Decision Support Systems xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

# A supply chain as a network of auctions ☆

Thierry Moyaux *, Peter McBurney, Michael Wooldridge

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

ARTICLE INFO

*Article history:*
Received 20 July 2007
Received in revised form 15 July 2010
Accepted 28 July 2010
Available online xxxx

*Keywords:*
Supply-chain modelling
SCOR model
Auctions
System dynamics
Agent-based simulation

ABSTRACT

We propose and study a model of supply chains as networks of auctions. Specifically, each company in our model is represented according to the Supply Chain Council's SCOR model, and the company's trading strategy is adapted from a model proposed by Steiglitz and colleagues. Our study of this model, implemented with the JASA auction simulator, shows that price dynamics are more complicated than simply balancing consumption demands, capacities for transformation, and raw material supplies. In addition, we identify three patterns of price dynamics, explain their cause, and propose rules linking initial market conditions with the occurrence of these patterns.



## 1. Introduction

SCM is one of the most widely studied problems in contemporary manufacturing and industrial management [22]. SCM involves the design, modelling, implementation, and coordinated control of networks of resources in order to supply goods and services to consumers. Typical goals are to build SCs 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 SCM 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 [4]. 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 auctions such as eBay[1]).

It is not surprising, thus, that researchers would investigate the use of market mechanisms for SCM. An interesting market-based system in which traders produce and consume goods was Steiglitz et al.'s [24]. In the present 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,[2] we have studied SCs 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 [25], e.g., [26]. An SC then consists of sets of market interactions involving three connected flows up and down the chain: demand, 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 three price signals) in the management of SCs. For this purpose, we have replicated the experiments from their three papers[3] [16,23,24] using JASA,[4] and study how these results scale to networks of auctions in which manufacturers are used to connect markets by buying (e.g., raw food) in some of them, then processing the purchased products in order to sell the obtained products (e.g., cooked food) in other markets. Our aim, therefore, is to understand if tools that are effective for managing

☆ This research was undertaken as part of the EPSRC-funded project *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.

* Corresponding author. Tel.: +33 472 43 79 94.
*E-mail addresses:* thierry.moyaux@insa-lyon.fr (T. Moyaux), mcburney@liverpool.ac.uk (P. McBurney), mjw@liverpool.ac.uk (M. Wooldridge).
*URLs:* http://liesp.insa-lyon.fr/v2/?q=fr/node/2161 (T. Moyaux), http://www.csc.liv.ac.uk/~peter/ (P. McBurney), http://www.csc.liv.ac.uk/~mjw/ (M. Wooldridge).

[1] See http://www.ebay.com/.
[2] See http://www.marketbasedcontrol.com/.
[3] We do not present our replication here of the results of the three papers by Steiglitz et al. Note that when we refer to these papers, we refer in fact to our replication of their models.
[4] Java Auction Simulator API (http://www.sourceforge.net/projects/jasa and http://www.essex.ac.uk/ccfea/staff/profile.aspx?ID=205).

0167-9236/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2010.07.013

the dynamics of a single auction remain effective in the presence of SC dynamics; that is, to understand if these tools can also handle the different streams (demand, products, and money) in SCs, as well as the interactions among these streams. With regard to such adaptation of tools, we have already stabilised the prices in an SC modelled as in this paper by means of speculation [17]. In addition, in [18], we planned to adapt the methodology proposed in [16] to broadcast different price signals with the aim of stabilising SCs. However, adapting such tools to SCs is not the topic of this paper, which focuses on the model and its dynamics. Nor are we considering the mechanism design problem for joint decision-making by entities in a SC, as in [11].

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 when there are two sequential markets. Finally, Section 6 discusses this study.

## 2. Background and related work

According to Dodd and Kumara [6], Fox was probably the first to model an SC as a multiagent system [8]. Many other applications of multiagent systems to SCM followed, both in SC control [14] and SC formation [3]. For example, Anthes [1] reports that Procter & Gamble[5] "*saves USD300 million annually on an investment of less than 1% of that amount*", as a result of agent-based simulation. While multi-agent market models have been widely investigated in the context of SCs, networks of markets have received little attention. Niu et al. [20] studied the competition of parallel markets (i.e., traders make two decisions: which market to bid in, and how much to bid) with a variant of the JASA software. This variant of JASA is used by the TAC-MD which also deals with the competition of parallel markets.[6] Other studies are more focussed on the use of market networks to model SCs, and hence closer to our work. In particular, another track of the TAC dedicated to SCM called TAC-SCM[7] is perhaps the best known 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. This may be the closest model to ours since it also involves SC dynamics, in contrast to other market-mediated SCs [7,12,13,21]. Nevertheless, Grieger [10] confirms in his literature review that very few models of market-mediated SCs exist, hence the novelty of our approach.

Such models of market-mediated SCs make it possible to investigate questions such as the long-term costs and benefits in a business-to-business (B2B) context of auctions in comparison with a long-term relationship. In other words, when is it better to have a market-mediated SC in which competition among traders leads to a short-term efficient solution, or a traditional RFQ/RFP (Request for Quote/Proposal) process in which learning may end up with a better long-term result? [9, p. 1147] In addition, markets are often thought to be efficient when they exhibit perfect competition, but are they able to handle the complexity of SC market dynamics and information transfers arising from their interconnected flows?

We now describe the model of Steiglitz et al. [24], on which we build. In this model, 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 stabilise the clearing price when no price bubbles are created. Subsequently, Steiglitz and Shapiro [23] extended this initial study of the model by analysing the occurrence of these price bubbles and interrupting them during their formation. In both papers, trading agents bid a price calculated as $P(t-1)*B\left(\overline{f},\overline{g}\right)$, where $P(t-1)$ is the previous price in the auction, and $B\left(\overline{f},\overline{g}\right)$ is a function of the internal state of the agent (this strategy will be detailed in this paper). Later, Steiglitz extended this model with Mizuta [16], in an attempt 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. 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 bid the price $P0(t-1)*B\left(\overline{f},\overline{g}\right)$, $P1(t-1)*B\left(\overline{f},\overline{g}\right)$ or $P2(t-1)*B\left(\overline{f},\overline{g}\right)$. When the auctioneer broadcasts $P0$, then $P$ slowly reaches its equilibrium price; when $P1$ is broadcast, then $P$ fluctuates forever; finally, using $P2$ causes rapid convergence to equilibrium.

These three Steiglitz models [16,23,24] assume that, in every round, traders decide to produce either food or gold. Such an assumption replicates an economy in which the traders are two-activity workers without specialization. In contrast, the workers in our model are highly specialized: either they are farmers who grow food, or else they are miners who dig for gold. Such specialization generates an interdependency among the two types of agents: farmers rely on miners to fulfill their need for 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. In this way, miners and farmers form the simplest SC. Technically, the main difference between our model and that of Steiglitz is the fact that even single-market SCs involve two types of agents, viz., end customers and raw material producers, while Steiglitz et al. use only one. The possibility to connect at least two markets by manufacturers further differentiates our model from previous work.

In this paper, we first study such a simple SC 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 type of agent, called *manufacturers*, who 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 competition: two kinds of products are exchanged among these three types of agents, that is, the first type of products in the marketplace between the end customers and the manufacturers, and the second type in the market between the manufacturers and the raw material producers. We think that such an improvement over Steiglitz et al.'s model sheds light on the different interdependencies in SCs. In fact, the different streams traveling across SCs cause both the markets and the different types of traders to be interdependent. For example, price fluctuations in the first market may affect price fluctuations in the second market, as we noted in [17].

## 3. The supply chain model

Our aim in this section is to describe the SC model in sufficient detail for an interested reader to replicate it. We subsequently investigate the properties of this model. The basic idea of the model is simply that the SC itself is represented as a chain of interconnected markets. Thus, for example, one market connects raw material producers to manufacturers, and another market connects end customers to manufacturers. Our belief is that by building SCs in

[5] See http://www.pg.com/.
[6] Trading Agent Competition - Market Design (http://www.marketbasedcontrol.com/cat).
[7] Trading Agent Competition - Supply Chain Management (http://www.sics.se/tac).

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

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 Section 3.1. The ordering strategy used by companies in our model is described in Section 3.2. The use of the JASA auctioneer in the model is described in Section 3.3. Finally, some definitions and parameter settings conclude this section.

### 3.1. The companies modelled with SCOR

There are three types of agents in our model of an SC, namely, the end-users (denoted *EndCustomer*0), the manufacturers (*Manufacturer*1), and the producers of raw materials (*RawMatProd*1 or *RawMatProd*2). These different entities are illustrated in Fig. 1. *Manufacturer*1 in Fig. 1b is modelled directly according to the first level of SCOR, while the other four companies in Fig. 1(a) and (b) are simplifications of this model. The three functions *Deliver*, *Make* and *Source* directly correspond to SCOR—but we ignore the *Plan* and the two *Return* in SCOR. Every *Deliver* or *Source* may be seen as an agent according to Steiglitz, i.e., holding products in inventory and bidding in an auction, which explains why we call them "inventory". Companies, such as *EndCustomer*0, are also agents (their activity is called *Make*) which encapsulate inventory-agents. We use "she" for *Source* inventory-agents, "he" for *Deliver* inventory-agents and "it" for their company-agent. In more detail, the companies in Fig. 1 have the following functions.

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

- *Make*0 produces $Make0Money = +M > 0$ units of money by adding this quantity to *Money*0, and consumes $Make0Products = -P < 0$ 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 by disappearing from the system, nor tries later on to consume more to compensate for a past shortage of food).
- *Source*0 is an inventory-agent who bids in *Market*01 in order to buy products to ensure 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. [24], described in Section 3.2.

*Manufacturer*1 in Fig. 1(b) has three functions:

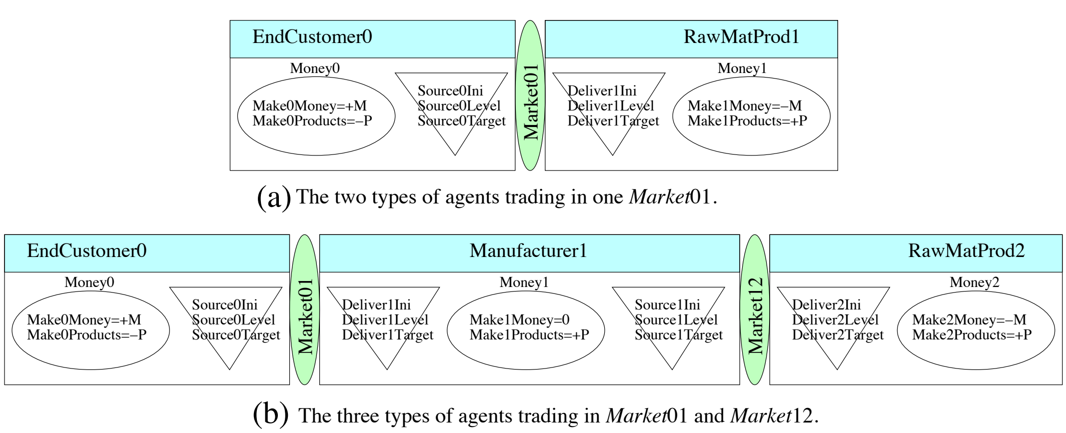
- *Deliver*1 is an inventory who uses Steiglitz et al.'s bidding strategy [24] to place ask shouts in *Market*01. The goal of *Deliver*1 is to sell products so that the level of *Deliver*1 stays at *Deliver*1*Target*. *Money*1 is shared among *Deliver*1 and *Source*1.
- *Make*1 is the production function of *Manufacturer*1 which transforms a quantity of *Make*1*Products* units in every round at a production cost of *Make*1*Money* (considered zero 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 Fig. 1(a) and *RawMatProd*2 in Fig. 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 his 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 Fig. 1(b), this results in the sequence: (i) *Source*0 and *Deliver*1 place a shout in *Market*01 (the order is not important), (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.

Finally, we use the following parameters throughout the paper, described here for Fig. 1(b): (i) the production of food is balanced with its consumption: $Make0Products = -100$ and $Make1Products = Make2Products = 100$, (ii) as well as for money: $Make0Money = 100$, $Make1Money = 0$ and $Make2Money = -100$ (iii) simulations start with $Money\{0,1,2\} = 1000$, (iv) all inventory targets are the same: $Source0Target = Deliver1Target = Source1Target = Deliver2Target = 1500$ (which may be summarised as $InventoryTarget = 1500$), and (v) initial inventory levels will be specified when necessary such that $InventoryIni \in \{500, 1499, 1500, 1501, 2500\}$.




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

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

**Fig. 1.** The two structures of supply chain considered in this paper.

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

*3.2. The bidding strategy from Steiglitz et al.*

As described above, 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 strategy proposed by Steiglitz et al. [24], we will also use their terminology, and explain how we combine this strategy with the SCOR model presented above. With regard to terminology, the model in [24] 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 defined in [24, p. 5] as

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

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]" [24, p. 5], and
- $B\left(\bar{f},\bar{g}\right) = \left[b_{0\infty} - (b_{0\infty} - b_{00})e^{-\gamma\bar{g}}\right]^{(1-\bar{f})}$ with $\gamma = ln\left(\frac{b_{0\infty}-b_{00}}{b_{0\infty}-b_{01}}\right)$. $B$ returns a value below one when the food inventory is above its target level, i.e., $\bar{f} > 1 \Rightarrow B < 1$, which makes the inventory-agent bid at a price lower than $P(t-1)$ in the hope to sell; $g$ amplifies the value returned by $B$ depending on the richness of the agent, e.g., the richer a buyer, the more she is ready to pay for her food. Finally, the scaling parameters of $B$ given in [23, p. 43] are: $b_{00} = B(0,0) = 4.0$, $b_{01} = B(0,1) = 8.0$ and $b_{0\infty} = B(0,\infty) = 16.0$.

An important comment must be made about $B\left(\bar{f},\bar{g}\right)$. This function makes sellers decrease prices and buyers increase prices, which is of course not what we might expect 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, while, (ii) 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 $B\left(\bar{f},\bar{g}\right)$ has to be defined in the counter-intuitive way it is now. If $B\left(\bar{f},\bar{g}\right)$ 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.[8] Besides, the initial Steiglitz model is based on another interpretation of $B\left(\bar{f},\bar{g}\right)$: the more sellers (i.e., producers) have in inventory, the less value these products have because of higher inventory holding costs. Steiglitz and colleagues have not pointed out this question,[9] and we do not aim to address it but only to adapt their model and stabilisation methods in [16,23,24] to SCs. Subsequently, 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 $Valuation\left(t,\bar{f},\bar{g}\right)$ by:

- $Valuation\left(t, \frac{Source0Level}{Source0Target}, \frac{Money0}{P01(t-1)*Source0Target}\right)$
  $= P01(t-1)*B\left(\frac{Source0Level}{Source0Target}, \frac{Money0}{P01(t-1)\ Source0Target}\right)$ for $Source0$,
- $Valuation\left(t, \frac{Deliver1Level}{Deliver1Target}, \frac{Money1}{P01(t-1)*Deliver1Target}\right)$
  $= P01(t-1)*B\left(\frac{Deliver1Level}{Deliver1Target}, \frac{Money1}{P01(t-1)*Deliver1Target}\right)$ for $Deliver1$.
- …
- $Valuation\left(t, \frac{Deliver2Level}{Deliver2Target}, \frac{Money2}{P12(t-1)*Deliver2Target}\right)$
  $= P12(t-1)*B\left(\frac{Deliver2Level}{Deliver2Target}, \frac{Money2}{P12(t-1)*Deliver2Target}\right)$ for $Deliver2$.

Next, the *bidding strategy* must calculate two values: the price and the quantity shouted. The price shouted is simply the true estimated $Valuation\left(t,\bar{f},\bar{g}\right)$, that is, the value of the good actually estimated by the agent without trying to pay less or be paid more. Besides the price, 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 ($Deliver\{1,2\}Target - Deliver\{1,2\}Level$) units.
  Since *Delivers* are not allowed to buy, and *Sources* not to sell, the quantity returned by these two subtractions is always positive.
- However, if an inventory (i.e., a *Source*, since *Delivers* can only sell) wants to buy while she belongs to a company not rich enough (i.e., if $Price_{shouted} * Quantity_{shouted} > Money_{company}$), then she tries to buy the maximum quantity she can afford at the placed price $Valuation\left(t,\bar{f},\bar{g}\right)$, that is, the quantity placed is the largest integer which is less than or equal to $Money\ /\ Valuation\left(t,\bar{f},\bar{g}\right)$.

*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. We shall explain how our auctioneer is different from those used by Steiglitz and his colleagues [16,23,24], and also the difference between the broadcast price $P$ and the clearing price $Pcl$.

*3.3.1. Calculation of the clearing price Pcl*

We now explain the operation of our auctioneer through the three examples in Table 1. Example 1 in Table 1(a) assumes four shouts, namely *ask*1, *ask*2, *bid*1 and *bid*2, which are ordered in this table 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 Table 1(a) that *ask*1 at the lowest sell price £1.1 can be matched with *bid*1 at the highest buy price £2.2. In contrast, in the second line, *ask*2 with the second lowest sell price £2.1 cannot be matched with *bid*2 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" [24,

[8] Notice that the bidding strategy allows the solution of the apparent paradox of suppliers decreasing instead of increasing $P$. When suppliers have more products in inventory (e.g., due to production), their $Valuation\left(t,\bar{f},\bar{g}\right)$ of the product decreases because of inventory holding costs—$Valuation\left(t,\bar{f},\bar{g}\right)$ is thus well defined. It is the role of the bidding strategy not to communicate this depreciation by being more "intelligent" than the truth telling strategy used by Steiglitz et al. and us.

[9] How to design a valuation function is related to the origin of the value of goods, which is a non-trivial question ([5], [15], Chap. VI). 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? We think that Steiglitz et al.'s $B\left(\bar{f},\bar{g}\right)$ implements the first of these three examples.

**Table 1**
Three examples of clearing by our JASA auctioneer.

| 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 $bidQuote = P_{bid2} = 1.2$ | |
| ($ask1$) 1 unit at £1.1 | ($bid1$) 1 units at £2.2 |
| (b) Example 2: $askQuote = P_{ask1} = 1.1$ and $bidQuote = P_{bid1} = 2.2$ | |
| ($ask1$) 1 unit at £1.1 | ($bid1$) 2 units at £2.2 |
| (c) Example 3: $askQuote = bidQuote = P_{bid1} = 2.2$ | |
| ($ask1$) 1 unit at £1.1 | ($bid1a$) 1 unit at £2.2 |
| | ($bid1b$) 1 unit at £2.2 |
| (d) Another representation of Example 3: $askQuote = bidQuote = P_{bid1a} = 2.2$ | |

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* [2]: *askQuote* is the price "buyers need to beat in order for their offers to get matched," and "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 $Pcl = 0.5 * askQuote + 0.5 * bidQuote = £1.65$.

Next, Examples 2 and 3 in Table 1(b) and (c) illustrate a case often encountered in the experiments described 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 more than £2.2 in order to get matched with $ask1$, otherwise $bid1$ will win instead of the new bid shout; thus $bidQuote = P_{bid1} = £2.2$. Similarly, an additional ask shout should propose less 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 as in Example 3. This scenario is described in Table 1(c), which may conveniently be rewritten as Table 1(d) in which $bid1$ is split into two shouts $bid1a$ and $bid1b$. As before, a new bid shout should propose more than £2.2 in order to get matched with $ask1$, otherwise $bid1$ will win instead of the new bid shout; thus $bidQuote = P_{bid1a} = £2.2$. The difference between Examples 2 and 3 is that a new ask shout should not propose less than $P_{ask1} = £1.1$ anymore, but more than $P_{bid1a} = £2.2$, 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 above, the buyer wants 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 a *Source* buyer and a *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. Example 3 illustrates a phenomenon encountered in the results in this paper when there is one buyer and one seller (we shall see this also happens when there are as many buyers as sellers) 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 $bid1$.

*3.3.2. 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 $P * B\left(\bar{f}, \bar{g}\right) = £0$. 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 [24, p. 9]:

$$
\begin{aligned}
P(t) &= Pcl(t) \text{ when } Pcl \neq £0; \\
&= askQuote(t) (\text{i.e., the lowest ask price}) \text{ when no agents buy}; \\
&= bidQuote(t) (\text{i.e., the highest bid price}) \text{ when no agents sell}; \\
&= P(t-1) \text{ when no agents trade.}
\end{aligned}
$$

We always start a simulation with $P(t-1) = P(-1) = 1$ in all markets. Finally, we call $P01$ the broadcast price $P$ and $Pcl01$ the clearing price $Pcl$ in *Market*01, and $P12$ and $Pcl12$ their equivalents in *Market*12.

*3.3.3. Definition of the equilibrium price Peq*

In [24, p. 11], the equilibrium price is defined as 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 $Peq$). In our setting, we modify that definition slightly.

In contrast to [24], the price has no influence on the production of food in our SC model. Specifically, $Peq$ is the ratio of the production of money over the production of products when:

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

In this particular case, $Peq$ is the same in the single market in Fig. 1(a) and in the two markets in Fig. 1(b): $P01eq = P12eq = P/M$. Since $M = P = 100$ in this paper, $P01eq = P02eq = 1$.

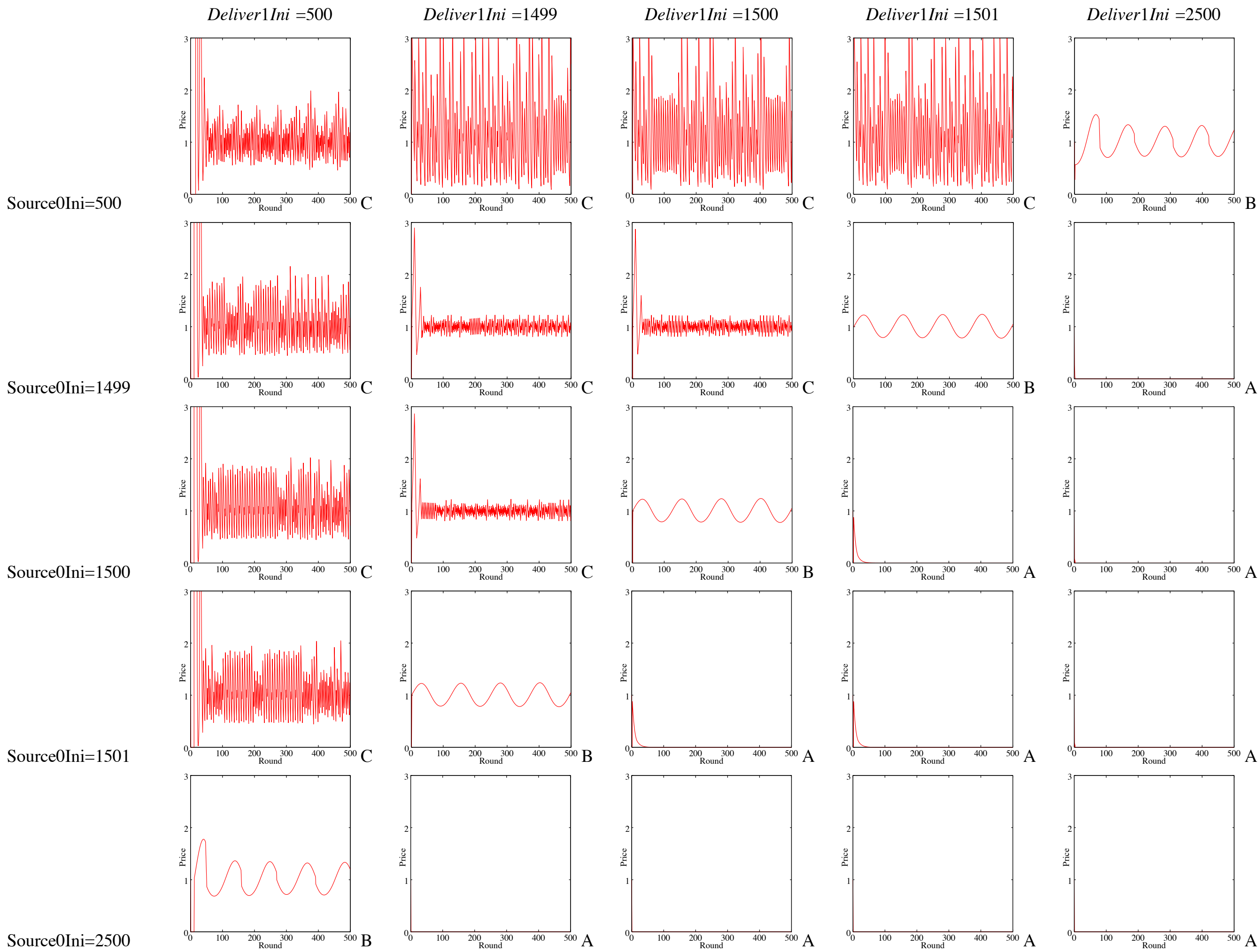



**Fig. 2.** Price dynamics in *Market*01 (with *Source*0*Target* = *Deliver*1*Target* = 1500).

However, this paper also considers scenarios violating the third condition, that is, with several companies per level of the SC (see Sections 4.2 and 5.3). As we will see, in this 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 Fig. 1(a). Let us recall that we set all *InventoryTargets* to 1500 in this paper but allow *InventoryInis* 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 SC possible with only two companies. Fig. 2 shows that initial conditions are very important in our SC 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 Fig. 2 when *Source*0*Ini* = *Source*0*Target* = 1500 and *Deliver*1*Ini* = *Deliver*1*Target* = 1500. With this configuration, *P*01 smoothly fluctuates around *P*01*eq* = 1. We refer to this pattern of smooth fluctuations as pattern “B”, because it forms the border between the two other patterns in Fig. 2. As soon as one of both *InventoryInis* (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 refer to this chaotic pattern as pattern “C”. In stark contrast, as soon as either of both *InventoryInis* increases, we obtain Pattern A, in which *P*01 falls to zero. To explain these three patterns, we should first notice that $\sum InventoryLevel(t) = \sum InventoryIni$ at any time *t* during all the duration of a simulation because (i) the total consumption is balanced with the total production of good, 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* ≠ *InventoryTarget* and bought/sold later on. With this in mind, we can describe the following characteristics of the three patterns:

1. *Pattern C*:
   (a) *How to make Pattern C happen*: Set $(\sum InventoryTarget - \sum InventoryIni) > 0$, e.g., *Source*0*Ini* = 501 with *Deliver*1*Ini* = 2500, and *Source*0*Ini* = 2500 with *Deliver*1*Ini* = 501 both incur Pattern C.
   (b) *Why Pattern C happens*: Pattern C is chaotic in the sense that 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:

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

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

- *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 he places ask shouts for less units than his company *RawMatProd*1 produces during the first rounds of the simulation, so that the lack is transferred 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 needed, and thus bids for less units than *Deliver*1.

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

  The system alternates between these two kinds of periods, depending on whether *Source*0 has enough money to buy all that she consumes (period of 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 with an example drawn from an actual simulation: *P*01 increases from Rounds 0 to 10, next decreases from 10 to 17, and increases from 17 on. Numbers in italics indicate the price chosen by the auctioneer. We can see that 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 Table 1(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), in the "period of decrease of *P*01", you may check in Table 2 that *Source*0 does not bid for all the units she needs because she is too poor to afford that quantity. Finally, we can also see in Table 2 that the initial conditions of the presented simulation outcomes are *Source*0*Ini* = 1500 with *Deliver*1*Ini* = 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) *How to make Pattern B happen*: Set ($\sum InventoryTarget - \sum InventoryIni$) = 0, e.g., *Source*0*Ini* = 501 with *Deliver*1*Ini* = 2499, and *Source*0*Ini* = 2499 with *Deliver*1*Ini* = 501 both incur Pattern B.
   (b) *Why 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 ($\sum InventoryTarget - \sum InventoryIni$) ≈ 0. As can be seen in Fig. 2, Pattern B is made of cycles of slow increases of *P*01, sometimes followed by sudden decreases of *P*01, immediately 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 simulations in which it occurs, it concludes a "period of slow increase of *P*01." This decrease resembles a sine wave. 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-

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

| Round | Start of round | | | | | | | | End of round | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *Source*0 | | | | *Deliver*1 | | | | Auctioneer | |
| | 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 |

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

way between the two proposed prices in the previous period. Such a choice causes *P*01 to cease 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 with an actual simulation run: *P*01 increases from Rounds 0 to 32, then decreases from 32 to 93, 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 $B\left(\bar{f},\bar{g}\right)$ 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*:

(a) *How to make Pattern A happen*: Set ($\sum InventoryTarget - \sum InventoryIni$)<0, e.g., *Source*0*Ini* = 499 with *Deliver*1*Ini* = 2500, and *Source*0*Ini* = 2500 with *Deliver*1*Ini* = 499 both lead to Pattern A.

(b) *Why Pattern A happens*: In all rounds, *Deliver*1 sells one unit more than *Source*0 buys, hence, the auctioneer chooses the price bid by *Deliver*1 as *P*. Since *Deliver*1 tries to reduce the price in the hope to sell, *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 soon 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 replace the truth telling strategy in the Steiglitz model by a more "intelligent" strategy.

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

In conclusion, the sign of ($\sum InventoryTarget - \sum InventoryIni$) makes it possible to determine the pattern of the dynamics of *P*01 when there is only one *Source*0 trading with only one *Deliver*1. We call this comparison as *Rule 2*:

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

- If ($\sum InventoryTarget - \sum InventoryIni$)>0, then *P*01 has a Pattern C;
- If ($\sum InventoryTarget - \sum InventoryIni$) = 0, then *P*01 has a Pattern B;
- If ($\sum InventoryTarget - \sum InventoryIni$)<0, then *P*01 has a Pattern A.

We may notice *P*01 in all the examples in this subsection revolves around *P*01*eq* = £1 (cf. Tables 2–4). The next subsection introduces *Rule 1* to apply before *Rule 2*, and slightly modifies *Rule 2* in order to accommodate with the scenario in which more than one *Source*0 and more than one *Deliver*1 trade in *Market*01. *P*01*eq* will not always be around £1 anymore.

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

| Round | Start of round | | | | | | | | End of round | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *Source*0 | | | | *Deliver*1 | | | | Auctioneer | |
| | 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 |

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

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

| Round | Start of round | | | | | | | | End of Round | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *Source*0 | | | | *Deliver*1 | | | | Auctioneer | |
| | 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 | 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 |
| … | … | … | … | … | … | … | … | … | … | … |

### 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 *InventoryTargets* are set to 1500 in this subsection. Since we noticed in the previous subsection that the sign of ($\sum$ *InventoryTarget* − $\sum$ *InventoryIni*) seems to be more important than the actual value of the different *InventoryTargets* and *InventoryInis* (*Rule 2*), the cases *InventoryIni* = 500 and *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 Fig. 2.

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. Notice that there are as many sellers as buyers with "111 111", but not with "211 111."

We first check that *Rule 2* is not enough to predict what pattern will happen when there are many agents. In fact, ($\sum$ *InventoryTarget*) − ($\sum$ *InventoryIni*) may be rewritten as $(\sum_{i=0}^{\#Source0} Source0_i Target + \sum_{i=0}^{\#Deliver1} Deliver1_j Target) - (\sum_{i=0}^{\#Source0} Source0_i Ini + \sum_{i=0}^{\#Deliver1} Deliver1_j Ini)$, where #*Source*0 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:

- $\sum_{i=0}^{\#Source0} Source0_i Target = Source0Target * \#Source0 = 1500*(1+1+1) = 4500;$
- $\sum_{j=0}^{\#Deliver1} Deliver1_j Target = Deliver1Target * \#Deliver1 = 1500*(1+2+1) = 6000;$
- $\sum_{i=0}^{\#Source0} Source0_i Ini = 1499 + 1500 + 1501 = 4500;$
- $\sum_{j=0}^{\#Deliver1} Deliver1_j Ini = 1499 + 1500 + 1500 + 1501 = 6000.$

$\Rightarrow$ ($\sum$ *InventoryTarget*) − ($\sum$ *InventoryIni*) = (4500 + 6000) − (4500 + 6000) = 0 ⇒ Pattern B.

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

Therefore, *Rule 2* is not enough because the relative numbers of sellers and buyers should also be taken into account. This is why Table 5 presents the number #*Source*0 of buyers and #*Deliver*1 of sellers. With these notations, the results in Table 5 seem to indicate that the three patterns A, B and C of *P*01 have the following characteristics:

1. *Pattern C*:
   (a) *When Pattern C happens*:
   - Either (#*Source*0 − #*Deliver*1) = 0 and ($\sum$ *InventoryTarget* − $\sum$ *InventoryIni*) > 0,
   - Or (#*Source*0 − #*Deliver*1) > 0.

   (b) *How Pattern C happens*: The first condition is very similar to the previous subsection, that is, the case (#*Source*0 = #*Deliver*1) = 1 in the previous subsection resembles the case (#*Source*0 = #*Deliver*1) > 1. Specifically, we can see these initial conditions as setting a system with #*Source*0 = #*Deliver*1 auctions running in parallel, where every auction has one *Source*0 matched with one *Deliver*1 (the matching may be different in every round), and where *Deliver*1s collectively force *Source*0s 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 (#*Source*0 − #*Deliver*1 > 0) is also quite similar to what happens in the previous subsection. More precisely, there are now more *Source*0s than *Deliver*1s which means that more products are consumed than produced. This imbalance leads to the same two kinds of periods:
   - *Periods of decrease of P*01: These periods are as in the previous subsection, that is, *Source*0s are too poor to afford all what they consume because *P*01 is too high. As a consequence, the total quantity ordered by the *Source*0s is lower than the total quantity ordered by the *Deliver*1s, which causes one of the prices proposed by a *Deliver*1 to be chosen as *P*01.
   - *Periods of increase of P*01: Basically, the total quantity consumed by buyers is greater than the total quantity produced by sellers, and, hence, 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 *P*01 is too high. This problem of wealth of the buyers does not apply (or, at least, is less acute) during a period of increase of *P*01. As a consequence, buyers now bid for a quantity higher than what is proposed by sellers.

   (c) *Example of Pattern C*: Tables 6 and 7 illustrate these two types of periods:
   - *Periods of decrease of P*01: Table 6 illustrates this "period of decrease of *P*01" with the first round of a

**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.

| # *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<5 | 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 |

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

**Table 6**
Example of decrease of *P*01 in Pattern C.

| Asks | Bids |
|---|---|
| (*ask*1) 100 units at £4.85256341 | (*bid*4) 91 units at £7.278684 |
| (*ask*2) 100 units at £4.85259734 | (*bid*3) 30 units at £7.094054 |
| (*ask*3) 100 units at £4.85263601 | (*bid*2) 30 units at £6.631892 |
| *(a) Ask and bid shouts.* | (*bid*1) 80 units at £4.852597 |
| | |
| (*ask*1*a*) 91 units at £4.85256341 | (*bid*4) 91 units at £7.278684 |
| (*ask*1*b*) 09 units at £4.85256341 | (*bid*3*a*) 09 units at £7.094054 |
| (*ask*2*a*) 21 units at £4.85259734 | (*bid*3*b*) 21 units at £7.094054 |
| (*ask*2*b*) 30 units at £4.85259734 | (*bid*2) 30 units at £6.631892 |
| (*ask*2*c*) 49 units at £4.85259734 | (*bid*1*a*) 49 units at £4.852597 |
| (*ask*3*a*) 31 units at £4.85263601 | (*bid*1*b*) 31 units at £4.852597 |
| (*ask*3*b*) 69 units at £4.85263601 | |
| *(b) Transformation of ask and bid shouts to see that askQuote* = *bidQuote* = $P_{ask3b}$ = *4.85263601 in panel (a).* | |

configuration "211 111" in which *P*01 decreases (Round 8) when there are four *Source*0s (starting with levels 1499, 1499, 1500 and 1501) and three *Deliver*1 (starting at levels 1499, 1500 and 1501). Table 6(a) presents the quantities and prices bid by the four *Source*0s and asked by the three *Deliver*1s. As in the examples in Table 1, asks are written in ascending order of price, and bids in descending order of price. Table 6(b) presents how the auctioneer splits these shouts. For example, *ask*1 is split into *ask*1*a* and *ask*1*b* so that *ask*1*a* can be matched with *bid*4 and *ask*1*b* with the part *bid*3*a* of *bid*3. With this representation, we can see that any new ask must be below $P_{ask3b}$ to get matched with *bid*1, i.e., to beat *ask*3, thus *bidQuote* = $P_{ask3b}$, and any new bid must be above $P_{ask3b}$ to afford some of the 69 units of *ask*3*b*, thus *askQuote* = $P_{ask3b}$.

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 $Valuation\left(t,\overline{f},\overline{g}\right)$, and, therefore, the only way to influence *P*01 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, $P_{ask3b}$) is used as *P*01, and since sellers always try to decrease the price, then $P01(t)<P01(t-1)$.

- *Periods of increase of P*01: Table 7 illustrates a round during a "period of increase of *P*01." The round considered is the fifteenth of the same simulation as Table 6, which corresponds to the first round of the second period of increase in the simulation of "211 111." More precisely, Table 7(a) presents the shouts placed by the seven traders, and Table 7 how we can split these shouts to make *askQuote* and *bidQuote* obvious. The main thing to notice is that *P*01 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 Table 6. Briefly, *P*01 suddenly "jumps", as in 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 *P*01 does not fluctuate smoothly. Notice that such "jumps" are due to the operation of the auctioneer, thus independent from the Steiglitz bidding function. 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 SC, as when there is 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) of products bid for more (respectively, for less),

**Table 7**
Example of increase of *P*01 in Pattern C.

| Asks | Bids |
|---|---|
| *(a) Ask and bid shouts.* | |
| (*ask*1) 229 units at £0.36408550 | (*bid*4) 229 units at £1.04207499 |
| (*ask*2) 216 units at £0.37159512 | (*bid*3) 272 units at £0.95046649 |
| (*ask*3) 100 units at £0.45293937 | (*bid*2) 280 units at £0.93427686 |
| | (*bid*1) 283 units at £0.91019291 |
| | |
| *(b) Transformation of ask and bid shouts to see that askQuote* = *bidQuote* = $P_{bid2}$ = *0.93427686 in panel (a).* | |
| (*ask*1) 229 units at £0.36408550 | (*bid*4*a*) 263 units at £1.04207499 |
| (*ask*2*a*) 34 units at £0.37159512 | (*bid*4*b*) 34 units at £1.04207499 |
| (*ask*2*b*) 182 units at £0.37159512 | (*bid*3*a*) 182 units at £0.95046649 |
| (*ask*3*a*) 90 units at £0.45293937 | (*bid*3*b*) 90 units at £0.95046649 |
| (*ask*3*b*) 10 units at £0.45293937 | (*bid*2*a*) 90 units at £0.93427686 |
| | (*bid*2*b*) 190 units at £0.93427686 |
| | (*bid*1) 283 units at £0.91019291 |

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

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 have their level at their *InventoryTarget*. Next, in every round after this equilibration period, every *Source*0 is matched with a *Deliver*1 and the same exchange takes place in each pair *Source*0/*Deliver*1 as in the previous subsection.

In summary, Pattern B happens again because buyers are alternatively richer then poorer than sellers. As Pattern A, Pattern B is due to the operation of the auctioneer, rather than to the used bidding function.

Notice that the parameters incurring Pattern B are very intuitive settings and this pattern will thus occur quite *often* in simulation, even though these conditions are *very uncommon* in practice.

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, both cases incurring Pattern A resemble their equivalent when $(\#Source0=\#Deliver1)>1$, in which *P*01 falls to zero because the sellers (instead of the single seller) are favoured by the auctioneer due to the fact they collectively sell more than the buyers.

In conclusion, the sign of $(\#Source0-\#Deliver1)$ allows the determination of the pattern of the dynamics of *P*01 when there are several *Source*0s trading with several *Deliver*1s. The reasons for this are almost the same as in the previous subsection, and are only due to the clearing mechanism rather than to the bidding function. We refer to the following comparison as *Rule 1*:

*Rule 1*: If some *Source*0s buy in *Market*01, and some *Deliver*1s sell in this market, then:

- If $(\#Source0-\#Deliver1)>0$, then *P*01 has a Pattern C;
- If $(\#Source0-\#Deliver1)=0$, then apply *Rule 2*;
- If $(\#Source0-\#Deliver1)<0$, then *P*01 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 $(\sum InventoryTarget-\sum InventoryIni)>0$, then *P*01 has a Pattern C;
- If $(\sum InventoryTarget-\sum InventoryIni)=0$, then *P*01 has a Pattern B;
- If $(\sum InventoryTarget-\sum InventoryIni)<0$, then *P*01 has a Pattern A.

## 5. The two market scenario

We now detail the price dynamics of *P*01 and *P*02 in the two auctions of the SC in Fig. 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 SC. 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

By way of comparison with the previous section, we consider the two auctions *Market*01 and *Market*12 instead of only *Market*01, which leads us to change the name of the raw material supplier from *RawMatProd*1 to *RawMatProd*2, and to add *Manufacturer*1 as an intermediary buying in *Market*12, transforming the bought products in order to sell the finished products in *Market*01.

### 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 the appendix in [19] for details). As a consequence, we can summarise the dynamics of *P*01 and *P*12 with Table 8. In fact, it is even possible to generate Table 8 from (any version of) *Rule 2*.[11] In order to illustrate this, let us consider the case $Source0Ini=Deliver1Ini=Source1Ini=1501$ and $Deliver2Ini=1499$, i.e., the lower right entry in Table 8 which has Pattern A twice. *Market*01 has Pattern A according to *Rule 2* because $Source0Ini+Deliver1Ini=1501+1501$ is greater than $Source0Target+Deliver1Target=1500+1500$. But there seems to be a problem with *Market*12 which should have Pattern B according to *Rule 2* (because $Source1Ini+Deliver2Init=1501+1499$ is equal to $Source1Target+Deliver2Target=1500+1500$), but is replaced by Pattern A in Table 8.

When the application of *Rule 2* does not match the results obtained by simulation, the pattern obtained by simulation is written in italics in Table 8. 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, hence, *P*12 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 8:

*Rule 3*: If a market (*Market*01 in our case) has Pattern A, then a market further from *EndCustomers* (*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.

### 5.3. Price dynamics in the two markets with many agents

Exploring the dynamics of *P*01 and *P*12 when there are several companies at both levels of the SC requires many simulations. Table 9 outlines a few of them by showing how Table 5 may be extended to two markets. Specifically, this table presents a small sample of configurations with 3, 4, 5 or 6 *Source*0s (respectively, *Source*1s) buying from 3, 4, 5 or 6 *Deliver*1s (respectively, *Deliver*2s). For instance, the first line is "3<4, 4<5, 111, 112, 121, 221, A A, A A", which means that:

- 3<4: 3 *Source*0s buy from 4 *Deliver*1s;
- 4<5: 4 *Source*1s buy from 5 *Deliver*2s;
- 111: The 3 *Source*0s start with 1499, 1500 and 1501 units in inventory respectively;
- and so on with 112 for *Deliver*1s, 121 for *Source*1s, and 221 for *Deliver*2s;
- A A, A A: Both *P*01 and *P*12 should have Pattern A according to *Rules 1*, *2* and 3, which is confirmed by simulation. This means that our three rules work in this specific configuration.

[11] *Rule 1* does not apply here because there is not more than one buyer and one seller per market.

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

**Table 8**
Price dynamics of *P*01 and *P*12.

| *Source*0*Ini* | *Deliver*1*Ini* | *Source*1*Ini* = 1499 | | | | | | *Source*1*Ini* = 1500 | | | | | | *Source*1*Ini* = 1501 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *Deliver*2*Ini* | | | | | | *Deliver*2*Ini* | | | | | | *Deliver*2*Ini* | | | | | |
| | | = 1499 | | = 1500 | | = 1501 | | = 1499 | | = 1500 | | = 1501 | | = 1499 | | = 1500 | | = 1501 | |
| | | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 | *P*01 | *P*12 |
| 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* |

The important thing to notice in Table 9 is that our three rules still apply. However, only a small sample of the configurations of the SC are tested, because: (i) we have not studied all the combinations of 1 and 2 for the twelve numbers in every row (there are between $2^9$ and $2^{12}$ of these combinations, since every *Manufacturer*1 has one *Deliver*1 and one *Source*1, thus #*Deliver*1 = #*Source*1), (ii) such combinations of 1 and 2 do not fully specify an SC. For example, the first line of Table 9 describes four *Manufacturer*1s without specifying explicitly their respective *InventoryIni*, e.g., one *Manufacturer*1 has *Deliver*1*Ini* = 1499, but is this the one with *Source*1*Ini* = 1499 or 1501, or one of the two with *Source*1*Ini* = 1500? Such a large space to explore suggests that sight recognition of Patterns A, B and C (and, perhaps, D, E, etc.) should be automated. This is left for future work in order to ensure that *Rules 1*, *2* and 3 are valid for more SC configurations than those in Table 9.

## 6. Discussion

The previous two sections explained the causes of the three observed price patterns, as summarised by *Rules 1* and *2*, and showed that these patterns and these two rules apply to scenarios with either one single or two connected markets. In the case of two connected markets, our model allows the exploration of linkages between these markets. If a market (*Market*01 in our case) has Pattern A, then a market further from *EndCustomers* (called *Market*12) will also have Pattern A. That is, we proposed *Rule 3* to describe how the fall of the price in our *Market*01 prevents money moving up to *Market*12, causing a price fall in *Market*12. More generally, Rule 3 seems to (partially) describe the propagation of (positive or negative) price bubbles. In this regard, we have observed in related work [17, p. 84] that both price stabilisation and price bubbles (i.e., the opposite of price falls) which arise from speculation may also propagate between connected markets. In future work, we may explore whether such propagation is uni-directional, as with the price falls, or not. This investigation of the linkages among markets thus sheds light on the models of Steiglitz [16,23,24].

In this paper, we also shed light on other features of these three Steiglitz models, e.g., on the differences between the two-activity companies modelled by Steiglitz and the supply chain (several companies, each with a single, particular, activity) (see Section 2), and on the reason for which $Valuation\left(t, \bar{f}, \bar{g}\right)$ makes sellers reduce instead of increasing the price (see Section 3.2). Another issue concerns the dependence of production on price by the producers in these models. While the Steiglitz models assume that the type of produced items depends on price, our model assumes no dependence of production on price which, in our case, would cause an increase or decrease of the production of our unique type of products. Although this is an unrealistic assumption, we have retained it in order that our results may be directly comparable with those of Steiglitz. We expect that adopting a more realistic assumption (i.e., allowing production to vary with price levels) would result in a model which avoids the price declines observed in Pattern A, or, if these declines still occur in a more realistic model, allows for the identification of their causes. Consequently, we think that Patterns B and C would remain, as well as Rules 1, 2 and 3 (where Rule 3 would correspond to price bubbles).

## 7. Conclusions

In this paper we have presented a model of market-mediated SC. As outlined in our literature review, our study seems to be one of the first to investigate the dynamics of market networks in which manufacturers buy products in one market, transform these purchased products into output products, then sell the output products in a second market. 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 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 using the JASA auction simulation platform and ran simulations with, variously, one market or two markets in sequence.

The results obtained from these simulations can be summarised as follows. First, only three patterns of price dynamics were obtained. Next, setting the parameters of a market-mediated SC 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 summarised 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 summarised in a third rule.

The price dynamics studied in this paper were observed as we sought a method to calculate the equilibrium price in every market of an SC. Our first method was based on the conventional economic idea that the equilibrium price is the price at which production (supply) equals consumption (demand). Unfortunately, this does not apply to our SC since neither production nor consumption depends on price yet. Our second method was to calculate the equilibrium price as the ratio of the total money available in a market divided by the total quantity of products requested or available in this market; this method was outlined in Section 3.3 for a simple setting, and will be applied in a more complex setting in future work. Before making production and consumption depend on price, the first extension of this paper will thus be an analytical description enabling the evaluation of the equilibrium price. Once this is done, we will

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013

**Table 9**
Dynamics of *P*01 and *P*12 under a small sample of configurations with 3, 4, 5 or 6 buying and selling inventories per level of the supply chain.

| # *Source*0 Vs. # *Deliver*1 | # *Source*1 Vs. # *Deliver*2 | # *Source*0*Ini* = 1499 | # *Source*0*Ini* = 1500 | # *Source*0*Ini* = 1501 | # *Deliver*1*Ini* = 1499 | # *Deliver*1*Ini* = 1500 | # *Deliver*1*Ini* = 1501 | # *Source*1*Ini* = 1499 | # *Source*2*Ini* = 1500 | # *Source*1*Ini* = 1501 | # *Deliver*2*Ini* = 1499 | # *Deliver*2*Ini* = 1500 | # *Deliver*2*Ini* = 1501 | Pattern of *P*01 predicted by our 3 rules | Obtained pattern of *P*01 | Pattern of *P*12 predicted by our 3 rules | Obtained pattern of *P*12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3<4 | 4<5 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | A | A | A | A |
| 3<4 | 4<6 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | A | A | A | A |
| 3<5 | 5<6 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | A | A | A | A |
| 3<5 | 5<6 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | A | A | A | A |
| 3<4 | 4=4 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | A | A | A | A |
| 3<5 | 5=5 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | A | A | A | A |
| 3<6 | 6=6 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | A | A | A | A |
| 4<5 | 5=5 | 2 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | A | A | A | A |
| 4<6 | 6>3 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | A | A | A | A |
| 5<6 | 6>4 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | A | A | A | A |
| 3<4 | 4>3 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | A | A | A | A |
| 3<5 | 5>3 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | A | A | A | A |
| 3=3 | 3<4 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | B | B | A | A |
| 3=3 | 3<5 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 2 | B | B | A | A |
| 3=3 | 3<6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | B | B | A | A |
| 4=4 | 4<5 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 2 | 1 | C | C | A | A |
| 3=3 | 3=3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | B | B | B | B |
| 4=4 | 4=4 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | C | C | A | A |
| 5=5 | 5=5 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | B | B | C | C |
| 6=6 | 6=6 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | B | B | B | B |
| 4=4 | 4>3 | 2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | C | C | C | C |
| 5=5 | 5>3 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | C | C | C | C |
| 5=5 | 5>4 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 1 | C | C | C | C |
| 6=6 | 6>3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | B | C | C | C |
| 5>4 | 4<5 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 2 | 1 | C | C | A | A |
| 5>4 | 4<6 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | C | C | A | A |
| 6>4 | 4<5 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | C | C | A | A |
| 6>4 | 4<6 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | C | C | A | A |
| 4>3 | 3=3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | C | C | B | B |
| 5>3 | 3=3 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | C | C | B | B |
| 5>4 | 4=4 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | C | C | C | C |
| 6>3 | 3=3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | C | C | B | B |
| 6>4 | 4>3 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | C | C | C | C |
| 6>5 | 5>3 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | C | C | C | C |
| 6>5 | 5>4 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | C | C | C | C |
| 6>5 | 5>4 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | C | C | C | C |

consider the case where production and consumption depend on price, and then adapt the aforementioned law of supply and demand to this new model.

Further research will also take account of more agent heterogeneity, for example: (i) companies should not all have the same inventory target since this level is one of the decisions companies have to make; (ii) companies should not all use the same strategy—e.g., one of the common automated trading strategies instead of the truth telling used in this paper; and (iii) the topology of the auction network should be closer to real-world networks rather than the sequential (straight-line) structure considered in this paper.

## References

[1] G. Anthes, Agents of change, Computer World27 January.
[2] S. Phelps, Evolutionary mechanism design, Ph. D Thesis, University of Liverpool, UK (2008).
[3] M. Babaioff, W.E. Walsh, Incentive-compatible, budget-balanced, yet highly efficient auctions for supply chain formation, Decision Support Systems 39 (2005) 123–149.
[4] S.H. Clearwater (Ed.), Market-based Control: A Paradigm for Distributed Resource Allocation, World Scientific, Singapore, 1996.
[5] M. Dobb, Theories of Value and Distribution since Adam Smith, second EditionCambridge University Press, UK, 1981.
[6] C. Dodd, S.R.T. Kumara, A Distributed Multi-agent Model for Value Nets, 14th Int. Conf. Industrial & Engineering Applications of AI & Expert Systems (IEA/AIE), Vol. 2070 of Lecture Notes in AI, Springer, Budapest, Hungary, 2001, pp. 718–727.
[7] M. Fan, J. Stallaert, A.B. Whinston, Decentralized mechanism design for supply chain organizations using an auction market, Information Systems Research 14 (1) (2003) 1–22.
[8] M.S. Fox, J.F. Chionglo, M. Barbuceanu, The Integrated Supply Chain Management, Report of the Enterprise Integration Lab. Industrial Eng. Dept., University of Toronto, Canada, 1993.
[9] A.M. Geoffrion, R. Krishnan, E-business and management science—mutual impacts (part 2 of 2), Management Science 49 (11) (2003) 1445–1456.
[10] M. Grieger, Electronic marketplaces: a literature review and a call for supply chain management research, European Journal of Operational Research 144 (2003) 280–294.
[11] Z. Guo, F. Fang, A.B. Whinston, Supply chain information sharing in a macro prediction market, Decision Support Systems 42 (3) (2006) 1944–1958.
[12] Z. Guo, G.J. Koehler, A.B. Whinston, A market-based optimization algorithm for distributed systems, Management Science 53 (8) (2007) 1345–1358.
[13] H. Lee, S. Whang, The impact of the secondary market on the supply chain, Management Science 48 (6) (2002) 719–731.
[14] F. Lin, H. Kuo, S. Lin, The enhancement of solving the distributed constraint satisfaction problem for cooperative supply chains using multi-agent systems, Decision Support Systems 45 (2008) 795–810.
[15] K. Marx, Wage-Labour and Capital & Value, Price and Profit, Int. Publishers, New York, 1976.
[16] H. Mizuta, K. Steiglitz, E. Lirov, Effects of price signal choices on market stability, Journal of Economic Behavior and Organization 52 (2003) 235–251.
[17] T. Moyaux, P. McBurney, Reduction of the Bullwhip Effect in Supply Chains through Speculation, in: C. Bruun (Ed.), Advances in Artificial Economics, Springer, Berlin, 2006, pp. 77–89.
[18] T. Moyaux, P. McBurney, Modelling a Supply Chain as a Network of Markets, IEEE Int. Conf. Service Systems & Service Management (ICSSSM 2006), France, 2006, pp. 816–821.
[19] T. Moyaux, P. McBurney, M. Wooldridge, A supply chain as a network of auctions, Technical Report ULCS-07-012, Department of Computer Science, University of Liverpool, UK (July 2007).
[20] J. Niu, K. Cai, S. Parsons, E. Sklar, Some Preliminary Results on Competition between Markets for Automated Traders, Workshop on Trading Agent Design & Analysis (TADA-07), Canada, 2007.
[21] C.R. Plott, Experimental Study of Interdependent Systems: Markets and Networks, Technical Report AFRL-IF-RS-TR-2002-252, California Institute of Technology for the Air Force Research Laboratory, 2002.
[22] D. Simchi-Levi, P. Kaminsky, E. Simchi-Levi, Designing and Managing the Supply Chain, McGraw-Hill, 2000.
[23] K. Steiglitz, D. Shapiro, Simulating the madness of crowds: price bubbles in an auction-mediated robot market, Computational Economics 12 (1998) 35–59.
[24] K. Steiglitz, M.L. Honig, L.M. Cohen, A Computational Market Model Based on Individual Action, in: S.H. Clearwater (Ed.), Market-based Control: A Paradigm for Distributed Resource Allocation, World Scientific, Singapore, 1996, pp. 1–27.
[25] J.D. Sterman, Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment, Management Science 35 (3) (1989) 321–339.
[26] F. Strozzi, J. Bosch, J. Zaldvar, Beer game order policy optimization under changing customer demand, Decision Support Systems 42 (2007) 2153–2163.

**Thierry Moyaux** received the Engineering degree in Industrial Engineering from the École Polytechnique Universitaire de Marseille, Marseille, France, and the M.Sc. degree in computer-integrated manufacturing and computer science from the Université d'Aix-Marseille 3 (Paul Cézanne), Marseille, both in 2000, and the Ph.D. degree in computer science from Université Laval, Quebec City, QC, Canada, in 2004. He was a Research Assistant in the Department of Computer Science at the University of Liverpool, Liverpool, U.K. He is now a Maître de Conférences (lecturer) in the Department of Industrial Engineering at the Institut National des Sciences Appliquées of Lyon, France. His research interests include multiagent systems and their application to supply chain management. He has authored several conference and journal papers in these areas.

**Peter McBurney** is a faculty member of the Agent Applications, Research and Technology (Agent ART) Group of the Department of Computer Science at the University of Liverpool in Liverpool, UK. Between January 2004 and March 2006, he was the Administrative Coordinator of the AgentLink network, an EC-funded Coordination Action supporting European research and development in agent-based computing. Since January 2007, he has been Co-Editor-in-Chief of the journal, Knowledge Engineering Review, published by Cambridge University Press. Peter McBurney was awarded the University Medal in Mathematical Statistics by the Australian National University, Canberra, and has a PhD in Computer Science from the University of Liverpool, UK. McBurney's research focuses on the design and use of multi-agent systems for complex decision-making domains, dealing particularly with agent communications languages and the use of argumentation. He has published over 70 papers (44 refereed) and 11 monographs, and co-edited 3 collections.

**Michael Wooldridge** is a Professor of Computer Science in the Department of Computer Science at the University of Liverpool, UK. He is a member of the Agent Applications, Research, and Technology research group (Agent ART), which carries out both pure and applied research in the area of autonomous agents and multiagent systems. He has published over two hundred articles in the theory and practice of agent-based systems, including thirteen books in the area. His main interests are in the use of formal methods of one kind or another for specifying and reasoning about multiagent systems, and computational complexity in multiagent systems. Other interests include agent-oriented software engineering and negotiation. In 2006, he was the recipient of the ACM Autonomous Agents Research Award. He is ranked as the 143rd most cited computer scientist (out of 770,000), according to the CiteSeer normalised ranking of August 2006. In 1997, he founded AgentLink, the EC-funded European Network of Excellence in the area of agent-based computing, and he coordinated AgentLink between 1997 and July 2000. He is co-editor-in-chief of the Journal Autonomous Agents and Multi-Agent Systems, an associate editor of the Journal of Artificial Intelligence Research (JAIR), and serves on the editorial boards of the Journal of Applied Logic, Journal of Logic and Computation, Journal of Applied Artificial Intelligence, and Engineering Applications of Artificial Intelligence.

Please cite this article as: T. Moyaux, et al., A supply chain as a network of auctions, Decis. Support Syst. (2010), doi:10.1016/j.dss.2010.07.013