Logist. Res. (2013) 6:25–41
DOI 10.1007/s12159-012-0094-9

ORIGINAL PAPER

Solving the order promising impasse using multi-criteria decision analysis and negotiation

Christoph Hempsch · Hans-Jürgen Sebastian · Tung Bui

Received: 21 December 2011 / Accepted: 30 October 2012 / Published online: 1 December 2012

Abstract In an inter-firm make-to-order production environment, it is not always possible to satisfy the available-to-promise (ATP) and capable-to-promise (CTP) conditions due to untimely and unmatched production capacity and customer demand. Therefore, it is important to quickly explore alternate solutions that would satisfy both the customers and the suppliers. The purpose of this paper is to present a multi-agent-based system for automated multi-attribute negotiation in order promising. Our proposed solution is based on concepts of evolutionary system design that advocates for continued exploration of new solutions until a satisfactory solution is found. Based on a number of real-life ordering situations—changes of delivery date, price adjustments, and addition/modifications of value-added services as part of the order package, we embed multi-attribute multi-utility simulations into a linear program to search for a negotiated solution when a typical ATP/CTP function of a supply chain management system fails to fulfill a customer order.

Keywords Supply chain management · Multiple-criteria analysis · Multi-agent systems · Heuristics · Group decision and negotiations · Simulation

C. Hempsch
Deutsche Post DHL, Bonn, Germany
e-mail: christoph.hempsch@deutschepost.de

H.-J. Sebastian (✉)
RWTH Aachen, Aachen, Germany
e-mail: sebastian@or.rwth-aachen.de

T. Bui
University of Hawaii, Honolulu, HI, USA
e-mail: tungb@hawaii.edu

1 Introduction

Despite the adoption of seamless integration of supply chain management and order management applications using near real-time data in many core industries, finding an alternate solution to an initial order that cannot be satisfied remains a challenge to manufacturers. For a supply chain planner, an effective sourcing strategy in a volatile market is critical to ensure cost-effective replenishment of products or services between participating firms in their global operations. As an extension to the available-to-promise (ATP) function, capable-to-promise (CTP) takes into consideration capacity information in the supply chain to generate order promising in a short-term, order-based production environment. For incoming customer orders, CTP decides whether or not it is possible to fulfill the desired order quantity and delivery date. With the use of online systems, an order promise depends on a number of complex factors: increasing number of products and services requested by an increasing number of customers, possibility of product customization during the ordering process, expected shorter cycle time and life cycles, and flexible pricing (e.g., Kilger and Meyr [31]. Under scarcity of raw materials and constrained production capacity, orders may be rejected or not fulfilled based on the initial terms of the customers. When an initial order is not fulfilled, the planner needs to quickly find an alternate and mutually agreeable solution between the supplier(s) and the customer(s). Yet, a major weakness of the most CTP functions currently available in supply chain systems is the difficulty in finding an alternate solution. The planner is typically given a set of attributes to specify (or to relax), such as the amount of backward/forward consumption to cover the current shortage or the time fence. This CTP procedure is arbitrary and time-consuming. It requires a skillful and experienced supply chain planner to successfully negotiate an alternate order.

This article combines and extends approaches presented by the co-authors at the AMCIS conference 2009 [11] and the HICSS 43 conference 2010 [12]. The paper shows how a multi-attribute negotiation process can be embedded in a linear program to solve the severe limitation of current computer-supported ATP functions. It discusses how to design a negotiation-augmented supply chain network to support CTP communication, bargaining, and negotiation activities. In the absence of an initial, negotiation heuristics are integrated in optimization routines to generate alternative offers.

The paper is organized as follows. First, we briefly introduce the domain of order promising in the context of a make-to-order production environment and suggest how negotiation can be applied to ATP/CTP calculations. We next extend our analysis to a production–distribution network. Proposed negotiation concepts are operationalized using multi-attribute utility functions.

2 Order promising in operational supply chain management

2.1 The standard case of ATP/CTP functions

As previously introduced, available-to-promise (ATP) and capable-to-promise (CTP) are known activities in the management of inter-organizational supply chains. Within a make-to-order or configure-to-order production environment, production or configuration is not initiated until the producer receives a customer order demanding the specific product. Due to the fact that the quantity of materials or components in stock or the production resources may be limited at a given point in time and cannot be replenished or extended before the desired date of delivery, the producer has to decide on the following:

- quantity,
- due date, and
- price

to commit to each customer order [30].

ATP and CTP functions are widely discussed in the supply chain literature (e.g., Ball et al. [3, Kilger and Schneeweiss 30, Shin and Leem 44]. ATP can be defined as a simple function that looks up the producers' finished products inventory and, if available, reserves the quantity to satisfy an incoming customer's order. CTP in turn takes the whole production process into consideration to look ahead what quantity may be available within a certain time frame. (Some authors denote the functionality of CTP as Advanced ATP [13].)

Figure 1 depicts a basic workflow of the ATP/CTP functions. The process is triggered by a customer order to the producer. A customer order typically contains a set of products (with order positions), the desired quantities, and a delivery due date. Sometimes, a price is specified with the order, but price is usually used as a negotiation item. To check whether the order can be fulfilled, the ATP and CTP functions are executed consecutively. If the ATP function is able to fulfill the ordered quantity from existing inventories in the supply chain, the requisition is created and the products should be delivered on time. If the order cannot be satisfied, the CTP function checks whether the ordered products can be produced within the time delivery constraint. If this is the case, the CTP function creates a production and delivery plan to fulfill the order. Otherwise, the customer is notified that his order cannot be executed, unless some modification to that must be negotiated.

Given the ATP/CTP functions, Fig. 2 shows how these functions operate in a simple pull-based make-to-order production where one single producer receives customer

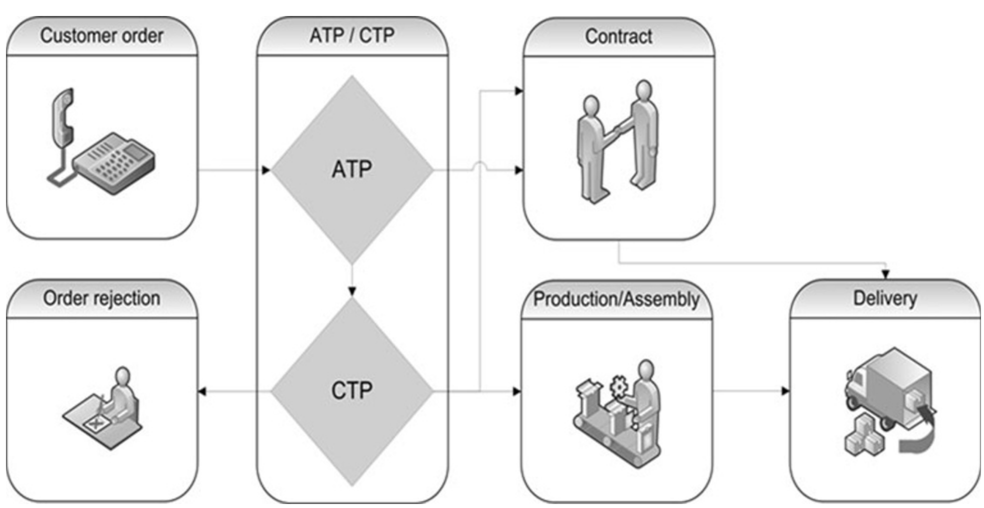


Fig. 1 Basic workflow of available-to-promise (ATP) and a capable-to-promise (CTP) functions

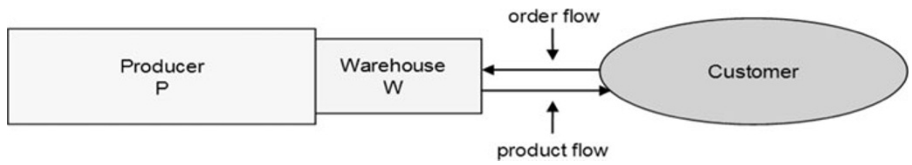


Fig. 2 Basic scenario of an aggregated view of a supply chain from the perspective of ATP/CTP functionality

orders. The producer P has an attached warehouse W for finished products. We will come back to this standard case in Sect. 2.3.

2.2 Order promising in a production–distribution network

In supply chain management, order promising refers to the ability of the SC planner to a global view of supply and demand in order for him to answer to customer orders. The global supply chain network consists of several production sites linked together by a distribution network. The logistics of these networks depend on the number and type of products (functional or innovative products), the type of production (make-to-stock, make-to-order, and configure-to-order), and the location of the push–pull boundary within the network. From the perspective of the supplier(s), the objectives of order promising are to meet customer orders in such way that minimizes costs (e.g., avoiding production delays or overtime, missing or excessive inventory, additional transportation costs due to last-minute shipping needs), increasing revenues by achieving higher capture rates given current production capacity, and ultimately improving customer satisfaction by keeping order promises accurate and by keeping lead time as short as possible.

2.2.1 A dynamic model of a production–distribution network

In this Sect. 2.2.1 we explain the network and define parameters in particular orders, decision variables, and constraints. This is the basis of a collection of submodels which will be introduced in Sect. 2.2.2 by selection of a subset of constraints from Sect. 2.2.1 and by adding two objective functions.

For the sake of clarity, we focus on a production–distribution network for make-to-order products with convergent manufacturing process. In this logistic network, both the supplier stage and the end-customer stage of the entire supply chain are not considered explicitly (see Fig. 3). The end-customer demand is modeled by customer orders which are generated by retailers.

Similar systems have been considered in the literature (e.g., Cohen et al. [14, Vidal and Goetschalckx 46]). Thus, and for the purpose of this paper that focuses more on the negotiation heuristics at the first level of actors, the supplier

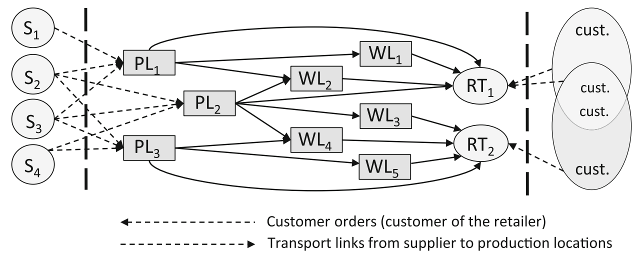


Fig. 3 An example of a production–distribution network

layer (S_1,…, S_4) and the customer layer of the supply chain in Fig. 3 will not be considered in the discussion below. The configuration of the production–distribution subsystem of the supply chain is characterized as follows:

- There is one producer P_{focal} (the focal enterprise of the SC) with n manufacturing facilities at different locations PL_i, $i = 1,\ldots, n$. Without loss of generalizability, we consider one functional product manufactured at all n production locations. Of course, there are several variants of the considered product, such as different configurations of a laptop computer, but we do not deal in this model with a configure-to-order production environment. We assume to have a make-to-stock production environment, and we assume that the order penetration point within the supply chain is given. For example, generic products are manufactured at the production sites, and the places where the variants are built are the warehouses.
  We consider a simple case of a pure push-oriented system (e.g., Olhager and Ostlund [39]. In this make-to-stock environment, the variants are manufactured at the production locations using demand forecasts and distributed to the retailers. However, as seen later, the included negotiation process will result in a combination of make-to-stock and make-to-order.
  The production locations PL_i have given capacities C_i (number of items produced per time unit) and given production costs K_i (unit per item), $i = 1,\ldots, n$. Each manufacturing facility PL_i has an inventory at its location (attached stock) with a stock level $I_t(\text{PL}_i)$, $i = 1,\ldots, n$, at time t which is available for direct delivery to the warehouses or to the retailers.
- There are retailers RT_k, $k = 1,\ldots, l$, that represent "the customers" in the supply chain. Each retailer predicts the demand of its customers and generates "customer orders" to the supply chain.

- The distribution system consists of warehouses WL_j, $j = 1,\ldots, m$, and transportation links.
- The inventory level of the considered product at time t is given by $I_t(\mathrm{WL}_j)$, $j = 1,\ldots, m$, measured in number of items.
- If we denote $P = \{\mathrm{PL}_i | i = 1,\ldots, n\}$, $W = \{\mathrm{WL}_j | j = 1,\ldots, m\}$ and $R = \{\mathrm{RT}_k | k = 1,\ldots,l\}$, we can represent the transportation relations or links as follows:
  - Producer–warehouse links: $(\mathrm{PL}_i, \mathrm{WL}_j) \in A_1 \subseteq P \times W$
  - Links between warehouses: $(\mathrm{WL}_{j1}, \mathrm{WL}_{j2}) \in A_2 \subseteq W \times W$
  - Warehouse–retailer links: $(\mathrm{WL}_j, \mathrm{RT}_k) \in A_3 \subseteq W \times R$
  - Direct transportation links: $(\mathrm{PL}_i, \mathrm{RT}_k) \in A_4 \subseteq P \times R$

Each transportation link has an assigned transportation time, which is assumed to be deterministic and known. The respective sets of links are denoted by A_1, A_2, A_3, and A_4.

An order $O_t^{(k)}$ of retailer RT_k at time t is represented by:

$$O_t^{(k)} = (Q_t^{(k)}, \mathrm{RLT}_t^{(k)}),$$

where $Q_t^{(k)}$ is quantity of the order of RT_k at time t and $\mathrm{RLT}_t^{(k)}$, requested lead time of the order

(We use (k) in order to assign the order to the retailer RT_k and to avoid confusion when considering the notations in Table 1. In Table 1 we introduce three types of order sets by aggregating with respect to the set of retailers. Therefore, k is not needed within the respective notations.)

At time t, the supply chain is faced with the following subsets of orders (Table 1).

Note that in the following we do not distinguish between the different sets of orders in order to keep the model tractable. Adding more variables to the supply chain network would unnecessarily increase the complexity in comparison with the aggregated basic scenario in Fig. 2. However, a subsequent extension of the model is possible.

To trigger the process, assume that the retailers RT_k, which represent the customers, generate a stream of orders at time t to the supply chain. To respond to this stream of orders, the focal enterprise P_{focal} can:

- accept all orders,
- accept only a subset of orders and reject the remaining, or
- start negotiation with the retailers to modify the existing orders (see Table 1).

A reasonable decision-making using a CTP system can be based on a quantitative model that takes into consideration both the production–distribution system introduced above and the order-stream. The detailed consideration of the network gives much more degrees of freedom (many decision variables) for such a network-CTP in comparison with a standard-CTP described above. For example, the decision space of a network-CTP includes the following:

- delivery from warehouses (from which warehouses, how many items)
- direct delivery from the producers inventories (from which producers, how many items)
- where to produce (how many items of the product), and
- replenishment for the warehouses.

This leads to the definition of decision variables in Table 2.

With the decision variables defined in Table 2, the decision model as the basis for the network-CTP can now be formulated as follows:

Constraints:

Warehouse replenishment:

$$\sum_{j=1}^{m} y_{i,j}(t + t_m(x_i(t))) \le x_i(t), \quad \text{where} \quad t_m(x_i(t)) = \frac{x_i(t)}{C_i} \tag{1}$$

for all $i = 1,\ldots, n$ and $t = 0, 1, 2, \ldots$

Delivery assignment of retailers:

$$\sum_{j=1}^{m} w_{j,k}(t) + \sum_{i=1}^{n} p_{i,k}(t) \le Q_t^{(k)} \tag{2}$$

for all $k = 1,\ldots, l$ and $t = 0, 1, 2, \ldots$

Inventory dynamics—warehouses and producers:

$$I_{t+1}(\mathrm{WL}_j) = I_t(\mathrm{WL}_j) + \sum_{i:(\mathrm{PL}_i, \mathrm{WL}_j)\in A_1} y_{ij}(t - d_{ij}) - \sum_{k:(\mathrm{WL}_j, \mathrm{RT}_k)\in A_3} w_{jk}(t) \tag{3}$$

for all $j = 1,\ldots, m$ and $t = 0, 1, 2,\ldots.$

$$I_{t+1}(\mathrm{PL}_i) = I_t(\mathrm{PL}_i) + \sum_{j:(\mathrm{PL}_i, \mathrm{WL}_j)\in A_1} y_{ij}(t) - \sum_{k:(\mathrm{PL}_i, \mathrm{RT}_k)\in A_4} p_{ik}(t) + x_i(t) \tag{4}$$

for all $i = 1,\ldots, n$ and $t = 0, 1, 2, \ldots$

Table 1 Possible order sets

$O_t = \{O_t^{(k)} \mid k = 1, \ldots, l\}$	"New set" of orders at time t
$O^{\mathrm{ac}}_{t' < t}$	Accepted orders at time $t' < t$, $\bigcup_{t'<t} O^{\mathrm{ac}}_{t'<t}$: accepted orders before t
$O^{\mathrm{re}}_{t' < t}$	Rejected orders at time $t' < t$, $\bigcup_{t'<t} O^{\mathrm{re}}_{t'<t}$: rejected orders before t
$O^{\mathrm{neg}}_{t' < t}$	Orders arrived at time t' and selected for negotiation, $\bigcup_{t'<t} O^{\mathrm{neg}}_{t'<t}$: orders selected for negotiation before t

Table 2 Decision variables

Decision	Variable	Description
Delivery from warehouse stock	$w_{j,k}(t)$	Quantity sent from warehouse WL_j to RT_k starting at time t (If transportation time is $d^*_{j,k}$, then the quantity arrives at the retailers place at time point $t + d^*_{j,k}$)
Direct delivery from the manufacturer's stock	$p_{i,k}(t)$	Quantity sent from producer PL_i to RT_k directly starting at time t (If transportation time is $d'_{i,k}$, then it arrives at time $t + d'_{i,k}$ at the retailers location)
Production orders for the manufacturers	$x_i(t)$	Quantity which producer PL_i starts to manufacture at time t (With C_i number of items produced by PL_i per time unit, the manufacturing time for any single order can be computed.)
Distribution (replenishment) decisions	$y_{i,j}(t)$	Quantity sent from producer PL_i to warehouse WL_j (in order to store the amount of goods there) at time t

Distribution decision constraints:

$$\sum_{k:(\mathrm{WL}_j,\mathrm{RT}_k)\in A_3} w_{jk}(t) \le I_t(\mathrm{WL}_j) + \sum_{i:(\mathrm{PL}_i,\mathrm{WL}_j)\in A_1} y_{ij}(t - d_{ij}) \tag{5}$$

for all $j = 1,\ldots,m$

$$\sum_{j:(PL_i,WL_j)\in A_1} y_{ij}(t) + \sum_{k:(PL_i,RT_k)\in A_4} p_{ik}(t) \le I_t(PL_i) + x_i(t) \tag{6}$$

for all $i = 1, 2, \ldots, m$ and, both, (5) and (6) for all $t = 0, 1, 2, \ldots$

Now, we discuss the constraints (1) to (6). PL_i produces a quantity $x_i(t)$ starting at time t. Then, the manufacturing time t_m of this quantity $x_i(t)$ (denoted by $t_m(x_i(t))$) becomes:

$$t_m(x_i(t)) = \frac{x_i(t)}{C_i} \tag{7}$$

We further assume that transportation of the produced good to the warehouses does not start before the whole production order quantity $x_i(t)$ has been produced and that there is no inventory available at the production site. Then, we get the constraint (1). Equation (1) shows that constraints are becoming more difficult to formulate if we assume a continuous time parameter t and if we explicitly consider time lags caused by production capacity constraints and the missing inventories from the producers. Therefore, we assume discrete time t, $t = 0, 1, 2, \ldots$, and inventories are integrated to the producers PL_i. We also assume $x_i(t) \le C_i$. Then, we replace constraint (1) with the constraints (4) and (6).

We also define the constraints related to an order quantity $Q_t^{(k)}$.

(a) Delivery from warehouse stock or direct delivery from manufacturers stock to a retailer RT_k should not be bigger than the ordered quantity $Q_t^{(k)}$ of this retailer. Formula (2) assumes that the respective links in the network exist. If not, the respective decision variable is set to 0. The delivery starts at time t provided that the inventory levels at time t allow the following:

$$\begin{aligned} &w_{j,k}(t) \le I_t(\mathrm{WL}_j) \quad \text{for all } j \text{ and } k \\ &p_{i,k}(t) \le I_t(\mathrm{PL}_i) \quad \text{for all } i \text{ and } k \end{aligned} \tag{8}$$

(b) To support multistage decision-making, our model is dynamic in nature with discrete time periods and state variables.

Equation (3) shows how the inventory levels of the warehouses are changing over time. Formula (4) models the state transformation for the producer inventories.

(c) If we use stock variables $I_t(\mathrm{PL}_i)$ and $I_t(\mathrm{WL}_j)$ to model producer and warehouse inventories, we have to make sure that these variables remain nonnegative over time. Therefore, from (3) and (4) we derive (5) and (6) with $I_{t+1}(\mathrm{WL}_j) \ge 0$ and $I_{t+1}(\mathrm{PL}_i) \ge 0$. With initial values $I_0(\mathrm{WL}_j) \ge 0$, $I_0(\mathrm{PL}_i) \ge 0$, and inequalities (5) and (6) fulfilled, nonnegative inventory levels are guaranteed.

Also, we have to take into account the capacity constraints:

$$0 \le x_i(t) \le C_i$$

where C_i are maximal production capacities per time interval.

2.2.2 *A simple strategic procedure for a CTP network*

In 2.2.1, we developed a dynamic model for the production–distribution network as a subsystem of a supply chain. This model might become very large in terms of the number of time-dependent variables and constraints. This model can be used in order to simulate the processes (i.e., production, storage, transportation) of the supply chain. For a given set of orders $O_t = \{O_t^{(1)},\ldots, O_t^{(l)}\}$ at time t, we can use the data $O_t^{(k)} = (Q_t^{(k)}, \mathrm{RLT}_t^{(k)})$, respectively. The ordered quantity and requested lead time define possible situations. Then we define a strategy which means mainly a sequence in which situations are checked. Perhaps, the most intuitive strategy related to the network is the four-step procedure below:

I. If possible, fulfill the order-set from warehouse inventories only.
II. If I is not possible, try to fulfill the order-set by using both the warehouse inventories and the goods which are in transport from the producers to the warehouse (regular replenishment processes of the warehouses)
III. If II is not possible, additionally select direct delivery from producers to the retailers.
IV. If III is not possible, additionally create new production at time t.

Each situation (I–IV) corresponds with a submodel that can be built from the set of decision variables and constraints we have introduced before. It is obvious that model I. is the simplest one (the network-ATP) and the other models become subsequently more complex in the sequence I, II, III, and IV.

To illustrate the proposed approach, we will first consider situation I (see Fig. 4).

$$\text{Situation } I \text{ is given, if} \quad \sum_{j=1}^{m} I_t(\mathrm{WL}_j) \geq \sum_{k=1}^{l} Q_t^{(k)} \tag{9}$$

In this case, it is possible to fulfill the overall ordered quantity by the aggregated warehouse stock $\sum_{j=1}^{m} I_t(\mathrm{WL}_j)$ at time t. However, inequality (9) does not check whether the delivery from warehouse inventories is possible or partly possible within the requested lead times. Also, there is no decision which quantities should be delivered from which warehouse to which retailer. In order to come up with optimal decisions from the supply chain from the point of view of focal enterprise (the producer P_{focal}), we define two objective functions by considering the cost K_t of delivery from the warehouses, and the lead time LT_t for this delivery process.

$$K_t = \sum_{j=1}^{m} \sum_{k=1}^{l} K_{jk} \cdot w_{jk}(t) \tag{10}$$

where K_{jk} denotes transportation cost per item with regard to the relation $\mathrm{WL}_j \rightarrow \mathrm{RT}_k$. (If the link $\mathrm{WL}_j \rightarrow \mathrm{RT}_k$ is not included in the network, we have $w_{jk}(t) = 0$ by definition.)

We define the lead time for order $O_t^{(k)} = (Q_t^{(k)}, \mathrm{RLT}_t^{(k)})$ of retailer RT_k by:

$$\mathrm{LT}_t^{(k)} = \max_{j=1,\ldots,m} \left\{ d_{jk}^{*} | w_{jk}(t) > 0 \right\} \text{and} \sum_{j=1}^{m} w_{jk}(t) = Q_t^{(k)}, \tag{11}$$

where d_{jk}^{*} denotes the transportation time from warehouse WL_j to retailer RT_k.

(Lead time refers to the maximal transportation time from the warehouses to RT_k, provided the entire ordered quantity is delivered.)

In sum, we have $l + 1$ objectives, if we do not aggregate lead times of the retailers. If we define an overall lead time by:

$$\mathrm{LT}_t = \max_k \mathrm{LT}_t^{(k)} \tag{12}$$

we get two objectives where both—costs and lead time—need to be minimized.

The task at hand for producer P_{focal}, as the focal enterprise in the supply chain, is to set the problem up as a series of multi-criteria problems and use the solutions of these problems initial offers to engage in a negotiation process between the producer $P = P_{\text{focal}}$ (the Supply Chain Agent) and the retailers $\mathrm{RT}_1, \ldots, \mathrm{RT}_l$ (the Retailer Agents). In order for us to embed negotiation in the model, we provide in the next section a brief discussion and justification of using negotiation processes in order promising.

3 Negotiation in order promising

3.1 Order promising as a compromise between producers and consumers

The main objective of the producer in order promising, of course, is to maximize revenue and earnings by manufacturing and selling as many products to as many customers as possible. As we are considering a pull-based production environment, customer satisfaction is of high importance in the long run. In general, there are three critical factors that determine the quality of an order promising system from a customer satisfaction point of view:

- Reaction time of the system: The duration of the decision-making process should be as short as possible to get the orders to the customers.

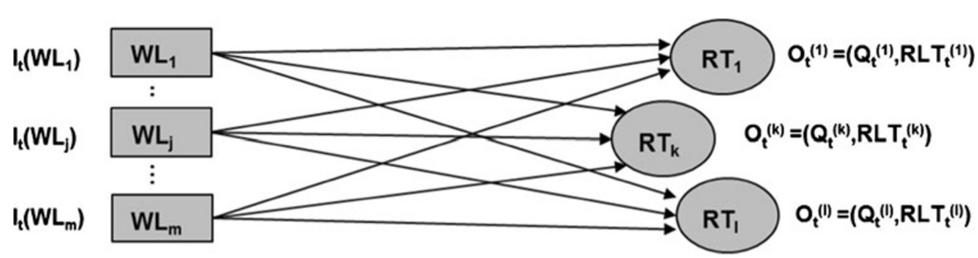


Fig. 4 The network-ATP (situation I)

- Quality of promised due date: The customer desires a short delivery time and a reliable prediction on it.
- Order acceptance rate: Only a small number of customer orders should be rejected unless the selection of accepted orders is solely based on short-term profit maximization considerations. A customer with a rejected order may choose to buy the product from another producer—and may not come back in the future.

There are multiple decisions apart from the ones mentioned above that are commonly incorporated into ATP and CTP to achieve these objectives. For example, to be able to accept more customer orders, order splitting or quantity splitting may be considered. Order splitting allows the delivery of order positions of a customer order at different dates. Quantity splitting allows splitting the ordered quantity into multiple orders and delivering these orders at different delivery dates. The workflow for the CTP function described above is rather common and is widely discussed in the literature. The issue in this paper, one that is not yet studied in depth in the current literature, is how to proceed once a customer order is rejected by the CTP function. This step requires some form of negotiation with the customers until an alternate solution is satisfactory for all parties.

3.2 Automated negotiation

Research in negotiation support tends to focus on two major areas: communication support and bargaining and group decision support. Experience has shown that the more antagonists engage in exchanging information and expression of their positions in a clear and concise manner, the more likely that they will move toward a solution that is acceptable to all. The underlying principle is that to steer or redirect communication leads to conflict to one that encourages conflict resolution, and better yet, collaboration [29]. A number of researchers propose the creation of computer-based platforms to support communications through structured language such as argumentation language (e.g., Bui et al. [7, Karacapilidis and Papdias 28]) or language to help structure negotiation issues (e.g., issue-based dialogue management) [15, 32]. Another area of research is to search for techniques to improve the negotiation outcomes (e.g., Bui [6], Yan et al. [48]). These techniques range from optimization to heuristics, from game theory to simulation (e.g., Bichler et al. [4] for a review). Among negotiation systems, there are negotiation support systems (NSS) and negotiation mediation systems (NMS). NSS are designed to assist negotiators in reaching mutually satisfactory decisions by providing a means of communication and through analysis of available information with a variety of appropriate decision methods. NMS implement negotiation processes between multiple entities. Their aim is to improve the efficiency of the negotiation processes through communications support and assistance toward integrative bargaining.

Most NSS seek to improve the outcome of the party that uses the system. In contrast, and as their name suggests, NMS are used to help the negotiation party to gain a more effective result. In this paper, we define negotiation in its broadest context, that is, any activity that helps avoid a solution impasse, or better yet, one that would yield a win–win situation to both customers and suppliers when the initial order by the customers cannot be met by suppliers. Acknowledging the existence of more than one issue in a typical negotiation, the general literature in multiple-attribute utility theory advocates for the continued exploration of solution until a compromise is found (e.g., Bui [6]). The exploration of “solution space” can be achieved by looking for new solutions that had not been thought of, adjusting or refocusing on views of the problem, or adding/replacing actors. This concept is known as evolutionary in the design of negotiation processes [10]. As shown in the next section, the consideration to split the quantity of an order that cannot be fulfilled or the adding of some additional services to a late delivery are examples of evolving the initial solutions to a new feasible set of possible solutions that are acceptable to all involved parties.

The notion of automated negotiation implies that some aspects of a negotiation are either conducted or at least supported by autonomous computer agents or parties [8]. In the context of ATP or CTP, this automated negotiation could be of routine procedures (e.g., fast and expanded search of “matching solutions,” quick estimation of delivery time, or instantaneous reporting of inventory levels). Furthermore, the agents could also be pre-programmed to act as a trained mediator looking for heuristic-based problem solving (e.g., Emerson and Piramuthu [17]). For example, the first procedural rule of an automated agent would be to immediately acknowledge the reception of a customer’s order and the generation of an alternate solution should the initial order cannot be satisfied. In a distributed platform linking customers to suppliers, the automation of negotiation processes could be implemented by a series of simple to more functional agents, thus a multi-agent system.

3.3 Potential benefits of introducing negotiation capabilities to CTP

As mentioned above, the success of an order promising system depends on three critical factors, that is, short reaction time, quality of promised due date, and a high acceptance rate. Unfortunately, the objectives of producers

Table 3 Multiple issues in CTP and conflicting objectives

Order Attributes	Producer	Customer
Due date	Late	Early
Quantity	High	Ordered amount
Price	Low	Low
Value-added services (VAS)	Low	High

and customers regarding the order attributes are, at least in some cases, divergent (see Table 3). This discrepancy needs to be taken into consideration by the focal producer whenever a customer order is rejected by the order promising system and a counteroffer is needed. As discussed earlier, any order that is not fulfilled is an opportunity loss. Table 4 shows suggested strategies on four typical negotiable order attributes for the producer. Obviously, negotiation concepts especially a negotiation support system for the computation of counteroffers may enhance the overall efficiency of the order promising system.

The potential need for, and benefit of, introducing negotiation support to the domain of order promising is discussed in recent literature (e.g., Rupp and Ristic [41]). Yet, most authors consider negotiation just for contracting before the ATP or CTP functions are executed, that is, producer and customer settle on fixed values or intervals for quantities and due dates (e.g., Sadeh et al. [42, Shin and Leem 43]). Other authors claim in their research that negotiation processes have been implemented, but they do not explain or even formalize these processes in details (e.g., Makatsoris et al. [34]). To our knowledge and at the time of this writing, very few specific negotiation processes or systems have been proposed to support post-ATP/CTP negotiation. Dudek and Stadtler [16] discuss a system for negotiation-based collaborative planning between supply chain partners, which consists of a supplier and a buyer. The supplier offers an initial quantity that can be revised by the buyer. Yet, their work assumes a collaborative partnership and not a situation in which producers and customers are confronted with divergent objectives.

Thus, there is potential for further research on enhancing the efficiency and effectiveness of order promising systems by introducing negotiation concepts and systems. Table 5 is a brief review of relevant literature supporting our research work. It classifies recent research literature on ATP/CTP, negotiation or Multi-Agent Systems (MAS), and, most relevant to this research, work that attempts to integrate these four topics.

3.4 Implementation of a multi-agent simulation framework for automated negotiation in order promising

Multi-agent systems (MAS) are information systems that have been of great interest in research over the last years. They consist of several intelligent agents that can exchange information or objects with each other. By doing so, agents can be designed to address complex problems which would be very difficult or impossible to solve with a single intelligent agent. In a distributed environment, MAS and their agents are naturally well suited to replicate real-world organizations or units. Agent-based technology can today be found in a wide range of applications like disaster response and modeling social systems (e.g., Jennings et al. [26]). The intelligent agents of a MAS share some important characteristics: They are mostly autonomous. They only have a limited, local view of the global environment. And there is no single agent that is able to control all the others. The agents are defined by their objectives, attributes, and behavior [27].

We have developed a multi-agent system (MAS) prototype that consists of different agents representing the retailers business and its customers as shown in Fig. 2. The implemented MAS focuses on the decision column of Table 4, but it can be extended to support activities during the pre- and post-decision phases.

Table 4 Suggested negotiation strategies to deal with CTP issues

Decision attributes	Pre-decision	Decision	Post-decision
Due Date	Forecast arriving orders and build stock and production capacity accordingly	Produce in advance or negotiate later date	Evaluate forecast accuracy, and if necessary adjust forecast techniques
Quantity	Forecast incoming orders and build stock and production capacity accordingly	Reduce quantity or split it	Evaluate forecast accuracy, and if necessary adjust forecast techniques
Price	Conduct market research on competitive pricing	Reduce price to compensate for later due date and/or smaller quantity	Check whether pricing was right
Value-added services (VAS)	Build up competence in customer services and preferences research	Offer customers value-added services to compensate for late delivery and/or delivery with smaller quantity	Assess customer satisfaction

Table 5 Selected literature on ATP/CTP, MAS and negotiation

	SCM	Order promising/ATP/CTP	Negotiation support	MAS
SCM/ATP/CTP				
Agatz et al. [1]	x	x		
Azevedo and Sousa [2]	x	x		
Bixby et al. [5]	x	x		
Chen et al. [13]	x	x		
Fischer [18]	x	x		
Meyr [35]	x	x		
Moses et al. [37]	x	x		
Rupp and Ristic [41]	x	x		
Kilger and Schneeweiss [30]	x	x		
Kilger and Meyr [31]	x	x		
Vidal and Goetschalckx [46]	x			
Wu and Liu [47]	x	x		
Zhao et al. [50]	x	x		
Argumentation and negotiation				
Bichler et al. [4]			x	x
Bui et al. [8]			x	x
Bui and Shakun [10]			x	
Karacapilidis and Papdias [28]			x	
Larsson [32]			x	
Louta et al. [33]			x	x
MAS				
Bui and Lee [9]				x
Julka et al. [27]	x	x		
SCM and MAS				
Fulkerson [20]	x	x		x
Sadeh et al. [42]	x	x		x
SCM and negotiation				
Gallien et al. [22]	x	x	x	
Grieger [24]	x		x	
Shin and Leem [43]	x	x	x	
Moodie and Bobrowski [36]	x	x	x	
Zhang et al. [49]	x	x	x	
SCM, MAS and negotiation				
Bui et al. [11]	x	x	x	x
Dudek and Stadtler [44]	x		x	x
Frey et al. [19]	x		x	x
Fung and Chen [21]	x	x	x	x
Makatsoris and Chang [34]	x	x	x	x
Richards et al. [40]	x	x	x	x
Stadtler [44]	x	x	x	x
Tan et al. [45]	x	x	x	x

3.5 System architecture

To illustrate our negotiation framework, we use the case of a computer retailer (producer). The customers place orders that typically consist of a specified number of computer systems. Since the retailer cannot accurately predict these orders with specific configurations and reliability of ordered products, it has little choice but adopting a make-to-order environment.

The *Order Collection* agent receives customer orders and passes them on to the *Supply Chain Central Control* unit. This agent communicates with *Production and Inventory Control* to get the necessary information to call the *CTP solver*. A linear program is used to decide whether

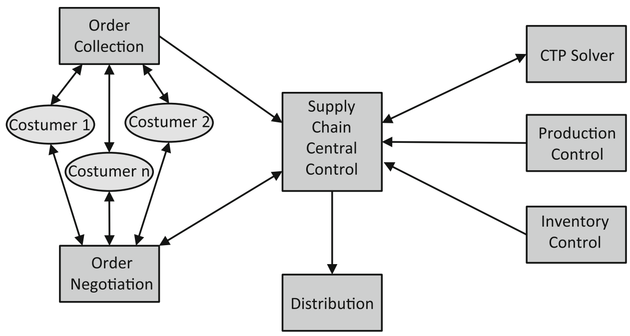


Fig. 5 System architecture of a negotiation-assisted make-to-order environment

or not the orders are accepted or rejected. These decisions are in turn returned to the *Central Control* agent. The latter attempts to derive alternatives for the rejected orders. These counteroffers are then passed on to the *Order Negotiation* agent that uses an algorithm described later to modify the counteroffers using new price and value-added services as terms of negotiation with the hope that they will be considered and accepted by the customer. The *Order Negotiation* then offers the counteroffers to the customers who are asked to take positions. In Fig. 5 a system architecture is given which is able to realize the negotiation-assisted make-to-order approach described above. Finally, the distribution agent takes care on realizing the decisions within the physical distribution network. A UML sequence diagram of this CTP and negotiation process shows the life span of and communication between the agent processes in [11].

3.6 Software framework and tools

The MAS was implemented using the Repast Simphony framework (North et al.'s Web site [38]). This Java-based environment provides a graphical user interface for running simulations within a MAS. The different agents are implemented using plain Java classes. The GNU Linear Programming Kit (GLPK Website) [23] and its Java interface (GLPKJNI website) [25] are used to solve the CTP model.

The system details as well as the results of computational experiments are published in [11]. As a proof of concept, the simulation of the automated negotiation showed that the number of rejected orders could be reduced while the overall revenue increased.

The successful simulation experiments encouraged us to continue the integration of multi-criteria approaches and negotiation techniques to the production–distribution network (see Fig. 3) and the model-based approach outlined in Sect. 2.2.2. The novelty here is to use interactive instead of automated negotiation and to develop a hybrid approach.

4 Outline of a negotiation approach within a CTP environment

Bui and Shakun [10] published a negotiation approach and software tool (NEGOTIATOR) to support distributed negotiation using multiple-criteria Pareto optimization. As discussed earlier, the underlying principle is the Evolutionary System Design approach that advocates for a systematic search of new solutions until a satisfactory is found. Although this evolutionary approach was originally applied to fields such as corporate strategies or public policies, the technique lends itself well to a CTP environment in supply chain management. Therefore, in order to show how a negotiation approach could be applied to a CTP environment, we explain the approach by Bui and Shakun [10] using an example. First, we apply the general concepts of the method to this application area.

4.1 A combined optimization: negotiation approach illustrated by an example

We consider the production–distribution network example from 2.2 with only one retailer.

4.1.1 Definition of values, goal variables, control variables, and weights

- General values are high performance, reliability in delivery, and safety.
- Operational expressions of the general values are formulated by goal variables, delivered quantity (of the requested good), lead time, cost, and price
- Notations of the control variables of the example-Supply Chain:

  Q_t—ordered quantity by the retailer (party A)
  OQ_t—offered quantity by the SC in reply to the order (SC—agent (this means the focal enterprise P_{focal}) is party B of negotiation)
  K_t—cost of the ordered quantity Q_t
  P_t—price of the ordered quantity Q_t
  LT_t—lead time of the ordered quantity Q_t
  t—is the time index $t \in \{0, 1, 2, \ldots\}$

The SC-agent computes cost and lead time using an optimization submodel (introduced in 2.2.2) and calculates a price P_t.

- Control variables of party A (retailer) are as follows: ordered quantity (number of items), price (e.g., EURO, Dollars), and lead time (e.g., days, weeks)
- Control variables of party B (Retailer) are as follows: offered quantity (number of items), price, and lead time

4.1.2 *The first round of negotiation*

To illustrate the first round of negotiation triggered by initial offers, and for the sake of clarity, we consider the following special numerical example:

Party A places an order of 500 units. The buyer (party A) does not propose a price, but the SC-agent (party B) has to. The SC-agent seeks to satisfy this order using price and lead time as decision variables. Each agent starts the negotiation with an initial offer. The initial offer of party B (Supply Chain) is derived from the solution of an optimization problem (see 2.2.2) which is considered next. Because the SC has to meet the ordered quantity of 500 items (or to reject the order), it uses submodel I and selects a solution which is closest to the requested lead time 4. Figure 6 shows a simple numerical example.

Initial offer from party A (Retailer)	Initial counteroffer from party B (Supply Chain)
Ordered Quantity: 500	Offered Quantity: 500
Price: –	Price: \$5,500
Lead-time: 4	Lead-time: 6

The condition for submodel I, $I_t(\mathrm{WL}_1) + I_t(\mathrm{WL}_2) + I_t(\mathrm{WL}_3) \geq Q_t$, is fulfilled for $Q_t = 500$.
The minimal-lead-time solution is $\widehat{w}_{11} = 400$, $\widehat{w}_{21} = 100$, $\widehat{w}_{31} = 0$
with related costs: $k_{11} \cdot 400 + k_{21} \cdot 100 = K_t(Max) = 4800$.

Therefore, the initial-offer delivery of the whole order of 500 items for a price of \$5,500 with lead time of 6 weeks was made.

The lead time of 6 for the whole order is much longer than the requested lead time of 4 weeks. Therefore, a negotiation with respect to lead time and price is required to avoid an impasse. Since negotiation parties have different views, we introduce weights that express the importance of these two attributes using a cardinal scale from [1,10] [10]. Weights can be normalized onto [0,1]. In the example, w_A and w_B denote the normalized weights, respectively, for Parties A and B

Weights of Party A:

Weights		Normalized Weights
Price:	5,	$w_A(P_t)$ = 0.357
Lead-time:	9,	$w_A(LT_t)$ = 0.643

Weights of Party B:

Weights		Normalized Weights
Price:	10,	$w_B(P_t)$ =0.666
Lead-time:	5,	$w_B(LT_t)$ = 0.333

4.1.3 *Ranges of the values of control variables (attributes)*

The ranges (intervals) of the control variables can be derived by solving an optimization problem (Fig. 7). For example, the SC-agent solves submodel I with the ordered quantity $Q_t(= 500)$ as input and gets.

This generates intervals $\widehat{K}_t \leq K_t \leq K_t(\mathrm{Max})$ and $\mathrm{L}\widehat{\mathrm{T}}_t \leq \mathrm{LT}_t \leq \mathrm{LT}_t(\mathrm{Max})$ for the control variables, provided the ordered set is $Q_t(= 500)$ and this quantity $Q_t(= 500)$ can be realized within submodel I.

The SC-agent derives a price interval $Pt \leq P_t \leq P_t(\mathrm{Max})$ from the given cost interval. In this numerical example, the requested lead time is smaller than the minimal lead time. Therefore, the SC-agent proposes the minimal lead time 6 and a price corresponding to $K_t(Max)$.

If we do not require that the ordered set of $Q_t(= 500)$ to be completely realized by submodel I, we can explore other combination of prices and lead times based on utility functions.

We assume now that there are known intervals:

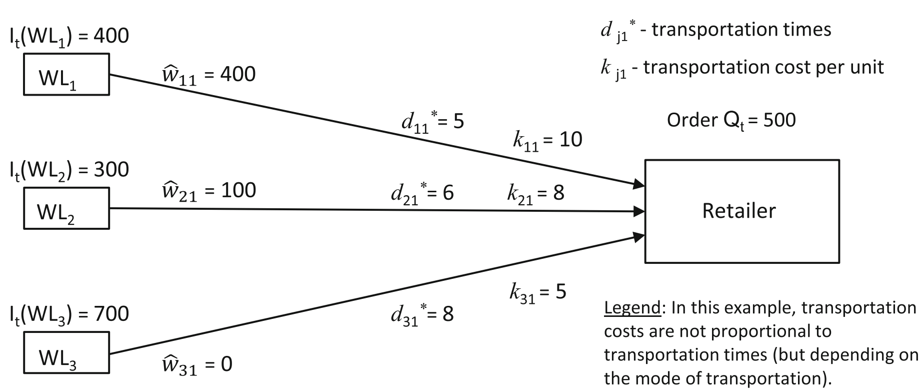
$P_t(\mathrm{Min}) \leq P_t \leq P_t(\mathrm{Max})$ and $\mathrm{LT}_t(\mathrm{Min}) \leq \mathrm{LT}_t \leq \mathrm{LT}_t(\mathrm{Max})$

4.1.4 *Utility functions*

Conditional utility functions:

The method is based on the use of utility functions of each party A and B.

Fig. 6 Computation of an initial offer; lead-time minimal solution



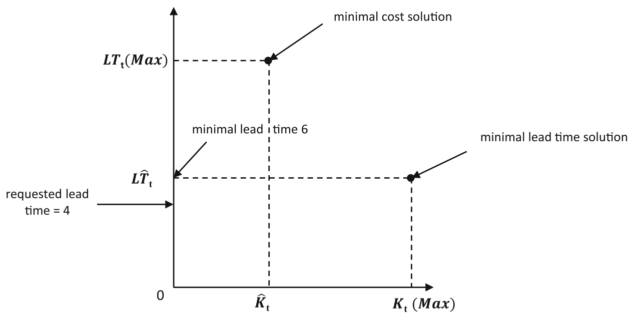


Fig. 7 Ranges of control variables

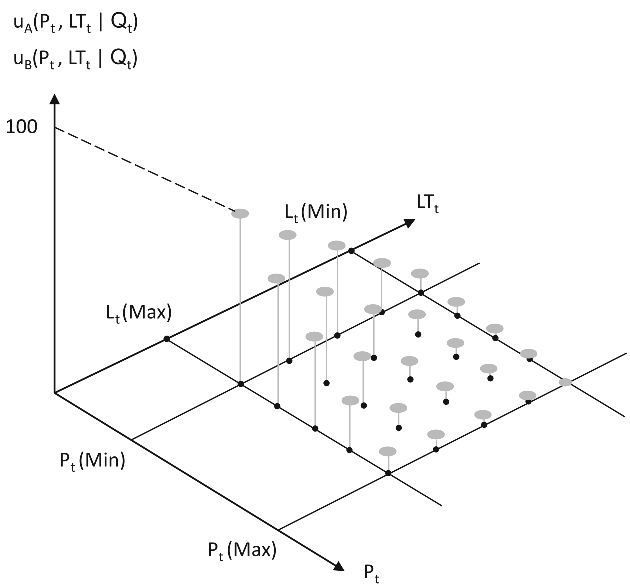


Fig. 8 Two-dimensional conditional utility function

Each party defines a weighted and conditional utility function (under the condition of ordered quantity Q_t(= 500)).

$u_A(P_t, \mathrm{LT}_t|Q_t)$, in our example $u_A(P_t, \mathrm{LT}_t|500)$
$u_B(P_t, \mathrm{LT}_t|Q_t)$, in our example $u_B(P_t, \mathrm{LT}_t|500)$
u_A, u_B: utility of party A, B of a price P_t and lead time LT_t under the condition that the order is Q_t(= 500) (defined over the ranges introduced in 4.1.3.)

It is very important to understand, that in this example, both utilities u_A and u_B depend on two variables P_t and LT_t. It means that it is not possible to define the utility of a particular value of the price variable without knowing the lead time values (Fig. 8).

If we deal with two-dimensional utilities, weights are, of course, not needed.

Weighted utility function:

We get the weighted utility functions, if we introduce one-dimensional utility functions $u_A(P_t|Q_t)$ and $u_A(\mathrm{LT}_t|Q_t)$ of the party A (and analogously for party B) with respect to only one variable P_t or LT_t, respectively (Fig. 9).

Then we get for the weighted utilities

$$u_A^w(P_t, \mathrm{LT}_t|Q_t) = w_A(P_t) \cdot u_A(P_t|Q_t) + w_A(\mathrm{LT}_t) \cdot u_A(\mathrm{LT}_t|Q_t)$$

$$u_B^w(P_t, \mathrm{LT}_t|Q_t) = w_B(P_t) \cdot u_B(P_t|Q_t) + w_B(\mathrm{LT}_t) \cdot u_B(\mathrm{LT}_t|Q_t)$$

Of course, we cannot expect that the weighted utility functions are identical with the conditional (two dimensional) utility functions u_A, u_B. Details of this weighted utility approach are discussed in [10].

In the following discussion, we consider simple examples related to the numbers in the initial offers and the respective submodel I (Fig. 9).

Joint utility functions:

Multiplication of u_A, u_B with normalized weights and adding the resulting curves of A and B results in "joint utilities of A and B with respect to each of the attributes price and lead time".

$$u_{\mathrm{Joint}}^{\mathrm{Price}}(P_t|Q_t = 500) = w_A(P_t) \cdot u_A(P_t|Q_t = 500) + w_B(P_t) \cdot u_B(P_t|Q_t = 500)$$

$$u_{\mathrm{Joint}}^{\mathrm{lead-time}}(\mathrm{LT}_t|Q_t = 500) = w_A(\mathrm{LT}_t) \cdot u_A(\mathrm{LT}_t|Q_t = 500) + w_B(\mathrm{LT}_t) \cdot u_B(\mathrm{LT}_t|Q_t = 500)$$

Using the weights given in Sect. 4.1.2 we get (Figs. 10,11):

$$u_{\mathrm{Joint}}^{\mathrm{Price}} = 0.357 \cdot u_A(P_t|500) + 0.666 \cdot u_B(P_t|500)$$

$$u_{\mathrm{Joint}}^{\mathrm{lead-time}} = 0.643 \cdot u_A(\mathrm{LT}_t|500) + 0.333 \cdot u_B(\mathrm{LT}_t|500)$$

Bui and Shakun (1996) use a simple additive function to derive the maximum of the joint or social utility functions in order to get a solution which maximizes the sum of weighted utility functions with respect to one attribute. In our example (see Figs. 10, 11), the highest joint utility for price is given by any value of the interval $[P_t^*, P_t(\mathrm{Max})]$ (Fig. 10). The highest joint utility for lead time lies in the interval $[\mathrm{LT}_t(\mathrm{Min}),\mathrm{LT}_t^*]$ where $LT_t^* = 5$ in our example.

For $P_t^* = 6{,}000$ and $P_t(\mathrm{Max}) = 7{,}000$, ($P_t(\mathrm{Min}) =$ 4,800), we find from Figs 10 and 11 that any pair $(P_t, LT_t) \in [6000, 7000] \times [4, 5]$ maximizes the joint utilities for prize and lead time as well and is, therefore, a candidate for a compromise solution.

This result is based on the one-dimensional utility analysis prescribed above.

Another approach would be to use the joint conditional two-dimensional utility functions by maximizing

$$Max\, u_A(P_t, LT_t|Q_t) + u_B(P_t, LT_t|Q_t)$$

with respect to (P_t, LT_t) over $[\mathrm{LT}_t(\mathrm{Min}), \mathrm{LT}_t(\mathrm{Max})] \times [P_t(\mathrm{Min}), P_t(\mathrm{Max})]$ and using the optimal solution (P_t^*, LT_t^*) for further negotiation. In that case, weights are not needed.

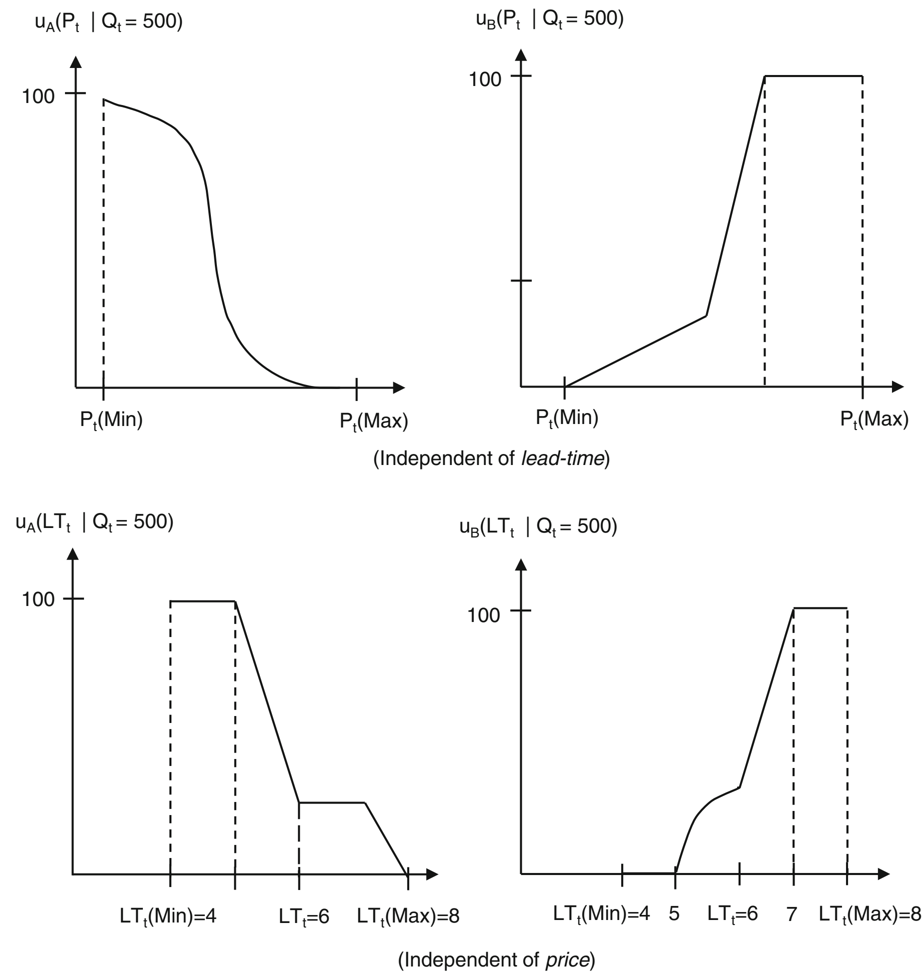


Fig. 9 Examples of one-dimensional utility functions

Price:	[\$6,000, \$7,000]
lead-time:	[4, 5]weeks

The negotiation support system offers a compromise solution in case that the whole ordered quantity 500 items will be delivered. (There are of course other possibilities, e.g., changing Q_t, splitting Q_t in subquantities with different lead times.) This concludes the first round of negotiation.

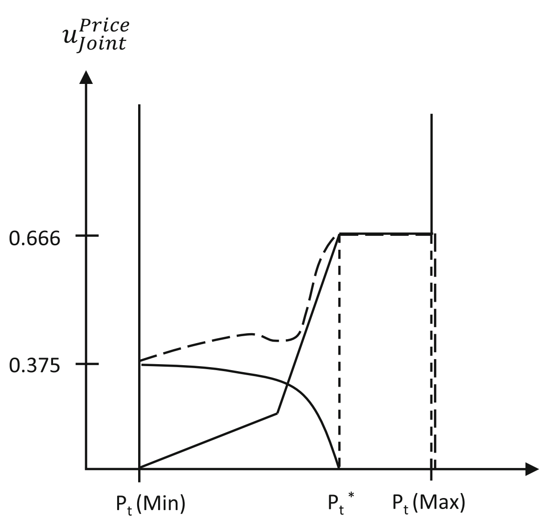


Fig. 10 Joint utility for price

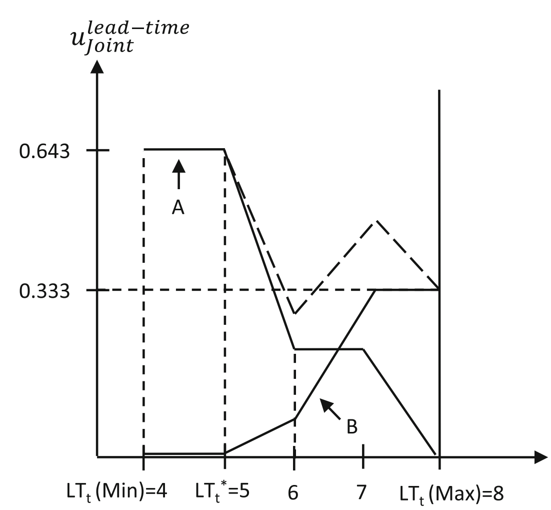


Fig. 11 Joint utility for lead time

4.1.5 Second round of negotiation using submodel III

The compromise reached in the first round of negotiation seems to be very reasonable for the SC (party A), because it is very close in terms of lead time compared to the initial offer. For party B, the new solution is still far away from the initial request (lead time of 6), because lead time of 5 is not possible in the ATP scenario (submodel I). However, the retailer (party A) only knows that the interval $4 \le LT_t \le 5$ has also for the SC maximal utility. For example, party A offers a price of \$6,000 for $Q_t = 500$ with lead time of 5. Therefore, let us assume that the retailer agent does not accept the compromise in terms of lead time but is willing to pay more instead.

That means the SC-agent is faced with an order of 500 for a price of \$6,000 and a lead time of 5 weeks. As seen in Fig. 6, it is clear that a lead time of 5 is not possible within submodel I. The SC-agent needs now to look for other submodels to find a solution.

To illustrate the approach, we assume $Q_{A_1}(\mathrm{In}-\mathrm{Transport})(t) = 0$ and consider submodel III. The respective subnetwork looks as follows (Fig. 12):

We analyze this network under the requirements $Q_t^{(1)} = 500$ and lead time $= 5$. With $\mathrm{Inv(Prod)}_t$ we denote the total inventory of all producers at time t.

Then, we have

$$I_t(\mathrm{WL}_1) = 400 < Q_t^{(1)} = 500, Q_{A_1}(\mathrm{In}-\mathrm{Transport}) = 0, \quad \mathrm{Inv(Prod)}_t = 600$$

and therefore:

$$I_t(\mathrm{WL}_1) + Q_{A_1}(\mathrm{In}-\mathrm{Transport}) + \mathrm{Inv(Prod)}_t \geq Q_t^{(1)} = 500$$

That means, the constraints for submodel III are now fulfilled. Party B tries to fulfill the required lead time $= 5$ with minimal cost.

After pre-processing using the retailer requirement: lead time $= 5$, we can set: $w_{21} = 0$, $w_{31} = 0$.

Therefore, we get: $w_{11}(t) + p_{11}(t) + p_{21}(t) = 500$

$$0 \le w_{11} \le 400, 0 \le p_{11} \le 200, 0 \le p_{21} \le 400$$
$$\text{such that} \quad 10 \cdot w_{11}(t) + 20 \cdot p_{11}(t) + 15 \cdot p_{21}(t) \to \mathrm{Min}.$$

The minimum-cost solution is: $\widehat{w}_{11}(t) = 400, \widehat{p}_{11}(t) = 0, \widehat{p}_{21}(t) = 100,$

and the assigned minimal cost value is: $10 \times 400 + 15 \times 100 = 4{,}000 + 1{,}500 = \$5{,}500$

With this new solution, the SC-agent will perhaps accept the retailer offer without further negotiation, because it realizes maximal joint utility.

Let us now consider another case: $d'_{21} = 6$ (instead of $d'_{21} = 5$, see Fig. 12).

Then we get $p_{21}(t) = 0$, and therefore, the optimization problem becomes:

$$w_{11}(t) + p_{11}(t) = 500$$
$$0 \le w_{11}(t) \le 400, 0 \le p_{11}(t) \le 200$$
$$\text{such that} \quad 10 \times w_{11} + 20 \times p_{11} \to \mathrm{Min}.$$

The optimal solution is: $\widehat{w}_{11}(t) = 400$, $\widehat{p}_{11}(t) = 100$, and minimal cost is \$6,000.

In this scenario, the SC-agent cannot offer a price of \$6,000. Therefore, parties might need to reconsider the problem (i.e., re-examine the values of the decision outcomes—price and lead time in their utility functions), and another round of negotiation is needed.

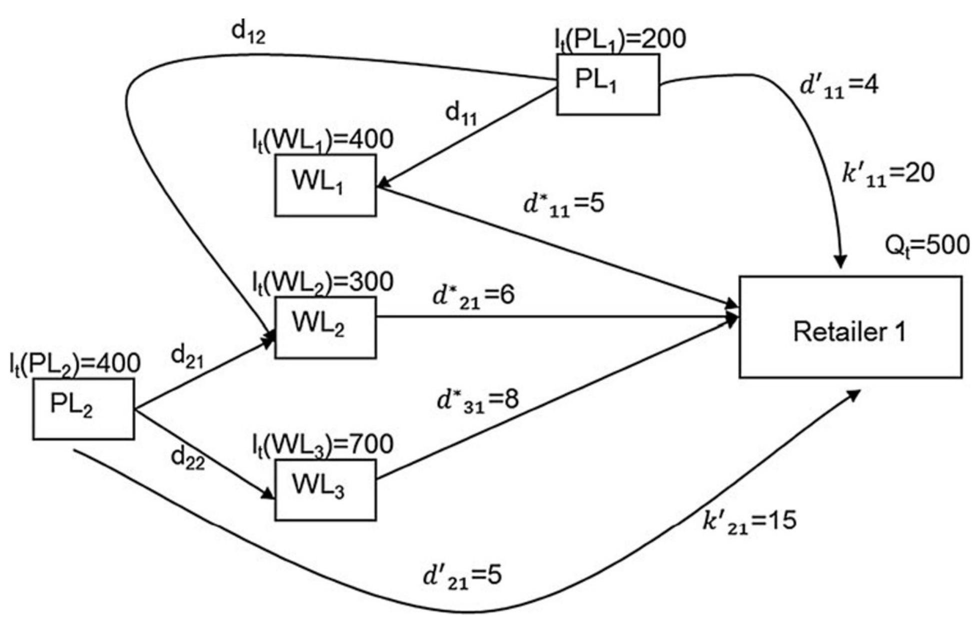


Fig. 12 Network for submodel III

The main reason is that the intervals for price and lead time are in a strong sense valid only for each of the submodels. If we change the submodels, we need to define new intervals and utilities. For example, within submodel I, a lead time of 5 is not possible given the overall ordered quantity, and a lead time of 4 is not feasible as well. Therefore, instead of defining $u_B(\mathrm{LT}_t|500) = 0$ for $4 \leq \mathrm{LT}_t \leq 5$, we could use negative utilities for agent B (see Fig. 13).

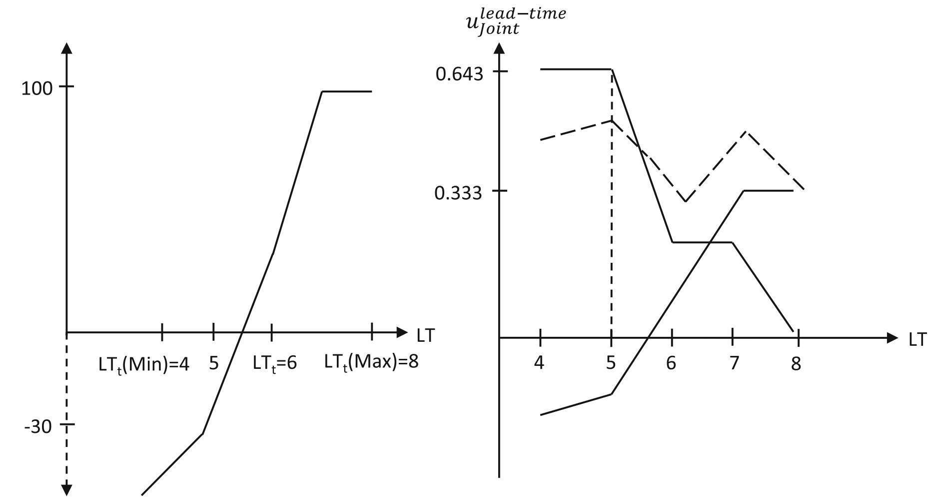


Fig. 13 Using negative utilities–utility function for lead time in the case of submodel I (lead times 4 and 5 are not possible within the scenario of submodel I)

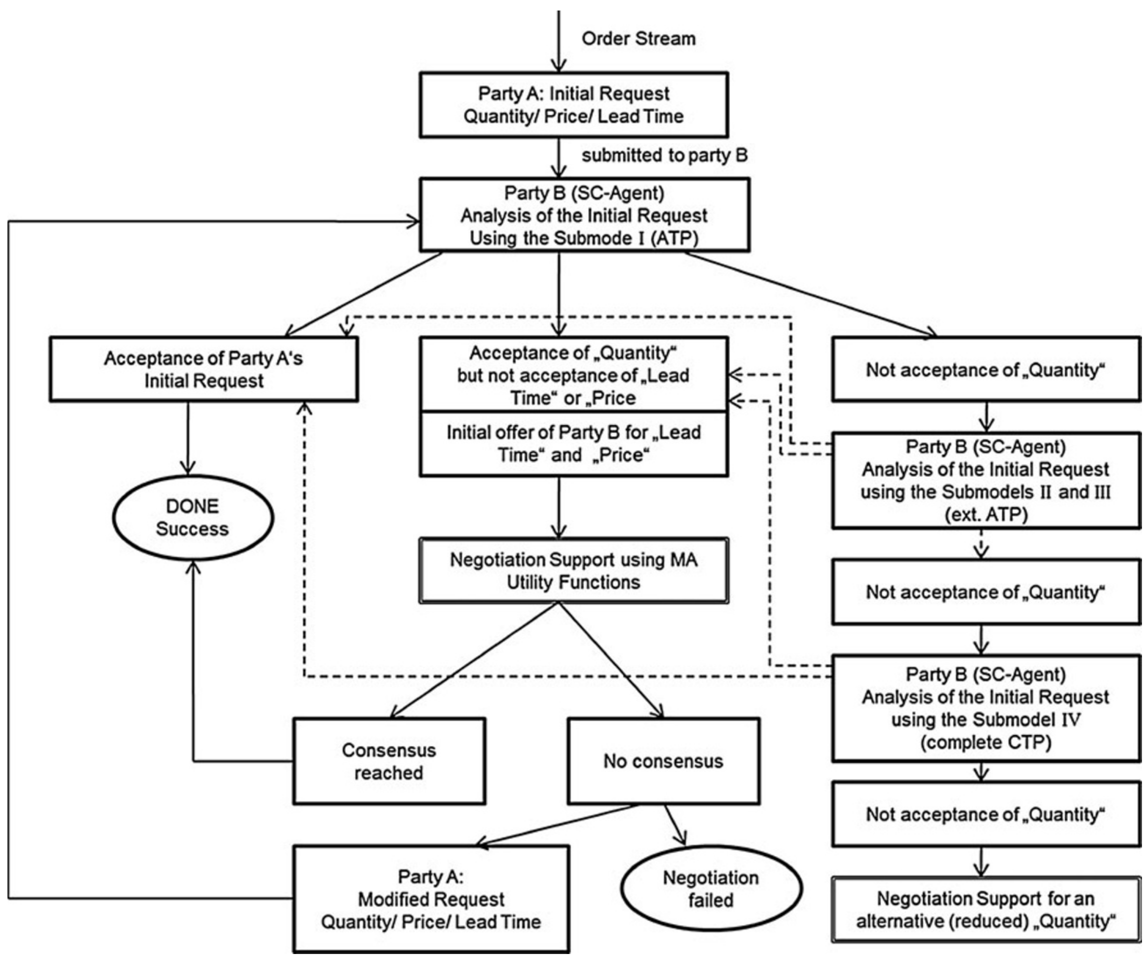


Fig. 14 An agent-based multi-attribute negotiation procedure for order promising

An utility function with negative values for an undesired subinterval is better than a smaller interval, because it gives more space for modified order fulfillment and negotiation.

This illustrates the approach. Of course, the description is not complete yet. After a positive result of the negotiation, a re-optimization of the whole supply chain is needed. Also, other types of offers are possible, for example, splitting the ordered quantity into parts with different lead times, etc.

In Fig. 14, we outline an algorithmic approach of the combined optimization–negotiation method we have introduced by a simple numeric example above.

5 Conclusion

With the increasing adoption of industry-wide supply chain systems, there is a real and urgent need to find negotiated solutions between suppliers and customers instead of rejecting orders that appear to be unable to match with supply capacity. In the context of available-to-promise (ATP) and capable-to-promise (CTP) functions, experience has shown that order promising has turned out to be a rather difficult process. It is rather common that, in an inter-firm supply chain, finding a solution (that includes delivery of available stocks and make-to-order production) to an incoming customer order is not possible. More often than not, orders are rejected, thus failing customers and incurring lost revenues. Acknowledging the impossibility of using optimization techniques to find an alternate solution, we have presented in this paper a multi-attribute, multi-utility model that can be integrated in a CTP solver to explore alternate solutions. First, we set up a basic supply chain model with the usual constraints: production, warehouse replenishment, and delivery and distribution. Next, we introduced the concept of weighted utilities for the suppliers and buyers to help them explore a variety of feasible solutions based on price, lead time, and quantity and demonstrated how the Pareto optimization can be used to search for solutions that maximize joint utilities. The proposed model has been implemented and tested using life-like scenarios. Results of our simulations suggest that negotiation procedures did reduce the number of rejected orders and increase the overall revenues when negotiation concepts and algorithms are applied. A major benefit of this research is that, unlike many suggested approaches in the recent literature, the proposed solution to solve order promising impasse is intuitive enough for supply chain participants to understand and appreciate and tractable enough to be implemented in a multi-agent system.

Acknowledgments This research is supported by the DAAD (German Academic Exchange Service) through the PPP-program.

References

1. Agatz NAH, Fleischmann M, van Nunen JAEE (2008) E-fulfillment and multi-channel distribution—a review. Eur J Oper Res 187:339–356
2. Azevedo AL, Sousa JP (2000) A component-based approach to support order planning in a distributed manufacturing enterprise. J Mater Process Technol 107:431–438
3. Ball MO, Chen C-Y, Zhao Z-Y (2004) Available to Promise. In: Simchi-Levi D, Wu D, Shen M (eds) Handbook of quantitative supply chain analysis: modeling in the E-business era. Kluwer, Boston, pp 447–484
4. Bichler M, Kersten G, Strecker S (2003) Towards a structured design of electronic negotiations. Group Decis Negotiat 12:311–335
5. Bixby A, Downs B, Self M (2006) A scheduling and capable-to-promise application for Swift and company. Interfaces 36(1):69–86
6. Bui T (1987) Co-oP: a multiple-criteria group decision support system. Springer, NewYork
7. Bui T, Bodard F, Ma PC (1998) ARBAS: an action-resource based argumentation systems. J Manag Inf Syst 14(3):223–237
8. Bui T, Gachet A, Sebastian H-J (2006) Web services for negotiation and bargaining in electronic markets: design requirements, proof-of-concepts, and potential applications to e-procurement. Group Decis Negotiat 15:469–490
9. Bui T, Lee J (1999) An agent-based framework for building decision support systems. Decis Support Syst 25:225–237
10. Bui TX, Shakun MF (1996) Negotiation processes, evolutionary systems design and NEGOTIATOR. Group Decis Negotiat 5(4–6):339–353
11. Bui T, Bosse T, Hempsch C, Sebastian H-J (2009) A multi-agent simulation framework for automated negotiation in order promising. In: Proceedings of AMCIS conference, San Francisco, August 2009
12. Bui T, Sebastian H-J (2010) Integration of multi-criteria decision analysis and negotiation in order promising. In: Proceedings of the 43rd HICCS conference, Kauai, January 2010
13. Chen C-Y, Zhao Z-Y, Ball MO (2001) Quantity and due date quoting available to promise. Inform Syst Front 3(4):477–488
14. Cohen M, Lee H (1988) Strategic analysis of integrated production-distribution systems: models and methods. Oper Res 36(2):216–228
15. Conklin J, Begeman M (1988) gBIS: a hypertext tool for exploratory policy discussion. ACM Trans Inform Syst 6(4):303–331
16. Dudek G, Stadler H (2005) Negotiation-based collaborative planning between supply chain partners. Eur J Oper Res 163:668–687
17. Emerson D, Piramuthu S (2002) Agent-based framework for dynamic supply chain configuration. In: Proceedings of the 37th Hawaii international conference on system sciences, IEEE, 0-7695-2056-1/04
18. Fischer ME (2001) Available-to-promise: Aufgaben und Verfahren im Rahmen Des supply chain managament. S. Roderer Verlag, Regensburg
19. Frey D, Stockheim T, Woelk P-O, Zimmermann R (2003) Integrated multi-agent-based supply chain management. In: Proceedings of the twelfth IEEE international workshop on enabling technologies: infrastructure for collaborative enterprises
20. Fulkerson B (1997) A response to dynamic change in the marketplace. Decis Support Syst 21:199–214
21. Fung RYK, Chen T (2005) A multiagent supply chain planning and coordination architecture. Int J Adv Manuf Technol 25:811–819

22. Gallien J, Tallec YL, Schoenmeyr T (2004) A model for make-to-order revenue management, Working Paper
23. GNU Linear Programming Kit, Version 4. http://www.gnu.org/software/glpk/
24. Grieger M (2003) Electronic marketplaces: a literature review and a call for supply chain management research. Eur J Oper Res 144:280–294
25. Interface, GLPK 4.8 J.: http://bjoern.dapnet.de/glpk/
26. Jennings NR, Sycara K, Wooldridge M (1998) A roadmap of agent research and development. Int J Auton Agents Multi-Agent Syst 1(1):7–38
27. Julka N, Srinivasan R, Karimi I (2002) Agent-based supply chain management-1: framework. Comput Chem Eng 26(12):1755–1769 (15)
28. Karacapilidis N, Papdias D (1999) Computer-supported argumentation and collaborative decision-making: the HERMES system. Inform Syst 26(4):259–277
29. Kellett PM (2007) Conflict dialogue: working with layers of meaning for productive relationships. Sage, New York
30. Kilger C, Schneeweiss L (2005) Demand fulfillment and ATP. In: Stadtler H, Kilger C (eds) Supply chain management and advanced planning, 3rd edn. Springer, New York, pp 179–195
31. Kilger C, Meyr H (2008) Supply chain management and advanced planning, 4th edn. Springer, New York
32. Larsson S (2002) Issue-based dialogue management. Working paper, Department of Linguistics, Goteborg University, Sweden
33. Louta M, Roussaki I, Pechlivanos L (2008) An intelligent agent negotiation strategy in the electronic marketplace environment. Eur J Oper Res 187:1327–1345
34. Makatsoris C, Chang YS (2008) Beyond CRM: a system to bridge the gap between the customer and the global manufacturing supply chain. In: Proceedings of the 41st Hawaii international conference on system sciences
35. Meyr H (2004) Supply chain planning in the german automotive industry. OR Spectrum 26:447–470
36. Moodie DR, Bobrowski PM (1999) Due date demand management: negotiating the trade-off between price and delivery. Int J Prod Res 37(5):997–1021
37. Moses S, Grant H, Gruenwald L, Pulat S (2004) Real-time due-date promising by build-to-order environments. Int J Prod Res 42(20):4253–4375
38. North M, Howe NCT, Howe HT, Vos R The Repast simphony runtime system. http://repast.sourceforge.net/
39. Olhager J, Ostlund B (1990) An integrated push-pull manufacturing strategy. Eur J Oper Res 45(2–3):135–142
40. Richards HD, Dudenhausen HM, Makatsoris C, de Ridder L (1997) Flow of orders through a virtual enterprise—their proactive planning and scheduling, and reactive control, Comput Control Eng J 173–179
41. Rupp T, Ristic M (2004) Determination and exchange of supply information for co-operation in complex production networks. Robot Auton Syst 49:181–191
42. Sadeh N, Hildum D, Kjenstad D (2003) Agent-based E-supply chain decision support. J Org Comput Electr Commer 13(3):225–241
43. Shin K, Leem CS (2002) A reference system for internet based inter-enterprise electronic commerce. J Syst Softw 60:195–209
44. Stadtler H (2005) Supply chain management and advanced planning—basics, overview and challenges. Eur J Oper Res 163:575–588
45. Tan GW, Shaw MJ, Fulkerson B (2000) Web-based supply chain management. Inform Syst Front 2(1):41–55
46. Vidal CJ, Goetschalckx M (1997) Strategic production-distribution models: a critical review with emphasis on global supply chain models. Eur J Oper Res 98:1–18
47. Wu H–H, Liu J-Y (2008) A capacity available-to-promise model for drum-buffer-rope systems. Int J Prod Res 46(8):2255–2274
48. Yan Y, Yen J, Bui T (2002) A multi-agent-based negotiation support system for distributed transmission cost allocation. Int J Intell Syst Acc Financ Manag 10(3):187–200
49. Zhang X, Ou J, Gilbert SM (2008) Coordination of stocking decisions in an assemble-to-order environment. Eur J Oper Res 189:540–558
50. Zhao Z, Ball MO, Kotake M (2005) Optimization-based available-to-promise with multi-stage resource availability. Ann Oper Res 135:65–85