A Supply Chain as a Network of Auctions

Thierry Moyaux, Peter McBurney and Michael Wooldridge

Department of Computer Science
University of Liverpool
Liverpool L69 7ZF, UK

{moyaux, p.j.mcburney, mjw}@csc.liv.ac.uk

July 17, 2007

Abstract

This paper proposes and evaluates a model of supply chains as networks of auctions. In this model, companies are represented according to the first level of the Supply Chain Council's SCOR model and the trading strategy of the agents is adapted from a model proposed by Steiglitz and colleagues. Specifically, the highest level of SCOR treats a company as comprising three functions, namely *Source*, *Make*, and *Deliver*. Our companies may also have these three functions, where *Source* and *Deliver* are modelled in a way similar to the agents proposed by Steiglitz and his colleagues, that is, they hold products in their inventory and place shouts to buy or sell products so that their inventory remains at its target level. After presenting our model, we study its behaviour with a simulation based on the JASA auction simulator. This study provides insights into supply chain dynamics and associated trading behaviours. In particular, the simulation results show that price dynamics are more complicated than simply balancing consumption, transformation capacities, and the supply of raw materials. In addition, we identify three patterns of price dynamics in our auctions, explain their cause, and propose rules linking initial conditions and the occurrence of such price patterns.

Keywords: Supply chain modeling, SCOR Model, Auctions, System dynamics, Agent-based Simulation.

1 Introduction

Supply chain management is one of the most widely studied problems in contemporary manufacturing and industrial management (Simchi-Levi *et al.*, 2000). Supply chain management involves the design, modeling, implementation, and coordinated control of networks of resources in order to supply goods and services to consumers. Typical goals are to build supply chains that are, for example, agile (able to respond rapidly to changing market circumstances), lean (with the smallest possible commitment to items in stock), and robust (resilient against unforeseen logistical problems). Improvements in supply chain management can yield significant competitive advantage for producers, hence the considerable interest this subject has aroused.

One increasingly popular approach to the design and management of complex systems is the use of *market mechanisms* — see e.g., Clearwater (1996). Markets are widely recognised as providing efficient mechanisms for resource management and allocation. Historically, the inevitable coordination and management overheads associated with implementing market-based systems have meant that their use has been reserved for large applications. However, the widespread availability of cheap networked computer systems has meant that the overheads associated with operating market systems are now sufficiently low that they can be much more widely used (witness, for example, the growth of online auction-houses such as eBay).

It is not surprising, therefore, that researchers would investigate the use of market mechanisms for supply chain management. An interesting market-based system in which traders produce and consume goods was Steiglitz *et al.* (1996). In this paper, we build on their work. As part of a larger project to apply concepts from economics to the design and management of distributed computational systems,[1] we have studied supply chains as *sequences of linked marketplaces*. In this model, entities in the chain exhibit buyer/seller behaviours, rather than, for example, order/deliver behaviours (as in the Beer Game (Sterman, 1989)). In this model, a supply chain then consists of sets of market interactions involving three connected flows up and down the chain: needs, goods, and money.

Our work builds on the prior work of Steiglitz *et al.*, as follows. Essentially, we have adapted their model to *networks* of auctions in order to utilize their tools (speculation, and the three price signals) in the management of supply chains. For that purpose, we have replicated the experiments in these three papers[2] using the JAVA Auction Simulator API (JASA)[3], and study how these results scale to networks of auctions. Our aim, therefore, is to understand if tools effective for management of the dynamics of a single auction remain effective in the presence of supply chain dynamics; that is, to understand if these tools can also handle the different streams (needs, products, and money) in supply chains, as well as the interactions among these streams. With regard to such adaptation of tools, we have already stabilised the prices in a supply chain modeled as in this paper by means of speculation (Moyaux and McBurney, 2006b). We also plan in (Moyaux and McBurney, 2006a) to adapt the methodology proposed by Mizuta *et al.* (2003) to broadcast different price signals in order to stabilise supply chains. However, adapting such tools to supply chains is not the topic of this paper, which focuses on the model and its dynamics.

The remainder of this paper is structured as follows. Following a survey of related work, Section 3 introduces our model. Section 4 presents the dynamics of the price when a single market is considered. In particular, we identify three patterns of price dynamics, explain their cause, and propose two rules linking such patterns with some initial conditions of the simulation. Section 5 extends these observations and explanations, and adds a rule in a scenario with two sequential markets. Finally, Section 6 concludes the paper.

2 Background and Related Work

According to Dodd and Kumara (2001), Mark S. Fox was probably the first to model a supply chain as a multiagent system (Fox *et al.*, 1993). Besides the construction of an agent-oriented software architecture (Fox *et al.*, 2000), Fox and a colleague of his have proposed COOL (COOrdination Language), a language based on KQML for coordinating industrial distributed applications (Barbuceanu and Fox, 1996). Many other applications of multiagent systems to supply chain management have followed. For instance, Cloutier *et al.* (2001) also worked on coordination by proposing CAT (Convention, Agreement and Transaction) to the bus manufacturer Prévost Car.[4] CAT allows business partners to share high level information, next interact (e.g. negotiate) with each other, in order to commit on punctual manufacturing or informational actions to perform in the future. Another example is provided by Anthes (2003) who reports that Procter & Gamble[5] "*saves USD300 million annually on an investment of less than 1% of that amount*" thank to agent-based simulations. In fact, Nutech Solutions (former Biosgroup[6]) provided Procter & Gamble with an agent-based simulation of a portion of its retail supply network in order to study the impact of certain policies applied by the companies in this network. Next,

[1] See `http://www.marketbasedcontrol.com/`

[2] We not present our replication here of the results of the three papers by Steiglitz. Note that when we refer to these papers, we refer in fact to our replication of their models.

[3] See `http://www.csc.liv.ac.uk/~sphelps/jasa/` and `http://jasa.sourceforge.net/`.

[4] See `http://www.prevostcar.com/`.

[5] See `http://www.pg.com/`.

[6] See `http://www.biosgroup.com/lit_featured.asp` and `http://www.nutechsolutions.com/lit_featured.asp`.

additional studies are more focussed on the use of markets in supply chains. The Trading Agent Competition - Supply Chain Management (TAC-SCM)[7] may be the most famous model in this category. As indicated by its name, this is a competition in which entrants propose software trading agents in order to buy components from several suppliers, assemble these components, and sell the finished products to end customers, all automatically.

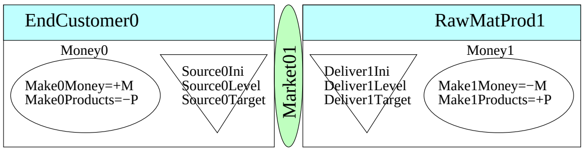
Such a model of a market-mediated supply chain allows studying questions as the long-term costs and benefits in a business-to-business (B2B) context of online auctions in comparison with a long-term relationship. In other words, is it better to have a market-mediated supply chain in which the competition among traders leads to the short-term optimal, or a traditional RFQ/RFP (Request for Quote/Proposal) process in which learning may end with a better long-term result? (Geoffrion and Krishnan, 2003, p. 1147) In addition, markets are often thought to be efficient when they are perfect, but are they able to handle the complexity of supply chain dynamics due to their interconnected flows? Markets may therefore be an efficient way to manage supply chains.

In the model of Steiglitz *et al.* (1996), a single type of agent produces food and gold, then trades food for gold via a market modelled as an auctioneer. Two kinds of speculators are also introduced, which both stabilise the clearing price when no price bubbles are created. Subsequently, Steiglitz and Shapiro (1998) extended this initial study of the model by analysing the occurence of these price bubbles and interrupting them during their formation. In both papers, trading agents bid a price calculated as $P(t-1) * B(\bar{f}, \bar{g})$, where $P(t-1)$ is the previous price in the auction, and where $B(\bar{f}, \bar{g})$ is a function of the internal state of the agent (this strategy will be detailed in this paper). Later, this model was extended to understand how an auctioneer can stabilise the price in a single auction by broadcasting more information about the state of the auction than simply the actual clearing price (Mizuta *et al.*, 2003). Specifically, the auctioneer broadcasts one of the following price signals: (i) $P0$ is the non-weighted average of the prices in all (bid and ask) shouts, (ii) $P1$ is the average of the prices in all shouts weighted with the quantity of these shouts, and (iii) $P2$ is another weighted average of the prices proposed by the traders. Next, the traders then bid the price $P0(t-1) * B(\bar{f}, \bar{g})$, $P1(t-1) * B(\bar{f}, \bar{g})$ or $P2(t-1) * B(\bar{f}, \bar{g})$. When the auctioneer broadcasts $P0$, then P slowly reaches its equilibrium; when $P1$ is broadcast, then P fluctuates forever; finally, using $P2$ causes rapid convergence to the equilibrium.

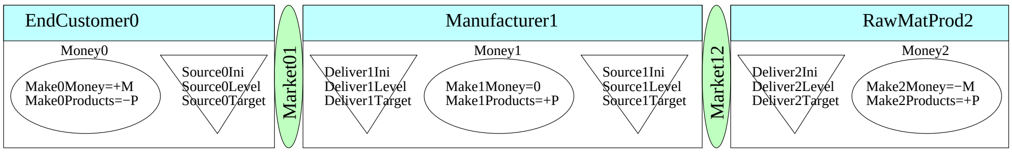
The problem with Steiglitz *et al.*'s model is the assumption that, in every round, traders decide to produce either food or gold. Such an assumption replicates an economy of ancient time in which the agents are not specialised on a specific production activity. In stark contrast, the agents in modern economies are very specialised: either they are farmers who grow food, or they are miners who dig for gold. Such a specialisation generates an interdepency among the two types of agents: farmers rely on miners to fulfil their needs in gold, while miners have to trade with farmers in order to obtain the food they consume. Since miners always sell gold and buy food, two streams flowing in opposite directions appear, namely a stream of food linked to a stream of gold. This way, miners and farmers form the most simple supply chain.

In this paper, we first study such a simple supply chain with only two types of agents, in which the miners are called end customers because they are the source of money, and the farmers are seen as raw material producers because they provide end customers with products. Then, we extend this model with a third kind of agents, called manufacturers, that transform the products bought from the raw material producers in order to sell the transformed items to the end customers. This model is thus quite similar to the TAC-SCM. Since two kinds of products are exchanged among these three types of agents, two markets are required, i.e. one market per product type. We think that such an improvement over Steiglitz *et al.*'s model does not only make it more realistic with regard to modern economies, but also sheds light on the different interdependencies in supply chains. In fact, the different streams traveling accross supply chains cause both the markets and the different types of traders to be interdependent. For instance, the fluctuations of the price in the first market may impact on the fluctuations of the price

[7] See `http://www.sics.se/tac/`.



(a) The two types of agents trading in one *Market*01.



(b) The three types of agents trading in *Market*01 and *Market*12.

Figure 1: The two structures of supply chain considered in this paper.

in the second market.

The model with two and three types of company-agents is now introduced.

3 The Supply Chain Model

Our aim in this section is to give a detailed description of the supply chain model whose properties we subsequently investigate. The basic idea of the model is simply that the supply chain itself is modelled as a chain of interconnected markets. Thus, for example, a market connects raw material producers to manufacturers, and another connects end customers to manufacturers. Our belief is that by building supply chains in this way, we can in particular make them more efficient and more responsive to prevailing market circumstances.

Our model makes use of the first level of the Supply-Chain Operations Reference-model (SCOR), and we describe how SCOR is used in our model in Subsection 3.1. The ordering strategy used by companies in our model is described in Subsection 3.2. The use of the JASA auctioneer in the model is described in Subsection 3.3. Finally, some definitions and the settings of parameters conclude this section.

3.1 The Companies Modelled with Supply-Chain Operations Reference-model (SCOR)

There are three types of entity in our model of a supply chain:

- End-users (denoted *EndCustomer*0),
- Manufacturers (*Manufacturer*1), and
- Producers of Raw Materials (*RawMatProd*1).

These different entities are illustrated in Figure 1(b). *Manufacturer*1 in Figure 1(b) is modelled directly according to the first level of SCOR, while the other four companies in Figures 1(a) and 1(b) are simplifications of this model. The three functions *Deliver*, *Make* and *Source* directly correspond to SCOR[8]. We will use "inventory" to refer to a *Deliver* or a *Source* as an agent holding products in inventory and bidding in an auction. Companies, such as *EndCustomer*0, are also agents (their activity is called *Make*) which encapsulate inventory-agents. We use "she" for *Source*s, "he" for *Deliver*s and "it" for their company. In more detail, the companies in Figure 1 have the following functions.

*EndCustomer*0 in Figures 1(a) and 1(b) has two functions:

[8]SCOR also considers two *Return* functions in parallel with *Deliver* and *Source*, and a *Plan* function controlling the five other functions.

- *Make*0 produces *Make*0*Money* $= +M > 0$ units of money by adding this quantity to *Money*0, and consumes *Make*0*Products* $= -P < 0$ units of food in every round. The consumption of products is achieved by removing them from the inventory *Source*0. If *EndCustomer*0 cannot consume the quantity *Make*0*Products*, then it forgets this fact in the future (i.e., it neither dies nor tries later on to consume more to compensate for a past lack of food).

- *Source*0 is an inventory-agent who bids in *Market*01 in order to buy products so that her inventory level *Source*0*Level* is kept at *Source*0*Target*. She starts the simulation at level *Source*0*Ini*. The products are paid with *Money*0. The bidding strategy is the one introduced by Steiglitz *et al.* (1996), and presented in Subsection 3.2.

*Manufacturer*1 in Figure 1(b) has three functions:

- *Deliver*1 is an inventory which uses Steiglitz *et al.* (1996)'s bidding strategy to place ask shouts in *Market*01. The goal of *Deliver*1 is to sell products so that the level of *Deliver*1 stays at *Deliver*1*Target*. *Money*1 is shared among *Deliver*1 and *Source*1.

- *Make*1 is the production function of *Manufacturer*1 which transforms a quantity of *Make*1*Products* units in every round at a production cost of *Make*1*Money* (considered null in this paper).

  Specifically, *Make*1 performs two actions in every round: (i) if the work-in-process inventory of *Make*1 is full with *Make*1*Products* items, then this content is moved into *Deliver*1 to simulate the end of the transformation of these items, and (ii) whenever *Source*1 contains more than *Make*1*Products* items, a new production batch is launched by moving a quantity of exactly *Make*1*Products* items from *Source*1 into *Make*1. When *Source*1 does not contain enough items, then nothing is moved, so that the work-in-process inventory *Make*1 is either empty or full, but never half-full.

- *Source*1 is similar to other inventories, that is, she holds products and bids in *Market*12 in order to purchase the raw materials which will next be transformed by *Make*1.

*RawMatProd*1 in Figure 1(a) and *RawMatProd*2 in Figure 1(b) have two functions, which reflect *EndCustomer*0:

- *Make*$\{1,2\}$ produces *Make*$\{1,2\}$*Products* $= +P > 0$ units of food every round by adding them into the inventory *Deliver*$\{1,2\}$, and consumes *Make*$\{1,2\}$*Money* $= -M < 0$ units of money every round. If it cannot consume this quantity of money, it forgets this fact in the future (i.e., *RawMatProd*$\{1,2\}$ neither dies nor tries to consume more money in the future).

- *Deliver*$\{1,2\}$ bids in *Market*$\{01,12\}$ in order to keep its level *Deliver*$\{1,2\}$*Level* at *Deliver*$\{1,2\}$*Target*, and starts the simulation at level *Deliver*$\{1,2\}$*Ini*.

The sequence of actions is as follows: (i) *Deliver*s and *Source*s place their shout first; next (ii) *Make*s produce, and *Market*01 is always invoked before *Market*12. In figure 1(b), this results in the sequence: (i) *Source*0 and *Deliver*1 place a shout in *Market*01 (in random order, i.e. either *Source*0 or *Deliver*1 first), (ii) *Market*01 is cleared, (iii) *Source*1 and *Deliver*2 place a shout in *Market*12 (in any order), (iv) *Market*12 is cleared, (v) *Make*0 is invoked, (vi) *Make*1 is called, (vii) *Make*2 is executed, and (i') another similar round starts by having *Source*0 and *Deliver*1 place a bid in *Market*01, etc.

3.2 The Bidding Strategy from (Steiglitz *et al.*, 1996)

As just described, companies do not bid directly in auctions; this is the role of their *Source* and/or *Deliver* inventories. We now describe the bidding strategy they use. Because we use the bidding strategies proposed by Steiglitz *et al.* (1996), we will also use their terminology, and explain how we combine these strategies with the SCOR model presented in the previous section. With regard to terminology, the model in (Steiglitz *et al.*, 1996) has the lowest possible number of goods to enable trade, that is, two goods, which are called "food" and "gold". In the remainder of the paper, we will call the first kind either "food", "good", "unit", "product" or "item", and the second type either "gold" or "money".

Next, JASA splits any bidding strategy into two parts, namely the valuation of the good and the bidding strategy itself. *The valuation of the good* is detailed in (Steiglitz *et al.*, 1996, p. 5) as

$$Valuation(t, \bar{f}, \bar{g}) = P(t-1) * B(\bar{f}, \bar{g}),$$

where:

- $P(t-1)$ is the price in the considered market in the previous round,
- $\bar{f}$ is the food inventory normalised by its target level,
- $\bar{g}$ represents the "gold inventory normalized by the current value of [the target level of the considered inventory]" (Steiglitz *et al.*, 1996, p. 5),
- $B(\bar{f}, \bar{g}) = [b_{0\infty} - (b_{0\infty} - b_{00})e^{-\gamma\bar{g}}]^{(1-\bar{f})}$ with $\gamma = ln(\frac{b_{0\infty}-b_{00}}{b_{0\infty}-b_{01}})$. B returns a value below one when the food inventory is above its target level, i.e. $B < 1$ when $\bar{f} > 1$, which makes the inventory-agent bid at a price lower than $P(t-1)$ in the hope to sell. $\bar{g}$ amplifies the value returned by B depending on the richness of the agent, e.g., the richer a buyer, the more expensive she is ready to buy her food. Finally, the scaling parameters of B are given in (Steiglitz and Shapiro, 1998, p. 43): $b_{00} = B(0,0) = 4.0$, $b_{01} = B(0,1) = 8.0$ and $b_{0\infty} = B(0,\infty) = 16.0$.

An important comment should be made about $B(\bar{f}, \bar{g})$. This function makes sellers decrease prices and buyers increase prices, which is of course not what we typically see in real life. The reason for this apparently strange design arises from the definitions of ask and bid shouts: (i) in bid shouts, buyers announce the maximum price they agree to spend on every item bought, (ii) while, in ask shouts, sellers announce the minimum price they want to be paid for every item sold. Next, any auctioneer clears the auction in more or less the same way by choosing a price higher than the price proposed in all matched ask shouts (and lower than any unmatched ask shout) and below the price proposed in all matched bid shouts. If we want matches to occur, then $B(\bar{f}, \bar{g})$ has to be defined in the counter-intuitive way it is now. If $B(\bar{f}, \bar{g})$ was designed according to intuition, then buyers would all propose a price below $P(t-1)$, sellers would propose a price above $P(t-1)$, and no shouts would ever be matched.[9] We do not aim at addressing this question but only at adapting Steiglitz *et al.* (1996)'s model and stabilisation methods to supply chains. However, we will pay attention to this limitation when interpreting simulation runs, since it makes all suppliers try to decrease P, while this should be the role of their clients.

Finally, $\bar{f}, \bar{g}$ and P need to be adapted to our model by replacing $Valuation(t, \bar{f}, \bar{g})$ by:

- $Valuation(t, \frac{Source0Level}{Source0Target}, \frac{Money0}{P01(t-1)*Source0Target})$
  $= P01(t-1) * B(\frac{Source0Level}{Source0Target}, \frac{Money0}{P01(t-1)*Source0Target})$ for $Source0$,
- $Valuation(t, \frac{Deliver1Level}{Deliver1Target}, \frac{Money1}{P01(t-1)*Deliver1Target})$
  $= P01(t-1) * B(\frac{Deliver1Level}{Deliver1Target}, \frac{Money1}{P01(t-1)*Deliver1Target})$ for $Deliver1$.
- ...

[9] How to design a valuation function is related to the origin of the value of goods, which is a large question (Dobb, 1981). For example, does value come (i) from the scarcity of goods, (ii) from the work necessary to produce goods, or (iii) from the utility drawn from using goods? $B(\bar{f}, \bar{g})$ implements the first of these three examples.

Asks	Bids
($ask1$) 1 unit at £1.1	($bid1$) 1 unit at £2.2
($ask2$) 1 unit at £2.1	($bid2$) 1 unit at £1.2

(a) Example 1: $askQuote = P_{ask2} = 2.1$ and $bidQuote = P_{bid2} = 1.2$

Asks	Bids
($ask1$) 1 unit at £1.1	($bid1$) 1 units at £2.2

(b) Example 2: $askQuote = P_{bid1} = 2.2$ and $bidQuote = P_{ask1} = 1.1$

Asks	Bids
($ask1$) 1 unit at £1.1	($bid1$) 2 units at £2.2

(c) Example 3: $askQuote = bidQuote = P_{bid1} = 2.2$

Asks	Bids
($ask1$) 1 unit at £1.1	($bid1-1$) 1 unit at £2.2
	($bid1-2$) 1 unit at £2.2

(d) Another representation of Example 3: $askQuote = bidQuote = P_{bid1} = 2.2$

Figure 2: Three examples of clearing by our JASA auctioneer.

- $Valuation(t, \frac{Deliver2Level}{Deliver2Target}, \frac{Money2}{P12(t-1)*Deliver2Target})$
  $= P12(t-1) * B(\frac{Deliver2Level}{Deliver2Target}, \frac{Money2}{P12(t-1)*Deliver2Target})$ for $Deliver2$.

The *bidding strategy* must calculate two values: the price and the quantity shouted. The price shouted is simply the true estimated $Valuation(t, \bar{f}, \bar{g})$, that is, the value of the good actually estimated by the agent without trying to pay less or be paid more. Next, the strategy calculates the quantity shouted in the following way:

- Essentially, the quantity bid is the one needed to keep $\bar{f} = 1$, i.e., to keep the inventory at its target level. That is, a *Source* who wants to buy proposes the quantity $(Source\{0,1\}Level - Source\{0,1\}Target)$, and a *Deliver* who wants to sell bids for $(Deliver\{1,2\}Target - Deliver1,2Level)$ units.

  Since *Deliver*s are not allowed to buy, and *Source*s not to sell, the quantity returned by the previous two subtractions is always positive.

- However, if an inventory (i.e. a *Source*, since *Deliver*s can only sell) wants to buy while it belongs to a company not rich enough (i.e., if the product of the price shouted by the quantity shouted is higher than the funds *Money* owned by the company), then she tries to buy the maximum quantity she can afford at the placed price $Valuation(t, \bar{f}, \bar{g})$, that is, the quantity placed is the largest integer which is less than or equal to $Money/Valuation(t, \bar{f}, \bar{g})$.

3.3 The Clearing House Auctioneer Provided with JASA

Besides the buyers and sellers, an institution is needed to match these two kinds of traders. In our model, this is a JASA auctioneer which calculates *P* in every round. that (Steiglitz *et al.*, 1996, p. 7) “no buyer [should] pay more than his bid” and “no seller [should] sell for less than his offer”. We shall explain how our auctioneer is different from those used by Steiglitz *et al.* (1996), Steiglitz and Shapiro (1998) and Mizuta *et al.* (2003), and also the difference between the broadcast price *P* and the clearing price *Pcl*.

Calculation of the Clearing Price *Pcl*

We now explain the operation of our auctioneer through the three examples in Figure 2. Example 1 in Figure 2(a) assumes four shouts, namely *ask*1, *ask*2, *bid*1 and *bid*2, which are ordered in this figure in ascending order of price for asks, and by descending order of price for bids. With this order, matched shouts are at the top of the table, and unmatched shouts at the bottom. In fact, we can see in the first line of Figure 2(a) that *ask*1 at the lowest sell price £1.1 can be matched with the *bid*1 at the highest buy price £2.2. On the contrary, in the second line, $ask2$ with the second lowest sell price £2.1 cannot be matched with $bid2$ at the second highest buy price £1.2. Since "no buyer [should] pay more than [her] bid" and "no seller [should] sell for less than his offer" (Steiglitz *et al.*, 1996, p. 7), then the auctioneer should choose the clearing price Pcl so that two conditions are satisfied:

- $1.1 \leq Pcl \leq 2.2$ (i.e. $P_{ask1} \leq Pcl \leq P_{bid1}$) in order to match $ask1$ with $bid1$ in the first line, and
- $1.2 < Pcl < 2.1$ (i.e., $P_{bid2} < Pcl < P_{ask2}$) in order *not* to match $ask2$ with $bid2$ in the second line.

Therefore, the auctioneer should choose Pcl so that $1.2 < Pcl < 2.1$. Then, where exactly to place the clearing price Pcl? JASA chooses Pcl by defining two numbers called *askQuote* and *bidQuote* (Phelps, 2007):

- *askQuote* is the price "buyers need to beat in order for their offers to get matched".
- "sellers need to ask less than *bidQuote* in order for their offers to get matched".

In Example 1, $askQuote = P_{ask2} = 2.1$ because a new buyer would have to place a bid shout with a price above P_{ask2} in order to be matched with the unmatched $ask2$. Similarly, a new seller needs to ask less than $bidQuote = P_{bid2} = 1.2$ to have her ask matched with the unmatched $bid2$. Pcl must necessarily be between *askQuote* and *bidQuote* to satisfy the two aforementioned conditions. In this paper, our auctioneer chooses Pcl so that $Pcl = 0.5 * askQuote + 0.5 * bidQuote = 1.65$. Finally, we call $P01$ the broadcast price P and $Pcl01$ the clearing price Pcl in $Market01$, and $P12$ and $Pcl12$ their equivalents in $Market12$.

Next, Examples 2 and 3 in Figures 2(b) and 2(c) illustrate a case often encountered later in this paper. In this case, there is only one buyer and one seller, their offers are matched, but the trader bidding for the highest quantity is favoured. To see this, Example 2 starts with a configuration in which both traders bid for the same quantity. It is easy to check that an additional bid shout should propose less than £2.2 in order to get matched with $ask1$, otherwise $bid1$ will win instead of the new bid shout; thus $bidQuote = P_{ask1} = 2.2$. Similarly, an additional ask shout should propose more than £1.1 to get matched with $bid1$ at the place of $ask1$; thus $askQuote = P_{ask1} = 1.1$. However, let us assume that $bid1$ is not for 1 but for 2 units. This scenario is described in Figure 2(c), which may conveniently rewritten be as Figure 2(d) in which $bid1$ is split into two shouts $bid1 - 1$ and $bid1 - 2$. As before, a new bid shout should propose less than £2.2 in order to get matched with $ask1$, otherwise $bid1$ will win instead of the new bid shout; thus $bidQuote = P_{ask1} = 2.2$. But the difference between Examples 2 and 3 is that a new ask shout should not propose more more than $P_{bid1} = £1.1$ anymore, but less than $P_{bid1-2} = £2.2$, to get matched with $bid1$. As a consequence, *bidQuote* increases up to £2.2, $bidQuote = askQuote$, and the buyer forces Pcl to move in the direction she wants. As explained before, the direction the buyer wants is to increase Pcl, conversely to what intuition states. However, some of the price dynamics analysed in Sections 4 and 5 come from this method used to clear the auction. Specifically, we often obtain smooth price fluctuations when the *Source* buyer and the *Deliver* seller bid for the same quantity, then the *price suddenly changes* because a trader decreases or increases the quantity he or she proposes while the other trader keeps proposing the same quantity. Of course, other auctioneers/clearing algorithms may cause other price dynamics than this. Examples 3 illustrates a phenomenon encountered in the results in this paper when there is only one buyer and one seller in a market: in this scenario, we see that the trader proposing the highest quantity forces the auctioneer to choose his or her price, while the exchanged quantity is proposed by the other trader (in Example 3, the quantity exchanged is the one proposed in $ask1$, and the clearing price is the one asked in $bid1$).

Definition of the Broadcast Price P

Examples 1, 2 and 3 illustrate how Pcl is chosen by the auctioneer when at least one ask shout can be matched with at least one bid shout. If no matches are possible, then $Pcl = 0$. However, choosing

$P = Pcl = 0$ is a problem for the bidding strategy used in this paper, because this makes *all* agents bid a price $P * B(\bar{f}, \bar{g}) = 0$. As a consequence, if $P(t) = 0$ in some round t, then $P(t+k) = 0$ in any round $(t+k), k > 0$. In order to avoid this problem, we make a distinction between the actual clearing price *Pcl* and the price *P* broadcast by the auctioneer. The three papers by Steiglitz do not make explicit this distinction between *P* and *Pcl*, but deal with $Pcl = 0$ in a way which can be described as (Steiglitz *et al.*, 1996, p. 9):

$P(t)$=$Pcl(t)$ when $Pcl \neq 0$;
=$askQuote(t)$ (i.e. the lowest ask price) when no agents buy;
=$bidQuote(t)$ (i.e. the highest bid price) when no agents sell;
=$P(t-1)$ when no agents trade.

Finally, we always start a simulation with $P(t-1) = P(-1) = 1$ in all markets.

3.4 Some Definitions

Balanced Supply Chain

We call "balanced" a supply chain in which (i) the total transformation capacity of all *Manufacturer*s is greater or equal to the total food consumption of all *EndCustomer*s, and (ii) the total production capacity of all *RawMatProd*s is equal to the total food consumption of all *EndCustomer*s, and (iii) the total consumption of money of all *RawMatProd*s is equal to the total production of money of all *EndCustomer*s.

We start all our simulations with a balanced supply chain, i.e. (i) the production of food is balanced with its consumption: $Make0Products = -100$ and $Make1Products = 100$, and (ii) the production of money is balanced with its consumption: $Make0Money = 100$, $Make1Money = -100$ (simulations start with $Money\{0,1,2\} = 1000$). In addition, all inventory targets are the same throughout the paper with $Source0Target = Deliver1Target = Source1Target = Deliver2Target = 1500$, or, shortly, all $InventoryTarget = 1500$.

Equilibrium Price *Peq*

The definition of the equilibrium price *Peq* needs to be adapted from that of (Steiglitz *et al.*, 1996, p. 11), where it is the "price at which just enough agents produce food to satisfy the need of all nonspeculating agents." The idea of this definition is that agents start producing food (respectively, money) when $P > Peq$ (respectively, $P < Peq$) because it is more cost-efficient than producing money (respectively, food), which eventually triggers an excess (respectively, a deficit) of food and thus a decrease of *P* below *Peq* (respectively, an increase of *P* above *Peq*).

Conversely to Steiglitz *et al.* (1996), the price has no influence on the production of food in our supply chain model. Specifically, *Peq* is the ratio of the production of money over the production of products when:

- $Make1Money = 0$ (see Figure 1 for notations),
- the productions of products and money are balanced with their consumption, and
- there is only one company per level of the supply chain: only one *EndCustomer*0, one *Manufacturer*1 and one $RawMatProd\{1,2\}$.

In this very particular case, *Peq* is the same in the single market in Figure 1(a) and in the two markets in Figure 1(b): $P01eq = P12eq = P/M$. In this paper, we always use $M = P = 100$ (e.g. $Make0Products = -100$ and $Make0Money = +100$ for every *EndCustomer*0), so that $P01eq = P02eq = 1$.

However, this paper also considers scenarios violating the third condition, that is, with several companies per level of the supply chain (see Subsections 4.2 and 5.3). In that case, *Peq* is much less trivial and will be studied in future work.

4 The Single Market Scenario

This section presents the price dynamics when some *EndCustomer*0s trade with some *RawMatProd*1s in *Market*01, which corresponds to Figure 1(a). Let us recall that we set all *InventoryTarget*s to 1500 in this paper, but allow *InventoryIni*s to change.

4.1 Price Dynamics in the Single Market with Two Agents

We start with only one *EndCustomer*0 and one *RawMatProd*1, that is, the most simple supply chain possible with only two companies. Table 1 shows that initial conditions are very important in our supply chain model, because the dynamics of *P*01 strongly depend on the initial value of the inventory levels. We now investigate this characteristic of our model and look for regularities in its behaviour. First of all, the most basic setting is in the center of table 1 when $Source0Ini = Source0Target = 1500$ and $Deliver1Ini = Deliver1Target = 1500$. With this configuration, *P*01 smoothly fluctuates around $P01eq = 1$. We call B this pattern of smooth fluctuations because it is the border between the two other patterns in Table 1. As soon as one of both *InventoryIni*s (i.e. either *Source*0*Ini* or *Deliver*1*Ini*) decreases (by one unit since it is the minimal change, because JASA uses integers to represent inventory levels), price fluctuations become chaotic; we call C this chaotic pattern. In stark contrast, as soon as either of both *InventoryIni*s increases, we obtain Pattern A in which *P*01 falls to zero. To explain these three patterns, we should first notice that the initial difference $(\sum InventoryTarget - \sum InventoryIni) > 0$ remains during all the duration of a simulation because (i) the supply chain is balanced, and (ii) if an inventory *Source*0/*Deliver*1 could not buy/sell all the units required to keep her/his level at *InventoryTarget*, then this is memorised in $InventoryLevel \neq InventoryTarget$ and bought/sold later on. With that in mind, we can find the following characteristics of the three patterns:

1. *Pattern C*:

   (a) *When Pattern C happens*: Set $(\sum InventoryTarget - \sum InventoryIni) < 0$, e.g. $Source0Ini = 501$ with $Deliver1Ini = 2500$, and $Source0Ini = 2500$ with $Deliver1Ini = 501$ both incur Pattern C.

   (b) *How Pattern C happens*: Pattern C is chaotic, that is, it looks like a random process, while it is not random at all since the simulation follows deterministic rules.[10] Next, we can describe Pattern C as a succession of two types of periods:

      - *Period of increase of P*01: In such periods, the auctioneer favours the buyer *Source*0 because she bids for more units than *Deliver*1. *Deliver*1 bids for less units because he controls where the initial lack $(\sum InventoryTarget - \sum InventoryIni)$ is, and forces this lack to be with *Source*0. This control works this way: (i) if *Deliver*1 has this lack at the beginning of the simulation, then she places ask shouts for less units than his company *RawMatProd*1 produces during the first rounds of the simulation, so that the lack is transfered to *Source*0, and (ii) if *Source*0 has this lack at the beginning of the simulation, then she places bid shouts for more than she consumes, but she does not receive all these products because *Deliver*1 only proposes what his company produces.
      - *Period of decrease of P*01: In such periods, the auctioneer favours the seller *Deliver*1, because *Source*0 is too poor (*P*01 is too high) to afford all the units, and bids thus for less units than *Deliver*1.

[10] The experiments reported in this paper use no pseudo-random number generators.

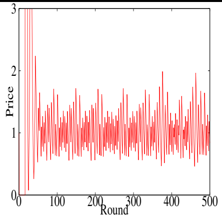
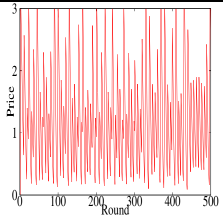
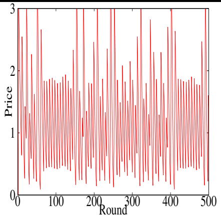
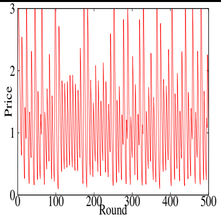
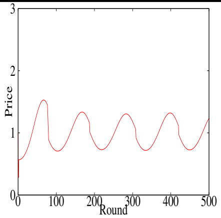
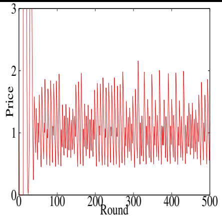
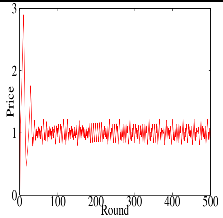
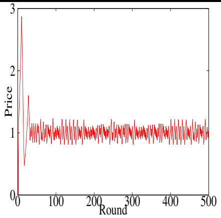
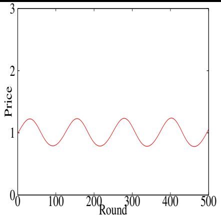
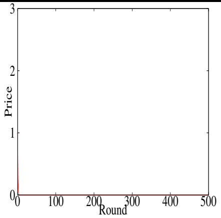
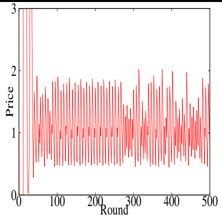
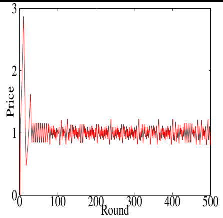
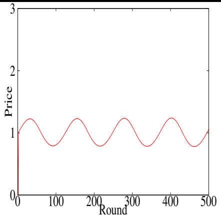
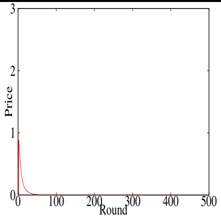
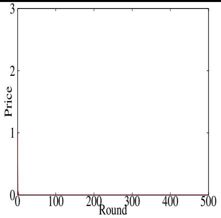
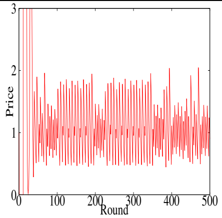
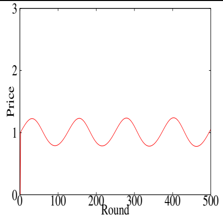
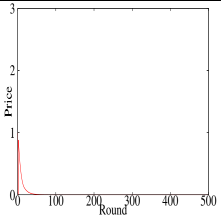
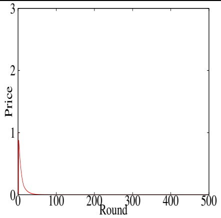
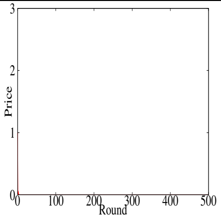
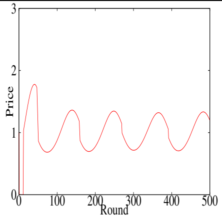
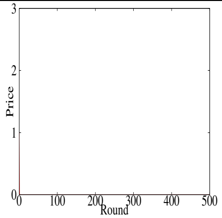
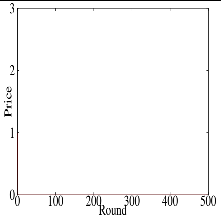
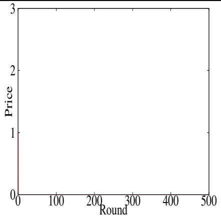
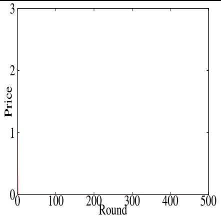
	$Deliver1Ini = 500$	$Deliver1Ini = 1499$	$Deliver1Ini = 1500$	$Deliver1Ini = 1501$	$Deliver1Ini = 2500$
$Source0Ini = 500$	 C	 C	 C	 C	 B
$Source0Ini = 1499$	 C	 C	 C	 B	 A
$Source0Ini = 1500$	 C	 C	 B	 A	 A
$Source0Ini = 1501$	 C	 B	 A	 A	 A
$Source0Ini = 2500$	 B	 A	 A	 A	 A

Table 1: Price dynamics in $Market01$ (with $Source0Target = Deliver1Target = 1500$).

	Start of round								End of Round	
	*Source*0				*Deliver*1				Auctioneer	
Round	Funds	*Source*0−*Level*	Quantity bid	Price bid	Funds	*Deliver*1−*Level*	Quantity asked	Price asked	Quantity exchanged	*P*01
0	1000	1500	0	0	1000	1499	0	0	0	1
1	1100	1400	100	*1.139*	900	1599	*99*	0.882	99	1.139
2	1087	1399	101	*1.296*	913	1600	*100*	1.007	100	1.296
3	1058	1399	101	*1.468*	942	1600	*100*	1.148	100	1.468
4	1011	1399	101	*1.657*	989	1600	*100*	1.302	100	1.657
5	945	1399	101	*1.863*	1055	1600	*100*	1.472	100	1.863
6	1141	1399	101	*2.087*	859	1600	*100*	1.656	100	2.087
7	750	1399	101	*2.329*	1250	1600	*100*	1.856	100	2.329
8	617	1399	101	*2.590*	1383	1600	*100*	2.072	100	2.590
9	458	1399	101	*2.868*	1542	1600	*100*	2.304	100	2.868
10	271	1399	*85*	3.164	1729	1600	100	*2.551*	85	2.551
11	155	1384	*54*	2.850	1845	1615	115	*2.220*	54	2.220
12	135	1338	*51*	2.592	1865	1661	161	*1.818*	51	1.818
13	142	1289	*63*	2.228	1858	1710	210	*1.387*	63	1.387
14	155	1252	*87*	1.769	1848	1747	247	*0.993*	87	0.993
15	168	1239	*130*	1.292	1832	1760	260	*0.682*	130	0.682
16	180	1269	*206*	0.869	1820	1730	230	*0.478*	206	0.478
17	181	1375	125	*0.548*	1819	1624	*124*	0.390	124	0.548

Table 2: Example of simulation trace of Pattern C (winning prices and quantities are in italics).

Since *Source*0 cannot buy all what *EndCustomer*0 consumes, she lacks more than $(\sum \mathit{InventoryTarget} - \sum \mathit{InventoryIni})$ units.

The system alternates between these two kinds of periods, depending on whether *Source*0 has enough money to buy all that she consumes (increase of *P*01), or not (decrease of *P*01). A consequence of this alternation is that the price *P*01 does not fluctuate in a smooth way because it is chosen as being alternatively the price proposed either by the seller or by the buyer.

(c) *Example of Pattern C*: Table 2 illustrates the two aforementioned types of periods: *P*01 increases from Rounds 0 to 10, next decreases from 10 to 17, and increases from 17 on. Numbers in italics indicate the value chosen by the auctioneer. We can see the auctioneer selects (i) the price bid by the *Source*0 buyer and the quantity asked by the *Deliver*1 seller during the increase of *P*01, and (ii) the other way around during the decrease of *P*01. As noted in Example 3 in Figure 2(d), the trader proposing the highest quantity forces the auctioneer to use his or her price, while the exchanged quantity is the one proposed by the other trader. Regarding (ii), remember that, in the “period of decrease of *P*01”, *Source*0 does not bid for all the units she needs because she is too poor to afford that quantity. Finally, we can also read in Table 2 that the initial conditions of the presented data are $\mathit{Source0Ini} = 1500$ with $\mathit{Deliver1Ini} = 1499$.

In summary, in Pattern C, the *Source*0 buyer is always favoured (i.e. *P*01 is the price she proposes), except when she lacks of money in which case the *Deliver*1 seller is favoured (i.e., *P*01 is his price). Switching between the prices proposed by *Source*0 and *Deliver*1 stabilises the price around *P*01*eq* because *Source*0 increases *P*01 as much as she can afford to, while *Deliver*1 decreases *P*01 until *Source*0 can afford to buy all what she consumes. Switching between the prices proposed by these two traders also causes the brutality of the fluctuations of *P*01.

2. *Pattern B*:

(a) *When Pattern B happens*: Set $(\sum \mathit{InventoryTarget} - \sum \mathit{InventoryIni}) = 0$, e.g. $\mathit{Source0Ini} =$

501 with *Deliver*1*Ini* $= 2499$, and *Source*0*Ini* $= 2499$ with *Deliver*1*Ini* $= 501$ both incur Pattern B.

(b) *How Pattern B happens*: Pattern B corresponds to a border between Patterns A and C. Since JASA only allows for integer inventory levels, it is not possible to investigate what happens close to this border, i.e. when $(\sum \mathit{InventoryTarget} - \sum \mathit{InventoryIni}) \approx 0$. As can be seen in Figure 1, Pattern B is made of cycles of slow increases of *P*01, sometimes followed by sudden decreases of *P*01, next always followed by slow decreases of *P*01:

- *Period of slow increase of P*01: In such periods, both *Source*0 and *Deliver*1 bid for the same quantity (100 units), i.e., the excess in one inventory is equal to the lack in the other inventory. Since bid quantities are equal, *Pcl*01 is chosen by the auctioneer half-way between the price proposed by these two inventories, and, because *Source*0 feels richer than *Deliver*1, the price proposed by *Source*0 raises quicker than the price proposed by *Deliver*1 decreases.
- *Sudden decrease of P*01: This is a short period (usually about five rounds) which does not happen with all initial conditions. In the simulation in which it occurs, it concludes a "period of slow increase of *P*01". Visually, this decrease corresponds to a shape different from a sine-like one. When this decrease occurs, it corresponds to the fact that *Source*0 cannot bid for all the products she needs because *P*01 is too high. As a consequence, the auctioneer uses the price proposed by *Deliver*1 as *P*, while it was the price proposed by *Source*0 in the "period of slow increase of *P*01". As a consequence, the quantities bid by both inventories stop to be equal and the auctioneer chooses *P*01 as the price proposed by *Deliver*1, while *P*01 was half-way between the two proposed prices in the previous period. Such a choice makes so that *P*01 stops to have the exponential shape of Function *B* and has instead a sudden decrease.
- *Period of slow decrease of P*01: This period is the opposite of a "period of slow increase of *P*01", i.e., *Deliver*1 feels richer than *Source*0 and makes thus the price decrease.
- *Sudden increase of P*01: We have never observed such an event, but it would correspond to a lack of products by *Deliver*1 (which is the opposite of a "sudden decrease of *P*01" which corresponds to a lack of money by *Source*0).

(c) *Example of Pattern B*: Table 3 illustrates two of the three aforementioned types of periods: *P*01 increases from Rounds 0 to 32, next decreases from 32 to 93, and increases from 93 on. The most noticeable thing in this table is that products do not seem to move because both inventories start and finish at the same level. For example, in every round, *Source*0 starts at 1400, consumes 100 units, purchases 100 units, and finishes at 1400. Next, there is no "Sudden decrease of *P*01", and, therefore, *P*01 is never chosen as the price proposed by either trader. In fact, *P*01 is always chosen half-way between the two propositions, and only the difference of speed of variation between these two proposed prices explains the slow fluctuations of *P*01. This difference of speed of variation is due to the function $B(\bar{f}, \bar{g})$ which depends on both the wealth $\bar{g}$ of the company and the inventory level $\bar{f}$, where only $\bar{g}$ changes while $\bar{f} = 1$ all the time (indeed, an exception is possible: $\bar{f} \neq 1$ during a "Sudden decrease of *P*01").

Essentially, the smooth fluctuations of *P*01 around *P*01*eq* in Pattern B are due to the fact that one inventory is richer (*Source*0 during increases of *P*01, *Deliver*1 during decreases) than the other one while both bid for the same quantity. There may be discontinuities of these smooth fluctuations; in the simulations in which they occur, such discontinuities correspond to a lack of money by the producer of money *EndCustomer*0 which manages *Source*0.

3. *Pattern A*:

	Start of round								End of Round	
	*Source*0				*Deliver*1				Auctioneer	
Round	Funds	*Source*0−*Level*	Quantity bid	Price bid	Funds	*Deliver*1−*Level*	Quantity asked	Price asked	Quantity exchanged	*P*01
0	1000	1500	0	0	1000	1500	0	0	0	1
1	1100	1400	*100*	1.139	900	1600	*100*	0.882	100	1.011
2	1099	1400	*100*	1.151	901	1600	*100*	0.891	100	1.021
3	1097	1400	*100*	1.163	903	1600	*100*	0.901	100	1.032
4	1094	1400	*100*	1.175	906	1600	*100*	0.911	100	1.043
...	...	...	...	...	...	...	...	...	...	...
30	696	1400	*100*	1.375	1304	1600	*100*	1.075	100	1.225
31	673	1400	*100*	1.375	1326	1600	*100*	1.076	100	1.225
32	651	1400	*100*	1.374	1349	1600	*100*	1.075	100	1.225
33	628	1400	*100*	1.373	1372	1600	*100*	1.075	100	1.224
...	...	...	...	...	...	...	...	...	...	...
91	658	1400	*100*	0.895	1342	1600	*100*	0.686	100	0.790
92	679	1400	*100*	0.895	1321	1600	*100*	0.686	100	0.791
93	699	1400	*100*	0.896	1301	1600	*100*	0.687	100	0.791
94	720	1400	*100*	0.898	1280	1600	*100*	0.687	100	0.793

Table 3: Example of simulation trace of Pattern B (winning quantities are in italic).

(a) *When Pattern A happens*: Set $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni}) > 0$, e.g. $\textit{Source}0\textit{Ini} = 499$ with $\textit{Deliver}1\textit{Ini} = 2500$, and $\textit{Source}0\textit{Ini} = 2500$ with $\textit{Deliver}1\textit{Ini} = 499$ both lead to Pattern A.

(b) *How Pattern A happens*: In all rounds, *Deliver*1 sells one unit more than *Source*0 buys, thus the auctioneer chooses the price bid by *Deliver*1 as *P*. Since *Deliver*1 tries to reduce the price in the hope to sell, then *P* decreases. This behaviour is indeed the exact opposite to a "Period of increase of *P*01" in Pattern C.

*P*01 never goes up because we never have the exact opposite of a "Period of decrease of *P*01" in Pattern C, which would be caused by a *Deliver*1 with too few products (which is the opposite of "*Source*0 is too poor"). This seems to indicate that a fourth pattern looking like Pattern C is possible when *InventoryTarget*s are set closer to zero.

Notice that a consequence of the decrease of *P*01 to zero is that *Deliver*1 is not able to acquire the money consumed by his company *RawMatProd*1, which quickly cannot have any of the gold units it is supposed to consume.

Finally, Pattern A looks very unrealistic because *P*01 falls to zero only because of the initial levels of the inventories. Since this would not happen in real life, simulations in which Pattern A occurs should be disregarded. The problem with this pattern is that it seems not to be specific to our auctioneer or to the bidding strategy, that is, it cannot be avoided by fixing something in the code of the simulator. One solution to avoid Pattern A would be to introduce a bidding strategy which looks into the past.

(c) *Example of Pattern A*: Table 4 illustrates how *P*01 decreases forever.

In conclusion, the sign of $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni})$ allows the determination of the pattern of the dynamics of *P*01 when there is only one *Source*0 trading with only one *Deliver*1. We call *Rule 2* this comparison:

> *Rule 2* (temporary version): If one *Source*0 buys in *Market*01 and one *Deliver*1 sells in this market, then:

- If $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni}) < 0$, then *P*01 has a Pattern C;
- If $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni}) = 0$, then *P*01 has a Pattern B;

	Start of round								End of Round	
	*Source*0				*Deliver*1				Auctioneer	
		*Source*0−	Quantity	Price		*Deliver*1−	Quantity	Price	Quantity	
Round	Funds	*Level*	bid	bid	Funds	*Level*	asked	asked	exchanged	*P*01
0	1000	1500	0	0	1000	1501	0	0	0	1
1	1100	1400	*100*	1.138	900	1601	101	*0.880*	100	0.880
2	1112	1400	*100*	1.006	888	1601	101	*0.773*	100	0.773
3	1134	1400	*100*	0.887	865	1601	101	*0.677*	100	0.677
4	1167	1400	*100*	0.780	833	1601	101	*0.592*	100	0.592
...	...	...	...	...	...	...	...	...	...	...

Table 4: Example of simulation trace of Pattern A (winning prices and quantities are in italic).

- If $(\sum \mathit{InventoryTarget} - \sum \mathit{InventoryIni}) > 0$, then $P01$ has a Pattern A.

The next subsection introduces *Rule 1* to apply before *Rule 2*, and slightly modifies *Rule 2* in order to accomodate with the scenario in which more than one *Source*0 and more than one *Deliver*1 trade in *Market*01.

4.2 Price Dynamics in the Single Market with Many Agents

We now study what happens when there are several *Source*0s buying from several *Deliver*1s. As in the rest of this paper, all *InventoryTarget*s are set to 1500 in this subsection. Since we noticed in the previous subsection that the sign of $(\sum \mathit{InventoryTarget} - \sum \mathit{InventoryIni})$ seems to be more important than the actual value of the different *InventoryTarget*s and *InventoryIni*s (*Rule 2*), the cases $\mathit{InventoryIni} = 500$ and $\mathit{InventoryIni} = 2500$ are not taken into account in this subsection. Table 5 proposes a small sample of all the possible combinations of several *Source*0s trading with several *Deliver*1s. First of all, we obtain the same three patterns A, B and C of *P*01 as in Figure 1.

Next, Table 5 should be understood as follows. The first line presents two configurations: the left one is “111 111” in which three *Source*0s (starting at levels 1499, 1500 and 1501) buy from three *Deliver*1s (starting at levels 1499, 1500 and 1501), which incurs Pattern B, while, the right configuration of the first line is “211 111” in which four *Source*0s (starting at levels 1499, 1499, 1500 and 1501) buy from three *Deliver*1s (starting at levels 1499, 1500 and 1501) and a Pattern C is obtained.

We first check that *Rule 2* is not enough to predict what pattern will happen when there are many agents. In fact, $(\sum \mathit{InventoryTarget}) - (\sum \mathit{InventoryIni})$ may be rewritten as $(\sum_{i=0}^{\#Source0} \mathit{Source0_iTarget} + \sum_{i=0}^{\#Deliver1} \mathit{Deliver1_jTarget})$ - $(\sum_{i=0}^{\#Source0} \mathit{Source0_iIni} + \sum_{i=0}^{\#Deliver1} \mathit{Deliver1_jIni})$, where $\#\mathit{Source0}$ is the number of *Source*0s. The entry “111 121” (left column in third line) provides us with an example showing that this reading of *Rule 2* does not work: Table 5 reports that the simulation exhibits Pattern A, while *Rule 2* would propose Pattern B:

- $\sum_{i=0}^{\#Source0} \mathit{Source0_iTarget} = 1500 * 3 = 4500$,
- $\sum_{j=0}^{\#Deliver1} \mathit{Deliver1_jTarget} = 1500 * 4 = 6000$,
- $\sum_{i=0}^{\#Source0} \mathit{Source0_iIni} = 1499 + 1500 + 1501 = 4500$,
- $\sum_{j=0}^{\#Deliver1} \mathit{Deliver1_jIni} = 1499 + 1500 + 1500 + 1501 = 6000$,
- $\Rightarrow (\sum \mathit{InventoryTarget}) - (\sum \mathit{InventoryIni}) = (4500 + 6000) - (4500 + 6000) = 0 \Rightarrow$ Pattern B.

This example demonstrates that adding one $\mathit{Deliver1_j}$ starting with $\mathit{Deliver1_jIni} = \mathit{Deliver1_jTarget}$ does not change the sign of $(\sum \mathit{InventoryTarget} - \sum \mathit{InventoryIni})$, while this $\mathit{Deliver1_j}$ proposes products to sell in *Market*01 and impacts thus on *P*01.

# *Source0Ini* = 1499	# *Source0Ini* = 1500	# *Source0Ini* = 1501	# *Deliver1Ini* = 1499	# *Deliver1Ini* = 1500	# *Deliver1Ini* = 1501	# *Source0* Vs. # *Deliver1*	Pattern of *P01*		# *Source0Ini* = 1499	# *Source0Ini* = 1500	# *Source0Ini* = 1501	# *Deliver1Ini* = 1499	# *Deliver1Ini* = 1500	# *Deliver1Ini* = 1501	# *Source0* Vs. # *Deliver1*	Pattern of *P01*
1	1	1	1	1	1	$3 = 3$	B		2	1	1	1	1	1	$4 > 3$	C
1	1	1	1	1	2	$3 < 4$	A		2	1	1	1	1	2	$4 = 4$	B
1	1	1	1	2	1	$3 < 4$	A		2	1	1	1	2	1	$4 = 4$	C
1	1	1	1	2	2	$3 < 5$	A		2	1	1	1	2	2	$4 < 5$	A
1	1	1	2	1	1	$3 < 4$	A		2	1	1	2	1	1	$4 = 4$	C
1	1	1	2	1	2	$3 < 5$	A		2	1	1	2	1	2	$4 < 5$	A
1	1	1	2	2	1	$3 < 5$	A		2	1	1	2	2	1	$4 < 5$	A
1	1	1	2	2	2	$3 < 6$	A		2	1	1	2	2	2	$4 < 6$	A
1	1	2	1	1	1	$4 > 3$	C		2	1	2	1	1	1	$5 > 3$	C
1	1	2	1	1	2	$4 = 4$	A		2	1	2	1	1	2	$5 > 4$	C
1	1	2	1	2	1	$4 = 4$	A		2	1	2	1	2	1	$5 > 4$	C
1	1	2	1	2	2	$4 < 5$	A		2	1	2	1	2	2	$5 = 5$	A
1	1	2	2	1	1	$4 = 4$	B		2	1	2	2	1	1	$5 > 4$	C
1	1	2	2	1	2	$4 < 5$	A		2	1	2	2	1	2	$5 = 5$	B
1	1	2	2	2	1	$4 < 5$	A		2	1	2	2	2	1	$5 = 5$	C
1	1	2	2	2	2	$4 < 6$	A		2	1	2	2	2	2	$5 < 6$	A
1	2	1	1	1	1	$4 > 3$	C		2	2	1	1	1	1	$5 > 3$	C
1	2	1	1	1	2	$4 = 4$	A		2	2	1	1	1	2	$5 > 4$	C
1	2	1	1	2	1	$4 = 4$	B		2	2	1	1	2	1	$5 > 4$	C
1	2	1	1	2	2	$4 < 5$	A		2	2	1	1	2	2	$5 = 5$	B
1	2	1	2	1	1	$4 = 4$	C		2	2	1	2	1	1	$5 > 4$	C
1	2	1	2	1	2	$4 < 5$	A		2	2	1	2	1	2	$5 = 5$	C
1	2	1	2	2	1	$4 < 5$	A		2	2	1	2	2	1	$5 = 5$	C
1	2	1	2	2	2	$4 < 6$	A		2	2	1	2	2	2	$5 < 6$	A
1	2	2	1	1	1	$5 > 3$	C		2	2	2	1	1	1	$6 > 3$	C
1	2	2	1	1	2	$5 > 4$	C		2	2	2	1	1	2	$6 > 4$	C
1	2	2	1	2	1	$5 > 4$	C		2	2	2	1	2	1	$6 > 4$	C
1	2	2	1	2	2	$5 = 5$	A		2	2	2	1	2	2	$6 > 5$	C
1	2	2	2	1	1	$5 > 4$	C		2	2	2	2	1	1	$6 > 4$	C
1	2	2	2	1	2	$5 = 5$	A		2	2	2	2	1	2	$6 > 5$	C
1	2	2	2	2	1	$5 = 5$	B		2	2	2	2	2	1	$6 > 5$	C
1	2	2	2	2	2	$5 < 6$	A		2	2	2	2	2	2	$6 = 6$	B

Table 5: Pattern of the dynamics of *P*01 when there are 3, 4, 5 or 6 *Source*0s trading with 3, 4, 5 or 6 *Deliver*1s.

Therefore, *Rule 2* is not enough because the relative numbers of sellers and buyers should also be taken into account. That's why Table 5 presents the number $\#Source0$ of buyers and $\#Deliver1$ of sellers. With these notations, the results in Table 5 seem to indicate that the three patterns A, B and C of $P01$ have the following characteristics:

1. *Pattern C*:

   (a) *When Pattern C happens*:

   - Either $(\#Source0 - \#Deliver1) = 0$ and $(\sum InventoryTarget - \sum InventoryIni) < 0$,
   - Or $(\#Source0 - \#Deliver1) > 0$.

   (b) *How Pattern C happens*: The first condition is very similar to the previous subsection, that is, the case $(\#Source0 = \#Deliver1 = 1)$ in the previous subsection resembles the case $(\#Source0 = \#Deliver1 > 1)$. Specifically, we can see these initial conditions as setting a system with $\#Source0 = \#Deliver1$ auctions running in parallel, where every auction has one $Source0$ matched with one $Deliver1$ (the matching is different in every round), and where $Deliver1$s collectively force $Source0$s to keep or receive the initial lack of products $(\sum InventoryTarget - \sum InventoryIni)$ at the beginning of the simulation. In other words, we observe the same two kinds of periods as for Pattern C in the previous subsection.

   The second condition $(\#Source0 - \#Deliver1 > 0)$ is also quite similar to what happens in the previous subsection. More precisely, there are now more $Source0$s than $Deliver1$s which means that more products are consumed than produced. This disbalance leads to the same two kinds of periods:

   - *Periods of decrease of* $P01$: These periods are as in the previous subsection, that is, $Source0$s are too poor to afford all what they consume because $P01$ is too high. As a consequence, the total quantity ordered by $Source0$s is lower than the total quantity ordered by $Deliver1$, which causes one of the prices proposed by a $Deliver1$ to be chosen as $P01$.
   - *Periods of increase of* $P01$: Basically, the total quantity consumed by buyers is greater than the total quantity produced by sellers, and thus, the total quantity to buy should be greater than the total quantity for sale. However, we have just seen that this does not work this way when $P01$ is too high. This problem of wealth of the buyers does not apply (or, at least, is is less acute) during a period of increase of $P01$. As a consequence, buyers now bid for a quantity higher than what is proposed by sellers.

   (c) *Example of Pattern C*: Figures 3 and 4 illustrate these two types of periods:

   - *Periods of decrease of* $P01$: Figure 3 illustrates this "period of decrease of $P01$" with the first round in which $P01$ decreases (round 8) when there are four $Source0$s (starting with levels 1499, 1499, 1500 and 1501) and three $Deliver1$ (starting at levels 1499, 1500 and 1501). Figure 3(a) presents the quantities and prices bid by the four $Source0$s and asked by the three $Deliver1$s. As in the examples in Figure 2, asks are written in ascending order of price, and bids in descending order of price. Figure 3(b) presents how the auctioneer splits these shouts. For example, $ask1$ is split into $ask1-1$ and $ask1-2$ so that $ask1-1$ can be matched with $bid4$ and $ask1-2$ with the part $bid3-1$ of $bid3$. With this representation, we can see that any new ask must be below P_{ask3-2} to get matched with $bid1$, i.e. to beat $ask3$, thus $bidQuote = P_{ask3-2}$, and any new bid must be above P_{ask3-2} to afford some of the 69 units of $ask3-2$, thus $askQuote = P_{ask3-2}$.

     This example illustrates how sellers are collectively favoured by the auctioneer because they sell a total quantity higher than the total demand. Notice that all the prices asked may be matched by all the prices bid by definition of $Valuation(t, \bar{f}, \bar{g})$, and, therefore,

Asks	Bids
($ask1$) 100 units at £4.85256341	($bid4$) 91 units at £7.278684
($ask2$) 100 units at £4.85259734	($bid3$) 30 units at £7.094054
($ask3$) 100 units at £4.85263601	($bid2$) 30 units at £6.631892
	($bid1$) 80 units at £4.852597

(a) Ask and bid shouts.

Asks	Bids
($ask1-1$) 91 units at £4.85256341	($bid4$) 91 units at £7.278684
($ask1-2$) 09 units at £4.85256341	($bid3-1$) 09 units at £7.094054
($ask2-1$) 21 units at £4.85259734	($bid3-2$) 21 units at £7.094054
($ask2-2$) 30 units at £4.85259734	($bid2$) 30 units at £6.631892
($ask2-3$) 49 units at £4.85259734	($bid1-1$) 49 units at £4.852597
($ask3-1$) 31 units at £4.85263601	($bid1-2$) 31 units at £4.852597
($ask3-2$) 69 units at £4.85263601	

(b) Transformation of asks and bids shouts to see that $askQuote = bidQuote = P_{ask3-2} = 4.85263601$ in Figure 3(a).

Figure 3: Example of decrease of $P01$ in Pattern C.

the only way to influence $P01$ is to propose more products, as done here by the sellers. In fact, the buyers would like to bid for the same quantity as what is proposed by the sellers, but are too poor to afford this quantity. As a consequence, the price proposed by one of these sellers (here, P_{ask3-2}) is used as $P01$, and since sellers always try to decrease the price, then $P01(t) < P01(t-1)$.

- *Periods of increase of* $P01$: Table 4 illustrates a round during a period of increase of $P01$. The round considered is the fifteenth of the same simulation as Figure 3, which corresponds to the first round of the second period of increase in this simulation. More precisely, Figure 4(a) presents the shouts placed by the seven traders, and Table 4(b) how we can split these shouts to make *askQuote* and *bidQuote* obvious. The main thing to notice is that $P01$ is now necessarily one of the P_{bid}s because buyers bid for a higher quantity, while it was one of the P_{ask}s in Figure 3.

Shortly, $P01$ suddenly “jumps”, as in the Pattern C in the previous subsection, from one of the P_{ask}s to one of the P_{bid}s when we change of period, which explains why $P01$ does not fluctuate smoothly. As a conclusion about Pattern A, we can say that this pattern occurs for same reasons when there is only one trader per level of the supply chain, and where there are more than one trader.

2. *Pattern B*:

   (a) *When Pattern B happens*:

   - Only when $(\#Source0-\#Deliver1) = 0$ and $(\sum InventoryTarget - \sum InventoryIni) = 0$.

   (b) *How Pattern B happens*: As with Pattern C, the case $(\#Source0 = \#Deliver1 = 1)$ of Pattern B resembles the case $(\#Source0 = \#Deliver1 > 1)$. Again, everything happens as if $\#Source0 = \#Deliver1$ simulations were carried out in parallel. In the first few rounds, traders with an excess (respectively, a lack) products bids for more (respectively, for less), and are able to transfer this excess (respectively, lack) to another inventory when this second inventory has a lack (respectively, an excess). If this transfer does not occur or is not completed in a round, it may take place in the next round, so that, all inventories eventually

Asks	Bids
($ask1$) 229 units at £0.36408550	($bid4$) 229 units at £1.04207499
($ask2$) 216 units at £0.37159512	($bid3$) 272 units at £0.95046649
($ask3$) 100 units at £0.45293937	($bid2$) 280 units at £0.93427686
	($bid1$) 283 units at £0.91019291

(a) Ask and bid shouts.

Asks	Bids
($ask1$) 229 units at £0.36408550	($bid4-1$) 263 units at £1.04207499
($ask2-1$) 34 units at £0.37159512	($bid4-2$) 34 units at £1.04207499
($ask2-2$) 182 units at £0.37159512	($bid3-1$) 182 units at £0.95046649
($ask3-1$) 90 units at £0.45293937	($bid3-2$) 90 units at £0.95046649
($ask3-2$) 10 units at £0.45293937	($bid2-1$) 90 units at £0.93427686
	($bid2-2$) 190 units at £0.93427686
	($bid1$) 283 units at £0.91019291

(b) Transformation of asks and bids shouts to see that $askQuote = bidQuote = P_{bid2} = 0.93427686$ in Figure 4(a).

Figure 4: Example of increase of $P01$ in Pattern C.

have their level at their *InventoryTarget*. Next, in every round after this equilibration period, every $Source0$ is matched with a $Deliver1$ and the same exchange takes place in each pair $Source0/Deliver1$ as in the previous subsection.

In a few words, Pattern B happens again because buyers are alternatively richer then poorer than sellers.

Notice that the conditions incurring Pattern B are the most intuitive way to set a simulation and this pattern will thus occur quite often, even though these conditions are very particular.

3. *Pattern A*:

   (a) *When Pattern A happens*:

   - Either $(\#Source0 - \#Deliver1) = 0$ and $(\sum InventoryTarget - \sum InventoryIni) > 0$,
   - Or $(\#Source0 - \#Deliver1) < 0$.

   (b) *How Pattern A happens*: Again, the case $(\#Source0 = \#Deliver1 = 1)$ of Pattern A resembles the case $(\#Source0 = \#Deliver1 > 1)$, in which $P01$ falls to zero because the sellers (instead of the single seller) are favoured by the auctioneer due to the fact they sell more than the buyers.

In conclusion, the sign of $(\#Source0 - \#Deliver1)$ allows the determination of the pattern of the dynamics of $P01$ when there are several $Source0$s trading with several $Deliver1$s. The reasons for this are almost the same as in the previous subsection. We call *Rule 1* this comparison:

Rule 1: If some $Source0$s buy in $Market01$, and some $Deliver1$ sell in this market, then:

- If $(\#Source0 - \#Deliver1) > 0$, then $P01$ has a Pattern C;
- If $(\#Source0 - \#Deliver1) = 0$, then apply *Rule 2*;
- If $(\#Source0 - \#Deliver1) < 0$, then $P01$ has a Pattern A.

In order to be used with *Rule 1*, *Rule 2* needs to be slightly rewritten as:

Rule 2: If as many *Source*0s buy in *Market*01 as many *Deliver*1s sell in this market, then:

- If $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni}) < 0$, then $P01$ has a Pattern C;
- If $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni}) = 0$, then $P01$ has a Pattern B;
- If $(\sum \textit{InventoryTarget} - \sum \textit{InventoryIni}) > 0$, then $P01$ has a Pattern A.

5 The Two Market Scenario

We now detail the price dynamics of $P01$ and $P02$ in the two auctions of the supply chain in Figure 1(b). For that purpose, we first sketch the changes in the considered scenario in comparison with the previous section. Next, we present the price dynamics when there is the minimal number of agents, i.e. one agent at each level of the supply chain. Finally, we outline how we expect to study scenarios with more agents in the future.

5.1 Presentation of the Two Markets and the Three Agents

In comparison with the previous section, we consider the two auctions *Market*01 and *Market*12 instead of only *Market*01, which leads us to add *Manufacturer*1, and to change the name of the raw material supplier from *RawMatProd*1 to *RawMatProd*2.

5.2 Price Dynamics in the Two Markets with Three Agents

The simulation of two auctions with one seller and one buyer per auction shows the same Patterns A, B and C as in the previous section (see Appendix A for details). As a consequence, we can summarise the dynamics of $P01$ and $P12$ with Table 6. In fact, it is even possible to generate Table 6 from (any version of) *Rule 2* (*Rule 1* does not apply here because there are not more than one buyer and one seller per market). In order to illustrate this, let us consider the case $\textit{Source}0\textit{Ini} = \textit{Deliver}1\textit{Ini} = \textit{Source}1\textit{Ini} = 1501$ and $\textit{Deliver}2\textit{Ini} = 1499$ (i.e. the lower right entry in Table 6, and the bottom left entry in Table 9) which has Pattern A twice. *Market*01 has Pattern A according to *Rule 2* because $\textit{Source}0\textit{Ini}+\textit{Deliver}1\textit{Ini} = 1501+1501$ is greater than $\textit{Source}0\textit{Target}+\textit{Deliver}1\textit{Target} = 1500+1500$. But there seems to be a problem with *Market*12 which should have Pattern B according to *Rule 2* (because $\textit{Source}1\textit{Ini} + \textit{Deliver}2\textit{Init} = 1501 + 1499$ is equal to $\textit{Source}1\textit{Target} + \textit{Deliver}2\textit{Target} = 1500 + 1500$), but is replaced by Pattern A in Table 6.

When the application of *Rule 2* does not match the results obtained by simulation, the pattern obtained by simulation is written in italics in Tables 6 (as well as Tables 7, 8 and 9). We can see that italics is only for "A"s in *Market*12. The explanation for this is that a Pattern A in *Market*01 makes so that *Manufacturer*1 is not able to attract money from the producer of money (i.e. *EndCustomer*0) because the price falls to zero. As a consequence, *Manufacturer*1 cannot send this money into *Market*12, and, therefore, $P12$ cannot have its normal pattern due to the fact that *Manufacturer*1 becomes poorer and poorer. This explains why the differences between the application of *Rule 2* and actual simulation results only (i) affect *Market*12, (ii) deal with Pattern A in *Market*01 and (iii) incur Pattern A in *Market*12 but never Patterns B or C. Eventually, we can infer *Rule 3* from Table 6:

Rule 3: If a market (*Market*01 in our case) has Pattern A, then a market further from *EndCustomer*s (*Market*12 in our case) will also have Pattern A.

Therefore, *Rule 2* should be applied first, next *Rule 3*. As described in the next subsection when there are several buyers and sellers in some market, whether *Rule 1* should be applied before *Rule 2* is left for future work.

		Source1Ini=1499						Source1Ini=1500						Source1Ini=1501					
		Deliver2Ini						Deliver2Ini						Deliver2Ini					
		=1499		=1500		=1501		=1499		=1500		=1501		=1499		=1500		=1501	
Source0Ini	Deliver1Ini	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$	$P01$	$P12$
=1499	=1499	C	C	C	C	C	B	C	C	C	B	C	A	C	B	C	A	C	A
	=1500	C	C	C	C	C	B	C	C	C	B	C	A	C	B	C	A	C	A
	=1501	B	C	B	C	B	B	B	C	B	B	B	A	B	B	B	A	B	A
=1500	=1499	C	C	C	C	C	B	C	C	C	B	C	A	C	B	C	A	C	A
	=1500	B	C	B	C	B	B	B	C	B	B	B	A	B	B	B	A	B	A
	=1501	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*
=1501	=1499	B	C	B	C	B	B	B	C	B	B	B	A	B	B	B	A	B	A
	=1500	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*
	=1501	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*	A	*A*

Table 6: Price dynamics of $P01$ and $P12$ (details in Tables 7, 8 and 9).

5.3 Price Dynamics in the Two Markets with Many Agents

Exploring the dynamics of $P01$ and $P12$ when there are several companies at both levels of the supply chain requires many simulations. We have not yet undertaken them because this would require automation of the recognition of Patterns A, B and C (and, perhaps, D, E, etc.). This is left for future work. However, initial results obtained in a few cases with sight recognition seem to show that applying *Rule 1*, then *Rule 2* and finally *Rule 3* allows determining the dynamics of $P01$ and $P12$.

6 Conclusions

In this paper we have presented a model of market-mediated supply chains. Our purpose is to study how conceptual tools designed to control a single market may be extended to the control of linked networks of markets. Specifically, our model is based on the single auction and the bidding strategy proposed by Steiglitz and his colleagues. We replace their agents by company-agents represented with the first level of Supply Chain Council's SCOR model. Finally, we implemented our model within JASA and ran simulations with one or with two markets in sequence.

The results obtained from these simulations can be summarised as follows. First, only three patterns of price dynamics are obtained. Next, setting the parameters of a market-mediated supply chain is more complicated than just balancing (i) consumption of products, transformation capacities and supply of products, and (ii) consumption and production of money. In fact, market dynamics also play a role. In our model, such dynamics are influenced by the difference between the initial and the target levels of the inventories used to trade in an auction. We have identified and explained the relations between these initial conditions of the inventories and the three observed price dynamics. These relations are summarized by two rules predicting price dynamics. Finally, we studied the impact of the price dynamics in one market on the price dynamics in the other market. Our insights are summarized in a third rule.

In this paper, all agents have the same inventory target. The first task in extending this work would be to continue the study of the regularities found between price dynamics and the initial conditions of the simulation in order allow for different inventory targets. Such an extension may introduce new patterns of price dynamics, because stock outs may occur for reasons not considered with the scenarios considered in this paper (i.e. stock outs only occur here when the price drops to zero, which makes manufacturers unable to receive money from one market in order to buy in the other market). Another interesting point here deals with the fact that inventory targets should be optimised so that the agents reduce their inventory holding costs. In fact, all the behaviours explored in this paper arise from an automatic procedure for placing shouts in auctions, but no more "intelligent" decisions than these are made. Finally, we plan to study different topologies of networks instead of the sequential (straight-line) structure considered in this paper; that is, we have so far only considered auctions linked in sequence and would also like consider auctions linked otherwise, for instance, in parallel. We believe these non-sequential topologies will require companies to each have several "source" and "deliver" inventories, instead of only one of each; these different inventories would trade in different auctions. For example, a manufacturer could have one "deliver" inventory selling in the bolt market, and two "source" inventories, one buying in the screw market and the other one in the nut market. Such an architecture is under construction for future research and publication.

Acknowledgments

This research was undertaken as part of the EPSRC-funded project on *Market-Based Control of Complex Computational Systems* (GR/T10664/01), and we are grateful for this support. We also thank Omar Baqueiro Espinosa, Andrew Byde, Andrew Dowell, Enrico Gerding, Nick Jennings, Tomasz Michalak and Steve Phelps for their comments and suggestions.

References

Anthes, G. (2003). Agents of change. *Computer World*. http://www.computerworld.com/printthis/2003/0,4814,77855,00.html (accessed 19 July 2006).

Barbuceanu, M. and Fox, M. S. (1996). Capturing and modeling coordination knowledge for multi-agent systems. *International Journal on Cooperative Information Systems*, **5**(2 & 3), 275–314.

Clearwater, S. H., editor (1996). *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific: Singapore.

Cloutier, L., Frayret, J.-M., D'Amours, S., Espinasse, B., and Montreuil, B. (2001). A commitment-oriented framework for networked manufacturing coordination. *International Journal of Computer Integrated Manufacturing*, **14**(6), 522–534.

Dobb, M. (1981). *Theories of Value and Distribution Since Adam Smith*. Cambridge University Press, 5th reprint of the 2d edition edition.

Dodd, C. and Kumara, S. R. T. (2001). A distributed multi-agent model for value nets. In *Proc. 14th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE)*, volume 2070 of *Lecture Notes in Artificial Intelligence*, pages 718–727, Budapest (Hungary).

Forrester, J. W. (1958). Industrial dynamics - A major breakthrough for decision-makers. *Harvard Business Review*, **36**(4), 37–66.

Fox, M. S., Chionglo, J. F., and Barbuceanu, M. (1993). The integrated supply chain management. Internal report of the Enterprise Integration Laboratory, Department of Industrial Engineering, University of Toronto, Ontario, Canada.

Fox, M. S., Barbuceanu, M., and Teigen, R. (2000). Agent-oriented supply-chain management. *International Journal of Flexible Manufacturing Systems*, **12**(2/3), 165–188.

Geoffrion, A. M. and Krishnan, R. (2003). E-business and management science - Mutual impacts (part 2 of 2). *Management Science*, **49**(11), 1445–1456.

Lee, H. L., Padmanabhan, V., and Whang, S. (1997a). The bullwhip effect in supply chain. *Sloan Management Review*, **38**(3), 93–102.

Lee, H. L., Padmanabhan, V., and Whang, S. (1997b). Information distortion in a supply chain: The bullwhip effect. *Management Science*, **43**(4), 546–558.

Mizuta, H., Steiglitz, K., and Lirov, E. (2003). Effects of price signal choices on market stability. *Journal of Economic Behavior and Organization*, **52**, 235–251.

Moyaux, T. and McBurney, P. (2006a). Modelling a supply chain as a network of markets. In *Proceedings of the IEEE International Conference on Service Systems and Service Managament, (ICSSSM 2006)*, Troyes, France.

Moyaux, T. and McBurney, P. (2006b). Reduction of the bullwhip effect in supply chains through speculation. In C. Bruun, editor, *Proceedings of the Symposium on Artificial Economics 2006, Lecture Notes in Economics and Mathematical Systems 584 (Springer)*, pages 77–89, Aalborg, Denmark.

Phelps, S. (2007). *Evolutionary Mechanism Design*. Ph. D thesis, University of Liverpool (U.K.). *Forthcoming*.

Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E. (2000). *Designing and Managing the Supply Chain*. McGraw-Hill Higher Education.

Steiglitz, K. and Shapiro, D. (1998). Simulating the madness of crowds: Price bubbles in an auction-mediated robot market. *Computational Economics*, **12**, 35–59.

Steiglitz, K., Honig, M. L., and Cohen, L. M. (1996). A computational market model based on individual action. In S. H. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*, pages 1–27. World Scientific: Singapore.

Sterman, J. D. (1989). Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science*, **35**(3), 321–339.

A Detail of the two Auctions with one Seller and Buyer per Auction

This appendix details the results summarised in Table 6. That is, Figures 7, 8 and 9 present the dynamics of $P01$ and $P12$ depending on the initial conditions of each of the four inventories $Source0$, $Deliver1$, $Source1$ and $Deliver2$. Table 6 only presented the type of price dynamics of $P01$ and $P12$, while Tables 7, 8 and 9 show what happens in detail. The upper graph in any entry in Tables 7, 8 and 9 represents $P01$, while the graph at the bottom is shows $Market12$.

The first point to note is that the price in these three figures has the same Patterns A, B and C, as summarised in Table 6. Next, we observe a relationship between $P01$ and $P12$. For example, we never have smooth fluctuations of both $P01$ and $P12$ at the same time. That is, if $P01$ and $P12$ both have Pattern B, then their fluctuations are never sine-like. To see that, consider the case in which all inventories start at their target levels (i.e. case $Source0Ini = Deliver1Ini = Source1Ini = Deliver2Ini = 1500$, which is in the center of Table 8). According to what we observed in Subsection 4.1, we should obtain a Pattern B in both markets. Indeed, this is what we obtain, except that the smooth fluctuations have lost their regularity. This can be informally interpreted in this way: (i) the smooth fluctuations are caused by a seller feeling richer when his buyer feels poorer, and the other way around (see explanations about Pattern B in Subsection 4.1), but (ii) $Manufacturer1$ is both a seller in $Market01$ and a buyer in $Market12$. In (Moyaux and McBurney, 2006b), we related such an impact of one market on another to a consequence of the "bullwhip effect" (Forrester, 1958; Lee *et al.*, 1997a,b) on prices.

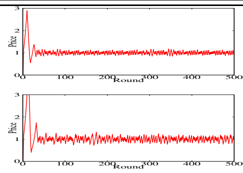
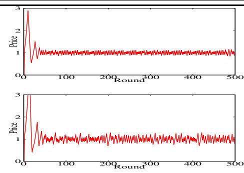
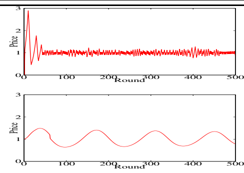
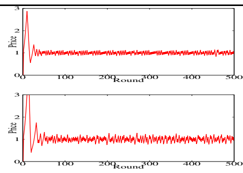
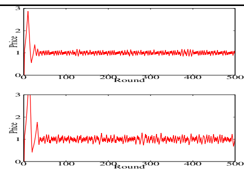
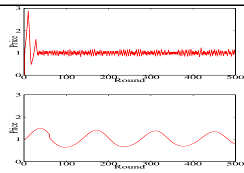
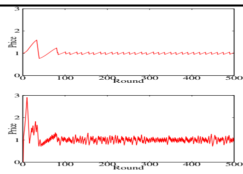
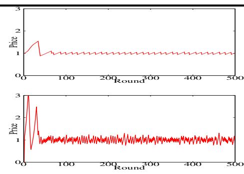
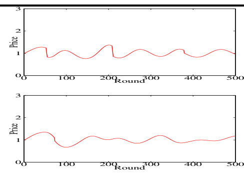
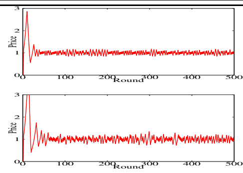
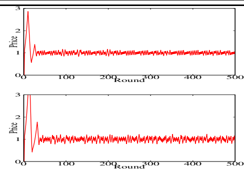
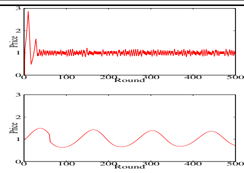
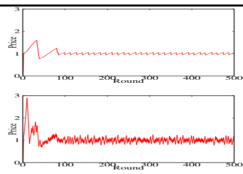
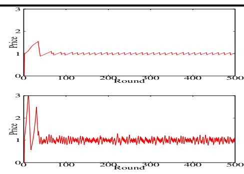
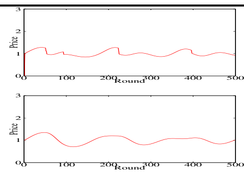
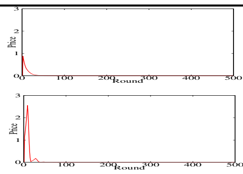
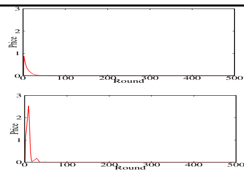
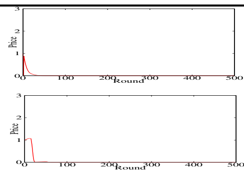
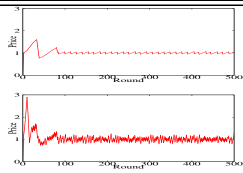
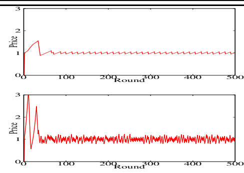
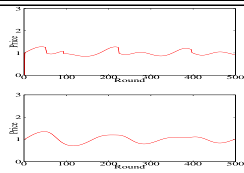
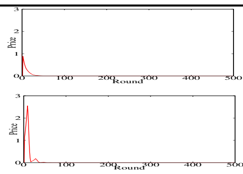
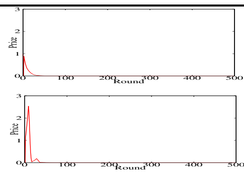
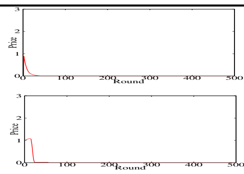
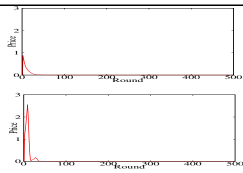
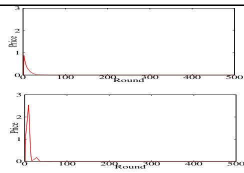
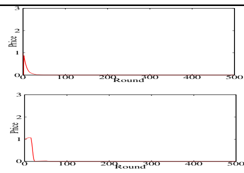
	$Source1Ini = 1499$, $Deliver2Ini = 1499$	$Source1Ini = 1499$, $Deliver2Ini = 1500$	$Source1Ini = 1499$, $Deliver2Ini = 1501$
$Source0Ini = 1499$ $Deliver1Ini = 1499$	 C C	 C C	 C B
$Source0Ini = 1499$ $Deliver1Ini = 1500$	 C C	 C C	 C B
$Source0Ini = 1499$ $Deliver1Ini = 1501$	 B C	 B C	 B B
$Source0Ini = 1500$ $Deliver1Ini = 1499$	 C C	 C C	 C B
$Source0Ini = 1500$ $Deliver1Ini = 1500$	 B C	 B C	 B B
$Source0Ini = 1500$ $Deliver1Ini = 1501$	 A *A*	 A *A*	 A *A*
$Source0Ini = 1501$ $Deliver1Ini = 1499$	 B C	 B C	 B B
$Source0Ini = 1501$ $Deliver1Ini = 1500$	 A *A*	 A *A*	 A *A*
$Source0Ini = 1501$ $Deliver1Ini = 1501$	 A *A*	 A *A*	 A *A*

Table 7: Detail of Table 6 showing the price dynamics of $P01$ on top and $P12$ at bottom (1/3).

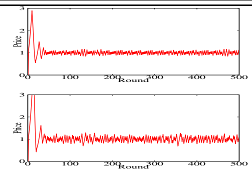
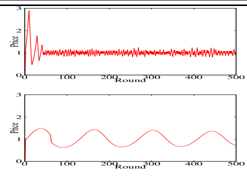
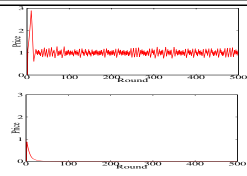
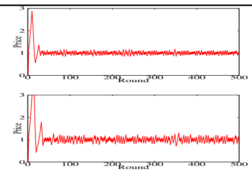
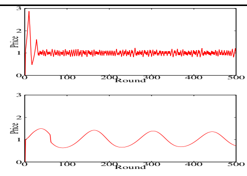
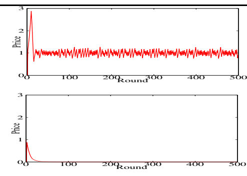
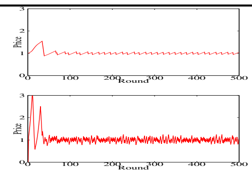
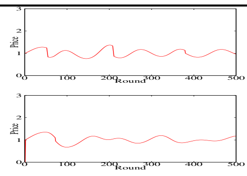
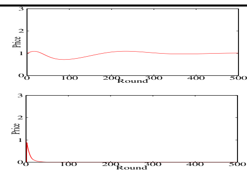
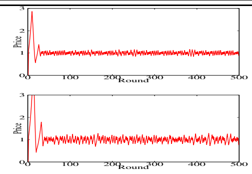
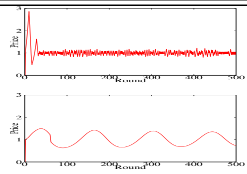
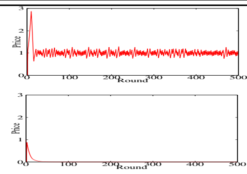
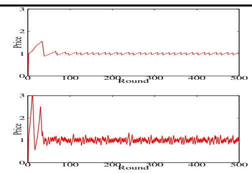
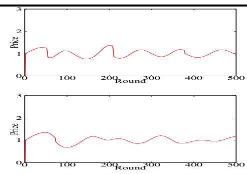
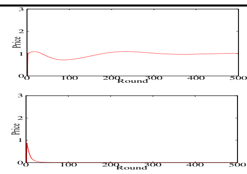
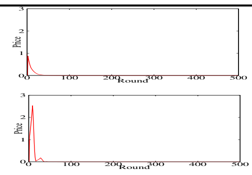
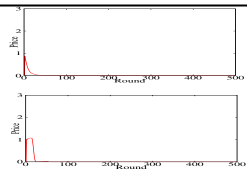
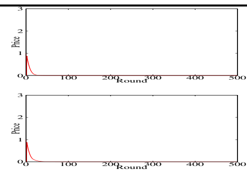
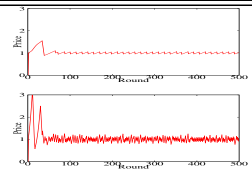
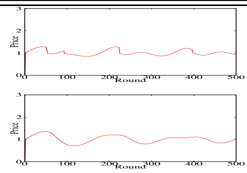
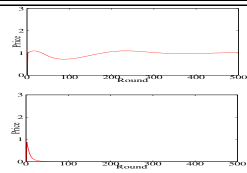
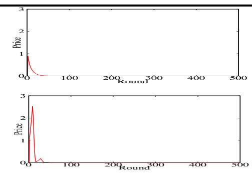
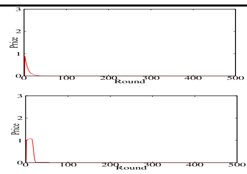
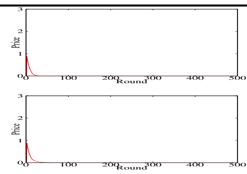
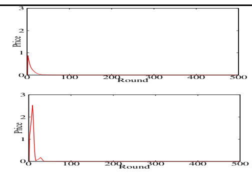
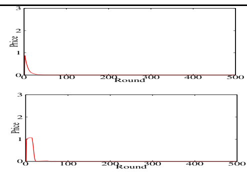
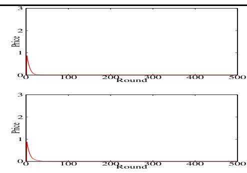
	$Source1Ini = 1500$, $Deliver2Ini = 1499$	$Source1Ini = 1500$, $Deliver2Ini = 1500$	$Source1Ini = 1500$, $Deliver2Ini = 1501$
$Source0Ini = 1499$ $Deliver1Ini = 1499$	 C C	 C B	 C A
$Source0Ini = 1499$ $Deliver1Ini = 1500$	 C C	 C B	 C A
$Source0Ini = 1499$ $Deliver1Ini = 1501$	 B C	 B B	 B A
$Source0Ini = 1500$ $Deliver1Ini = 1499$	 C C	 C B	 C A
$Source0Ini = 1500$ $Deliver1Ini = 1500$	 B C	 B B	 B A
$Source0Ini = 1500$ $Deliver1Ini = 1501$	 A *A*	 A *A*	 A A
$Source0Ini = 1501$ $Deliver1Ini = 1499$	 B C	 B B	 B A
$Source0Ini = 1501$ $Deliver1Ini = 1500$	 A *A*	 A *A*	 A A
$Source0Ini = 1501$ $Deliver1Ini = 1501$	 A *A*	 A *A*	 A A

Table 8: Detail of Table 6 showing the price dynamics of $P01$ on top and $P12$ at bottom (2/3).

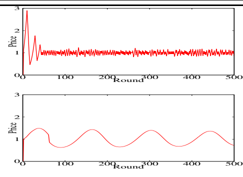
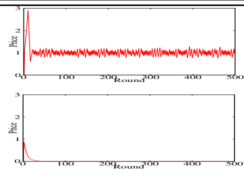
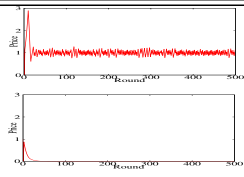
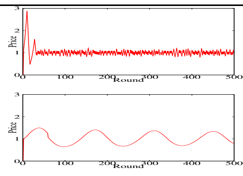
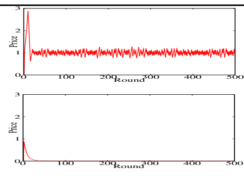
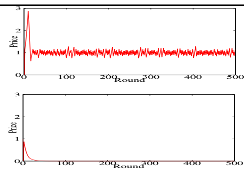
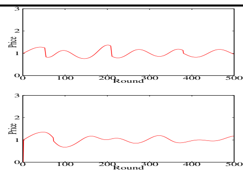
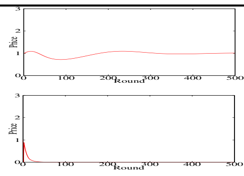
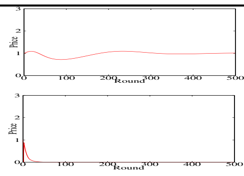
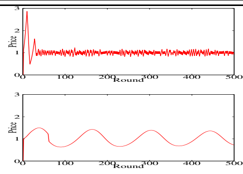
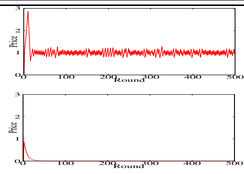
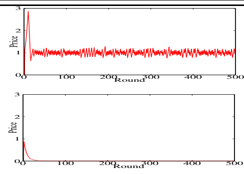
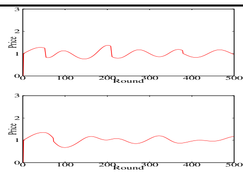
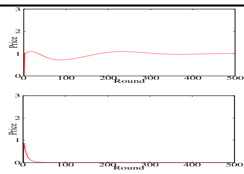
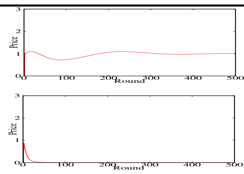
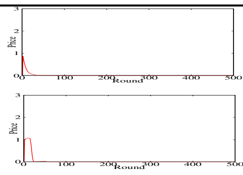
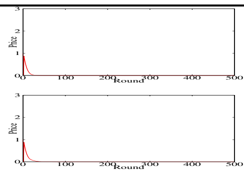
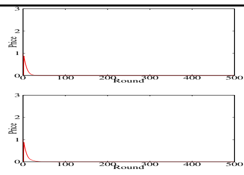
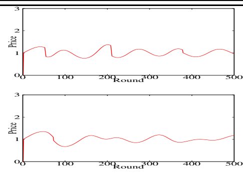
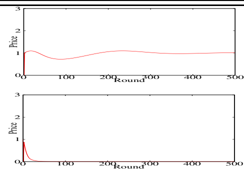
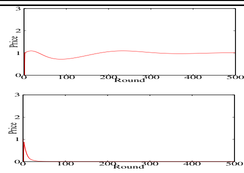
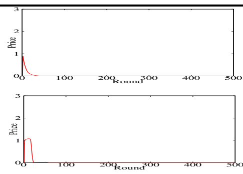
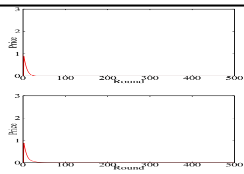
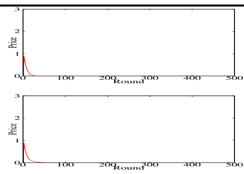
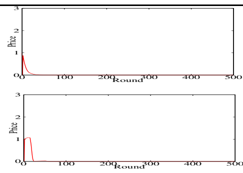
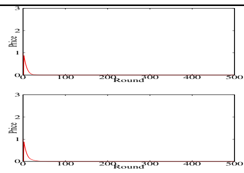
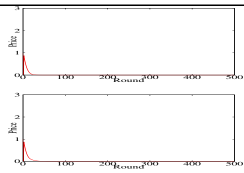
	$Source1Ini = 1501$, $Deliver2Ini = 1499$	$Source1Ini = 1501$, $Deliver2Ini = 1500$	$Source1Ini = 1501$, $Deliver2Ini = 1501$
$Source0Ini = 1499$ $Deliver1Ini = 1499$	 C B	 C A	 C A
$Source0Ini = 1499$ $Deliver1Ini = 1500$	 C B	 C A	 C A
$Source0Ini = 1499$ $Deliver1Ini = 1501$	 B B	 B A	 B A
$Source0Ini = 1500$ $Deliver1Ini = 1499$	 C B	 C A	 C A
$Source0Ini = 1500$ $Deliver1Ini = 1500$	 B B	 B A	 B A
$Source0Ini = 1500$ $Deliver1Ini = 1501$	 A A	 A A	 A A
$Source0Ini = 1501$ $Deliver1Ini = 1499$	 B B	 B A	 B A
$Source0Ini = 1501$ $Deliver1Ini = 1500$	 A A	 A A	 A A
$Source0Ini = 1501$ $Deliver1Ini = 1501$	 A A	 A A	 A A

Table 9: Detail of Table 6 showing the price dynamics of $P01$ on top and $P12$ at bottom (3/3).