2,\ldots,m \quad \text{and} \quad h_k(\mathbf{X}_{i+1}), \quad k = 1,2,\ldots,p$

If $g_j(\mathbf{X}_{i+1}) \leq \varepsilon$ for $j =1,2,\ldots,m$, and $|h_k(\mathbf{X}_{i+1})| \leq \varepsilon$, $k =1,2,\ldots,p$, and

$\max |\mathbf{X}_{i+1} - \mathbf{X}_i| \leq \varepsilon$,

where ε is a prescribed small positive tolerance, all the original constraints can be assumed to have been satisfied. Hence stop the procedure by taking $\mathbf{X}_{opt} \cong \mathbf{X}_{i+1}$

If $g_j(\mathbf{X}_{i+1}) \geq \varepsilon$ for some j, or $|h_k(\mathbf{X}_{i+1})| \geq \varepsilon$, for some k, find all the elements which are negative, reset them 0. Then, set the new iteration number as $i = i + 1$, and go to Step 2.

4.5 Computational experiment

In order to make any meaningful statements about the performance of the proposed algorithm, we performed a computational experiment. As there are currently no benchmark problems in the spatial price equilibrium, we resorted to random generation of problem parameters. We followed an approach similar to Harker(1984) in the generation of the parameters. Parameters were generated in this fashion to ensure that an equilibrium existed in each region in the absence of trade. The production cost functions, inverse demand functions and O-D transportation cost functions are given by $C_l^r(S_l^r) = \alpha_l^r S_l^r + \beta_l^r S_l^{r^2}$, $\theta_l^r(D_l^r) = \sigma_l^r - \delta_l^r D_l^r$, and

$$c_{ij}^{r} = \phi_{ij}^{r} + \mu_{ij}^{r} V_{ij}^{r^2} + \sum_{k \neq r} \omega_{ij}^{k} V_{ij}^{k} \ .$$

The numerical example we solved is three regions and two commodities. Table I lists the coefficients used for this example. In order to prove the convergence and uniqueness of the solution, we provide three initial starting vectors for every model. Tables II, III, and IV represent the convergence of three different initial starting vectors for perfect competition, respectively. Tables V, VI, and VII represent the convergence of different initial starting vectors for monopoly, respectively. Tables VIII, IX, and X represent the convergence of different initial starting vectors for oligopoly, respectively. As we mentioned before, we would like to show how transportation is used to influence market. We also provide the equilibrium solution for each commodity when they are single commodity in the market. Table XI lists the equilibrium solution for each commodity before they compete in the same market. Table XII lists the equilibrium solution when these commodities are in the same market. Table XIII summarizes the results of congestion of various models.

As we can see, the best initial starting vector is always based on the isolated flows (i.e., the absence of flows between regions; no export and no import). It always reduces a lot of iterations and computation. We also notice that the more commodities in total are shipped, the more congestion there is(see Table XIII), especially in an oligopoly model. Therefore,the transportation has much more impact in oligopoly than the other two models.

Table I Coefficients

Region	α_l^1	β_l^1	α_l^2	β_l^2	σ_l^1	δ_l^1	σ_l^2	δ_l^2
1	1.0	0.5	2.0	0.3	19.0	0.2	27.0	0.3
2	2.0	0.4	1.5	0.5	27.0	0.01	30.0	0.2
3	1.5	0.3	1.0	0.4	30.0	0.3	19.0	0.01
O-D Pair (i , j)	ϕ_{ij}^1	μ_{ij}^1	ϕ_{ij}^2	μ_{ij}^2	ω_{ij}^1		ω_{ij}^2	
(1 , 2)	1.0	0.1	2.0	0.4	0.02		0.02	
(1 , 3)	2.0	0.4	1.0	0.1	0.03		0.03	
(2 , 1)	1.0	0.2	3.0	0.3	0.01		0.01	
(2 , 3)	3.0	0.3	1.0	0.2	0.04		0.04	
(3 , 1)	1.0	0.1	4.0	0.4	0.03		0.03	
(3 , 2)	4.0	0.4	1.0	0.1	0.02		0.02	

Table II Perfect Competition (initial starting vector based on isolated flow)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6
V_{11}^1	2.306	8.018	6.146	6.362	6.373	6.373
V_{12}^1	9.048	7.917	8.873	8.849	8.85	8.85
V_{13}^1	5.105	2.382	1.752	1.517	1.502	1.502
V_{21}^1	2.454	-7.086	0	0	0	0
V_{22}^1	24.859	40.211	30.726	30.728	30.729	30.729
V_{23}^1	3.448	-2.515	0	0	0	0
V_{31}^1	2.947	-2.519	0	0	0	0
V_{32}^1	5.254	3.041	2.282	2.145	2.139	2.139
V_{33}^1	23.348	31.363	29.562	29.731	29.74	29.74
V_{11}^2	18.578	25.468	32.274	26.089	26.101	26.101
V_{12}^2	5.404	3.102	2.859	2.533	2.515	2.515
V_{13}^2	4.337	0.364	-7.787	0	0	0
V_{21}^2	3.609	-1.307	0	0	0	0
V_{22}^2	15.857	25.197	22.242	22.286	22.289	22.289
V_{23}^2	3.343	-2.22	0	0	0	0
V_{31}^2	4.511	1.305	-3.633	0	0	0
V_{32}^2	7.193	5.85	6.187	6.248	6.25	6.25
V_{33}^2	10.568	15.178	19.796	16.051	16.049	16.049

Table III Perfect Competition (all initial starting elements are 200)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6	It. 7	It. 8	It. 9	It. 10
V_{11}^{1}	-185.115	-85.331	-35.737	-11.438	0.158	6.662	6.051	6.345	6.373	6.373
V_{12}^{1}	100.215	50.554	26.163	14.734	10.068	8.254	8.829	8.847	8.85	8.85
V_{13}^{1}	99.996	50.007	25.034	12.576	6.352	3.075	1.91	1.539	1.502	1.502
V_{21}^{1}	99.843	49.622	24.234	10.994	3.206	-5.04	0	0	0	0
V_{22}^{1}	-168.874	-68.531	-17.937	8.202	23.09	36.801	30.725	30.728	30.729	30.729
V_{23}^{1}	99.89	49.758	24.537	11.61	4.458	-1.119	0	0	0	0
V_{31}^{1}	99.79	49.558	24.202	11.089	3.748	-1.578	0	0	0	0
V_{32}^{1}	100.001	50.016	25.051	12.623	6.514	3.616	2.447	2.16	2.139	2.139
V_{33}^{1}	-168.157	-67.971	-17.692	7.792	21.222	29.656	29.399	29.713	29.74	29.74
V_{11}^{1}	-172.128	-72.049	-21.941	3.263	16.227	23.781	28.544	26.092	26.101	26.101
V_{12}^{1}	100.022	50.06	25.128	12.748	6.679	3.715	2.804	2.528	2.515	2.515
V_{13}^{1}	99.903	49.83	24.743	12.082	5.465	1.332	-3.268	0	0	0
V_{21}^{1}	99.912	49.791	24.579	11.682	4.666	-0.181	0	0	0	0
V_{22}^{1}	-176.072	-75.875	-25.551	0.051	13.766	23.106	22.249	22.287	22.289	22.289
V_{23}^{1}	99.857	49.695	24.437	11.465	4.297	-1.041	0	0	0	0
V_{31}^{1}	99.956	49.908	24.835	12.202	5.7	2.08	-1.37	0	0	0
V_{32}^{1}	100.085	50.259	25.594	13.705	8.413	6.259	6.199	6.249	6.25	6.25
V_{33}^{1}	-177.815	-77.937	-28.192	-3.656	8.163	13.982	17.493	16.051	16.049	16.049

Table IV Perfect Competition (all initial starting elements are 0.1)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6	It. 7	It. 8	It. 9	It. 10	It. 11
V_{11}^1	7.352	4.186	5.807	6.435	6.575	6.583	6.583	6.295	6.367	6.373	6.373
V_{12}^1	7.838	8.667	8.814	8.883	8.899	8.9	8.9	8.834	8.849	8.85	8.85
V_{13}^1	8.768	8.309	2.218	1.395	1.211	1.201	1.201	1.6121	1.511	1.502	1.502
V_{21}^1	-35.824	0	0	0	0	0	0	0	0	0	0
V_{22}^1	154.008	30.757	30.755	30.755	30.754	30.754	30.754	30.722	30.728	30.729	30.729
V_{23}^1	-88.284	0	0	0	0	0	0	0	0	0	0
V_{31}^1	-1.319	0	0	0	0	0	0	0	0	0	0
V_{32}^1	-53.839	0	0	0	0	0	0	2.677	2.189	2.14	2.139
V_{33}^1	94.944	30.23	30.927	31.202	31.263	31.266	31.266	29.345	29.704	29.74	29.74
V_{11}^2	124.436	27.778	27.778	27.778	27.778	27.778	27.778	25.567	26.044	26.1	26.101
V_{12}^2	-19.419	0	0	0	0	0	0	3.316	2.601	2.517	2.515
V_{13}^2	-76.14	0	0	0	0	0	0	0	0	0	0
V_{21}^2	-52.661	0	0	0	0	0	0	0	0	0	0
V_{22}^2	27.214	2.168	13.454	19.68	25.023	22.655	`22.655	22.171	22.277	22.289	22.289
V_{23}^2	43.794	21.477	9.915	3.284	-2.809	0	0	0	0	0	0
V_{31}^2	-46.196	0	0	0	0	0	0	0	0	0	0
V_{32}^2	42.97	22.109	12.201	8.002	6.404	6.569	6.567	6.16	6.24	6.25	6.25
V_{33}^2	25.808	0.121	10.05	14.279	15.932	15.734	15.736	16.138	16.059	16.049	16.049

Table V Monopoly (initial starting vector based on isolated flow)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5
V_{11}^1	7.357	7.944	8.721	8.788	8.788
V_{12}^1	10.887	6.879	5.791	5.697	5.697
V_{13}^1	-0.58	0	0	0	0
V_{21}^1	-6.344	0	0	0	0
V_{22}^1	41.686	30.266	30.293	30.295	30.295
V_{23}^1	-5.458	0	0	0	0
V_{31}^1	-0.176	0	0	0	0
V_{32}^1	2.065	2.195	2.196	2.196	2.196
V_{33}^1	25.825	22.653	22.652	22.652	22.652
V_{11}^2	21.862	18.698	18.799	18.801	18.801
V_{12}^2	0.967	1.729	1.598	1.594	1.594
V_{13}^2	1.799	2.542	2.472	2.472	2.472
V_{21}^2	-2.926	0	0	0	0
V_{22}^2	22.806	19.241	19.322	19.324	19.324
V_{23}^2	-1.374	0	0	0	0
V_{31}^2	-1.899	0	0	0	0
V_{32}^2	1.21	2.179	2.027	2.023	2.023
V_{33}^2	22.613	19.763	19.914	19.917	19.917

Table VI Monopoly (all initial starting elements are 400)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6	It. 7	It. 8	It. 9	It. 10	It. 11	It. 12	It. 13	It. 14
V_{11}^1	-387.147	-187.171	-87.228	-37.342	-12.558	-0.407	5.556	11.522	9.129	8.788	8.788	8.788	8.788	8.788
V_{12}^1	200.04	100.107	50.235	25.482	13.451	8.018	6.011	4.809	5.688	5.697	5.697	5.697	5.697	5.697
V_{13}^1	199.994	99.99	49.988	24.985	12.477	6.203	3.005	1.102	0.063	0	0	0	0	0
V_{21}^1	199.968	99.924	49.846	24.697	11.907	5.075	0.618	-11.17	0	0	0	0	0	0
V_{22}^1	-369.452	-169.378	-69.253	-19.014	6.454	19.902	28.32	45.06	30.294	30.295	30.295	30.295	30.295	
V_{23}^1	199.97	99.936	49.881	24.778	12.075	5.41	1.363	-3.945	0	0	0	0	0	0
V_{31}^1	199.963	99.933	49.895	24.834	12.226	5.796	2.395	1.067	-1.012	0	0	0	0	0
V_{32}^1	200.003	100.011	50.029	25.065	12.636	6.523	3.64	2.312	2.25	2.196	2.196	2.196	2.196	2.196
V_{33}^1	-376.215	-176.185	-76.147	-26.081	-0.958	11.782	18.549	23.482	25.163	22.652	22.652	22.652	22.652	22.652
V_{11}^2	-379.146	-179.134	-79.123	-29.108	-4.078	8.496	14.969	19.028	26.92	17.549	18.54	18.78	18.801	18.801
V_{12}^2	200.003	100.011	50.025	25.053	12.606	6.448	3.46	2.002	0.725	1.631	1.555	1.591	1.594	1.594
V_{13}^2	199.988	99.994	50.022	25.084	17.705	6.668	3.858	2.527	0.364	4.938	3.031	2.516	2.472	2.472
V_{21}^2	199.983	99.96	49.92	24.845	12.198	5.666	1.978	-1.031	0	0	0	0	0	0
V_{22}^2	-379.611	-179.583	-79.545	-29.477	-4.348	8.405	15.182	19.696	24.289	19.235	19.327	19.324	19.324	19.324
V_{23}^2	199.974	99.953	49.924	24.876	12.285	5.865	2.424	0.274	-6.252	0	0	0	0	0
V_{31}^2	199.985	99.97	49.945	24.9	12.313	5.894	2.434	0.113	-13.262	0	0	0	0	0
V_{32}^2	199.993	99.999	50.02	25.066	12.654	6.557	3.648	2.208	1.144	2.295	2.049	2.024	2.023	2.023
V_{33}^2	-378.027	-178.017	-78.014	-28.015	-3.07	8.498	15.864	19.619	33.917	15.592	19.878	19.915	19.917	19.917

Table VII Monopoly (all initial starting elements are 0.5)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6
V_{11}^1	6.936	6.947	8.519	8.78	8.788	8.788
V_{12}^1	14.493	8.274	6.074	5.707	5.697	5.697
V_{13}^1	-2.018	0	0	0	0	0
V_{21}^1	-10.12	0	0	0	0	0
V_{22}^1	49.343	30.232	30.286	30.295	30.295	30.295
V_{23}^1	-9.617	0	0	0	0	0
V_{31}^1	-0.342	0	0	0	0	0
V_{32}^1	1.896	2.203	2.194	2.196	2.196	2.196
V_{33}^1	28.79	22.649	22.653	22.652	22.652	22.652
V_{11}^2	25.053	17.499	18.571	18.784	18.801	18.801
V_{12}^2	0.177	3.334	2.01	1.632	1.594	1.594
V_{13}^2	0.58	3.335	2.516	2.466	2.472	2.472
V_{21}^2	-5.257	0	0	0	0	0
V_{22}^2	25.775	18.464	19.159	19.313	19.324	19.324
V_{23}^2	-2.474	0	0	0	0	0
V_{31}^2	-3.94	0	0	0	0	0
V_{32}^2	0.188	3.291	2.183	2.021	2.023	2.023
V_{33}^2	25.658	18.659	19.76	19.92	19.917	19.917

Table VIII Oligopoly (initial starting vector based on isolated flow)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6
V_{11}^1	1.654	4.965	4.223	4.27	4.285	4.285
V_{12}^1	9.203	8.7	9.141	9.127	9.124	9.124
V_{13}^1	5.196	3.166	2.742	2.714	2.707	2.707
V_{21}^1	2.376	-6.245	0	0	0	0
V_{22}^1	24.453	37.19	30.33	30.331	30.331	30.331
V_{23}^1	3.623	-0.783	0	0	0	0
V_{31}^1	4.052	2.163	1.019	0.903	0.851	0.851
V_{32}^1	5.635	3.952	3.765	3.765	3.769	3.769
V_{33}^1	16.702	20.096	20.672	20.738	20.763	20.763
V_{11}^2	13.124	16.263	17.164	18.17	17.389	17.394
V_{12}^2	5.64	3.939	3.256	3.411	3.523	3.522
V_{13}^2	5.512	3.877	3.651	3.058	3.365	3.356
V_{21}^2	3.996	0.912	0.832	-1.01	0	0
V_{22}^2	12.983	18.268	25.517	19.758	19.017	19.017
V_{23}^2	3.828	0.039	-9.711	0	0	0
V_{31}^2	4.541	1.738	0.03	-1.273	0	0
V_{32}^2	6.863	5.933	5.024	5.834	5.858	5.858
V_{33}^2	10.712	14.419	17.094	17.465	16.195	16.196

Table IX Oligopoly (all initial starting elements are 200)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It. 6	It. 7	It. 8	It. 9	It. 10	It. 11	It. 12
V_{11}^{1}	-243.288	-114.365	-50.195	-18.663	-3.839	2.491	8.651	3.982	4.253	4.284	4.285	4.285
V_{12}^{1}	-149.433	74.343	37.18	19.326	11.637	9.247	7.523	9.322	9.133	9.124	9.124	9.124
V_{13}^{1}	149.551	74.355	36.812	18.144	8.991	4.709	2.663	2.768	2.72	2.707	2.707	2.707
V_{21}^{1}	149.553	74.242	36.415	17.162	6.848	0.002	-25.801	0	0	0	0	0
V_{22}^{1}	-264.959	-116.224	-41.624	-3.863	15.951	28.083	57.415	30.329	30.331	30.331	30.331	30.331
V_{23}^{1}	149.54	74.274	36.569	17.573	7.786	2.279	-1.939	0	0	0	0	0
V_{31}^{1}	148.541	72.821	34.984	16.129	6.886	2.782	4.311	1.674	0.97	0.853	0.851	0.851
V_{32}^{1}	149.824	74.768	37.303	18.694	9.624	5.487	3.682	3.698	3.759	3.769	3.769	3.769
V_{33}^{1}	-200.205	-87.201	-30.739	-2.591	11.301	17.868	19.572	20.372	20.705	20.762	20.763	20.763
V_{11}^{2}	-203.373	-90.518	-34.173	-6.159	7.558	13.956	16.613	17.922	17.9	17.392	17.394	17.394
V_{12}^{2}	149.686	74.569	37.089	18.499	9.473	5.38	3.823	3.426	3.477	3.522	3.522	3.522
V_{13}^{2}	149.208	73.868	36.31	17.744	8.847	4.998	3.715	3.174	3.127	3.362	3.356	3.356
V_{21}^{2}	149.358	74.017	36.308	17.384	7.811	2.882	0.433	-1.663	0	0	0	0
V_{22}^{2}	-235.869	-106.823	-42.27	-9.94	6.319	14.634	19.314	20.239	19.02	19.017	19.017	19.017
V_{23}^{2}	149.548	74.29	36.6	17.639	7.945	2.704	-0.9	0	0	0	0	0
V_{31}^{2}	149.681	74.515	36.92	18.1	8.651	3.871	1.373	0.107	-1.474	0	0	0
V_{32}^{2}	149.367	74.157	36.762	18.462	10.012	6.751	5.818	5.714	5.881	5.858	5.858	5.858
V_{33}^{2}	-273.446	-124.902	-50.822	-14.151	3.538	11.495	14.901	16.233	17.613	16.195	16.196	16.196

Table X Oligopoly (all initial starting elements are 0.1)

Variable	It. 1	It. 2	It. 3	It. 4	It. 5	It.6
V_{11}^1	-6.787	-1.606	3.246	4.234	4.285	4.285
V_{12}^1	29.699	15.866	10.431	9.197	9.125	9.124
V_{13}^1	0.203	3.979	2.807	2.703	2.707	2.707
V_{21}^1	-16.006	0	0	0	0	0
V_{22}^1	51.191	30.223	30.311	30.33	30.331	30.331
V_{23}^1	-5.718	0	0	0	0	0
V_{31}^1	4.001	2.014	1.087	0.864	0.851	0.851
V_{32}^1	10.509	5.823	4.095	3.78	3.769	3.769
V_{33}^1	17.874	18.836	20.457	20.752	20.763	20.763
V_{11}^2	14.828	16.878	17.929	17.391	17.394	17.394
V_{12}^2	8.034	4.668	3.551	3.521	3.522	3.522
V_{13}^2	3.439	3.045	3.085	3.364	3.356	3.356
V_{21}^2	2.769	0.395	-1.656	0	0	0
V_{22}^2	19.639	18.449	30.211	19.017	19.017	19.017
V_{23}^2	-5.513	0	0	0	0	0
V_{31}^2	-1.695	0	0	0	0	0
V_{32}^2	10.712	6.715	5.755	5.858	5.858	5.858
V_{33}^2	13.179	15.363	16.298	16.195	16.196	16.196

Table XI : Equilibrium solutions before commodities compete in the market

Variable	Pure -competition	Oligopoly	Monopoly
V_{11}^{1}	6.358	4.252	8.785
V_{12}^{1}	8.862	9.136	5.701
V_{13}^{1}	1.509	2.741	
V_{21}^{1}			
V_{22}^{1}	30.728	30.330	30.295
V_{23}^{1}			
V_{31}^{1}		0.852	
V_{32}^{1}	2.204	3.806	2.203
V_{33}^{1}	29.694	20.736	22.648
V_{11}^{2}	26.077	17.332	18.796
V_{12}^{2}	2.551	3.537	1.605
V_{13}^{2}		3.465	2.47
V_{21}^{2}			
V_{22}^{2}	22.279	19.008	19.313
V_{23}^{2}			
V_{31}^{2}			
V_{32}^{2}	6.276	5.905	2.05
V_{33}^{2}	16.023	16.148	19.891

Table XII : Equilibrium solutions when commodities compete in the market

Variable	Pure -competition	Oligopoly	Monopoly
V_{11}^{1}	6.373	4.285	8.788
V_{12}^{1}	8.85	9.124	5.697
V_{13}^{1}	1.502	2.707	
V_{21}^{1}			
V_{22}^{1}	30.729	30.331	30.295
V_{23}^{1}			
V_{31}^{1}		0.851	
V_{32}^{1}	2.139	3.769	2.196
V_{33}^{1}	29.74	20.763	22.652
V_{11}^{2}	26.101	17.394	18.801
V_{12}^{2}	2.515	3.522	1.594
V_{13}^{2}		3.356	2.472
V_{21}^{2}			
V_{22}^{2}	22.289	19.017	19.324
V_{23}^{2}			
V_{31}^{2}			
V_{32}^{2}	6.25	5858	2.023
V_{33}^{2}	16.049	16.196	19.917

Table XIII : Result of Competition

Variable	Pure -competition	Oligopoly	Monopoly
V_{11}^{1}	0.2%	0.8%	0.035%
V_{12}^{1}	-0.1%	-0.1%	-0.076%
V_{13}^{1}	-0.5%	-0.13%	
V_{21}^{1}			
V_{22}^{1}	0.0017%	0.019%	0.001%
V_{23}^{1}			
V_{31}^{1}		-0.095%	
V_{32}^{1}	-2.9%	-1%	-0.3%
V_{33}^{1}	0.2%	0.1%	0.018%
V_{11}^{2}	0.092%	0.3%	0.025%
V_{12}^{2}	-1.4%	-0.5%	-0.7%
V_{13}^{2}		-2.2%	0.065%
V_{21}^{2}			
V_{22}^{2}	0.045%	0.048%	0.056%
V_{23}^{2}			
V_{31}^{2}			
V_{32}^{2}	-0.4%	-0.8%	-1.3%
V_{33}^{2}	0.2%	0.3%	0.1%

V. Conclusion

5.1 Introduction

This thesis has demonstrated that the same basic solution algorithm can be used to solve three conceptually different models of spatial competition. As we all know, the modeling involves lots of simplification of reality. Thus, all models are flawed. Besides, no one set of economic assumptions completely describes the workings of the economic system under study; the market may exhibit traits of both perfect and imperfect competitions. Thus, in making predictions about the future state of such an economic system, we cannot rely on any one model.

5.2 Research Summary

- Analysis of the problem domain.

  Before developing the algorithm, an intensive study was accomplished to gain an understanding of the terms, concepts, and philosophies for spatial price equilibrium problem.

- Market Structure

  We provided an overview of market structure in order for the reader to have better understanding, and clear concepts for the difference among these three different models.

- Detailed Expanding Algorithm

  The algorithm we developed was based on the expanding algorithm for single linear spatial price equilibrium[47]. Our algorithm uses Expanding

Algorithm as a series subproblem. We also offered the most important tool, network data structure, with which we can solve large-scale problems.

- Implementation and testing

  The implementation stage was to create an algorithm for General spatial price equilibrium problem(could be multi-commodity, and nonlinear shipping cost function), and provided numerical examples for all three of these models to test the alorithm.

5.3 Future research

First, the models presented in this thesis are static, and hence do not introduce entry/exit issues. If firms are making economic profits, it is very likely that new firms will enter the market and perturb the established economic equilibrium of supplies, demands and flows. Introducing entry/exit issues is a fruitful area of research in that it not only impacts policy modeling, but is also useful in facility location decisions.

Second, all of the spatial models that are currently available assume $c_{ij}^r(V_{ij}^r)$ is a constant or increasing function. But, in reality, there exists shipping cost function, as freight rate discounts for large shipments make this a decreasing function of V_{ij}^r, which leads to nonconvex optimization problems. Can uniqueness be assured with a weaker condition than strict monotonicity? Can convergence of the solution algorithms be shown under weaker conditions than those presented?

Third, the simple supply and demand functions that are used in GSPE may not be capable of capturing complex market behaviors, Future research must be directed towards the inclusion of more sophisticated supply/demand models.

Fourth, besides theoretical challenges, clearly the equilibrium process is computationally intensive. Further research could possibly reduce the computational complexity of the solution of GSPE by a large amount.

Finally, we cannot help but emphasize the importance of transportation, especially in an oligopoly market. Transportation is a vital part of all economy. It not only affects the availability and prices of goods sold at market, but it also has a major impact on energy usage, national defense matters, and many other national concerns. The growth of any economy is limited without adequate transportation support.

5.4 Conclusion

In conclusion, this thesis documented an algorithm based on the expanding algorithm (Jones[47]) to solve general spatial price equilibria for all three models. This work forms the baseline for future efforts for different models. This algorithm has shown that it not only solves all three of the models without any change, but it is also easy to implement.

It is very helpful to develop an algorithm on MATHCAD. It not only provides me everything that I need, but it is also visible. That is the reason why I picked it, especially, in my personal situation. MATHCAD has made it possible for me to implement the algorithm in urgent time. It indeed bought me a lot of time.

Appendix A Data Structures for Network Program (Kennington[25])

Labels for Rooted Trees

Let $\Im=[\Re, \wp]$ be a rooted tree with root node l. There is a unique path linking any node i $\neq$ l to node l, and we denote this path by P(i). Node i will be called a successor of node n, if n is in P(i). We denote the set of successors of node n by U(n) and the number of successors of node n by t_n.

We define a label for $\Im$ to be a mapping with domain $\Re$. The distance label,denoted by d_i, is given as follows :

$$d_i = \begin{cases} 0 & if \quad i = l \\ length \quad of \quad P(i) & otherwise \end{cases}$$

The predecessor label, denoted by P_i , is given by

$$P_i = \begin{cases} 0 & if \quad i = 1 \\ P(i) & otherwise \end{cases}$$

For any one-to-one mapping from $\Re$ onto $\Re$, say s_i, we define the family of maps by the recursion

$$s^l(i) = s_i,$$

$$s^{j+1} (i) = s^j (s_i).$$

Then s_i is called a thread label if U(i) = { s^j (i) : j = 1, ..., t_i }, when $t_i \neq 0$. For a given rooted tree, many such maps can typically be defined. Given a thread label, the preorder distance label, denoted by g_i , is a mapping from $\Re$ to $\Re$ such that

$$g_i = \begin{cases} 1 & i = 1 \\ j+1 & i = s^j(l) \end{cases}$$

Given a thread label, the last successor label, denoted by n_i, is given by

$$n_i = \begin{cases} i & if \quad U(i) = \phi \\ s^j(i) \quad otherwise; & s^j(i) \in U(i), and\, s^{j+1}(i) \notin U(i) \end{cases}$$

To illustrate these mappings, Table A.1 gives the labels for the rooted tree of Figure B.1.

the data structures that follow all represent $\Im$ using the predecessor and the thread labels plus various combinations of the other labels. Note

Table XIV Labels for The Rooted Tree of Figure I

Node i	Distance d_i	Predecessor P_i	Thread s_i	Last Successor n_i	No. of Successor t_i	Preorder Distance g_i
1	1	11	2	10	10	2
2	2	1	3	10	9	3
3	3	2	4	7	5	4
4	4	3	5	4	1	5
5	4	3	6	6	2	6
6	5	5	7	6	1	7
7	4	3	8	7	1	8
8	3	2	9	8	1	9
9	3	2	10	10	2	10
10	4	9	11	10	1	11
11	0	0	1	10	11	1

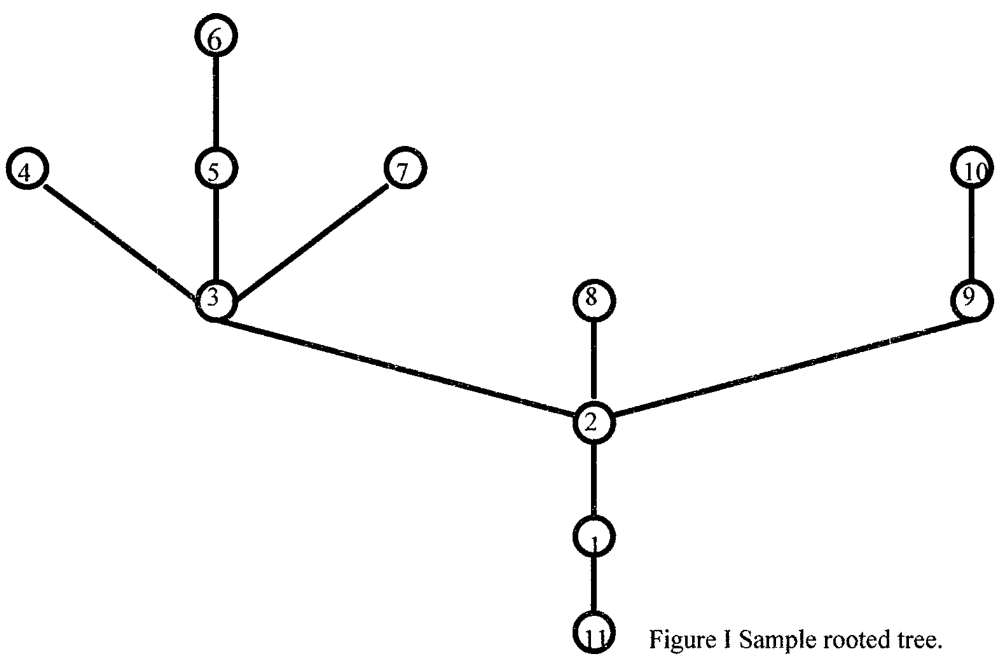


Figure I Sample rooted tree.

that each label used in the data structure requires a node-length array. furthermore, in general it is true that an efficient implementation using a data structure with k+1 node -length arrays will result in faster solution times than one with only k such arrays. Hence, in the absence of budgetary and other design restrictions, the appropriate data structure for a given problem is a function of the core storage available. We now present the data structures and corresponding algorithms for implementation using two,three,and four node-length arrays for representing $\mathfrak{I}$ has $\bar{I}$ - 1 arcs and the root arc. Therefore the pertinent information about the arcs is also carried in node-length arrays, where for $i \neq 1$ the information concerning the arc connecting nodes i and P_i is associated with node i. Suppose arc e_k connects nodes i and P_i . To facilitate the computations, we make use of an oriented arc identifier, m_i , which is defined to be k if $e_k = (P_i, i)$ and -k if $e_k = (i, P_i)$. The flow on e_k is denoted by α_i. To implement the pricing operation, it is desirable to maintain the values of the dual variables. Thus three additional node-length arrays are required, which may also be considered as labels.

Appnedix B . Convex & Concave & Positive Definite

B1. Convex and Concave Functions

Convex and concave functions play an important role in optimization problems. These functions naturally arise in linear optimization problems when dealing with parametric analysis. A function f of the vector $(x_1, x_2, \ldots, x_n)$ is said to be convex if the following inequality holds for any two vectors x_1 and x_2 :

$$f(\lambda x_1 + (1-\lambda)x_2) \le \lambda f(x_1) + (1-\lambda) f(x_2) \quad \text{for all} \quad \lambda \in [\,0, 1\,]$$

Figure II shows an example of a convex function. Note that the foregoing inequality can be interpreted as follows : $\lambda f(x_1) + (1-\lambda) f(x_2)$ where $\lambda \in [\,0, 1\,]$ represents the height of the chord joining $(x_1, f(x_1))$ and $(x_2, f(x_2))$ at the point $\lambda x_1 + (1-\lambda)x_2$. Since $f(\lambda x_1 + (1-\lambda)x_2) \le \lambda f(x_1) + (1-\lambda) f(x_2)$, then the height of the chord is at least as large as the height of the function itself.

A function f is concave if and only if -f is convex. This can be restated as follows :

$$f(\lambda x_1 + (1-\lambda)x_2) \ge \lambda f(x_1) + (1-\lambda) f(x_2) \quad \text{for all} \quad \lambda \in [\,0, 1\,]$$

for any given x_1 and x_2 Figure III shows an example of a concave function. An example of a function that is neither convex nor concave is depicted in Figure IV

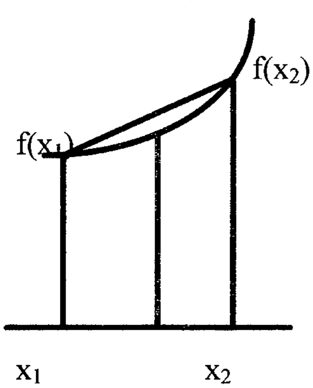


Figure II

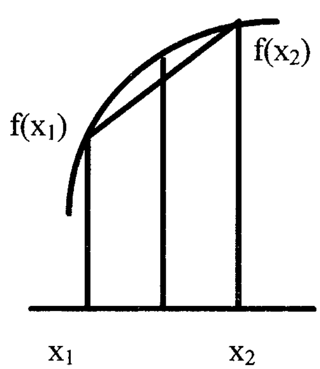


Figure III

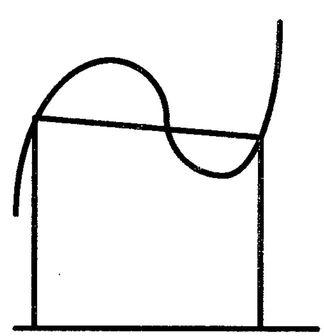


Figure IV

B.2 Positive Definite

It is known from matrix algebra that the quadratic form of Eq.(B.1) or (B.2) will be positive for all **h** if and only if **[J]** is positive definite at $\mathbf{X} = \mathbf{X}^*$. This means that a sufficient condition for the stationary point $\mathbf{X}^*$ to be a relative minimum is that the Hessian matrix evaluated at the same point be positive. This completes the proof for the minimization case. By proceeding in a similar manner, it can be proved that the Hessian matrix will be negative definite if $\mathbf{X}^*$ is a relative maximum point.

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{n} h_i h_j \frac{\partial^2 f}{\partial x_i \partial x_j}\Big|_{x=x^*} \tag{B.1}$$

$$Q = \mathbf{h}^T \mathbf{J}\, \mathbf{h} \,\big|_{\mathbf{X}=\mathbf{X}^*} \tag{B.2}$$

where

$$\mathbf{J}\,\big|_{\mathbf{X}=\mathbf{X}^*} = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\Big|_{X=X^*} \right] \tag{B.3}$$

is the matrix of second partial derivatives and is called the Hessian matrix of f(**X**).

Note: A matrix **A** will be positive definite if all its eigenvalues are positive; that is , all the values of λ that satisfy the determinantal equation

$$|\mathbf{A} - \lambda \mathbf{I}| = 0 \tag{B.4}$$

should be positive. Similarly, the matrix [**A**] will be negative definite if its eigenvalues are negative.

Another test that can be used to find the positive definiteness of a matrix **A** of order n involves evaluation of the determinants

$$\mathbf{A}_1 = |a_{11}| \qquad \mathbf{A}_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

$$\mathbf{A}_3 = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}, \ldots, \qquad \mathbf{A}_n = \begin{vmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & & a_{nn} \end{vmatrix}$$

The matrix **A** will be positive definite if and only if all the values $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, ... , $\mathbf{A}_n$ are positive. The matrix **A** will be negative definite if and only if the sign of $\mathbf{A}_j$ is $(-1)^j$ for j = 1,2, . . . , n. If some of the $\mathbf{A}_j$ are positive and the remaining $\mathbf{A}_j$ are zero, the matrix **A** will be positive semidefinite.

Appendix C. Variational Inequality & Complementarity Problems

C.1 Varational Inequality Problem

The problem of finding $x^* \in k$ such that

$F(x^*)^T(x - x^*) \geq 0$ for all $x \in k$

where $F(x): k \rightarrow R^n$, $k \subset R^n$, is a variational inequality problem (VIP)

Let $F : R^n \rightarrow R^n$ be continuous, $g : R^n \rightarrow R^m$ be differentiable, and $h : R^n \rightarrow R^p$ be linear affine. Let

$$k = \{x \in R^n | g(x) \geq 0, h(x) = 0\}$$

We then want to find a solution x^* to the variational inequality

$$F(x^*)^T(x - x^*) \geq 0 \text{ for all } x \in k \qquad \text{(C.1)}$$

Theorem 1 (Necessary conditions for solution). If the vector $x^* \in k$ is a solution to the variational inequality (C.1) and the gradients $\nabla g_i(x^*)$,for i such that $g_i(x^*) = 0$, and $\nabla h_i(x^*)$,for i = 1, . . ., p, are linearly independent, then there exists $\lambda \in R^m$ and $\mu \in R^p$ such that

$$F(x^*) - \nabla g(x^*)^T\lambda - \nabla h(x^*)^T\mu = 0 \qquad \text{(C.2)}$$

$$\lambda^T g(x^*) = 0 \qquad \text{(C.3)}$$

$$\lambda \geq 0 \qquad \text{(C.4)}$$

Theorem 2 (Sufficient conditions for solution). If $g_i(x)$ for i = 1, ... , m are concave and $x^* \in k$, $\lambda^* \in R^m$ and $\mu^* \in R^p$ satisfy (C.2), (C.3) and (C.4), then x^* is a solution to the variational inequality (C.1).

Theorem 3 (Sufficient conditions for a locally unique solution). If the conditions of Theorem 2 hold and in addition if F is differentiable and

$y^T \nabla F(x^*) y > 0 \quad \text{for all } \; y \neq 0$

such that

$$\nabla g_i(x^*) y \geq 0 \quad \text{for all } \; i \text{ such that } g_i(x^*) = 0$$

$$\nabla g_i(x^*) y = 0 \quad \text{for all } \; i \text{ such that } \lambda^* > 0$$

$$\nabla h_i(x^*) y = 0 \quad \text{for } \; i = 1, ..., p,$$

then x^* is a locally unique solution to variational inequality (C.1).

C.2 COMPLEMENTARITY PROBLEM

The problem of finding $x \in R^n$ such that

$$F(\mathbf{X})^T \mathbf{X} = \mathbf{0}$$

$$F(\mathbf{X}) \geq 0$$

$$\mathbf{X} \geq 0$$

Appendix D. GINO & GRG2 (general reduced gradient 2)

D.1 GINO

GINO is a modeling program which can be used to solve optimization problems and sets of simultaneous linear and nonlinear equations and inequalities. Thus, GINO can be used to solve problems in many areas such as resource allocation, strategic planning, economic analysis, and engineering design and analysis. GINO has the capability to not only evaluate formulate but also to run a formula backwards. Actually, GINO can solve simultaneous equations, inequality relations, and in addition can maximize or minimize the value of a specified variable (so-called optimization).

D.2 GRG2 (General Reduced Gradient 2)

GRG2, the portion of GINO which solves the model, uses a version of the generalized reduced gradient (GRG) algorithm. GRG was first developed in the late 1960's by Jean Abadie, and has since been refined by several other researchers. This section discusses the fundamental ideas of GRG and describes the version of GRG that is implemented in GRG2. More complete information regarding GRG ideas and the structure of GRG2 is contained in the following references: Abadie (1978), Lasdon,Waren, Jain, and Ratner (1978),

The generalized reduced gradient (GRG) method is an extension of the reduced gradient method that was presented originally for solving problems with linear constraints only[D.11]. To see the details of the GRG method, consider the nonlinear programming problem:

$$\text{Minimize } f(X) \tag{D.1}$$

subject to

$$h_j(X) \le 0, \qquad j = 1,2,\ldots,m \tag{D.2}$$

$$l_k(X) = 0, \qquad k = 1,2,\ldots,l \tag{D.3}$$

$$x_i^{(l)} \le x_i \le x_i^{(u)} \qquad i = 1,2,\ldots,n \tag{D.4}$$

By adding a nonnegative slack variable to each of the inequality constraints in Eq.(D.2), the problem can be stated as

$$\text{Minimize } f(X) \tag{D.5}$$

subject to

$$h_j(X) + x_{n+1} = 0, \qquad j = 1,2,\ldots,m \tag{D.6}$$

$$h_k(X) = 0, \qquad k = 1,2,\ldots,l \tag{D.7}$$

$$x_i^{(l)} \le x_i \le x_i^{(u)} \qquad i = 1,2,\ldots,n \tag{D.8}$$

$$x_{n+1} \ge 0, \qquad j = 1,2,\ldots,m \tag{D.9}$$

with n + m variables ($x_1, x_2,\ldots, x_n, x_{n+1},\ldots, x_{n+m}$). The problem can be rewritten in a general form as:

$$\text{Minimize } f(X) \tag{D.10}$$

subject to

$$g_j(X) = 0, \qquad k = 1,2,\ldots,m+l \tag{D.11}$$

$$x_i^{(l)} \le x_i \le x_i^{(u)} \qquad i = 1,2,\ldots,n+m \tag{D.12}$$

where the lower and upper bounds on the slack variable, x_i are taken as 0 and a large number (infinity), respectively ($i = n+1,n+2,\ldots,n+m$). The GRG method is based on the idea of elimination of variables using the equality constraints. Thus, theoretically, one variable can be reduced from the set x_i ($i = 1,2,\ldots,n+m$) for each of the m + l equality constraints given by Eqs. (D.6) and (D.7). It is convenient to divide the n+m design variables arbitrarily into two set as

$$X = \begin{bmatrix} Y \\ Z \end{bmatrix} \tag{D.13}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \text{design or independent variables} \tag{D.14}$$

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{m+1} \end{bmatrix} = \text{state or dependent variables} \tag{D.15}$$

and where the design variables are completely independent and the state variables are dependent on the design variables used to satisfy the constraints $g_j(X) = 0$, $j = 1,2,\ldots,m+1$. Consider the first variations of the objective and constraint functions:

$$df(X) = \sum_{i=1}^{n-1} \frac{\partial f}{\partial y_i} dy_i + \sum_{i=1}^{m+1} \frac{\partial f}{\partial z_i} dz_i = \nabla_Y^T f dY + \nabla_Z^T f dZ \tag{D.16}$$

$$dg_i(X) = \sum_{j=1}^{n-1} \frac{\partial g_i}{\partial y_j} dy_j + \sum_{j=1}^{m+1} \frac{\partial g_i}{\partial z_j} dz_j \tag{D.17}$$

or $$dg = [C]dY + [D]dZ \tag{D.18}$$

where

$$\nabla_Y f = \begin{bmatrix} \frac{\partial f}{\partial y_1} \\ \frac{\partial f}{\partial y_2} \\ \vdots \\ \frac{\partial f}{\partial y_{n-1}} \end{bmatrix} \tag{D.19}$$

$$\nabla_Z f = \begin{bmatrix} \frac{\partial f}{\partial z_1} \\ \frac{\partial f}{\partial z_2} \\ \vdots \\ \frac{\partial f}{\partial z_{m+1}} \end{bmatrix} \tag{D.20}$$

$$[C] = \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & \cdots & \frac{\partial g_1}{\partial y_{n-1}} \\ \vdots & & \vdots \\ \frac{\partial g_{m+1}}{\partial y_1} & \cdots & \frac{\partial g_{m+1}}{\partial y_{n-1}} \end{bmatrix} \tag{D.21}$$

$$[D] = \begin{bmatrix} \frac{\partial g_1}{\partial z_1} & \cdots & \frac{\partial g_1}{\partial z_{n-1}} \\ \vdots & & \vdots \\ \frac{\partial g_{m+1}}{\partial z_1} & \cdots & \frac{\partial g_{m+1}}{\partial z_{n-1}} \end{bmatrix} \tag{D.22}$$

$$dY = \begin{bmatrix} dy_1 \\ dy_2 \\ \vdots \\ dy_{n-1} \end{bmatrix} \tag{D.23}$$

$$dZ = \begin{bmatrix} dz_1 \\ dz_2 \\ \vdots \\ dz_{m+1} \end{bmatrix} \tag{D.24}$$

Assuming that the constraints are originally satisfied at the vector X, (g(X) = 0),any change in the vector dX must correspond to dg = 0 to maintain feasibility at X+dX. Eq.(D.17) can be solved to express dZ as

$$dZ = -[D]^{-1}[C]dY \tag{D.25}$$

The change in the objective function due to the change in X is given by Eq.(D.16), which can be expressed, using Eq.(D.24), as

$$df(X) = (\nabla_Y^T f - \nabla_Z^T [D]^{-1}[C])dY \tag{D.26}$$

or

$$\frac{df}{dY}(X) = G_R \tag{D.27}$$

where

$$G_R = \nabla_Y f - ([D]^{-1}[C])\nabla_Z f \tag{D.28}$$

is called the generalized reduced gradient. Geometrically, the reduced gradient can be described as a projection of the original n-dimensional gradient onto the (n-m) dimensional feasible region described by the design variables.

We know that a necessary condition is that the components of the gradient vanish. Similarly, a constrained function assumes its minimum value when the appropriate components of the reduced gradient are zero. This condition can be verified to be the same as the Kuhn-tucker conditions to be satisfied at a relative minimum. In fact, the reduced gradient G_R can be used to generate a search direction S to reduce the value of the constrained objective function similar to the gradient ∇f that can be used to generate a search direction S for an unconstrained function. A suitable step length λ

is to be chosen to minimize the value of f along the search Z is updated using Eq.(D.24). Noting that Eq(D.24) is based on using a linear approximation to the original nonlinear problem, we find that the constraints may not be exactly equal to zero at λ, that is, $dg \neq 0$. Hence, when Y is held fixed, in order to have

$$g_i(X) + dg_i(X) = 0 \qquad i = 1,2, \ldots, m+l \qquad \text{(D.29)}$$

we must have

$$g(X) + dg\ (X) = 0 \qquad \text{(D.30)}$$

Using Eq.(D.17) for dg in Eq.(D.29), we obtain

$$dZ = [D]^{-1}(-g(X) - [C]dY) \qquad \text{(D.31)}$$

The value of dZ given by Eq.(D.30) id used to update the value of Z as

$$Z_{update} = Z_{current} + dZ \qquad \text{(D.32)}$$

The constraints evaluated at the updated vector X, and the procedure [of finding dZ using Eq. (D.31)] is repeated until dZ is sufficiently small. Note that Eq.(D.31) can be considered as Newton's method of solving simultaneous equations for dZ.

Algorithm :

1. Specify the design and state variables. Start with an initial trial vector X. Identify design and state variables (Y and dZ) for the problem using the following guidelines.
   (a) The state variables are to be selected to avoid singularity of the matrix, [D].
   (b) Since the state variables are adjusted during the iterative process to maintain feasibility, any component of X that is equal to its lower or upper bound initially is to be designated a design variable.

(c) Since the slack variables appear as linear terms in the (originally inequality) constraints, they should be designated as state variables. However, if the initial value of any state variable is zero (its lower bound value), it should be designated a design variable. ¢

2. Compute the generalized reduced gradient. The GRG is determined using Eq.(D.27) can be evaluated numerically, if necessary.

3. Test for convergence. If all the components of the GRG are close to zero, the method can be considered to have converged and the current vector X can be taken as the optimum solution of the problem. For this , the following test can be used:

$$\|G_R\| \le \varepsilon$$

where ε is a small number. If this relation is not satisfied, we go to step 4.

4. Determine the search direction. The GRG can be used similar to a gradient of an unconstrained objective function to generate a suitable search direction, S. The techniques such as steepest descent, Fletcher-Reeves, Davidon- Fletcher-Powell, or Broydon-Fletcher-Goldfarb-Shanno methods can be used for this purpose. For example, if a steepest descent method is used, the vector S is determined as

$$S = - G_R \qquad \text{(D.33)}$$

5. Find the minimum along the search direction. Although any of the one-dimensional minimization procedures can be used to find a local minimum of f along the search direction S, the following procedure can be used conveniently.

(a) Find an estimate for λ as the distance to the nearest side constraint. When design variables are considered, we have

$$\lambda = \begin{cases} \dfrac{y_i^{(u)} - (y_i)_{old}}{s_i} & if \quad s_i > 0 \\ \dfrac{y_i^{(l)} - (y_i)_{old}}{s_i} & if \quad s_i < 0 \end{cases} \tag{D.34}$$

where s_i is the ith component of S. Similarly, when state variables are considered, we have from Eq.(D.24)

$$dZ = -[D]^{-1}[C]dY \tag{D.35}$$

Using dY = λS, Eq.(D.35) gives the search direction for the variables Z as

$$T = -[D]^{-1}[C]S \tag{D.36}$$

Thus

$$\lambda = \begin{cases} \dfrac{z_i^{(u)} - (z_i)_{old}}{t_i} & if \quad t_i > 0 \\ \dfrac{z_i^{(l)} - (z_i)_{old}}{t_i} & if \quad t_i < 0 \end{cases} \tag{D.37}$$

where t_i is the ith component of T.

(b) The minimum value of λ given by Eq.(D.34), λ_1, makes some design variable attain its lower or upper bound. Similarly, the minimum value of λ given Eq. (D.34), λ_2,will make some state variable attain its lower or upper bound. The smaller of λ_1 or λ_2 can be used as an upper bound on the value of λ for initializing a suitable one-dimensional minimization procedure. The quadratic interpolation method can be used conveniently for finding the optimal step length λ^*.

(c) Find the new vector X_{new}:

$$X_{new} = \begin{Bmatrix} Y_{old} + dY \\ Z_{old} + dZ \end{Bmatrix} = \begin{Bmatrix} Y_{old} + \lambda^{*} S \\ Z_{old} + \lambda^{*} T \end{Bmatrix} \tag{D.38}$$

If the vector X_{new} corresponding to λ^* is found infeasible, the Y_{new} is held constant and Z_{new} is modified using Eq.(D.31) with $dZ = Z_{new} - Z_{old}$. Finally, when convergence is achieved with Eq.(D.31), we find that

$$X_{new} = \begin{Bmatrix} Y_{old} + \Delta Y \\ Z_{old} + \Delta Z \end{Bmatrix} \tag{D.39}$$

and go to step 1.

Appendix E. Numerical Example for Oligopoly (Regions: 3; Commodities: 2)

ORIGIN ≡ 1

parameters:

$$\alpha := \begin{pmatrix} 1 & 2 \\ 2 & 1.5 \\ 1.5 & 1 \end{pmatrix} \quad \sigma := \begin{pmatrix} 19 & 27 \\ 27 & 30 \\ 30 & 19 \end{pmatrix} \quad \mu_1 := \begin{pmatrix} 0 & .1 & .4 \\ .2 & 0 & .3 \\ .1 & .4 & 0 \end{pmatrix} \quad \phi_1 := \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 1 & 4 & 0 \end{pmatrix}$$

$$\beta := \begin{pmatrix} .5 & .3 \\ .4 & .5 \\ .3 & .4 \end{pmatrix} \quad \delta := \begin{pmatrix} .2 & .3 \\ .01 & .2 \\ .3 & .01 \end{pmatrix} \quad \mu_2 := \begin{pmatrix} 0 & .4 & .1 \\ .3 & 0 & .2 \\ .4 & .1 & 0 \end{pmatrix} \quad \phi_2 := \begin{pmatrix} 0 & 2 & 1 \\ 3 & 0 & 1 \\ 4 & 1 & 0 \end{pmatrix} \quad \omega := \begin{pmatrix} 0 & 0.01 & 0.03 \\ 0.01 & 0 & 0.04 \\ 0.03 & 0.02 & 0 \end{pmatrix}$$

Reg : amount of regions,

Com : amount of commodities,

$(\theta^r)_i$: demand price per unit of commodity r at region i,

$(C^r)_i$: total production cost of commodity r at firm i,

$(c^r)_{i,j}$: shipping cost per unit of commodity r form firm i to region j,

$(\psi^r)_i$:supply price per unit of commodity r at firm i,

$(S^r)_i$: amount of supply of commodity r from firm i,

$(D^r)_i$: amount of demand of commodity r at region i,

$$(\theta^r)_i = (\sigma^r)_i - (\delta^r)_i \cdot (D^r)_i \qquad \text{demand function}$$

$$(C^r)_i = (\alpha^r)_i \cdot (S^r)_i + (\beta^r)_i \cdot \left[(S^r)_i\right]^2 \qquad \text{supply function}$$

$$(c^r)_{i,j} = (\phi^r)_{i,j} + (\mu^r)_{i,j} \cdot \left[(V^r)_{i,j}\right]^2 + \sum_{(i \neq r)} (\omega^i)_{i,j} \cdot (V^i)_{i,j} \qquad \text{shipping cost function}$$

Reg := rows(α) Com := cols(α) r := 1 .. Com·Reg s := 1 .. Com·Reg

Problem :

MAX

$$\sum_{r=1}^{Com}\sum_{i=1}^{Reg}\left(\theta^r\right)_i\cdot(D^r)_i - \sum_{r=1}^{Com}\sum_{i=1}^{Reg}(C^r)_i - \sum_{r=1}^{Com}\sum_{i=1}^{Reg}\sum_{j=1}^{Reg}(c^r)_{i,j}\cdot(V^r)_{i,j}$$

subject to

$$\sum_{r=1}^{Com}\sum_{i=1}^{Reg}(S^r)_i - \sum_{r=1}^{Com}\sum_{i=1}^{Reg}(D^r)_i = 0$$

$$\sum_{i=1}^{Reg}(S^r)_i - \sum_{i=1}^{Reg}(D^r)_i = 0 \qquad \text{for all r}$$

$$(V^r)_{i,j}\cdot\left[\left[(MR^r)_{i,i} + (c^r)_{i,j}\right] - (MR^r)_{j,i}\right] = 0$$

$$(V^r)_{i,j} \geq 0 \qquad \text{for all (i,j), and r}$$

where

$$(S^r)_i = \sum_{j=1}^{Reg}(V^r)_{i,j}$$

$$(D^r)_i = \sum_{i=1}^{Reg}(V^r)_{i,j}$$

$$(MR^r)_{j,i} = \left[\frac{d}{dV_{i,j}}\left[\left(\theta^r\right)_j\cdot(D^r)_j\right]\right] = \left(\sigma^r\right)_j - \left(\delta^r\right)_j\cdot\left[(D^r)_j + (V^r)_{i,j}\right]$$

$$(MR^r)_{i,i} = \left(\alpha^r\right)_i + 2\cdot\left(\beta^r\right)_i\cdot(S^r)_i \qquad \text{if i = j}$$

Initial guess : flows := Isolated_initial

Iteration 1

flows := multi(flows)

SOL_2_1 := SOL(flows)

flows =

1.654	9.203	5.196	0	0	0
2.376	24.453	3.623	0	0	0
4.052	5.635	16.702	0	0	0
0	0	0	13.124	5.64	5.512
0	0	0	3.996	12.983	3.828
0	0	0	4.541	6.863	10.712

Iteration 2

flows := multi(flows)

SOL_2_2 := SOL(flows)

flows =

4.965	8.7	3.166	0	0	0
0	37.19	0	0	0	0
2.163	3.952	20.096	0	0	0
0	0	0	16.263	3.939	3.877
0	0	0	0.912	18.268	0.039
0	0	0	1.738	5.933	14.419

Iteration 3

flows := multi(flows)

SOL_2_3 := SOL(flows)

flows =

4.223	9.141	2.742	0	0	0
0	30.33	0	0	0	0
1.019	3.765	20.672	0	0	0
0	0	0	17.164	3.256	3.651
0	0	0	0.832	25.517	0
0	0	0	0.03	5.024	17.094

Iteration 4

flows := multi(flows)

SOL_2$_4$:= SOL(flows)

flows =

4.27	9.127	2.714	0	0	0
0	30.331	0	0	0	0
0.903	3.765	20.738	0	0	0
0	0	0	18.17	3.411	3.058
0	0	0	0	19.758	0
0	0	0	0	5.834	17.465

Iteration 5

flows := multi(flows)

SOL_2$_5$:= SOL(flows)

flows =

4.285	9.124	2.707	0	0	0
0	30.331	0	0	0	0
0.851	3.769	20.763	0	0	0
0	0	0	17.389	3.523	3.365
0	0	0	0	19.017	0
0	0	0	0	5.858	16.195

Iteration 6

flows := multi(flows)

SOL_2$_6$:= SOL(flows)

flows =

4.285	9.124	2.707	0	0	0
0	30.331	0	0	0	0
0.851	3.769	20.763	0	0	0
0	0	0	17.394	3.522	3.356
0	0	0	0	19.017	0
0	0	0	0	5.858	16.196

$$\text{Flag}(1, \text{flows}) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \qquad \text{Flag}(2, \text{flows}) = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Flow_M := flows

Iteration(SOL_2) := rows(SOL_2)   Iteration(SOL_2) = 6

Tab2 := Tab(SOL_2, Iteration(SOL_2))   Conv2 := Conv(percent(Tab(SOL_2, Iteration(SOL_2))))

Table XV. Results of Iterations

Tab2 =

	1	2	3	4	5	6	7	8
1	4.965	8.7	3.166	-6.245	37.19	-0.783	2.163	3.952
2	4.223	9.141	2.742	0	30.33	0	1.019	3.765
3	4.27	9.127	2.714	0	30.331	0	0.903	3.765
4	4.285	9.124	2.707	0	30.331	0	0.851	3.769
5	4.285	9.124	2.707	0	30.331	0	0.851	3.769
6	4.285	9.124	2.707	0	30.331	0	0.851	3.769

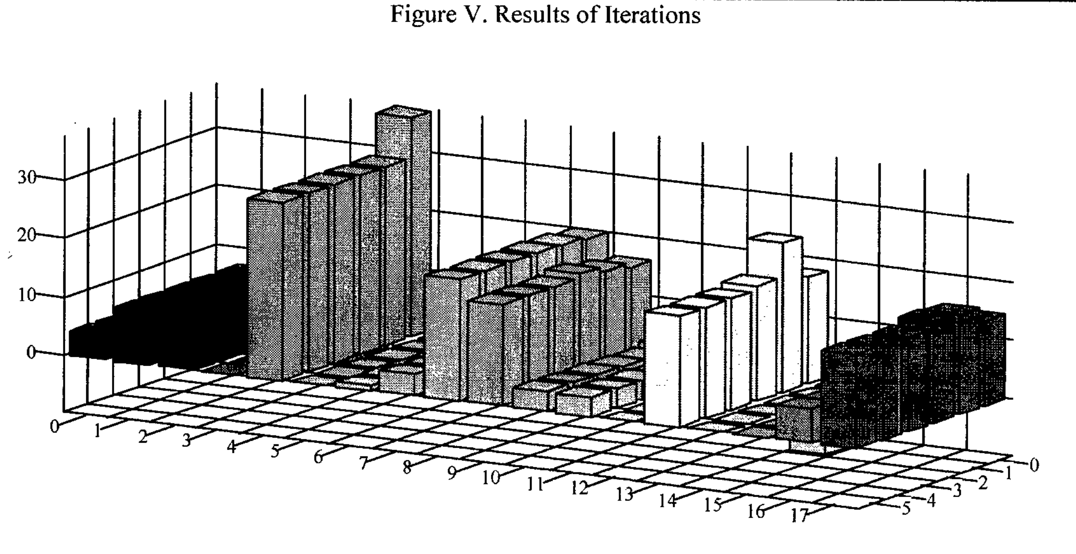
$$Tab2^{<9>} = \begin{bmatrix} 20.096 \\ 20.672 \\ 20.738 \\ 20.763 \\ 20.763 \\ 20.763 \end{bmatrix}$$

Tab2 =

	10	11	12	13	14	15	16	17
1	16.263	3.939	3.877	0.912	18.268	0.039	1.738	5.933
2	17.164	3.256	3.651	0.832	25.517	-9.711	0.03	5.024
3	18.17	3.411	3.058	-1.01	19.758	0	-1.273	5.834
4	17.389	3.523	3.365	0	19.017	0	0	5.858
5	17.394	3.522	3.356	0	19.017	0	0	5.858
6	17.394	3.522	3.356	0	19.017	0	0	5.858

$$Tab2^{<18>} = \begin{bmatrix} 14.419 \\ 17.094 \\ 17.465 \\ 16.195 \\ 16.196 \\ 16.196 \end{bmatrix}$$

Figure V. Results of Iterations



Tab(SOL_2, Iteration(SOL_2))

Table XVI. Results of Convergence

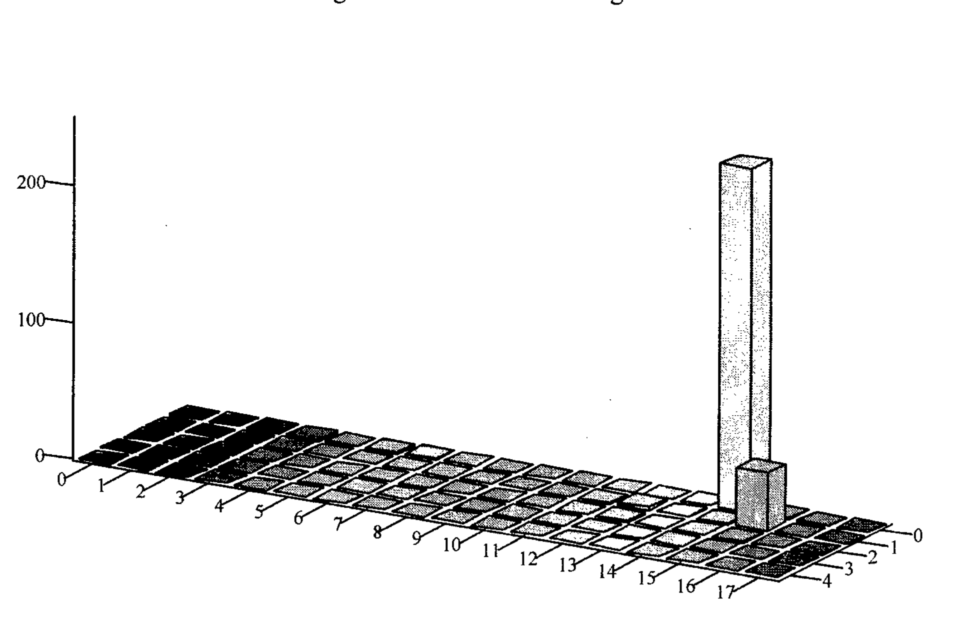
Conv2 =

	1	2	3	4	5	6	7	8	9
1	0.149	0.051	0.134	-1	0.184	-1	0.529	0.047	0.029
2	0.011	0.002	0.01	0	0	0	0.114	0	0.003
3	0.003	0	0.003	0	0	0	0.057	0.001	0.001
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0

Conv2 =

	10	11	12	13	14	15	16	17
1	0.055	0.173	0.058	0.088	0.397	252.368	0.982	0.153
2	0.059	0.047	0.162	2.214	0.226	-1	42.76	0.161
3	0.043	0.033	0.101	-1	0.037	0	-1	0.004
4	0	0	0.003	0	0	0	0	0
5	0	0	0	0	0	0	0	0

Figure VI. Results of Convergence



Conv(percent(Tab(SOL_2, Iteration(SOL_2))))

Flows of commodities 1 and 2 (before they compete in the market)

$$Flow\_S_1 := \begin{pmatrix} 4.252 & 9.136 & 2.741 \\ 0 & 30.330 & 0 \\ 0.852 & 3.806 & 20.736 \end{pmatrix} \qquad Flow\_S_2 := \begin{pmatrix} 17.347 & 3.539 & 3.433 \\ 0 & 19.008 & 0 \\ 0 & 5.904 & 16.149 \end{pmatrix}$$

$$Pre(Flow\_S) = \begin{bmatrix} 4.252 & 9.136 & 2.741 & 0 & 0 & 0 \\ 0 & 30.33 & 0 & 0 & 0 & 0 \\ 0.852 & 3.806 & 20.736 & 0 & 0 & 0 \\ 0 & 0 & 0 & 17.347 & 3.539 & 3.433 \\ 0 & 0 & 0 & 0 & 19.008 & 0 \\ 0 & 0 & 0 & 0 & 5.904 & 16.149 \end{bmatrix}$$

$P := Pre(Flow\_S)$

Comparsion:

Commodity 1 :

Single

$Profit(1,P) = 835.616$

$Total(1,P) = 71.853$

$$S(1,P) = \begin{pmatrix} 16.129 \\ 30.33 \\ 25.394 \end{pmatrix} \quad D(1,P) = \begin{pmatrix} 5.104 \\ 43.272 \\ 23.477 \end{pmatrix} \quad \psi(1,P) = \begin{pmatrix} 17.129 \\ 26.264 \\ 16.736 \end{pmatrix} \quad \theta(1,P) = \begin{pmatrix} 17.979 \\ 26.567 \\ 22.957 \end{pmatrix}$$

Multi

$Profit(1,Flow\_M) = 836.656$

$Total(1,Flow\_M) = 71.83$

$$S(1,Flow\_M) = \begin{pmatrix} 16.116 \\ 30.331 \\ 25.384 \end{pmatrix} \quad D(1,Flow\_M) = \begin{pmatrix} 5.136 \\ 43.224 \\ 23.47 \end{pmatrix} \quad \psi(1,Flow\_M) = \begin{pmatrix} 17.116 \\ 26.264 \\ 16.73 \end{pmatrix} \quad \theta(1,Flow\_M) = \begin{pmatrix} 17.973 \\ 26.568 \\ 22.959 \end{pmatrix}$$

Commodity 2 :

Single

$Profit(2,P) = 726.317$

$Total(2,P) = 65.38$

$$S(2,P) = \begin{pmatrix} 24.319 \\ 19.008 \\ 22.053 \end{pmatrix} \quad D(2,P) = \begin{pmatrix} 17.347 \\ 28.451 \\ 19.582 \end{pmatrix} \quad \psi(2,P) = \begin{pmatrix} 16.591 \\ 20.508 \\ 18.642 \end{pmatrix} \quad \theta(2,P) = \begin{pmatrix} 21.796 \\ 24.31 \\ 18.804 \end{pmatrix}$$

Multi

$Profit(2,Flow\_M) = 727.286$

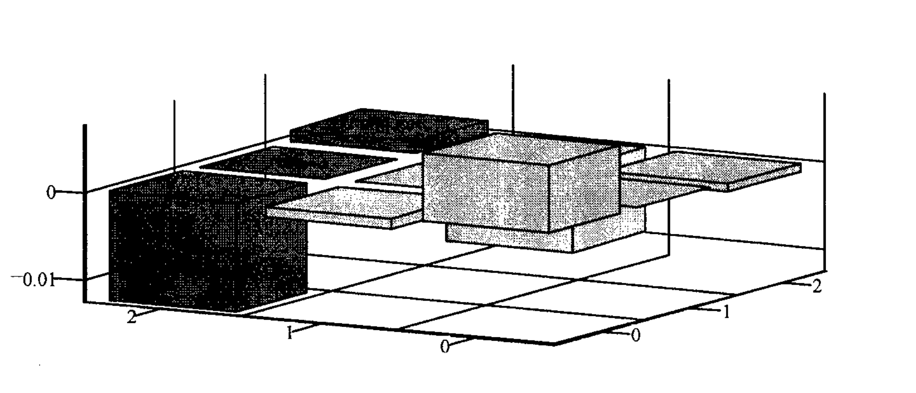
$Total(2,Flow\_M) = 65.343$

$$S(2,Flow\_M) = \begin{pmatrix} 24.272 \\ 19.017 \\ 22.053 \end{pmatrix} \quad D(2,Flow\_M) = \begin{pmatrix} 17.394 \\ 28.397 \\ 19.552 \end{pmatrix} \quad \psi(2,Flow\_M) = \begin{pmatrix} 16.563 \\ 20.517 \\ 18.643 \end{pmatrix} \quad \theta(2,Flow\_M) = \begin{pmatrix} 21.782 \\ 24.321 \\ 18.804 \end{pmatrix}$$

Results of Congestion:

$$\text{Congestion}(1,\text{Flow\_M},\text{Pre}(\text{Flow\_S})) = \begin{bmatrix} 0.008 & -0.001 & -0.013 \\ 0 & 1.866\cdot 10^{-5} & 0 \\ -9.471\cdot 10^{-4} & -0.01 & 0.001 \end{bmatrix}$$

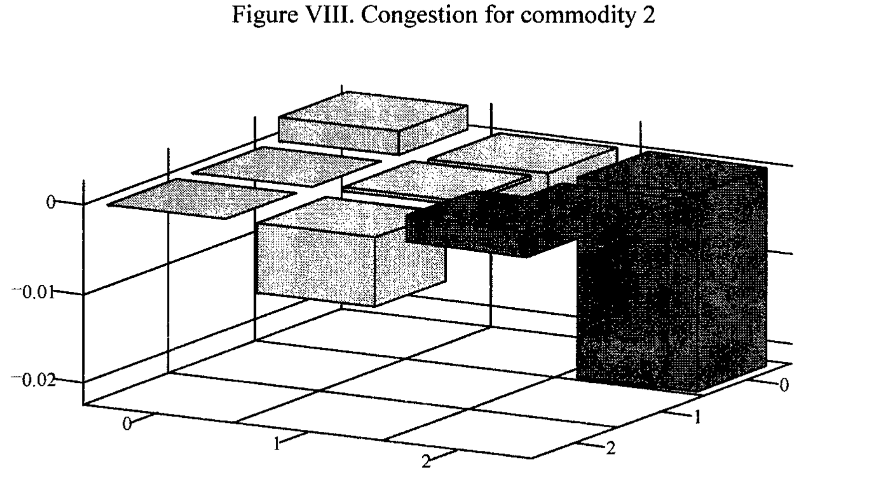
Figure VII. Congestion for commodity 1



Congestion(1, Flow_M, P)

$$\text{Congestion}(2,\text{Flow\_M},\text{Pre}(\text{Flow\_S})) = \begin{bmatrix} 0.003 & -0.005 & -0.022 \\ 0 & 4.842\cdot 10^{-4} & 0 \\ 0 & -0.008 & 0.003 \end{bmatrix}$$

Figure VIII. Congestion for commodity 2



Congestion(2, Flow_M, P)

Bibliography

1. Abdulaal, Mustafa and Larry J. LeBlanc. (1979) "Methods of Combining Modal Split and Equilibrium Assignment Models" Transportation Science 13, pp. 292-314.

2. Ahn, B. H. (1981) "Solution of Nonsymmetric Linear Complementarity Problems by Iterative Methods" Journal of Optimization Theory and Applications 33, pp. 175-185.

3. Ahn, B. H. and Seong, B. S. (1984) "A parametric network method for computing nonlinear spatial equilibria" Department of Management Science, Korea advanced Institute of Science and Technology (Seoul, Korea).

4. Aoyagi, M; Okabe, A (1993) "Spatial competition of firms in a two-dimensional bounded market" Regional Science and Urban Economics 23, pp. 259-289.

5. Barondes, C; Chan, Y (1991) "User Vs system optimization of network flow: Multicriteria decision-making" Working paper; Department of Operational Science, Air Force Institute of Technology, Wright Patterson AFB, Ohio.

6. Brandeau, M L; Chiu, S S (1994) "Location of competing facilities in a user-optimizing environment with market externalities" Transportation Science 28, pp. 125-140.

7. C. R. Glassey (1978) "*A* Quadratic Network Optimization Model for Equilibrium Single Commodity Trade Flows" Mathematical Programming 14, pp. 98-107.

8. Dafermos, S. (1983) "*An* iterative scheme for variational inequalities"

Mathematical Programming 26, pp. 40 - 47.

9. Dafermos, S., A. Naguarney (1984) "Sensitivity Analysis for the General Spatial Equilibrium Problem" Opns. Res. 32, pp. 1069-1086.

10. Dafermos, S., and A. Nagurney (1987) "Oligopolistic and Competitive Behavior of Spatially Separated Markets" Reg. Sci. and Urban Econ. 17, pp. 245-254.

11. Dafermos, S., and A. Nagurney (1989) "Supply and Demand Equilibration Algorithms for a Class of Market Equilibrium Problems" Trans. Sci. 23, pp. 118-124.

12. Dantzig, D. G. (1951) "Programming of Interdependent Activities: Mathematical Model" in Koopmans, T. C. "Analysis of Production and Allocation" John Willy and Sons, New York.

13. Fang, S. C. (1979) "An Iterative Method for Generalized Complementarity Problems" Mathematics Research Report 79-11, Department of Mathematics, University of Maryland, Cantonsville, MD.

14. Florian, M., and M. Los. (1982) "A New Look at Static Spatial Price Equilibrium Models" Reg. Sci. and Urban Econ.12, pp. 579-597.

15. Friesz, Terry L., Roger L. Tobin (1981) "An Equivalent Convex Optimization Problem for Network Equilibrium with Derived Demand" Report CUE-FNEM-1981-9-1, Department of Civil and Urban Engineering, University of Pennsylvania.

16. Friesz, Terry L., Roger L. Tobin, and Patrick T. Harker (1981) "Variational Inequalities and Convergence of Diagonalization Methods for Derived Demand Network Equilibrium Problems" Report CUE-FNEM-1981-10-1, Department of

Civil and Urban Engineering, University of Pennsylvania.

17. Friesz, T L; Harker, P T (1983) "Multicriteria spatial price equilibrium network design: theory and computational results" Trans. Research 17B, pp. 411- 426.

18. Harker, P T (1986) "Alternative models of spatial competition" Operations Research 34, No 3.

19. Infante-Macias, R; Munoz-perez, J (1995) "Competitive location with rectilinear distances" European Journal of Operational Research 80, pp. 77-85.

20. J. S. Pang and D. Chan (1982) "Iterative Methods for Variational and Complementarity Problems" Mathematical Programming 24, pp. 284-313.

21. J. S. Pang and P. S. C. Lee (1981) "A Parametric Linear Complementarity Technique for the Computation of Equilibrium Prices in a Single Commodity Spatial Model" Mathematical Programming 20, pp. 81-102.

22. J. S. Pang (1984) "Solution of the General Multi-commodity Spatial Equilibrium Problem by Variational and Complememtarity Methods" Journal of Regional Science 24, pp. 405-414.

23. J. S. Pang (1981) "A Hybrid Method for the Solution of Some Multi-Commodity Spatial Equilibrium Problems" Management Science 27, pp. 1142-1157.

24. Jones, P C; Theise, E S (1989) "On the equivalence of competitive transportation markets an congestion in spatial price equilibrium models" Transportation Science 23, pp. 112-117.

25. J. L. Kennington and R. V. Helgason (1980) "Algorithms for Network

Programming" Wiley , New York.

26. Knoblauch, V (1991) "Generalizing location games to a graph" The Journal of Industrial Economics 39, pp. 683-688.

27. Koopmans, T. C. (1949) "Optimal Utilization of the Transport System" Econometrica 17, pp. 136 - 146.

28. MacKinnon, J. G. (1975) "An Algorithm for the Generalized Transportation Problem" Regional Science and Urban economics 5, pp. 445 - 464.

29. Mamagasarian, O. L. (1977) "Solution of Symmetric Linear Complementarity Problems by Iterative Methods" Journal of Optimization Theory and applications 22, pp. 465-485.

30. M. H. Schneider (1986b) "The Expanding Equilibrium Algorithm" ORSA/TIMS Meeting Bulletin 22.

31. Miller, T C; Tobin, R L; Friesz, T L (1991) "Stackelberg games on a network with Cournot-Nash oligopolistic competitors" Journal of Regional Science 31, pp. 435-454.

32. Miller, T; Tobin, R L; Friesz, T L (1992) "Network facility-location models in Stackelberg-Nash-Cournot spatial competition" Papers in Regional Science 71, pp. 277-291.

33. Nagurney, A. (1986) "An Algorithm for the Single Commodity Spatial Price Equilibrium Problem" Regional science and Urban Economics 16, pp. 573-588.

34. Nagurney, A. (1987b) "Competitive Equilibrium Problems ,Variational Inequalities, and Regional Science" J. Reg. Sci. 27, pp. 55-76.

35. Nagurney, A; Aronson, J (1989) "A general dynamic spatial price network equilibrium model with gains and losses" Networks 21, pp. 109-132.

36. Nagurney,A; Zhao, L (1991) "A network equilibrium formulation of market disequilibrium and variational inequalities" Networks 21, pp. 109-132.

37. P. A. Samuelson (1952) "Spatial Price Equilibrium and Linear Programming" American Economic Review 42, pp. 283-303.

38. P. C. Jones, R. Saigal, and M. H. Schneider (1985) "Demand homotopies for computing nonlinear and multi-commodity spatial equilibria" Lecture Notes in Economics and Mathematical systems 249, pp. 118 - 135.

39. P. C. Jones, R. Saigal, and M. H. Schneider (1985a) "Computing Nonlinear Network Equilibria" Mathematical Programming 31, pp. 57-66.

40. P. C. Jones, G. Morrison, J. C. Swarts , and E. S. Theise (1987) "Nonlinear Spatial Equilibrium Algorithms: A Computational Comparison" Department of Industrial Engineering and Management Sciences Technical Report # 87-11. Northwestern University, Evanston, IL.

41. P. C. Jones, R. Saigal, and M. H. Schneider (1984) "A Variable Dimension Homotopy for Computing spatial Equilibria" Operations Research Letters 3, pp. 19-24.

42. Qiu, Y. (1991) "Solution Properties of Oligopolistic Network Equilibria" Networks 71, pp. 565-580.

43. Rowse, John. "Solving the Generalized Transportation Problem" Regional Science

and Urban Economics 11, pp. 57 - 68.

44. Samuelson, P. A. (1952) "A Spatial price equilibrium and linear programming" Am. Econ. Rev. 42, pp. 283-303.

45. S. Dafermos and A. Nagurney (1984) "A Network Formulation of Market Equilibrium Problems and Variational Inequalities" Operations Research Letters 3, pp. 247-250.

46. S. Enke (1951) "Equilibrium among Spatially Separated Markets : Solution by Electric Analogue" Econometrica 19, pp. 40 -47.

47. Theise, Eric S (1987) "A Turbo Pascal Implementation of the Expanding Equilibrium Algorithm" Department of Industrial Engineering and Management Sciences Technical Report 87-04, Northwestern University, Evanston, IL.

48. Theise, E S; Jones, P C (1988) "Alternative implementations of a diagonalization algorithm for multiple commodity spatial price equilibria" Technical Report 88-06 Department of Industrial Engineering and Management Sciences,Northwestern University, Evanston, IL.

49. Takayama, Takashi and George G. Judge. (1964b) "Equilibrium among Spatially Separated Markets: A Reformulation" Econometrica 32, pp. 510-524.

50. Takayama, Takashi and George G. Judge. (1971) "Spatial and Temporal Price and Allocation Models" North-Holland, Amsterdam.

51. T. L. Friesz, R. L. Tobin, and P. T. Harker (1984) "Alternative Algorithms for the General Network Spatial Price Equilibrium Problem" J. of Reg. Sci. 24, pp. 475-507.

52. T. L. Friesz, R. L. Tobin, T. E. Smith, and P. T. Harker (1983) "A Nonlinear Complementarity Formulation and Solution Procedure for the General Derived Demand Network Equilibrium Problem" J. of Reg. Sci. 23, pp. 337 - 359.

53. Theise, E S; Jones, P C (1988) "Nonlinear, single commodity spatial price equilibria and the expanding equilibrium algorithm: Two strategies for implementation" Technical report 88-16, Department of Industrial Engineering and Management Science, Northwestern university, Evanston, IL.

54. Tobin, Roger L. and T. L. Friesz (1983) "Formulating and Solving the Derived Demand Network Equilibrium Problem in Terms of Arc Variables" Journal of Regional Science 23, pp. 187-198.

55. Tobin, Roger L. (1988) "A Variable-dimension Solution Approach for the General Spatial Price Equilibrium Problem" Mathematical Programming 40, pp. 33 - 51.

56. Thore,S., Nagurney, A. and J. Pan. (1992) "General economic equilibrium and variational inequalities" O.R. Letts.10, pp. 369-376.

57. Udomkesmalee, M. (1985) "Modles and algorithms for spatial price equilibria with congestion" Ph.D. Dissertation, Department of Industrial Engineering, Northwestern University.

58. Vickerman, R.W. (1980) "Spatial Economic Behavior" St. Martin's, New

59. Wardrop, J G (1952) "Some theoretical aspects of road traffic research" Proceedings of the Institute of Civil Engineers, Part 2, United Kingdom. pp. 325-378.

Vita

Major Yu-Shen Ke was born on 18 August 1963, Taiwan, Republic of China (R.O.C.). He received a Bachelor of Science in Engineering Science degree at the National Cheng Kung University in Tainan, a southern city of Taiwan, in June 1989. After graduation, he worked as a teacher in CAF. In Jan 1995, he was accepted into the R.O.C. Air Force Education Program and entered the School of Engineering at Air Force Institute of Technology (AFIT) in Jan, 1995. Two weeks before coming to United States, Major Ke got married to Hui-Hung(Julia) Lin .

Permanent address: 32 SheaDa St. KungSang

Kaohsiung, Taiwan, R.O.C.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1997

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Alternative Implementations of Expanding Algorithm for Multi-Commodity Spatial Price Equilibrium

5. FUNDING NUMBERS

6. AUTHOR(S)
Yu-Shen Ke, Maj. ROCAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFIT/ENS
2950 Street
Wright-Patterson AFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/97M-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of Defense (R55)
9800 Savage Road
Fort George CT. Meade, MD
20755-6000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This thesis presents an algorithm based on the expanding algorithm to solve spatial price equilibrium problems for three different models: perfect competition, monopoly, oligopoly. The expanding algorithm (Jones[44]) is used to solve the linear single commodity spatial price equilibrium problem for perfectly competitive markets. In this thesis, we consider the general multi-commodity spatial price equilibrium problems with all nonlinear functions, and variable shipping cost. We also show that more commodities in total are shipped, and there is more congestion, especially in oligopoly model.

14. SUBJECT TERMS
spatial price equilibrium, perfect competition, monopoly, oligopoly, perfectly competitive market, multi-commodity, congestion

15. NUMBER OF PAGES
86

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL