Service Customization under Capacity Constraints: An Auction-Based Model

Chun Wang and Farnaz Dargahi

Concordia Institute for Information Systems Engineering

Concordia University, Montreal, Quebec, Canada

cwang@ciise.concordia.ca,

Abstract *In mass customization, companies strive to enhance customer value by providing products and services that are approximate to customers' needs. A company's strategy of allocating its limited capacity to meeting diverse customer requirements directly impact customer perceived value in terms of available options, cost, and schedule. Proposed in this paper is an auction-based mass customization model for solving the problem of service customization under capacity constraints (SCCC). The proposed model integrates customers' customization decision making with the allocation of company's capacity through multilateral negotiation between the company and its customers. The negotiation is conducted through a combinatorial iterative auction designed to maximize the overall customer value given limited capacity. The auction is incentive-compatible in the sense that customers will follow the prescribed myopic best-response bidding strategy. Experimental results indicate that customization solutions computed by the proposed model are very close to the optimal one. Revenue performance is also adequate when there is sufficient competition in the market.*

Keyword: Mass customization, capacity allocation, service customization, combinatorial iterative auction, multilateral negotiation, design by customers

1. Introduction

Mass Customization aims at producing what the customers need with near mass production efficiency. It can be seen as a collaborative optimization process between a company and its customers with the goal of finding the best match between the company's capabilities and the customers' needs. A company's core capabilities are the basis of its product families and their successive platforms (Meyer and Utterback, 1993). These capabilities are reflected in the people and assets applied to the development of new products. A company's capabilities can be represented by its Product Family Architecture (PFA) (Tseng and Jiao, 1996; Jiao and Tseng, 1999) which consists of a common base, a differentiation enabler, and a configuration mechanism. While PFA can serve as a systematic protocol for customers to navigate through the company's capabilities and define their own requirements, capabilities can also be organized and presented using scalable product family design (Simpson et al., 2001) and configurational product family design (Du *et al*., 2001; Ulrich, 1995). In scalable product family design, a variety of customer needs is satisfied through the configuration of scaling variables which are used to "stretch" or "shrink" the product platform in one or more dimensions. Configurational product family design, on the other hand, aims at developing a modular product platform on which product family members are derived by adding, substituting, and/or removing one or more functional modules. The pursue for a better match between the company's capabilities and customer needs has been the central theme in the mass customization literature (Jiao et. al, 2007; Simpson, 2004; Da Silveira et. al, 2001). In this paper a different perspective is taken to examine the impact of a company's *capacity* on product customizability and customer value. Here the term *capacity* is defined as a company's ability to produce customized products for a group of customers within a predefined time schedule.

It is imperative to consider a company's capacity constraints in customization decision making when production schedules are of importance to customers. This is particular true in service customization. Unlike product manufacturing, service production usually involves customer labor in the process (i.e., co-production) or requires the physical presence of the customer. Common examples can be seen in health care offices, buffet restaurants, and travel services. For service customers, it is desirable to have convenient production schedules because they need to physically present during service production. In addition, the service provider's capacity is perishable because service operations cannot rely on inventories to adjust to demand fluctuations. Perishability alludes to the time-sensitive nature of a service provider's capacity to produce the service (Sampson, 2001). In service customization, capacity constraints directly affect customers' satisfaction and the provider's profitability. Therefore, capacity constraints should be integrated into the service customization decision making.

To motivate the research from a practical perspective, consider the case of mass customization of travel packages. Major online travel brands such as Expedia Inc. (Expedia.com), Opodo (Opodo.com), and Orbitz Worldwide (Orbitz.com) are giving their customers tools to customize their own adventures in the form of "build your own package". Compared with pre-packaged vacations, customized packages are more attractive to customers because everyone's travel experience is unique and personal. A customized vacation package usually includes one or more of the following components: flight reservation, hotel reservation, car rental, and tickets to entertainment events. For a specific destination and a specific time window, the capacity limits of these components restrict customers' options and affect the customizability of travel products. This is particularly the case during high seasons when the capacity of service providers is heavily demanded. Similar situations occur in manufacturing mass customization. For example, in configurational product family design, a customer customizes its individual product by adding a group of functional modules to a base product. If a particular module takes excessively longer time to obtain due to the manufacture's capacity constraints, the customer may switch to an alternative module or even cancel the function.

This paper is concerned with the capacity aspect of mass customization. Specifically, it answers the question: Given limited capacity, how can a company maximize the value provided to its customers by coordinating customers' customization requirements? The main objective of the proposed approach is to maximize value across a large group of customers, which is, in economics terms, to maximize the *social welfare* (Mas-Colell et al., 1995). To facilitate clear formulation of the problem and meaningful presentation of the solution, the scope of the paper is restricted to service customization settings. However, the proposed model can be applied to manufacturing customization. In this paper, service customization under capacity constraints is modeled as an optimization problem; a *design-by-customers* approach is implemented using an auction. The contribution to the literature is two-fold. First, customers' customization decision making is integrated with company's capacity constraints, which is of particular relevance in service customization settings where a provider's capacity is perishable and often expensive to expand. Second, at system level, the overall value provided to customers is maximized by coordinating customers' customization requirements through auction-based multilateral negotiation. It is assumed that a company's objective is to maximize overall customer value. This objective is desirable because, in the long run, a company can improve its profit only by providing customers with high value added products and services. The rest of the paper is organized as follows. In Section II related literature is reviewed with an emphasis on the mass customization approaches that consider capacity constraints. In Section III a formulation for the SCCC problem is provided. In Section IV an auction-based negotiation model for service customization is resented. In Section V analytical results of the application of the proposed model to a special case of the SCCC problem are provided. The performance of the model on general SCCC problem is evaluated through a computational study. Conclusion and future research directions are presented in Section VI.

2. Literature Review

In mass customization manufacturing, *capacity* constraints are usually addressed from the customization management perspective in the context of manufacturing planning and scheduling. Focus has been placed on managing variety in production planning and using more flexible distributed coordination models for resource allocation (Tseng and Jiao, 2001). Typical distributed coordination models in manufacturing planning and scheduling include agent-based manufacturing (Shen et al, 2006), holonic manufacturing (Guo et al., 1994), holonic-based architecture for process manufacturing (Chokshi and McFarlane, 2008), and market-based mechanisms (Shaw, 1988; Baker, 1991; Lin and Solberg, 1992; Wellman et al, 2001). These distributed coordination models usually have a high level of responsiveness and can be supported by real-time resource availability estimation algorithms (Moses et al., 2008). Given that manufacturing planning and scheduling are usually considered as back-end issues confined within the boundary of the manufacture, in existing mass customization manufacturing approaches, capacity allocation is not explicitly integrated into front-end customer requirement definition and negotiation.

Compared with that in manufacturing sectors, customization research in services is relatively limited. As pointed out in Da Silveira et al. (2001), the lack of studies dealing with mass customization in service operations is perhaps one of the main gaps in the current mass customization literature. Anderson et al. (1997) contend that customizing a firm's offering to meet diverse needs of individual consumers is more important for satisfying service customers than for satisfying consumers of goods. Voss and Hsuan (2009) propose that service customization can be either combinatorial (i.e., the combination of a set of service processes and products to create a unique service) or menu driven (the selection of one or more services from a set of existing services/products to meet customer needs). Central to the combinatorial and menu driven approaches to service customization is the concept of modularity, a building block of modularized service system architecture. It is argued that the service architecture considered as a system at various levels of decomposition is the basis for service customization. Based on the concept of modularity and decomposition, Cao et al. (2006) propose an interactive service customization model which allows users to incrementally define the customized service process through a series of operations, including activation of goal decomposition, reusable component selection, and process composition. The on-demand process composition is supported by reusable standardized process components. Another stream of service customization research applies principles from product family design to service development. Moon et al. (2010) develop a module-based service model to represent the relationships between functions and processes in a service and to facilitate customized service design. In their service customization model, a module selection problem for platform design is considered as a strategic module-sharing problem under a collaboration situation. A coalitional game is used to model potential module sharing and determine which modules used in the platform create the most value. Similar to their manufacturing customization counterparts, existing service customization approaches are also "capability-oriented". In addition, they focus more on product modularity and decomposition.

The proposed customization model can be categorized as a *design-by-customers* approach. Tseng and Du (1998) present a two-phased design-by-customers process: customer needs acquisition phase and product design phase. In the customer needs acquisition phase, customers are first informed of the design options in terms of the company's PFA and then asked to prioritize desired product configurations in terms of the value they attach to product attribute levels. Product design phase involves an iterative refinement procedure in which customers can modify the attributes of the product through configuring the available building blocks. The objective of the iterative refinement procedure is to find a good building blocks configuration such that the customer gets a satisfactory utility from the customized product. Different from Tseng and Du (1998), the focus of this paper is rather on the integration of company's producing capacity with product's customizability. It is assumed that the company's capabilities are given and the customers know their value on different configuration of the product options. In addition, the proposed approach is mainly tailored to services customization.

In the extant literature, the most relevant study to the proposed approach is a negotiation-based methodology for custom product co-design proposed by Chen and Tseng (2007). In their paper, a bargaining procedure is proposed to explore and align demand and supply flexibilities. Customers' requirements are treated as a range of negotiable options. The supply side also exhibits some level of flexibility. Design decision making is taken as distributed and interactive problem solving with each side alternately making offers and counteroffers and collectively searching for mutually satisfactory solutions. Different from our approach, the focus of Chen and Tseng (2007) is on capability (features and attributes) negotiation rather than capacity negotiation. In addition, the negotiation model proposed in Chen and Tseng (2007) is a bilateral negotiation mechanism, whereas in this paper a multilateral model for service capacity negotiation is proposed. Chen and Tseng (2010) also propose an auction-based negotiation approach for procuring customized products. In their approach, after evaluating each proposed solution and assigning a bidding credit based on the solution's value premium, the customer conducts a reverse English auction in which manufacturers bid openly with incrementally lower prices (with product specifications being fixed). The lowest-price bidder will be awarded the contract and receive its bid price plus bidding credit as the final payment. While both Chen and Tseng (2010) and the proposed approach use auctions to address the incentive compatibility issue in a distributed environment, the auction proposed in Chen and Tseng (2010) is a standard single item reverse auction, whereas the proposed model is an iterative combinatorial auction with non-linear and non-anonymous package pricing. The proposed model is suitable for customers whose preferences exhibit complementarity over customized packages. While auctions accommodate complementary preferences have been well studied in the combinatorial auction literature (de Vries and Vohra, 2003), the proposed model is an iterative combinatorial auction specifically designed for service mass customization. The properties of the model and its effectiveness under various levels of product customizability have also been analyzed. Blecker et al. (2004) propose an auction-based framework for variety formation and steering in mass customization. In their approach, the main product building blocks are modeled as autonomous rational agents participating in an auction market where they compete by bidding to form product variants. To ensure their self-preservation, agents have to compute suitable bidding strategies with the best chances to meet customers' requirements. This is a distributed coordination model aiming at matching the company's capability with customer's requirements.

Travel package auctions have become one of the most popular service auctions on the Internet. Pre-packaged vacations are sold in travel auction websites, including eBay Travel (http://www.ebay.com), Luxury Link (http://www.luxurylink.com), and Sky Auction (http://www.skyauction.com). Some of these websites allow customers to select their own travel date ranges. Different from our auction-based customization model, existing online travel auctions usually do not provide the flexibility of bidding for a customized package.

3. Service Customization under Capacity Constraints

This section provides a formulation of the SCCC problem which consists of a group of customers and a service provider. Customers want to customize the service products. To provide a common design domain, the provider is assumed to adopt a configurational product family design approach (Du *et al*., 2001; Ulrich, 1995) such that it can present its capabilities in the form of a set of building blocks (services). Customers can customize the product by choosing a base product (a pre-defined group of services) and adding optional services according to their preferences. A customized product is a package of services chosen by a customer. For example, a vacation package can include transportation services, accommodation services, and additional entertainment activities. For a provider, a service has a capacity limit which is defined as the number of customers the service can accommodate during a specified time window. For each package of services, the customer attaches a *value* to it. This paper follows the *private value model* introduced by Vickrey (1961). According to the private value model, a customer has a value for each package, and this value does not depend on other customers' private information. A customer's payoff is linear in the customer's valuation of the package and the price paid for it. To maintain a positive payoff, the customer is willing to pay up to their value to obtain the package. It is important to note that a customer's value is fixed and hence not a function of the price paid for a package. However, a customer's payoff decreases when the price of the package increases.

Formally, the SCCC problem consists of a set of n customers and a set of m services. A customer can configure its service package by selecting a group of services. A service package has to include a pre-configured set of services, that is, the *base configuration*, denoted $\bar{S}$. For service i, its capacity is limited by $capacity(i)$. Let E_j be the set of service packages which are acceptable by customer j (i.e., *feasible packages*) and E be the union of the sets of acceptable service packages from all customers, $E = \bigcup_{j=1\dots n} E_j$. Let $v_j(B)$ be the value of customer j attached to the service package $B \in E$. $v_j(B) > 0$ if $B \in E_j$; $v_j(B) = 0$ otherwise. Let $x_j(B) = 1$ if the package $B \in E$is allocated to customer j and zero otherwise. The SCCC problem involves the selection of a set of service packages for customers such that the service provider's capacity constraints are respected and, at the same time, the sum of customer value (social welfare, in terms of microeconomics) derived from the selected packages is maximized. The problem can be formulated as the following integer programming.

$$max \sum_{j=1}^{n} \sum_{B \in E} x_j(B) v_j(B)$$

subject to

$$\sum_{B \in E} x_j(B) \leq 1, \quad j = 1, \dots, n \qquad (1)$$

$$\sum_{B \ni i} \sum_{j=1}^{n} x_j(B) \leq capacity(i), \; i = 1 \dots m \qquad (2)$$

$$\sum_{B \in E} x_j(B) = \sum_{B \in E_j} x_j(B), \quad j = 1, \dots, n \qquad (3)$$

$$\sum_{B \in E} x_j(B) = \sum_{B \supseteq \bar{S}} x_j(B), \quad j = 1, \dots, n \qquad (4)$$

$$x_j(B) = \{0,1\}, \quad B \in E, \quad j = 1, \dots, n \qquad (5)$$

Constraints (1) ensure that a customer can only obtain one service package. Constraints (2) ensure that the allocation of a service to customers does not exceed the capacity limit of the service provider. The set of constraints (3) ensure that if a package is assigned to a customer, it must belong to the set of product configurations acceptable by the customer. These constraints prevent the provider from assigning customers packages which they are not willing to accept. Constraints (4) enforce the selection of the base configuration in each awarded packages. Constraints (5) are a set of integer constraints. The provider's SCCC problem is NP-hard as stated in the following theorem.

Theorem 1: *The problem of service customization under capacity constraints (SCCC) is NP-hard.*

Proof: To show that SCCC is NP-hard, consider a special case in which $E_j = E$ for all $j = 1, \dots n$ and $\bar{S} = \phi$. In this case, Constraints (3) and Constraints (4) always hold. The relaxed model is a set packing problem, which is NP-complete (Karp, 1972). It follows that, as a general case, SCCC problem is NP-hard■.

The SCCC is an integer programming model which takes customer value as input. The key question to be asked here is how the value which each customer assigns to the package can be obtained. The way of computing value from product configurations can be customer specific. One approach suggested by Tseng and Du (1998) is to use methods designed to measure consumer preferences in marketing research, such as conjoint analysis (IntelliQuest, 1990). Conjoint analysis assumes that a product could be described as vectors of attributes, and each attribute can include several discrete levels. To apply conjoint analysis to SCCC, each service is modeled as an attribute and the discrete levels of attributes are restricted to 1 (service included) and 0 (service not included). As SCCC requires customers' complete valuation on all feasible packages, computing value for each and every configuration may become impractical when the space of feasible packages becomes large. Although customers can determine the value of feasible packages, they may be reluctant to report the value back to the service provider because, by the definition of private value model, value is the highest price that a customer is willing to pay for a given package. In many cases, these prices are sensitive private information. The following section proposes an auction-based service customization model which computes high quality solutions to SCCC without requiring valuations from customers.

4. The Auction-Based Service Customization Model

Auctions have long been considered as an effective way of allocating limited resources to competing users and of discovering market prices for products and services. In recent years, the pervasive inter-connectivity provided by the Internet has made auctions a popular mechanism that directly links the capacities of service providers with end customers. The proposed auction is a price mechanism in which a provider coordinates the customization requirements among its customers by adjusting the prices of service packages. In this section, the design of the auction is first described and then a worked example is presented to demonstrate the application of the auction to travel package customization. We also discuss important incentive and implementation considerations of the auction design in this section.

4.1. The Auction Model

The auction is designed as an iterative bidding procedure. A customer's bid is represented as a price-package pair $\langle package,\ bidding\ price \rangle$, where $package$ is the set of services that the customer wants and the $bidding\ price$ is the price that the customer is willing to pay for the services to be delivered. The bidding price is customer-dependent. A non-anonymous price structure, in which there is no common public price for a package, is deployed in the model. This structure allows the provider to price the same package differently for different customers, which is a common practice in many service industries. The bidding procedure consists of four components, namely initialization, availability update and bidding, termination checking, and winner determination.

4.1.1. Initialization

Before the bidding starts, the provider presents to the customers the set of available services and the base configuration that must be included in a customized package. Customers compute their respective sets of feasible packages E_j. For each package in E_j, the customer computes their value attached to it. The provider usually has a reservation price for each of the packages, which is the lowest price that the package can be bought. The initial bidding price for a package is set to be equal to its reservation price. If the reservation prices are unknown to the customers as in some online auctions, the initial bidding prices are set to zero. Knowing the values and initial bidding prices of packages in their E_j, a customer computes the payoff of each package. As explained previously, under the private value model, a customer's payoff for a package is the difference between their value and the bidding price. To maintain a positive payoff, the customer is willing to pay up to their value to get the package. After obtaining payoffs of their feasible packages, the customer selects the package with the highest payoff (breaking ties randomly) as their first package to bid.

4.1.2. Price Update and Bidding

At the beginning of round t ($t > 1$), customers need to update their bidding prices for the packages submitted at round $t - 1$, based on the provisional allocation determined at round $t - 1$. If a customer's bid was not awarded in the provisional allocation at round $t - 1$, the customer has three price updating options at round t: (1) it can increase its bidding prices by ε on the package it bid for at round $t - 1$ or rounds before $t - 1$, where ε is the minimum price increment imposed by the provider. Since customers are assumed to be rational in maximizing their payoffs, they, in general, do not bid with an increment more than ε; (2) it can also keep the bidding prices unchanged (taking an ε discount). However, if a customer takes this ε discount, the provider will consider the customer has entered into *final bid status* and the customer is forbidden from increasing the bidding prices on any of its packages in future rounds; and (3) the customer can, of course, withdraw from bidding. If a customer is included in the provisional allocation at round $t - 1$, it can keep its bidding price unchanged at round t. That is, it is allowed to repeat the same bids at round -1 . However, the bidding procedure does not prevent the customer from entering a higher bid.

After updating bidding prices, a customer needs to compute its set of payoff maximizing packages based on the updated bidding prices. In computing such a set, a customer *j* solves a maximization problem $max_{B\in E_j}\left[v_j(B) - p_j^t(B)\right]$ and obtains the set of packages that equally maximizes the payoff, where $p_j^t(B)$ is the bidding price for B at round t. That is, for any two packages B and B' in the payoff maximizing set, $v_j(B) - p_j^t(B) = v_j(B') - p_j^t(B')$. After obtaining the set of payoff maximizing packages, the customer randomly picks one and submits it to the provider with the updated bidding price. If a customer has entered into final bid status, it is no longer allowed to increase its bidding price. However, the customer can repeat its final bid in future rounds until termination. The purpose for this *final bid repeating* arrangement is to boost the provider's revenue. During the iterative bidding process, some bids can be temporarily "excluded" from the provisional allocation by a particular combination of allocation constraints and resource requirements from other bids with higher combined value. After several rounds, that particular combination may have changed to allow the space for previously excluded bids to be included in the allocation. However, without final bid repeating, those bids will not be submitted again if their valuations have been reached during the "excluded" periods. Therefore, they would not be included, even though capacity becomes available for them in subsequent provisional allocations.

4.1.3. Bids Screening and Termination

After bids are received from the customers, the provider first screens out invalid bids. Those bids will not be considered in the following winner determination procedure. Invalid bids are defined as having (1) any bidding price for a package which is below the highest bidding price for that same package received in previous rounds, (2) increased prices from customers who have already declared their final bidding status in previous rounds, and (3) packages which do not contain the base configuration or violate other configuration rules.

The provider then checks the termination condition against the valid bids. The bidding terminates if there are no price updates for all valid bids in the current round. That is, all customers that bid in the last round have repeated their bids. After the bidding terminates, the provider implements the final allocation and the customers pay their bidding prices. If the termination condition is not satisfied, the provider will take the set of valid bids as input and solve the winner determination model. After winner determination, the auction goes back to price update and bidding stage.

4.1.4. Winner Determination

The provider needs to compute a new provisional allocation in each round as long as the bidding is not terminated. The winner determination model is to select a subset of the bids submitted by customers such that the overall bidding price of the provisional allocation is maximized and the capacity constraints of the provider are not violated. Let N^t be the set of customers submitted their bids at round t and $p(B_j^t)$ be the bidding price of B_j^t, where B_j^t is the package submitted by customer j at round t, $j \in N^t$. Let $Z_j = 1$ if customer j wins and $Z_j = 0$ otherwise. The winner determination model can be expressed using the following integer programming.

$$max \sum_{j \in N^t} Z_j p(B_j^t)$$

$$subject\ to$$

$$\sum_{\substack{j \in N^t \\ B_j^t \ni i}} Z_j \leq capacity(i),\ \ i = 1 \ldots m \qquad (6)$$

$$Z_j = \{0,1\}, \quad j \in N^t \qquad (7)$$

Constraints (6) ensure that the bids awarded in a provisional allocation do not violate the provider's capacity constraints. Constraint (7) is a set of integer constraints.

The winner determination problem is a general form of the set packing problem which is NP-hard. In this paper, we use the commercial optimization package ILOG CPLEX 10.2 to solve it. Although winner determination problems in combinatorial auctions are in general NP-hard, many of them can be solved quickly by modern optimization algorithms to fairly large sizes. It is reported in Andersson et al. (2000) that CPLEX 6.5 performs very well in terms of running time for many of the common winner determination problem benchmarks distributions. The solving speed is comparable to the special-purpose winner determination algorithms, such as those in Fujishima et al. (1999) and Sandholm (2002). It is also shown in Sandholm et al. (2005) that some winner determination distributions with thousands of bids in an instance can be solved by CPLEX 8.0 within a couple of seconds. Our experiments in Section 5.2 also confirm that CPLEX 10.2 can quickly solve the SCCC problem instances with 1000 customers. However, for some problems with more domain specific constraints, such as those in Wang et al. (2009), CPLEX is much slower (an order of magnitude on average) than specially designed algorithms.

4.2. A Worked Example

In this subsection a worked example in a travel package customization setting is presented to demonstrate the application of the auction-based customization model. Suppose a travel agency offers “build your own package” tool for its customers to customize their vacation packages for a 7-day holiday season at a popular destination. Customers should travel to the destination on Day 1 and return on Day 7. The agency offers a list of travel components including flight reservation, hotel reservation, car rental, and tickets to entertainment events. There are multiple services for each of the components to accommodate various customer preferences. For example, Departure Ticket (DT) can be scheduled in the morning (DT-1), afternoon (DT-2), or evening (DT-3). For illustrative purposes, an example of an unrealistically small number of customers (five customers) is designed. In table 1 available services and their respective capacity are summarized. Table 2 shows customers’ feasible packages and their valuations on them, where B(a, b) represents the feasible package b from customer a. Customer value is generated using the method described in the computational study section (Section 5.2.1). The base configuration includes one and only one service of each of the components DT, RT, and HL. Customers can have one to five services from the component ET. To limit the number of rounds of bidding, high reservation prices are set for the packages (see Table 2). Submitted bids, provisional allocation, provider’s revenue, and customer’s value at each round of bidding are summarized in Table 3. ε is set to be 5. The auction terminates at round 12 with overall customer value at 7370. Compared with the optimal value 7790, the auction reaches 95% efficiency in this example. The sum of the prices paid by customers (i.e., provider revenue) is 7240, which is close to the overall solution value due to competition among customers. The provisional allocations along the bidding process manifest the heuristic search guided by the changing package bidding prices.

Table 1 Summary of service capacity

Service ID	Service Description	Capacity
DT-1	Departure Ticket in the morning of Day 1	3
DT-2	Departure Ticket in the afternoon of Day 1	2
DT-3	Departure Ticket in the evening of Day 1	2
RT-1	Return Ticket in the morning of Day 7	2
RT-2	Return Ticket in the afternoon of Day 7	2
RT-3	Return Ticket in the evening of Day 7	3
HL-1	First-class hotel	1
HL-2	Second-class hotel	3
HL-3	Motel	2
ET-1	Sporting event ticket	2
ET-2	Performing arts ticket	2
ET-3	Museum ticket	3
ET-4	Cruise trip ticket	3
ET-5	Fine dining ticket	2

Table 2 Customers' feasible packages and corresponding reservation prices and value

Customer	Feasible Packages	Reservation Price ($)	Value ($)
Cus#1	B (1,1) = {"DT3" "RT3" "HL3" "ET2" "ET3"}	1410	1445
Cus#1	B (1,2) = {"DT2" "RT1" "HL1" "ET1""ET3" "ET4" "ET5" }	2200	2250
Cus#1	B (1,3) = {"DT3" "RT2" "HL2" "ET4"}	1830	1870
Cus#2	B (2,1) = {"DT1" "RT3" "HL1" "ET1" "ET2" }	2120	2145
Cus#2	B (2,2) = {"DT1" "RT1" "HL1" "ET1" "ET4" "ET5"}	2320	2360
Cus#2	B (2,3) = {"DT3" "RT3" "HL2" "ET1" "ET3""ET4" }	2060	2085
Cus#3	B (3,1) = {"DT3" "RT3" "HL1" "ET1" "ET2" "ET3" "ET5"}	2210	2235
Cus#3	B (3,2) = {"DT1" "RT2" "HL1" "ET1" "ET3""ET4" }	2360	2370
Cus#4	B (4,1) = {"DT2" "RT1" "HL3" "ET1""ET2" "ET3" "ET4"}	1660	1695
Cus#4	B (4,2) = {"DT2" "RT1" "HL2" "ET1" "ET3"}	1730	1740
Cus#5	B (5,1) = {"DT3" "RT3" "HL2" "ET5"}	1610	1660
Cus#5	B (5,2) = {"DT1" "RT3" "HL1" "ET1" "ET3" "ET4" "ET5"}	2360	2375
Cus#5	B (5,3) = {"DT2" "RT2" "HL1" "ET2""ET4" "ET5" }	2130	2135
Cus#5	B (5,4) = {"DT3" "RT3" "HL3" "ET3" "ET5"}	1290	1295
Cus#5	B (5,5) = {"DT1" "RT3" "HL2" "ET3""ET4"}	1910	1945

Table 3 Submitted bids, provisional allocation, provider's revenue, and customer's value at each round of bidding

Round #	Submitted Bids	Provisional Allocation	Provider Revenue ($)	Customer Value ($)
1	B (1,2) , B (2,2) , B (3,1) , B (4,1) , B (5,1)	B (2,2), B (4,1) , B (5,1)	5590	5715
2	B (1,2) , B (2,2) , B (3,1) , B (4,1) , B (5,1)	B (2,2), B (4,1) , B (5,1)	5590	5715
3	B (1,2) , B (2,2) , B (3,1) , B (4,1) , B (5,1)	B (2,2), B (4,1) , B (5,1)	5590	5715
4	B (1,3) , B (2,2) , B (3,1) , B (4,1) , B (5,1)	B (1,3), B (2,2), B (5,1)	5760	5890
5	B (1,3) , B (2,2) , B (3,2) , B (4,1) , B (5,1)	B (1,3), B (3,2), B (5,1)	5800	5900
6	B (1,3) , B (2,2) , B (3,2) , B (4,1) , B (5,1)	B (1,3), B (3,2), B (5,1)	5800	5900
7	B (1,3) , B (2,2) , B (3,2) , B (4,1) , B (5,1)	B (1,3), B (3,2), B (5,1)	5800	5900
8	B (1,3) , B (2,1) , B (3,2), B (4,1) , B (5,1)	B (1,3) , B (2,1), B (4,1) , B (5,1)	7240	7370
9	B (1,3) , B (2,1) , B (3,1), B (4,1) , B (5,1)	B (1,3) , B (2,1), B (4,1) , B (5,1)	7240	7370
10	B (1,3) , B (2,1) ,B (3,2) , B (4,1) , B (5,1)	B (1,3) , B (2,1), B (4,1) , B (5,1)	7240	7370
11	B (1,3) , B (2,1) , B (3,1) , B (4,1) , B (5,1)	B (1,3) , B (2,1), B (4,1) , B (5,1)	7240	7370
12	B (1,3) , B (2,1) , B (3,1) , B (4,1) , B (5,1)	B (1,3) , B (2,1), B (4,1) , B (5,1)	7240	7370

4.3. Implementation Considerations

The efficiency of auctions largely depends on the level of competition among customers. While the Internet provides pervasive accessibility to virtually any electronic market, customers may come at different time. To aggregate demand and facilitate competition, Internet auctions usually span a couple of days or even longer. Customers can enter the auction and place bids at any time before the auction ends. To spare customers the trouble of continuously monitoring the bidding process and repeatedly placing their bids, Internet auctions allow bidders to provide direct value information to an automated bidding agent called proxy agent which bids on the behalf of customer.

In the proposed iterative auction for SCCC, a proxy agent needs to manage a set of feasible packages of the customer and decides which package to submit, at which round, and at what price. Thus, the customer should inform the agent its value on each of the feasible packages. In the meantime, the agent should be equipped with the algorithm to update bidding prices and select the payoff maximization package along the bidding process. If the customer prefers, the agent can also inform the customer regarding the bidding status and allow the customer to update its value before the auction ends. For easy access, customers may install the proxy agent on a personal computer, a smart phone, or other mobile devices.

Many online travel auctions, including those mentioned previously in this paper, provide a “buy it now” option to accommodate those buyers who cannot wait until the auction ends. A buyer can purchase the item immediately by paying the buy-it-now price. However, the buy-it-now price is usually a regular retail price which can be much higher than the final auction price. Rigorously, we should not consider buy-it-now as part of the auction design.

4.4. Incentive Issues

Given the customers’ private value model we have assumed, no customer bids above their valuation. In all cases, customers will not get negative payoffs, which encourage them to participate in the auction. However, understanding the incentives that a company has for setting up and conducting the proposed auction requires some explanations on the company’s objectives for auction design. In auction design there are two common objectives an auctioneer may have. The first is economic efficiency, and the second is revenue maximization (de Vries and Vohra, 2003). An auction is *economically efficient* if the allocation of objects to bidders chosen by the auctioneer maximizes the overall values of bidders. Economic efficiency is supported by well-developed auction theories. A typical example is the canonical Vickrey-Clarke-Groves mechanism (Vickrey 1961; Clark 1971; Groves 1973) which simultaneously achieves incentive compatibility and efficiency and has guided the design of many auctions. As a result, the majority of the auction literature takes economic efficiency as the design objective.

It is argued in Parkes and Kalagnanam (2005) that the goal of economic efficiency is well suited for the design of stable long-term markets that will form the basis for repeated trade. They expect that efficient markets will come to dominate the electronic market landscape based on their experience with procurement auctions deployed with a large chocolate manufacturer (Hohner et al., 2003). In the context of mass customization, economic efficiency is also desirable for a company which wants to build long-term business relationship with their customers. It is agreed in mass customization literature that one of the major objectives of mass customization is to improve customer value. In the long run, a company can only improve its profit by providing customers with high value added products and services. The long term benefits brought by the efficient auction design provide an incentive for companies to adopt economic efficiency as their auction design objective.

The objective of revenue maximization (optimal auction design), on the other hand, maximizes the auctioneer's revenue. Optimal auctions maximize seller's revenue in each transaction, which are perhaps more appropriate for a one-shot procurement problem, and in a setting in which the buyer has considerable market power (Parkes and Kalagnanam, 2005). Even a company only cares about short term benefits and wants to get the most out of every transactions, an efficient auction design is still a reasonable choice, especially when iterative bidding is used as an implementation structure. This is because there are no known optimal (i.e. revenue-maximizing) general-purpose combinatorial auctions, iterative or otherwise (Parkes, 2006). In fact the dynamic exchange of value information between bidders that is enabled within iterative combinatorial auctions is known to enhance revenue and efficiency in single item auctions with correlated values (Milgrom and Weber, 1982). One should expect efficient iterative combinatorial auctions to retain this benefit over their sealed-bid counterparts (Parkes, 2006). Therefore, from both long term and short term perspectives, a company has the incentives to deploy an efficient combinatorial auction.

The proposed auction is an efficient auction design which is implemented using an iterative bidding process. The bidding process is guided by a price mechanism. The revenue that the auctioneer collects is the sum of the bidding prices from awarded customers at winner determination. Given the design of the bidding procedure, the company's revenue is guaranteed to increase along the bidding process and reach its highest at termination. Despite the formulation of the economic efficiency objective of SCCC, the iterative bidding structure itself achieves high seller revenue in the same spirit of many real-world iterative auction applications, which supports our claim that the proposed model provides incentives to the seller.

5. Properties of the Auction-Based Customization Model

In this section the performance of the auction-based customization model is evaluated. The game-theoretic property of the model and efficiency analysis on the application of the model to a special case of the SCCC problem is first provided. The performance of the model on general SCCC problems is also evaluated through a computational study.

5.1. Analytical Results

In this paper, customers are treated as self-interested agents in the sense that they maximize their own objectives without considering the overall system performance. Requiring customers to reveal their valuations on packages is not practical because customers may fear that the provider will take advantage of the information and consequently charge higher prices for their favorite packages. In microeconomics, there are two approaches to modeling agent behaviors. The first model is game-theoretic and is based on mechanism design theory. In this model the equilibrium state is defined by the condition that agents play a best-response strategy to each other and cannot benefit from a unilateral deviation to an alternative strategy. The second model of agent behavior is price-taking, or myopic best-response, and relates to competitive equilibrium theory. In this second model, the equilibrium state is defined by the condition that an agent plays a best-response to the current price and allocation in the market, without modeling either the strategies of other agents or the effect of its own actions on the future state of the market. Mechanism design theory and game-theoretic modeling are most relevant when there is a small number of agents and when agents are expected to be rational and well-informed about the likely preferences of other agents. Competitive equilibrium theory and price-taking modeling are most relevant in large systems in which the effect of an agent's own strategy on the state of a market is small, or when there is considerable uncertainty about agent preferences and behaviors and no useful mechanism with dominant strategy equilibrium. In the design of the auction-based customization model for SCCC, customers are modeled as price takers because in mass customization environments it is reasonable to assume that the number of customers is not small. Therefore, the proposed auction is incentive compatible in the sense that customers will follow the myopic best-response bidding strategy prescribed by the proposed auction protocol.

The rest of the section is devoted to the evaluation of the quality of the customization solutions computed using the auction model. Since the objective is to maximize social welfare, quality here is defined as the sum of the value provided to all customers in a customization solution. Analytical results are developed for a special case of SCCC, in which a customer only has one feasible package. As stated in the following proposition, the proposed auction procedure computes optimal solutions for the special case.

Proposition 1: For a special case of the SCCC problem, where each of the customers only has one feasible service package and their value on the package is congruent to the reservation price of the package modulo ε, the iterative bidding procedure with final bid repeating always maximizes the sum of customers' valuations at its termination.

Proof: Since customers are assumed to take a private value model, a customer is willing to pay up to its value to get a package. Therefore, if a customer is not awarded in a provisional allocation, the customer will keep increasing its bidding prices in future rounds until it is awarded or it reaches its valuation. In the case of final bid repeating, customers repeat their previous bids at termination (round T). Therefore, all customers that are not included in the final allocation (denoted S^T) have bids with their valuations and the customers that have room to increase their bidding prices at termination are all included in S^T. The proposition is proved by showing that S^T is identical to the optimal allocation S^* computed by solving the winner determination problem using all customers' valuations as input.

The customers' bidding prices are constructed for an additional round (round $\mathrm{T}+1$) as follows. Pick a customer $l \in S^T$ with a bidding price at termination (denoted as p_l^T) that is smaller than their valuation. Let $p_l^{T+1} = p_l^T + n\varepsilon$. n is selected to make sure that p_l^{T+1} is the valuation of l. Since it is assumed that customers' valuations are congruent to the reservation price modulo ε, n must be an integer. For any other customer $j \in S^T$ and $\neq l$, $p_j^{T+1} = p_j^T$. Let S^{T+1} be the resultant allocation generated by the winner determination for round $\mathrm{T} + 1$. The first step is to prove $S^{T+1} = S^T$ by contradiction. Suppose $S^T \neq S^{T+1}$, consider the following two cases.

Case #1: $l \in S^{T+1}$. Because S^T is the allocation that maximizes the provider's revenue given the set of bidding prices at round T and it is assumed that $S^T \neq S^{T+1}$, it follows that $\sum_{j \in S^T} p_j^T > \sum_{j \in S^{T+1}} p_j^T$. By adding $n\varepsilon$ to both sides, it follows that $\sum_{j \in S^T \backslash l} p_j^T + p_l^T + n\varepsilon > \sum_{j \in S^{T+1} \backslash l} p_j^T + p_l^T + n\varepsilon$. That is, $\sum_{j \in S^T \backslash l} p_j^T + p_l^{T+1} > \sum_{j \in S^{T+1} \backslash l} p_j^T + p_l^{T+1}$. Because $\sum_{j \in S^{T+1}} p_j^{T+1} = \sum_{j \in S^{T+1} \backslash l} p_j^T + p_l^{T+1}$, it follows that $\sum_{j \in S^T \backslash l} p_j^T + p_l^{T+1} > \sum_{j \in S^{T+1}} p_j^{T+1}$, which means S^{T+1} does not contain the set of customers whose bidding prices at round $T + 1$ maximize the provider's revenue. This is a contradiction to the assumption.

Case #2: $l \notin S^{T+1}$. Because S^T is the allocation that maximizes the provider's revenue given the set of bidding prices at round T and $S^T \neq S^{T+1}$ is assumed, it follows that $\sum_{j \in S^T} p_j^T > \sum_{j \in S^{T+1}} p_j^T$. Since $\sum_{j \in S^T} p_j^T = \sum_{j \in S^T \backslash l} p_j^T + p_l^T$, it is clear that $\sum_{j \in S^T \backslash l} p_j^T + p_l^T + n\varepsilon > \sum_{j \in S^{T+1}} p_j^T$. Given the way that bidding prices at round $T + 1$ are constructed and $l \notin S^{T+1}$, it is followed that $\sum_{j \in S^T \backslash l} p_j^{T+1} + p_l^{T+1} > \sum_{j \in S^{T+1}} p_j^{T+1}$, which means S^{T+1} does not contain the set of customers whose bidding prices at round $T + 1$ maximize the provider's revenue. This is also a contradiction to the assumption.

By deriving two contradictions in case #1 & #2, it can be concluded that $S^T = S^{T+1}$. It is now ready to prove that S^T is optimal, that is, $S^T = S^*$. Note that S^* is a schedule computed using all customers' valuations as input. In S^{T+1}, customer l has a bid with its valuation. Since l was an arbitrary pick, $S^T = S^{T+1}$ can be a general conclusion for all other customers included in S^T. By repeating the above process for each of the customers, the bidding procedure can reach a final round where all customers included in S^T bid with their valuations. Note that, by definition, the resultant allocation at this final round is S^*. Therefore, $S^T = S^*$. It follows that S^T maximizes the sum of customers' valuations. █

5.2. Value and Revenue Performance under Various Product Customizability

Products with a higher level of customizability will likely meet individual customer needs better. However, a higher level of customizability often leads to higher costs. To manage the customization costs and improve operational efficiency, service providers usually restrict customers' freedom in choosing any combination of the services by imposing configuration rules. The proposed customization model allows providers to adjust the customizability of packages by defining different base configurations. When customizing a package, a customer is required to incorporate the services defined in the base configuration into the package. In terms of platform-based product development, the base configuration serves as a base product on which customers build their customized products. In this subsection the value and revenue performance of the auction-based customization model is validated under various levels of product customizability imposed by the service provider. The proposed model is also compared with the commonly used First-Come-First-Served capacity allocation approach in terms of solution values. The design of the set of testing data used for the experiments is described as follows.

5.2.1. Design of the testing data

The customization environment in which the computational study is conducted is the one described in the worked example. However, to demonstrate the practical relevance of the experiments, the number of customers and the capacity of services are now increased to a realistic scale. Customer value is also generated from common pricing schemes found in online travel auctions. In travel auction websites, such as eBay Travel (http://www.ebay.com), Luxury Link (http://www.luxurylink.com), and Sky Auction (http://www.skyauction.com), a package to be sold has a "buy it now" price which is usually its regular retail price. A customer can purchase the package immediately at the regular retail price if unwilling to wait until the termination of the auctions. However, if the customer wants a bargain, it must participate in the auction. The final auction price is determined by the market competition at the termination of the auction. A package also has a reservation price. The reservation price is often unknown to the customers. In the design of the testing data, it is assumed that there is a regular retail price for each of the available services and the retail price for a package is the sum of the retail prices of services included in the package. The reservation price for a package is set to be 40% of its retail price since it is common in the online travel auctions that the termination price can be as low as 60% discount from the regular retail price. It is assumed that customers who enter the auction expect some discount. They are not interested in purchasing the package at a price higher than the regular retail price. It is also assumed that reservation prices are hidden from customers.

Therefore, customer value on a package are randomly drawn from a uniform probability distribution between zero and its regular retail price. 10 SCCC problem groups are generated with the customer number ranging from 100 to 1000. For each group, 10 instances are randomly generated. Service capacity is also allocated in proportion to the number of customers such that, for most of the instances, around 80%-90% of the customers will be awarded a feasible package. For all instances, feasible packages of a customer must contain one of DT, one of RT, and one of HL.

5.2.2. Experimental Results

The auction-based customization model is evaluated in terms of its value and revenue performance under various levels of product customizability imposed by the service provider. For the computational study, three levels of product customizability are considered. The three levels are defined by different base configurations: Config#1= {one of DT, one of RT, one of HL}, Config#2= {one of DT, one of RT, one of HL, one of ET}, Config#3= {one of DT, one of RT, one of HL, three of ET}. The numbers of services contained in the three configurations are 3, 4, and 6. The solutions computed under Config#1 are used as the baseline for comparison. For each group of the problem instances, optimal solution value under Config#1 is computed by solving the SCCC integer programming model presented in Section 3. The SCCC model is coded in ILOG Optimization Programming Languages (http://www-01.ibm.com/software/websphere/products/optimization/) and the 10 groups of problem instances are solved using ILOG CPLEX. The flow control of the iterative bidding is coded in the OPL (Optimization Programming Languages) script language. A desktop PC with 2.4G Intel CPU and 8 GB memory is used to run the experiments. For all problem instances including those contain 1000 customers, an iteration of bidding and winner determination takes less than one second. This level of responsiveness is sufficient for the vast majority of service customization applications.

Table 4 Customer value and provider revenue generated at different levels of package customizability

Group	Base-Config#1				Base-Config#2		Base-Config#3	
	(1) Optimal value	(2) Auction value	(3) Auction revenue	(4) First-come-first-served Value	(5) Auction value	(6) Auction revenue	(7) Auction value	(8) Auction revenue
1	\$211,705	\$210,535	\$174,380	\$166,420	\$110,080	\$96,585	\$73,085	\$57,890
2	\$421,970	\$418,100	\$333,470	\$326,270	\$221,990	\$197,225	\$129,240	\$102,740

3	$633,215	$618,880	$482,370	$493,610	$336,620	$294,485	$173,650	$137,970
4	$848,365	$846,295	$691,550	$662,700	$448,860	$397,790	$211,955	$166,980
5	$1,055,680	$1,039,410	$814,790	$816,505	$563,895	$503,075	$279,435	$219,160
6	$1,269,615	$1,245,415	$963,130	$954,235	$676,915	$599,330	$333,085	$259,360
7	$1,473,780	$1,453,190	$1,130,300	$1,128,480	$787,980	$696,880	$390,545	$303,210
8	$1,688,120	$1,680,505	$1,354,670	$1,294,280	$900,455	$802,365	$453,520	$353,030
9	$1,907,200	$1,889,915	$1,476,390	$1,497,350	$1,014,995	$899,630	$515,165	$402,940
10	$2,114,810	$2,101,835	$1,681,890	$1,655,410	$1,126,325	$994,805	$568,815	$443,030

The solutions computed by the auction-based customization model are compared against the optimal ones computed by ILOG CPLEX. The first column of Table 4 shows the average optimal solution values for the 10 groups of testing problems. The second column and the third column show the solution value and revenues computed by the auction-based customization model. All customers are assumed to adopt final-bid-repeating and $\varepsilon = 20$ for all bidding. It is observed that the auction-based customization model can achieve on average 98% of the optimal value across the 10 groups of problem instances. The average revenue computed is approximately 78% of the optimal value.

To evaluate the impacts of package customizability on customer value, the testing problems are solved again with Config#2 and Config#3. When conducting the iterative bidding, all biding packages which do not satisfy Config#2 and Config#3 configuration requirements are excluded at the bids screening stage. Column five and Column six of Table 4 show the solution value and revenues with Config#2. It is observed that, on average, the solution value decreases to 53% of that with Config#1 and revenues decrease to 59% of that with Config#1. If Config#3 is applied, solution value will decrease to 27% of that with Config#1 and revenues will decrease to 28% of that with Config#1. It is evident from the experimental results that reducing product customizability can significantly decrease customers' overall value and provider's revenue.

The proposed customization approach is also compared against the commonly used first-come-first-served capacity allocation policy. For example, "build your own package" applications in travel industry usually allocate a provider's capacity on a first-come-first-served basis combined with dynamic pricing strategies. This approach is easy to implement and performs reasonably well in terms of enhancing revenue when capacity supply and demand are balanced. However, when demand exhibits strong seasonality, auction-based policy may perform better. Again, take travel package customization as an example. During high seasons, a service provider's capacity is often over-demanded. A first-come-first-served policy allocates capacity according to the customer arrival order rather than customers' value. It does not maximize overall customer value. To compare the performance of an auction-based policy against that of a first-come-first-served capacity allocation policy, each policy is applied to the 10 groups of SCCC testing problems. In the first-come-first-served policy scenario, customers in an instance are first randomly ordered. Capacity is allocated according to their position in the sequence until no more customers can be satisfied. Column 4 of Table 4 shows the solution value of the first-come-first-served policy over the testing problems under Config#1. It is observed that first-come-first-served policy achieves on average 78% of the value obtained by the auction-based customization model.

6. Conclusion

Previous study has developed the mechanism which aligns the capability flexibilities between a customer and a supplier through bilateral negotiation (Chen and Tseng, 2007). The proposed approach optimizes customer value by exploiting the capacity flexibilities among a group of customers. To this end, this paper describes an auction-based multilateral negotiation model which coordinates customers' customization requirements such that the overall customer value is maximized. The approach is incentive-compatible in the sense that customers will follow the myopic best-response bidding strategy prescribed by the auction protocol. The results of our experiments indicate that customization solutions computed by the proposed model are very close to the optimal one. Revenue performance is also adequate when there is sufficient competition in the market. The scalability of the approach is tested by applying the model to instances with up to 1000 customers. Based on the testing results, it can be concluded that the auction-procedure combined with ILOG CPLEX is capable of dealing with service customization problems of realistic scales.

Throughout the paper, it has been assumed that the service provider's capacity is known and fixed. In the future we will study settings where a provider's capacity is expandable and subject to dynamic changes. For applying the proposed approach to manufacturing mass customization, more efforts are warranted in capacity representation under various product family architectures.

Acknowledgements

Thanks to Dr. Yong Zeng for his helpful discussions and suggestions.

References

Anderson, E.W., Fornell, C., & Rust, R. T. (1997). Customer satisfaction, productivity, and profitability: Differences between goods and services. Marketing Science, 16(2), 129–145.

Baker, A. D. (1991), "Manufacturing Control with a Market-Driven Contract Net", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY.

Blecker, T., Abdelkafi, N., Kreutler, G., & Kaluza, B. (2004). Auction-based Variety Formation and Steering for Mass Customization. Electronic Markets, 14(3), 232-242.

Cao, J., Wang, J., Law, K., Zhang, S., & Li, M. (2006). An interactive service customization model. Information and Software Technology, 48(4), 280-296.

Chen, S., & Tseng, M. M. (2007). Aligning demand and supply flexibility in custom product co-design. International Journal of Flexible Manufacturing Systems, 19(4), 596-611.

Chen, S., & Tseng, M. M. (2010). A Negotiation-Credit-Auction mechanism for procuring customized products. International Journal of Production Economics, 127(1), 203-210.

Chokshi, N., & McFarlane, D. (2008). A distributed architecture for reconfigurable control of continuous process operations. Journal of Intelligent Manufacturing, 19(2), 215-232.

Clarke, E. H. (1971). Multipart pricing of public goods. Public choice, 11(1), 17-33.

Da Silveira, G., Borenstein, D., & Fogliatto, F. S. (2001). Mass customization: Literature review and research directions. International Journal of Production Economics, 72(1), 1-13.

de Vries, S., & Vohra, R. V. (2003). Combinatorial auctions: a survey. INFORMS Journal on Computing, 15(3), 284-309.

Du, X., Jiao, J., & Tseng, M. M. (2001). Architecture of product family: Fundamentals and methodology. Concurrent Engineering: Research and Application, 9(4), 309–325.

Fujishima, Y., Leyton-Brown, K., & Shoham, Y. (1999). Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches, Sixteenth International Joint Conference on Artificial Intelligence (IJCAI) (pp. 548–553).

Groves, T. (1973). Incentives in teams. Econometrica: Journal of the Econometric Society, 617-631.

Guo, L., Hasegawa, T., Luh, P. B., Tamura, S., and Oblak, J.M. (1994), "Holonic Planning and Scheduling for a Robotic Assembly Testbed," in Proceedings of the 4th International Conference on CIM and Automation Technology, (Troy, NY, October), IEEE, New York, pp. 142-149.

Hohner, G., Rich, J., Ng, E., Reid, G., Davenport, A. J., Kalagnanam, J. R., et al. (2003). Combinatorial and quantity-discount procurement auctions benefit Mars, Incorporated and its suppliers. Interfaces, 33(1), 23-35.

IntelliQuest, (1990), Conjoint Analysis: A Guide for Designing and Integrating Conjoint Studies, Marketing Research Technique Series Studies, American Marketing Association, Market Research Division, TX.

Jiao, J., & Tseng, M. M. (1999). A methodology of developing product family architecture for mass customization. Journal of Intelligent Manufacturing, 10(1), 3-20.

Jiao, J., Simpson, T., & Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. Journal of Intelligent Manufacturing, 18(1), 5-29.

Lin, G. Y., and Solberg, J. J. (1992), "Integrated Shop Floor Control Using Autonomous Agents," IIE Transactions, Vol. 24, No. 3, pp. 57-71.

Mas-Colell, A., Whinston, M. D., & Green, J. R. (1995). Microeconomics: Oxford University Press.

Meyer, M. H., & Utterback, J. M. (1993). The product family and the dynamics of core capability. Sloan Management Review, 34(3), 29-47.

Milgrom, P. R., & Weber, R. J. (1982). A theory of auctions and competitive bidding. Econometrica: Journal of the Econometric Society, 1089-1122.

Moon, S. K., Shu, J., Simpson, T. W., & Kumara, S. R. T. (2010). A module-based service model for mass customization: service family design. IIE Transactions, 43(3), 153-163.

Moses, S., Gruenwald, L., & Dadachanji, K. (2008). A scalable data structure for real-time estimation of resource availability in build-to-order environments. Journal of Intelligent Manufacturing, 19(5), 611-622.

Parkes, D. C., & Kalagnanam, J. (2005). Models for iterative multiattribute procurement auctions. Management Science, 51(3), 435-451.

Parkes, D. (2006). Iterative combinatorial auctions. In P. Cramton, Y. Shoham & R. Steinberg (Eds.), Combinatorial Auctions (pp. 41–77): Cambridge, MA: MIT Press.

Sampson, S. E. (2001). Understanding service businesses: Applying principles of the unified services theory, Second edition. John Wiley & Sons, New York, New York.

Sandholm, T. (2002). Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence, 135(1-2), 1-54.

Sandholm, T., Suri, S., Gilpin, A., & Levine, D. (2005). CABOB: a fast optimal algorithm for winner determination in combinatorial auctions. Management Science, 51(3), 374-390.

Simpson, T.W., Maier, J.R.A.,& Mistree, F. (2001). Product platform design: Method and application. Research in Engineering Design, 13(1), 2–22.

Simpson, T. W. (2004). Product platform design and customization: Status and promise. AIEDAM, 18(1), 3–20.

Shaw, M. J. (1988), “Dynamic Scheduling in Cellular Manufacturing Systems: A Framework for Networked Decision Making,” Journal of Manufacturing Systems, Vol. 7, No. 2, pp. 83-94.

Shen, W., Wang, L., & Qi, H. (2006). Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 36(4), 563-577.

Tseng, M. M., and Jiao, J. (1996), “Design for Mass Customization,” Annals of the CIRP, Vol. 45, No. 1, pp. 153-156.

Tseng, M. M., & Du, X. (1998). Design by Customers for Mass Customization Products. CIRP Annals - Manufacturing Technology, 47(1), 103-106.

Tseng, M. M., & Jiao, J. (2001). Mass Customization. In G. Salvendy (Ed.), Handbook of industrial engineering: Technology and operations management (pp. 684-708): Wiley-Interscience.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. Research Policy, 24(3), 419–440.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance, 16(1), 8-37.

Voss, C. (1985). Operations Management in Service Industries and the Public Sector: Test and Cases, Wiley, Chichester.

Voss, C. A., & Hsuan, J. (2009). Service Architecture and Modularity. Decision Sciences, 40(3), 541-569.

Wang, C., Ghenniwa, H. H., & Shen, W. (2009). Constraint-based winner determination for auction-based scheduling. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 39(3), 609-618.

Wellman, M. P., Walsh, W. E., Wurman, P. R., & MacKie-Mason, J. K. (2001). Auction protocols for decentralized scheduling. Games and Economic Behavior, 35(1-2), 271-303.