

A Study on Resource Allocation Optimization in a Multi-commodity Virtual Enterprise Network

Jayeola Femi OPADIJI¹⁾ and Toshiya KAIHARA²⁾

¹⁾Division of Systems Design, Graduate School of Science & Technology, Kobe University

²⁾Department of Computer & Systems Engineering, Kobe University

Received on 29th August 2006

Abstract

Finding an optimal solution to various constrained optimization problems in the supply network of organizations is clearly a Herculean task considering the fact that some of the objectives to be optimized have conflicting requirements. A popular method used in tackling issues aimed at improving the performance of a supply network is to focus on a particular planning horizon and explore activities where improvements can be made. Optimization problems within a supply network vary from strategic to operational domains and a number of models have been developed to address these issues including probabilistic models, mathematical algorithms and some meta-heuristic methodologies. We propose a strategic decision support model based on Multi-Agent System (MAS) paradigm to find a pareto-optimal resource allocation among a group of trading agents in a multi-commodity market scenario. We employ the price mechanism model in microeconomics to create a trading floor for a set of reactive agents to search for a solution to a resource allocation problem in a virtual enterprise network. While the concept of Competitive General Equilibrium (CGE) is not new in the neo-classical economic domain, its application to solving resource allocation problem within a supply chain structure may not be as common. We first study some existing algorithms for CGE based on the Walrasian market model and then introduce an adaptation of the Market-Oriented Programming (MOP) paradigm to the supply network problem. We compare the performance of these algorithms and discuss application areas of the competitive market based approach.

Keywords: Virtual Enterprise Network, Multi-Agent System, Computable General Equilibrium, Market-Oriented Programming, Resource Allocation

1 Introduction

Time-based competition among firms has resulted in them looking for new ways to improve the flow of resources in their value chain. Over the years, such methods of improvement have evolved along with the changing nature of various functional units and processes in these organizations. In fact, the concept of Business Process Re-engineering (BPR) [1, 2] has shifted the focus away from the various functional units that make up an organization, to the process that lead to the creation of value. A focus on these processes necessitates both an efficient integration of these functional units as well as a seamless coordination of flows among them. These two main challenges form the crux of the problem facing supply network managers today [2].

Finding an optimal solution to various constrained optimization problems in the supply network of organizations is clearly a Herculean task considering the fact that some of the objectives to be optimized have conflicting requirements. A popular method used in tackling issues aimed at improving the performance of a supply network is to focus on a particular planning horizon and explore activities where improvements can be made. Optimization problems within a supply network vary from strategic to operational domains and a number of models have been developed to address these issues including probabilistic models, mathematical algorithms and some meta-heuristic methodologies [3, 4, 5]. However, most of the mathematical models which are used in solving the supply chain optimization appear to be very rigid and may not take into account some dynamic properties of the supply network. These dynamic properties stem from the autonomous nature of the various units that make up the supply network. One way in which researchers have addressed this problem is to turn the field of Distributed Artificial Intelligence (DAI) [6] in order to develop new heuristics that will aid in better understanding the inner workings of

various supply network configurations and to seek better solutions for the multi-objective supply network problem.

Our focus is on the use of Multi-Agent System (MAS) architecture in developing a framework for resource allocation problems within a supply network. In using the MAS paradigm, characteristics of autonomous units operating in an environment as well as the properties of such an environment have to be clearly drawn out. Other issues, such as mechanisms for interaction and communication among the agents and with the environment have to be defined. The field of MAS is still in the developmental stage but it is rapidly growing and a number of literatures are available to introduce the fundamentals of MAS-based architectures [6, 7].

This research work proposes a strategic decision support model based on Multi-Agent System (MAS) paradigm to find a pareto-allocation of resources among a group of trading agents in a multi-commodity market scenario. We employ the price mechanism model in microeconomics [8] to create a trading floor for a set of reactive agents to search for a solution to a resource allocation problem in a virtual enterprise network. Due to the competitive market structure used in our model, we define our agents as purely reactive agents (price-takers) which have unbounded rationality in terms of trading quantity decisions and whose decisions are rule-based [9]. While the concept of Competitive General Equilibrium (CGE) [8, 10] is not new in neo-classical economics domain, its application for solving resource allocation problem within a supply chain structure may not be as common. We first studied some existing algorithms for CGE based on the Walrasian market model, and then we introduce an adaptation of the Market-Oriented Programming (MOP) [9, 11] paradigm to the supply network problem. We compare the performance of these algorithms and discuss application areas of the competitive market based approach.

In the next section, we discuss fundamental issues in Computable General Equilibrium (CGE) theory and introduce some previous analytical methods used in finding equilibrium price vector among a set of commodities in a competitive market. The third section describes in detail our research methodology while in the succeeding section, experimental procedures and simulations for both our proposed model and a chosen analytical process are discussed. Results of simulations carried out are reported in the fifth section of this paper. This section is also devoted to a comparison between the two methods used; we also check the performance of our algorithm in changing environmental conditions. We conclude this paper by discussing future direction of research activities using the proposed model in the strategic planning of different supply network configurations. Mention is also made of some of the limitations of this model as it relates to the operation planning domain which may not accommodate some of the assumptions made in the market model.

2 Computable General Equilibrium (CGE)

The goal of a CGE model is to find an equilibrium price vector P at which both the quantities of supply and demand of an array of goods in a competitive market is equal. The equilibrium point derived from the price vector is also used to determine the equilibrium allocation of resources among the traders in the market. The notion of the existence of a unique equilibrium price vector for a multi-commodity market was first explored by Leon Walras in the nineteenth century [11]. His idea gave birth to what is now known as the Walrasian market. The Walrasian market is based on some fundamental axioms of gross substitutability among the goods in the market, profit and utility maximization among trading agents which are often referred to as weak axioms in neo-classical economics. Formally, CGE model is defined by:

$g_i \subset G \forall i = 1, 2, 3 \dots n$ where G is a set of goods in the market and $|G| = n$
 g_i is associated with $p_i \subset P \forall i = 1, 2, 3 \dots n$ given that P is a vector of prices

$x_j \subset X \forall j = 1, 2, 3 \dots m$: X is a set of trading agents in the market

$e_k^j \subset E \forall k = 1, 2, 3 \dots n$ and e_k^j = endowment of trader j on market resource k and $|E| = m \times n$.

Our goal is to find a price vector $p^* \in P$ for which

$$\sum_{i=1}^m d_i^g = \sum_{i=1}^m s_i^g \text{ for all } g \in G (\text{set of goods}) \text{ and } m \text{ is the total number of trading agents} \quad (1)$$

i.e., the point where demand is equal to supply for all market resources

In this formulation, we assume a pure exchange economy where trading is the only form of economic activity.

2.1 Walrasian Market

The Walrasian market is based on two social accounting rules: conservation of products and conservation of values. Conservation of products refer to a condition where all the factors of production in an economy are absorbed for the production of goods and services by trading agents called suppliers while all the goods produced by these supplier agents are consumed by trading agents called consumers. Conservation of value simply requires that income in the economy must be equal to expenditure. These two conditions can be represented by a social accounting matrix (SAM).

Given a set of n consumers and m suppliers in an economy with k goods, budget of a consumer c is represented as:

$$B^c = \sum_{i=1}^k p_i e_i^c \quad (2)$$

Where $p_i \in P \forall i = 1, 2, 3, \dots, k$ is price of resource i and $e_i^c \in E \forall i = 1, 2, 3, \dots, k$ is the endowment of agent c on resource i .

The constrained optimization problem for a consumer agent c is:

$$\max U^c(g_1, g_2, g_3, \dots, g_k) \quad (3)$$

Where $U(\cdot)$ is utility function

Subject to a budget constraint given as:

$$\sum_i p_i g_i^c \leq B^c \quad (4)$$

and a non-negative price constraint

$$p_i \geq 0 \text{ for all } p \in P \forall 1 \leq i \leq k \text{ where } P \text{ is price vector} \quad (5)$$

The rationality of suppliers dictates that they seek to maximize profit. Therefore, the constraint optimization problem of a supplier s is:

$$\max \pi^s = p_s g_s - \sum_{\substack{i=1 \\ i \neq s}}^k p_i g_i \quad (6)$$

Where π^s is profit of supplier s

Subject to technology of supplier:

$$T^s(g_1, g_2, \dots, g_{s-1}, g_{s+1}, \dots, g_k) = g_s \quad (7)$$

and non-negative demand constraint

$$g_i \geq 0 \text{ for all } g \in G \forall 1 \leq i \leq k \quad (8)$$

2.2 Finding Equilibrium Price

A number of computational methods can be used to solve the constraint optimization problems of equations (3) – (8) as long as the utility and technology functions are well-behaved, that is, each function is both

monotonic and homogenous [8]. Price tatonnement algorithm [11] and Scarf algorithm [12, 13 and 14] are popular methods used in solving for equilibrium price vector.

The “price tatonnement” algorithm was proposed by Leon Walras as a decentralized relaxation method based on conventional price adjustment. Prices of individual products are adjusted upward or downward depending on excess demand of that in the economy. The simple price adjustment formula used is:

$$P_{t+1} = P_t + \alpha \left(\sum_{i=1}^n x_i - \sum_{i=1}^n e_i \right) \quad (9)$$

Where P_t = price of market resource at iteration t , α = price adjustment constant, x_i = market demand of product by trader i and e_i is market supply of product by trader i .

The speed of convergence to equilibrium price and the accuracy of the results depend on how small the value of α is. The smaller the value of α , the slower the convergence but the higher the accuracy of the results.

Scarf Algorithm is based on the Fixed-point theorem. This algorithm uses a function $f(\cdot)$ to map a simplex (whose corners are price vectors) into itself. The mapping seeks to find a point in the geometric space in which all the corner points of the simplex gives a complete mapping of the price vector into the simplex. The number of divisions of the simplex determines both the accuracy of the results as well as the speed of convergence. The higher the number of divisions of the simplex, the higher the accuracy and the slower the speed of convergence.

The existence of a unique equilibrium price p^* requires monotonicity in the preferences of traders. This condition means that if the preferences of traders are revealed, the marginal rate of substitution (MRS) must be the same for all trading agents for market clearing to occur [8], i.e. $MRS_i = C \quad \forall 1 \leq i \leq k$ where C is a constant positive value. The MRS is formally defined in equation (10) below.

$$\frac{dg_i}{dg_j} = - \frac{\frac{\partial u(\mathbf{g})}{\partial g_j}}{\frac{\partial u(\mathbf{g})}{\partial g_i}} \quad (10)$$

Where $u(\mathbf{g})$ = utility on all market resources and g_i = demand of resource i

3 Research Methodology

As stated earlier, in this research work, we employ the MAS paradigm in solving for equilibrium price vector in a walrasian market. This therefore requires that we define our agents, the nature of the environment and the interaction protocol for the system. Since the focus of this research is to use the price mechanism in the walrasian market model for optimal resource allocation in a supply network, we also define the structure of our supply network and remodel it to take up the form of a competitive market in order to search for the unique solution which we require.

3.1 Trading Agents

Consumer agent and supplier agent architectures are defined as an extended class of a super class trading agents as shown in figure 1 below:

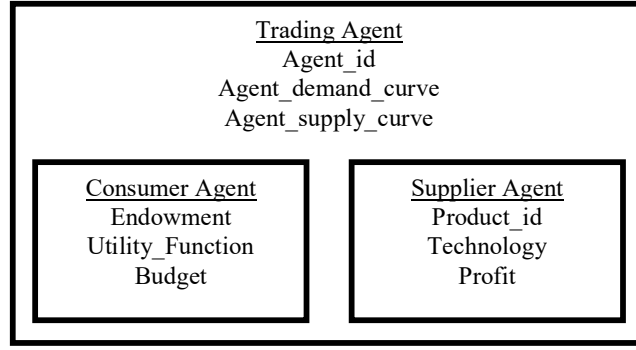


Figure 1: Trading Agent Class

Given market conditions as defined in section 2.1, we use Cobb-Douglas function to represent the utility function of the consumer and also the technology function of the supplier. The Cobb-Douglas function is a special case of the Constant Elasticity of Substitution (CES) function having a Technical Rate of Substitution (TRS) = 0. We assume that each trading agent as a constant returns to scale.

3.1.1 Consumer Agent

A consumer agent is treated as a special producer that produces a special type of good called utility U and bids for economic goods by solving its constraint optimization problem:

$$\max U^c = R_c \prod_{i=1}^k g_i^{\alpha_i^c} \quad (11)$$

$$\text{Given that } \sum_{i=1}^k \alpha_i^c = 1 \text{ for all } 1 \leq c \leq n \quad (12)$$

Subject to

$$\sum_{i=1}^k p_i e_i^c \leq B^c \quad (13)$$

$$\text{and } p_i \geq 0 \text{ for all } p \in P \quad (14)$$

Where R_c = consumption scale of consumer c

α_i^c = preference index of consumer c of good i

p_i = price of good i

e_i^c = endowment of consumer c of good i

g_i = bid of good i by consumer c

Solving the constraint optimization problem using the method of Lagrange multiplier [8], the bidding function for a consumer agent as:

$$g_i(p_i) = \frac{\alpha_i^c * B^c}{p_i} \quad (15)$$

3.1.2 Supplier Agent

Like the consumer agent, a supplier agent bids for goods in the economy by solving its constraint optimization problem:

$$\max \pi^s(P) = p_s g_s - \sum_{\substack{i=1 \\ i \neq s}}^k p_i g_i \quad (16)$$

Subject to

$$g_s(P) = R_s \prod_{\substack{i=1 \\ i \neq s}}^k g_i^{\beta_i^s} \quad (17)$$

Given that $\sum_{i=1}^k \beta_i^s = 1$ for all $1 \leq s \leq m$ (18)

Using same method as for the consumer agent problem, the bidding function for a supplier agent as:

$$g_i(p_i) = \left(\frac{p_i}{R_s \beta_i^s p_s} \right)^{\frac{1}{\beta_i^s - 1}} \quad (19)$$

Where β_i^s = technology index of good i for agent s

Both the consumer agents and the supplier agents are constructed as reactive agents (price-takers) and are stimulated by the current price vector in the market as computed by the market mechanism. The rules for actions are the bidding functions. The agents do not interact directly because they are in a competitive state. They only interact with the market mechanism.

3.2 Market Mechanism

Figure 2 illustrates the market mechanism. Operating under the axioms stated in the Walrasian market structure, all consumer agents supply all their endowments to the market. The suppliers make use of these endowments to produce goods which the consumers use to produce their utility. Going by the conservation of value axiom, all the profits made by the suppliers are returned to the consumers.

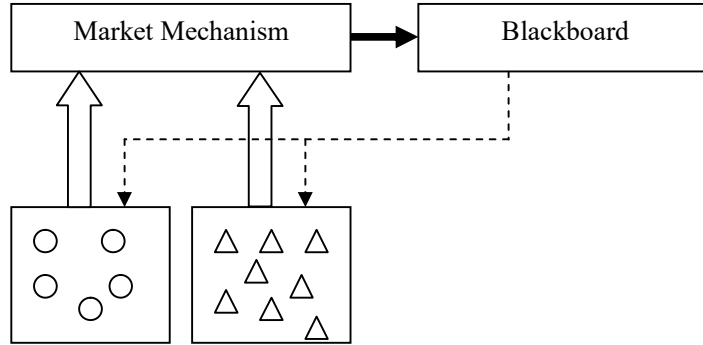


Figure 2: Blackboard Architecture for Competitive market

Current prices of goods in the market are displayed on the blackboard and agents compute their bids based on this price vector. The bids are submitted to the market mechanism which computes the total demand and supply for each good and also finds the equilibrium price for the commodity. It posts the new price vector on the blackboard and agents react to this new information by computing new bids and sending it to the market mechanism. This process continues until no agent is willing to change its bid any longer. At this point, market is said to clear and a Pareto allocation of goods is achieved. This means that the market has reached a point in which all the trading agents in the market have executed their best possible strategy in bidding for market resources based on the current market condition and it is not possible for any agent to have a better bid without another agent's bid deteriorating. Figure 3 shows a simple flowchart describing this process.

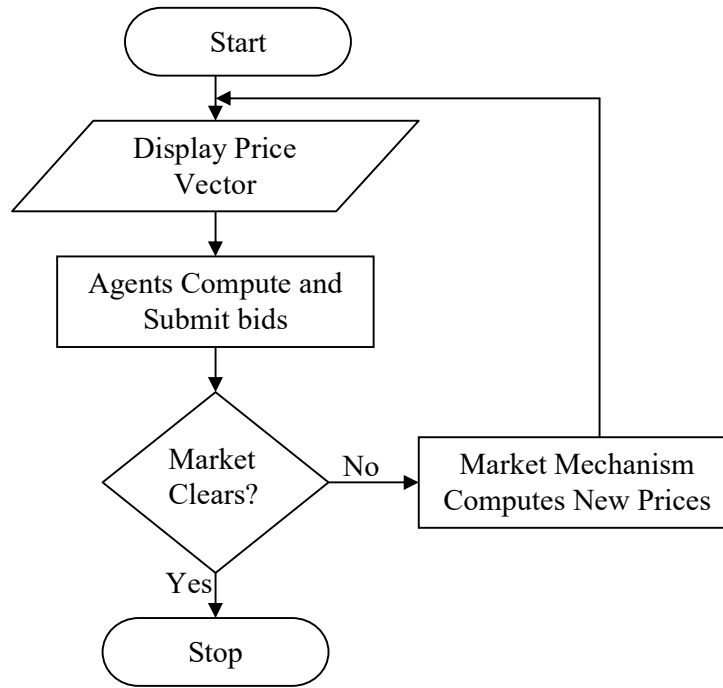


Figure 3: Competitive Market Mechanism

3.3 Supply Network Model

The supply network model we use in this research is the multilayer feed-forward model shown in figure 4. By feed-forward, we mean that the flow of value is unidirectional even though the flow of information may be bi-directional. We operate under the assumption that goods or money are not returned once committed to transaction.

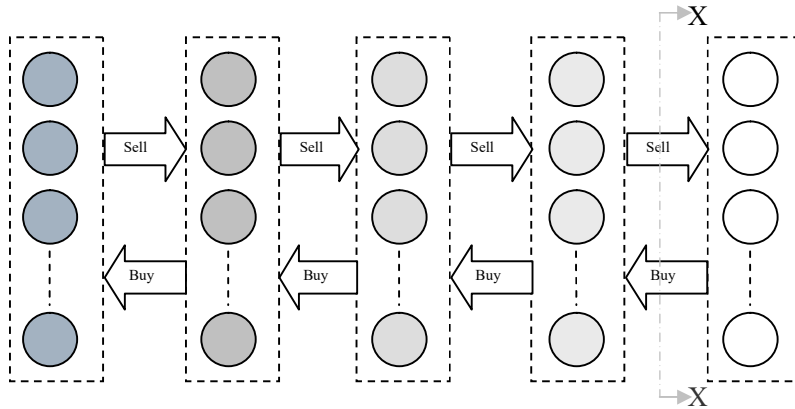


Figure 4: Multilayer Feed-forward supply network

In each of these layers, there are trading agents which have unique characteristics in terms of their technology and utility functions. In order to adapt this model to the competitive market structure, we need to remold the agents in each of the layers so as to have two distinct groups of agents in the market. We discuss this in the next section.

3.3.1 Remolding the Supply Network

In remolding the supply network, we first group the agents in the network into two by creating a split in the network along section X – X. This split categorizes the agents into pure consumers and suppliers. The next

step is to split each supplier agent into two comprising of a pure supplier and a neo-consumer. The new consumer section of each agent is responsible for holding the stock of resources of the agent while the pure supplier section is responsible for the resource flow (transformation) within the agent. Figures 5(a-b) illustrate this process.

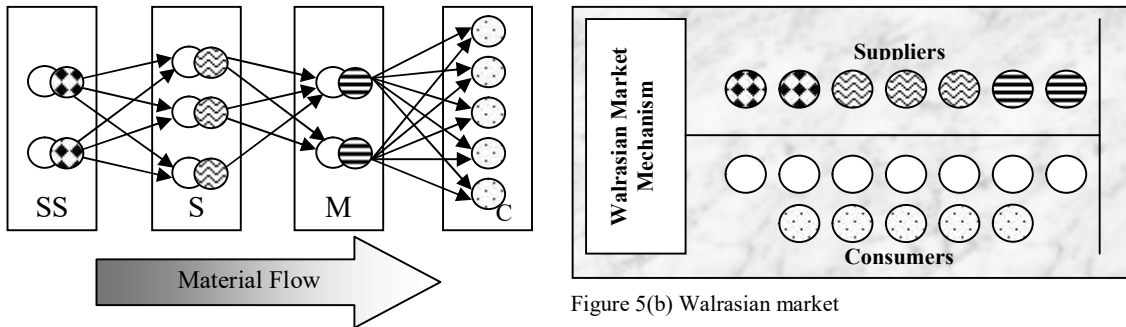


Figure 5(b) Walrasian market

Figure 5(a) Supply network (SS-Supplier's supplier, S-Supplier, M-Producer, D-Distributor, C-Consumer)

After the remolding, trading can begin in the walrasian market. In our model, the profit of each consumer is determined by their contributions to the market. Therefore, we can determine the profitability of a supplier agent by checking the profit that is accrued to its neo-consumer section.

4 Experimental Procedure

In this section, we describe how data for the agents are represented and how the simulation is carried out. We make use of a four-layered supply network like that in figure 5(a). The network consists of two supplier's supplier agent, three supplier agents, two producer agents and five pure consumer agents. The outputs of the producer layer are the final products of the supply network. The outputs of the other layers are inputs for the layers following them and they are referred to as intermediate factors of production in our model. Also, there are two primary factors in the market – capital and labour. These two primary factors are in the economy in the form of endowments. Tables 1 – 2 show in the input parameters of the agents.

4.1 Agent Properties

	Supplier's Supplier(ss1)				
	Investment		Technology		
	Preference	Endowment	Scale	Index	
Goods1(ss1)	0	0	0	0	
Goods2(ss2)	0	0	0	0	
Goods3(s1)	0	0	0	0	
Goods4(s2)	0	0	0	0	
Goods5(s3)	0	0	0	0	
Goods6(m1)	0	0	0	0	
Goods7(m2)	0	0	0	0	
Capital	0	0	25	0.2	
Labour	0	0	47	0.8	

	Supplier(s1)				
	Investment		Technology		
	Preference	Endowment	Scale	Index	
Goods1(ss1)	0	0	25	0.3	
Goods2(ss2)	0	0	23	0.3	
Goods3(s1)	0	0	0	0	
Goods4(s2)	0	0	0	0	
Goods5(s3)	0	0	0	0	
Goods6(m1)	0	0	0	0	
Goods7(m2)	0	0	0	0	
Capital	0.65	50	70	0.25	
Labour	0.35	30	40	0.15	

	Supplier(s3)				
	Investment		Technology		
	Preference	Endowment	Scale	Index	
Goods1(ss1)	0	0	60	0.2	
Goods2(ss2)	0	0	10	0.4	
Goods3(s1)	0	0	0	0	
Goods4(s2)	0	0	0	0	
Goods5(s3)	0	0	0	0	
Goods6(m1)	0	0	0	0	
Goods7(m2)	0	0	0	0	
Capital	0.8	60	37	0.3	
Labour	0.2	40	49	0.1	

	Producer(m2)				
	Investment		Technology		
	Preference	Endowment	Scale	Index	
Goods1(ss1)	0	0	0	0.2	
Goods2(ss2)	0	0	0	0.4	
Goods3(s1)	0	0	43	0	
Goods4(s2)	0	0	57	0	
Goods5(s3)	0	0	0	0	
Goods6(m1)	0	0	0	0	
Goods7(m2)	0	0	0	0	
Capital	0.8	60	49	0.3	
Labour	0.2	40	47	0.1	

	Supplier's Supplier(ss2)			
	Investment		Technology	
	Preference	Endowment	Scale	Index
Goods1(ss1)	0	0	0	0
Goods2(ss2)	0	0	0	0
Goods3(s1)	0	0	0	0
Goods4(s2)	0	0	0	0
Goods5(s3)	0	0	0	0
Goods6(m1)	0	0	0	0
Goods7(m2)	0	0	0	0
Capital	0	0	74	0.7
Labour	0	0	23	0.3

	Supplier(s2)			
	Investment		Technology	
	Preference	Endowment	Scale	Index
Goods1(ss1)	0	0	40	0.4
Goods2(ss2)	0	0	30	0.1
Goods3(s1)	0	0	0	0
Goods4(s2)	0	0	0	0
Goods5(s3)	0	0	0	0
Goods6(m1)	0	0	0	0
Goods7(m2)	0	0	0	0
Capital	0.7	40	60	0.2
Labour	0.3	60	34	0.3

Table1 (a-g): Properties of Supplier Agents in the Network

	Producer(m1)			
	Investment		Technology	
	Preference	Endowment	Scale	Index
Goods1(ss1)	0	0	0	0.2
Goods2(ss2)	0	0	0	0.4
Goods3(s1)	0	0	35	0
Goods4(s2)	0	0	65	0
Goods5(s3)	0	0	40	0
Goods6(m1)	0	0	0	0
Goods7(m2)	0	0	0	0
Capital	0.8	60	63	0.3
Labour	0.2	40	51	0.1

Preference Indices of Pure Consumers					
	Consumer8	Consumer9	Consumer10	Consumer11	Consumer12
Goods1(ss1)	0	0	0	0	0
Goods2(ss2)	0	0	0	0	0
Goods3(s1)	0	0	0	0	0
Goods4(s2)	0	0	0	0	0
Goods5(s3)	0	0	0	0	0
Goods6(m1)	0.27	0.18	0.36	0.29	0.44
Goods7(m2)	0.73	0.82	0.64	0.71	0.56
Capital	0	0	0	0	0
Labour	0	0	0	0	0

Table 2(a): Preference Indices of Pure Consumers

Endowment of Pure Consumers					
	Consumer8	Consumer9	Consumer10	Consumer11	Consumer12
Goods1(ss1)	0	0	0	0	0
Goods2(ss2)	0	0	0	0	0
Goods3(s1)	0	0	0	0	0
Goods4(s2)	0	0	0	0	0
Goods5(s3)	0	0	0	0	0
Goods6(m1)	0	0	0	0	0
Goods7(m2)	0	0	0	0	0
Capital	110	270	180	200	100
Labour	300	150	160	70	120

Table 2(b): Endowments of Pure Consumers

Table2 (a-b): Properties of Pure Consumer Agents in the Network

Note that we denote the pure consumers by suffices 8 – 12. Suffices 1 – 7 are reserved for the neo-consumers to be introduced to the walrasian market. The nomenclature has no effect on the trading process. The total number of commodities in the market is nine – two primary factors and seven intermediate goods.

4.2 Simulation

Using the parameters in section 4.1, simulations were run using our program developed in JAVA programming language. We ran simulations based on our model and also based on the tatonnement algorithm proposed by Leon Walras. We went a step further in creating a dynamic scenario for our model by gradually increasing the number of trading agents and inspecting the performance of the algorithm in terms of computational speed. The results of these simulations are presented in the following section.

5 Results and Discussion

5.1 Market Price-vector Variation

The results presented here are normalized prices of resources, i.e., the relative prices of commodities in the market; the summation of which is equal to unity. Figures 6 – 7 show how prices of goods fluctuate to arrive at market clearing. We give initial price vector as:

$P^T = \{0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.15, 0.15\}$. Iteration represents a bidding process by the trading agents. We run the simulation for thirty iterations. The market clearing price for our model is shown in Table 3 below:

Goods1	Goods2	Goods3	Goods4	Goods5	Goods6	Goods7	Capital	Labour
8.89E-03	8.71E-03	4.53E-02	6.93E-02	2.41E-02	0.149774	0.304991	0.272753	0.114914

Table 3: Market Clearing Price vector

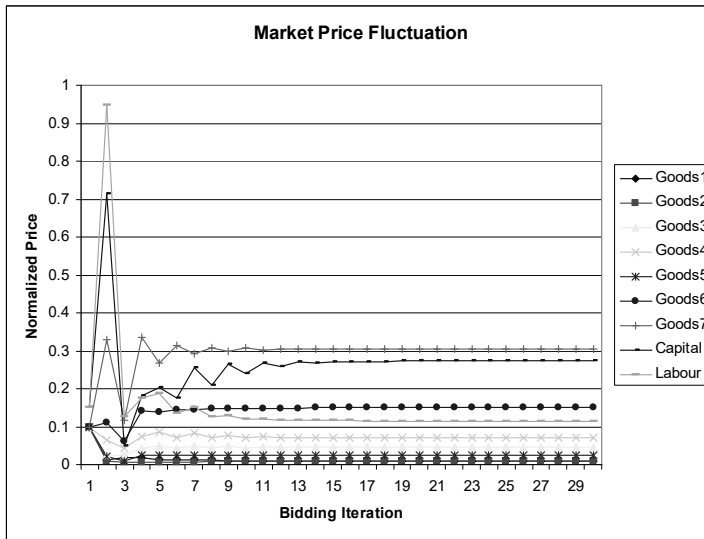


Figure 6: Price Fluctuation for primary factors and intermediate goods

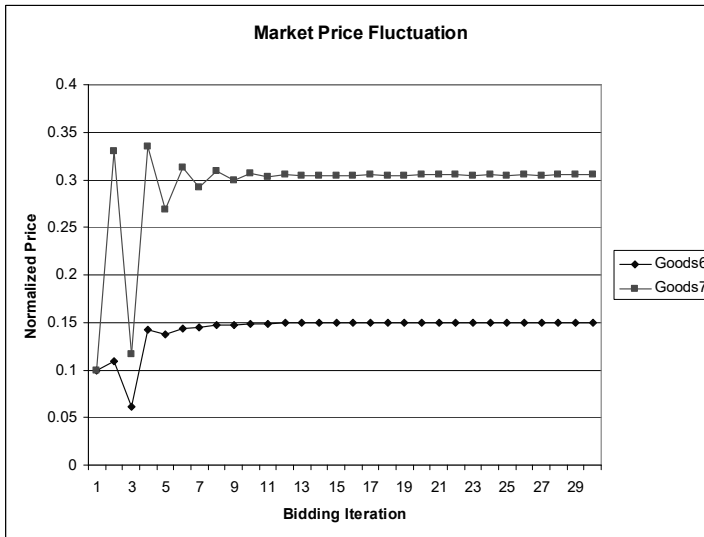


Figure 7: Price Fluctuation for final products

Figure 7 is extracted from figure 6 and shown for the purpose of clarity and since delivery of the final products of the supply network is the main objective of planning.

Simulation was run for the same market using the price tatonnement algorithm. We show the result for fifty thousand iterations which resulted in a market clearing price vector given in Table 4 below:

Goods1	Goods2	Goods3	Goods4	Goods5	Goods6	Goods7	Capital	Labour
8.90E-03	8.81E-03	4.56E-02	6.93E-02	2.42E-02	0.149791	0.305216	0.273307	0.114921

Table 3: Market Clearing Price vector for Tatonnement algorithm

The value of the price adjustment constant used is $\alpha = 0.0005$

5.2 Market Demand and Supply Variation

Next, we show demand and supply patterns during the bidding process. We only present the graphs for the two final products in figures 8 and 9.

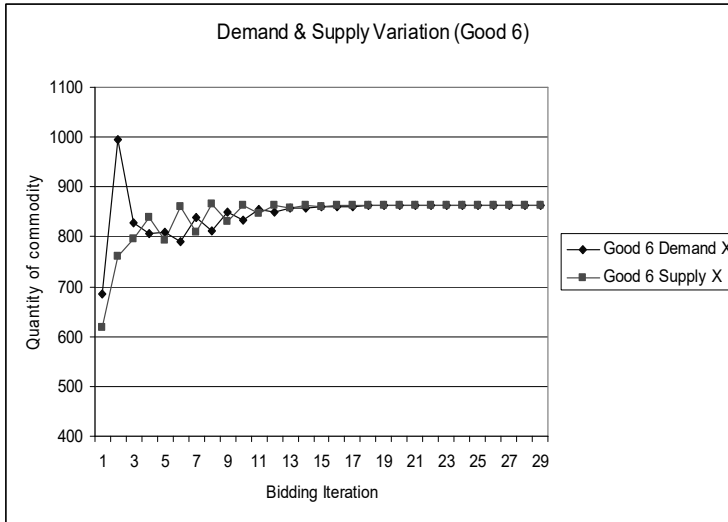


Figure 8: Demand and Supply Fluctuation for Goods6

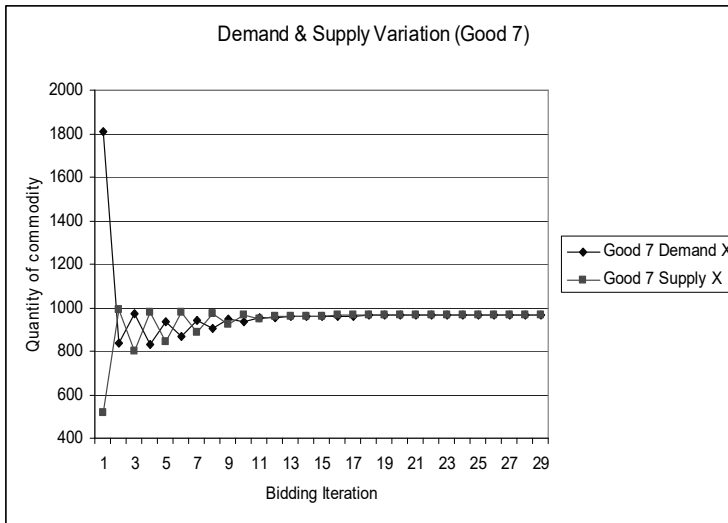


Figure 9: Demand and Supply Fluctuation for Goods7

5.3 Resource Allocation

The Pareto allocation of resources is shown in Tables 4 – 5.

	Supplier's Supplier(ss1)		Supplier's Supplier(ss2)		Supplier(s1)		Supplier(s2)	
	Demand	Supply	Demand	Supply	Demand	Supply	Demand	Supply
Goods1(ss1)	0	3378.29242	0	0	182.4757	0	3118.091	0
Goods2(ss2)	0	0	0	255.734237	166.9868	0	34.02171	0
Goods3(s1)	0	0	0	0	0	372.1739	0	0
Goods4(s2)	0	0	0	0	0	0	0	1194.041
Goods5(s3)	0	0	0	0	0	0	0	0
Goods6(m1)	0	0	0	0	0	0	0	0
Goods7(m2)	0	0	0	0	0	0	0	0
Capital	0.10365	0	5.347434	0	4.15826	0	4.035965	0
Labour	209.1001	0	0.395965	0	2.759407	0	13.42647	0

Table 4(a): Resource Allocation Table for SS1 – S2

	Supplier(s3)		Producer(m1)		Producer (m2)	
	Demand	Supply	Demand	Supply	Demand	Supply
Goods1(ss1)	77.92757	0	0	0	0	0
Goods2(ss2)	55.2666	0	0	0	0	0
Goods3(s1)	0	0	28.75714	0	344.1156	0
Goods4(s2)	0	0	350.3974	0	843.7952	0
Goods5(s3)	0	278.8105	279.6705	0	0	0
Goods6(m1)	0	0	0	863.1987	0	0
Goods7(m2)	0	0	0	0	0	964.2484
Capital	0.976781	0	11.23317	0	14.73053	0
Labour	1.034156	0	3.543563	0	78.30045	0

Table 4(b): Resource Allocation Table for S3 – M2

	Consumer 1		Consumer 2		Consumer 3		Consumer 4	
	Demand	Supply	Demand	Supply	Demand	Supply	Demand	Supply
Goods1(ss1)	0	0	0	0	0	0	0	0
Goods2(ss2)	0	0	0	0	0	0	0	0
Goods3(s1)	0	0	0	0	0	0	0	0
Goods4(s2)	0	0	0	0	0	0	0	0
Goods5(s3)	0	0	0	0	0	0	0	0
Goods6(m1)	0	0	0	0	0	0	0	0
Goods7(m2)	0	0	0	0	0	0	0	0
Capital	0	0	0	0	80.26309	50	90.04026	40
Labour	0	0	0	0	102.9297	30	91.87494	60

Table 5(a): Resource Allocation Table for Consumer 1 – Consumer 4

	Consumer 5		Consumer 6		Consumer 7		Consumer 8	
	Demand	Supply	Demand	Supply	Demand	Supply	Demand	Supply
Goods1(ss1)	0	0	0	0	0	0	0	0
Goods2(ss2)	0	0	0	0	0	0	0	0
Goods3(s1)	0	0	0	0	0	0	0	0
Goods4(s2)	0	0	0	0	0	0	0	0
Goods5(s3)	0	0	0	0	0	0	0	0
Goods6(m1)	0	0	0	0	0	0	229.3702	0
Goods7(m2)	0	0	0	0	0	0	304.5047	0
Capital	121.1952	60	388.4472	250	279.6462	150	0	110
Labour	72.15774	40	308.4545	30	117.5503	40	0	300

Table 5(b): Resource Allocation Table for Consumer 5 – Consumer 8

	Consumer 9		Consumer 10		Consumer 11		Consumer 12	
	Demand	Supply	Demand	Supply	Demand	Supply	Demand	Supply
Goods1(ss1)	0	0	0	0	0	0	0	0
Goods2(ss2)	0	0	0	0	0	0	0	0
Goods3(s1)	0	0	0	0	0	0	0	0
Goods4(s2)	0	0	0	0	0	0	0	0
Goods5(s3)	0	0	0	0	0	0	0	0
Goods6(m1)	86.1772	0	190.757	0	135.0919	0	222.3829	0
Goods7(m2)	192.7665	0	166.516	0	162.4005	0	138.9743	0
Capital	0	70	0	80	0	100	0	90
Labour	0	150	0	160	0	70	0	120

Table 5(c): Resource Allocation Table for Consumer 9 – Consumer 12

It can be seen from the results that there is a conservation of product as required by the walrasian market structure.

5.4 Computational Effort

We ran a simulation on an Intel P4 (3.2GHz) machine, varying the number of traders in the market from the original number of 19 traders. Table 6 and Figure 10 show this variation.

Traders	19	24	29	34	39	44	49	54	59	64	69
Time (sec)	21.359	36.734	70.36	86.078	104.219	125.39	142.25	152.828	169.891	195.172	204.329

Table 6: Variation of Population of Traders with Computational Time

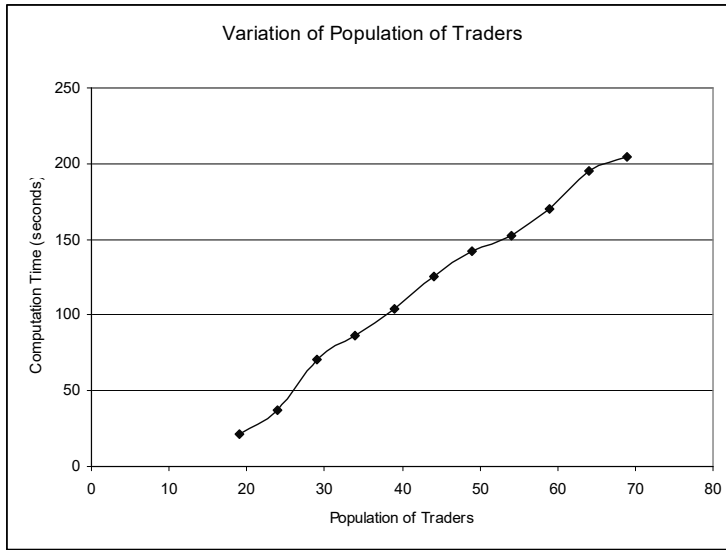


Figure 10: Variation of Population of Traders with Computational Time

5.5 Discussion

5.5.1 Correctness of Solution

First we confirm the correctness of the results we have obtained from our market model by comparing with analytical conditions for market clearing. For market to clear, it has been proven that the Marginal Rate of Substitution (MRS) between two goods for all the market participants must be the same. We confirmed this by finding the price ratio for the two final goods (goods 6 & 7) demanded only by the pure consumers and the results showed that:

$$\frac{p_6}{p_7} = \frac{\alpha_6^c}{\alpha_7^c} * \frac{g_7^c}{g_6^c} \quad (20)$$

From the price vector in table 3 and resource allocation in table 5, we find the price ratio to be approximately equal to [0.491076] which is the same as the right hand side of equation 19 for consumers 8 – 12.

	Consumer 8	Consumer 9	Consumer 10	Consumer 11	Consumer 12
Index (α_6)	0.27	0.18	0.36	0.29	0.44
Index (α_7)	0.73	0.82	0.64	0.71	0.56
Demand (g_6)	229.3702	86.1772	190.757	135.0919	222.3829
Demand (g_7)	304.5047	192.7665	166.516	162.4005	138.9743
M.R.S	0.491019	0.491019	0.491019	0.491019	0.491019

Table 7: Marginal Rate of Substitution for Goods6 & 7

We also confirmed the axiom of conservation of products in the walrasian market to confirm our result. The total number of the final goods demanded by the pure consumers is approximately equal to that which is supplied by producer agents m1 and m2.

	Demand	Supply
Goods6	863.7792	863.1987
Goods7	965.1621	964.2484

Table 8: Total Demand & Supply of Goods 6 & 7

Furthermore, comparing our results with the price tatonnement algorithm, we were able to confirm the correctness of the resource allocation matrix.

5.5.2 Efficiency of Algorithm

Comparing the speed of execution of our algorithm with the price tatonnement algorithm, we found our algorithm to be at least three times faster than the price tatonnement algorithm running under the same condition. When we compared the resource allocation result with that of the price tatonnement algorithm, we got an accuracy of 1 good while that of the price tatonnement is about 2 goods. As shown in Figure 10, we have an almost linear time variation with number of traders in the computational time measure of our algorithm.

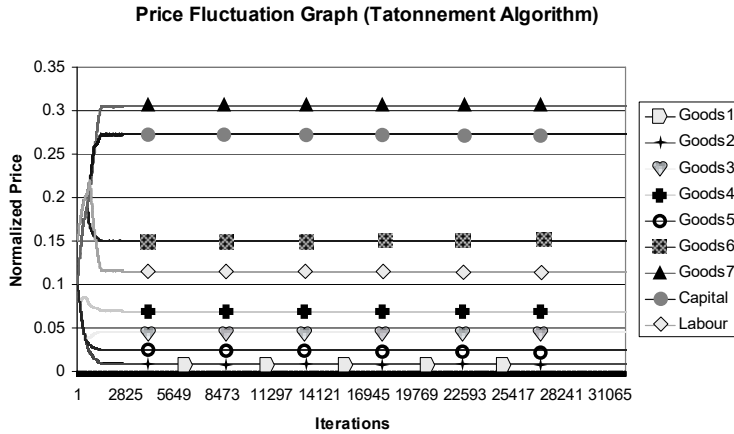


Figure 11: Price Fluctuation Graph for Tatonnement Algorithm

In figure 11, we represent the nature of the price fluctuation that takes place when we use the tatonnement algorithm in order to compare the results with what is shown in figure 6. From the results shown on the graph, it can be seen that the two algorithms arrive at the same solution but the major difference lies in the number of iterations needed to accomplish this.

In figures 12 and 13 we show how equilibrium price vector is reached in a three-dimensional plot for simulation runs using the MOP algorithm and the tatonnement algorithm respectively. In these simulations, we make use of three goods in the market just for validation purposes. While the MOP algorithm shows a situation in which equilibrium point is approached in big steps (less than 30 iterations), the tatonnement shows a movement towards equilibrium in relatively smaller steps (requiring about 50,000 iterations) even though, the movement appears to be more rapid. The big jumps in the MOP algorithms accounts for its quicker arrival at the market clearing price vector.

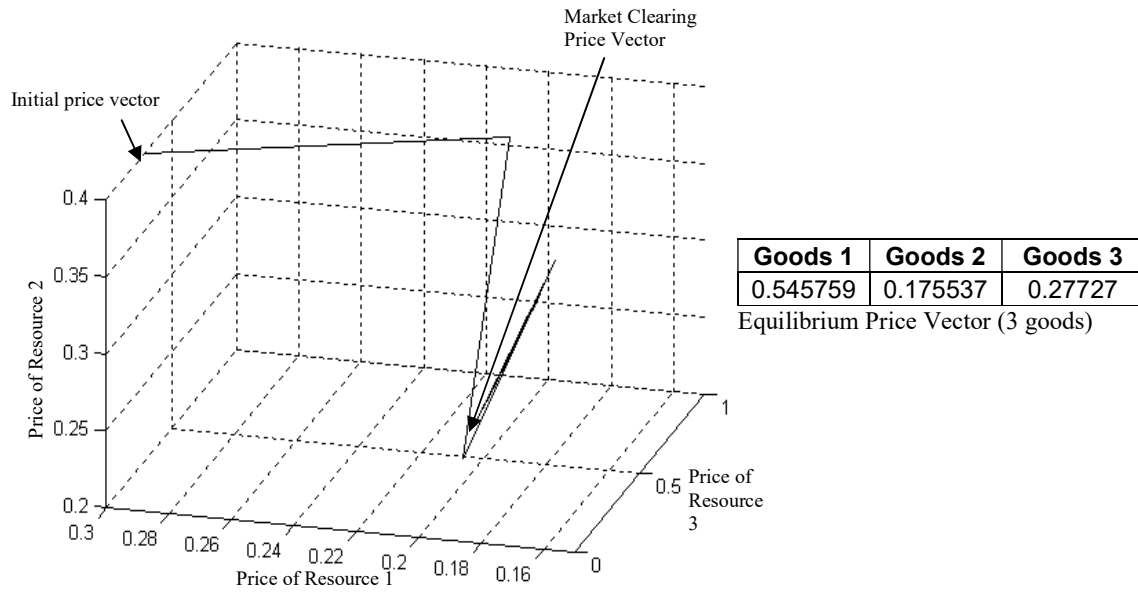


Figure 12: MOP Simulation with 3 goods

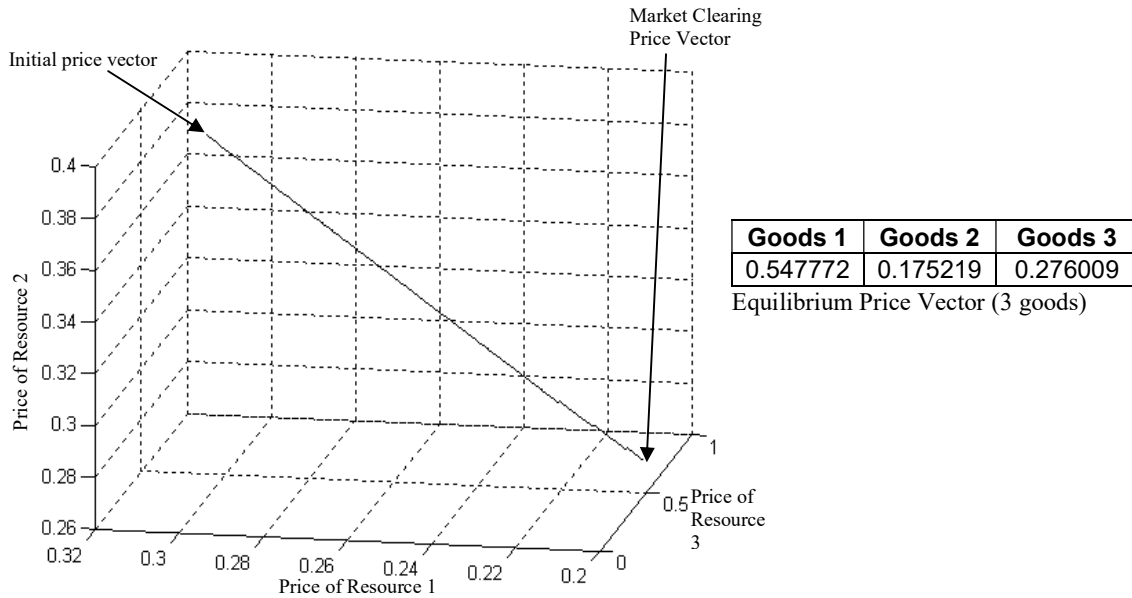


Figure 13: Tatonnement Algorithm Simulation with 3 goods

6 Conclusion

In this paper, we have introduced a model which makes use of the Multi-Agent System (MAS) paradigm to obtain an optimal resource allocation among a group of trading agents in a virtual enterprise network. We implemented simulations using the Market-Oriented Programming (MOP) technique based on the walrasian market model.

From the results we have obtained, it is clear that the market oriented model that we propose is more efficient than the existing methodology which we compared it with. It has already been confirmed from previous research that our algorithm is also faster than the scarf algorithm mentioned earlier in this paper [9]. The computational time expended also varies linearly with the number of traders in the market, which makes this model a first-order polynomial time algorithm.

The proposed model may be used as a strategic decision support model. The only major challenge is selecting the appropriate parameters for the agents in the network which will be a true representation of the actual market situation. This problem can be solved by collecting data on consumer behaviour and upstream customer activities over a long period of time and using this data, one can characterize every participant of the supply network. Again, due to the weak axioms of profit maximization, revealed preference and gross substitutability of products, this model might only be applicable for estimation and strategic planning only to give a holistic view of the supply network. The axioms may make problems in the operational planning domain somewhat oversimplified.

In subsequent research work, we intend to investigate how this model can be applied for monitoring the effects of some macroeconomic indicators on a virtual enterprise network. Another area of application is on how profit distribution in a supply network is affected by endowment distribution and consumer preference distribution in the network.

References

- [1] P. Fingar, J. Belini: *The Real Time Enterprise*, Meghan-Kiffer Press, Tampa Florida. (2004)
- [2] H. Stadtler, C. Kilger (Eds.): *Supply Chain Management and Advanced Planning-Concepts, Models, Software and Case Studies* (3rd Edition), Springer Press, Berlin. (2005)
- [3] M. Babaioff, N. Nisan: *Concurrent Auctions Across the Supply Chain*. *Journal of Artificial Intelligence research* 21, pp. 596 - 629. (2004)
- [4] M. Fan, J. Stallert, A.B. Whinston: *Decentralized Mechanism for Supply Chain Organization Using an Auction Market*. *Information Systems Research [INFORMS]*, Vol. 14, No. 1, pp. 1-22. (2003)
- [5] F.S. Hillier, G.J. Lieberman: *Introduction to Operations Research* (Fifth Edition) McGraw-Hill Publishing Company, USA. (1990)
- [6] G. Weiss (Ed.): *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press. (1999)
- [7] W. Shen, D.H. Norrie, J.A. Barthes: *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*. Taylor & Francis Inc. USA. (2001)
- [8] H.R. Varian: *Microeconomic Analysis* (Third Edition). W.W. Norton & Company, Inc. USA. (1992)
- [9] T. Kaihara, S. Fujii, *A Study on Virtual Market for Optimal Mediation in Economic Society, EES: Experiments in Economic Sciences-New Approaches to Solving Real-world Problems*, pp.494-499. (2004)
- [10] M. Anjan: *An Introduction to General Equilibrium Analysis: Walrasian and Non-Walrasian Equilibria*. Oxford Press, India. (2002)

- [11] M.P. Wellman: A Market-Oriented Programming Environment and its Application to Distributed Multi-commodity Flow Problems, *Journal of Artificial Intelligence Research* 1, pp.1-23. (1993)
- [12] K.J. Arrow, T.J. Kehoe: Distinguished Fellow Herbert Scarf's Contribution to Economics. *Journal of Economic Perspective*, Vol. 8, No. 4, pp 161 - 181. (1994)
- [13] H. Scarf: Approximation of Fixed Points. *SIAM Journal on Applied Mathematics*, Vol. 15, No. 5. pp 1328 - 1343. (1967)
- [14] H. Scarf: An Example of An Algorithm for Calculating General Equilibrium Prices. Cowles Foundation Discussion Paper No. 276. Cowles Foundation for Research in Economics at Yale University, USA. (1969).