

# Adaptive attribute selection for configurator design via Shapley value

YUE WANG AND MITCHELL M. TSENG

Advanced Manufacturing Institute, Hong Kong University of Science and Technology, Hong Kong

(RECEIVED April 4, 2010; ACCEPTED October 29, 2010)

## Abstract

Configurators have been generally accepted as important tools to elicit customers' needs and find the matches between customers' requirements and company's offerings. With product configurators, product design is reduced to a series of selections of attribute values. However, it has been acknowledged that customers are not patient enough to configure a long list of attributes. Therefore, making every round of configuring process productive and hence reducing the number of inputs from customers are of substantial interest to academic and industry alike. In this paper, we present an efficient product configuration approach by incorporating Shapley value, which is a concept used in game theory, to estimate the usefulness of each attribute in the configurator design. This new method iteratively selects the most relevant attribute that can contribute most in terms of information content from the remaining pool of unspecified attributes. As a result from product providers' perspective, each round of configuration can best narrow down the choices with given amount of time. The selection of the next round query is based on the customer's decision on the previous rounds. The interactive process thus runs in an adaptive manner that different customers will have different query sequences. The probability ranking principle is also exploited to give product recommendation to truncate the configuration process so that customers will not be burdened with trivial selection of attributes. Analytical results and numerical examples are also used to exemplify and demonstrate the viability of the method.

**Keywords:** Attribute Selection; Configurator; Probability Ranking Principle; Shapley Value

## 1. INTRODUCTION

Today, global marketplace is becoming increasingly competitive and diversified. Offering tailored products and in the mean time delivering products quickly to customers become major challenges for current manufacturing industry (McCutcheon et al., 1994). In this new situation, product configurator systems have been explored to handle the so-called "customization–responsiveness squeeze" phenomena, that is, providing tailored products with short delivery time (Schierholt, 2001; Salvador & Forza, 2004). Basically, a product configurator consists of a set of predefined attributes for customers to choose from. Some constraints on these attributes are also included to ensure that the selected attributes work compatibly. It takes a customer's specifications as input and the output is the customer's target product(s). With product configurators, product design is reduced to a series of selections of attribute values (Darr & Birmingham, 2000). In the studies of customers' decision-making processes, it has been

shown that customers have higher satisfaction with the outcomes of configuring process than traditional selection process (Kurniawan et al., 2003). Today, product configurators have not only been studied in academia but also been widely adopted in industries. It is reported that configurators have greatly improved manufacturers' responsiveness in product customization and reduced the cost of customer integration (Piller, 2004). Product configuration systems have been accepted as a viable strategy to bridge the gap between customers' needs and companies' offerings.

The history of product configurators can be traced back to 1970s. Digital Equipment Corporation (DEC) developed a program called R1 (later called XCON) in 1978 to configure VAX computer systems to customer specifications (McDermott et al., 1980). It was first put to use in 1980, and by 1986, it had processed 80,000 orders, achieving 95–98% accuracy. It was estimated to be saving DEC \$25 million a year by reducing the need to give customers free components when technicians made errors, by speeding up the assembly process, and by increasing customer satisfaction. Ever since then, a large number of configuration expert systems had been developed and put into use, such as Cossack (Frayman et al., 1987), BLADES (Elturky et al., 1986), and MICON

Reprint requests to: Yue Wang, Advanced Manufacturing Institute, RM2591, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: yacewang@ust.hk

(Birmingham et al., 1988). Because of the development of information technology in the past decade, it is possible for companies to acquire immediate information about customers' requirements and meet them by delivering customized products or related information efficiently. One of the most cited successful modern configurators cases is Dell Computer, which is able to deliver customized personal computers and notebooks within 1 week, with prices lower than its mass producing competitors. By using online configurator-based product customization system, Dell Computer has gained the so-called first-mover advantage and maintained high profitability and growth in a hypercompetitive industry for a long period.

Using configurators can streamline and automate the configuring process, reduce configuration errors, and enhance flexibility and responsiveness (Sabin & Weigel, 1998). However, there are still some limitations and shortcomings that have not been paid enough attention in previous research (Tseng & Piller, 2003). First, most product configurators still rely on fixed query sequences that entail sets of rigid interactive procedures. Although some configurators can capture customers' specifications in the order determined by customers, there is still no systematic study on adaptively eliciting customer needs according to customers' specifications in previous configuration steps. Therefore, the configuration process can be characterized as a one-way information flow from customers to designers, instead of an interactive and adaptive customer needs elicitation process. Second, product configuring process can be tedious and time consuming, especially when the product is complex. The configuring procedure may require seemingly redundant or trivial dialogues between customers and product development team. However, it has been widely acknowledged that customers are impatient to specify a long list of attributes (Enos, 2001). Therefore, it is necessary to elicit customers' needs in an efficient manner. Third, customers may have little knowledge about what a manufacturer is offering, including products features, design, limitations, cost, and delivery. Furthermore, they may even be unable to articulate their needs. Sometimes they are unclear about what they really want when facing a large number of options provided by companies. Customers may fail to understand or appreciate manufacturers' offerings (Simonson, 2005). They may find the configuration process unpleasant or even stressful (Schwartz, 2004). In summary, it is crucial for configurators to capture customers' specifications efficiently with less demand for customers' attention and time.

To overcome the limitations, we develop an approach for attribute selection task in product configuration process. Product configuring is considered as a sequential Q&A process. From designers' perspective, a customer's specifications to a product's attributes are unknown before the configuration process. Designers' objective is to elicit the customer's needs efficiently and accurately. During configuring process, designers can discover the customer's needs gradually based on the customer's partial specifications to some attributes. The more attributes the customer configures, the more information about the customer's needs is obtained. Thus, in product configuration process, designers' uncertainty about the customer's needs is decreasing. In

this sense, the configuration process is an *uncertainty elimination process* from designers' point of view. We want to eliminate the most uncertainty about a customer's needs in each configuration round so that designers can capture the customer's needs efficiently. In this paper, Shapley value is deployed to evaluate the relevance or usefulness level of each attribute. The method iteratively selects the most relevant attribute from the unspecified attributes pool and proposes it for the customer to configure. The selection of the next round query is based on the customer's decision in previous rounds. Thus, it solicits customers' specifications in an adaptive manner in the sense that different customers may have different query sequences. The customized one-to-one configuring procedure is presented and the final configuration can converge to a customer's target with fewer interactions between the customer and designers. In this sense, the configuration process is no longer a traditional process of passively accepting customers' specifications, but a bidirectional information flow procedure. This paper extends the methods in Wang and Tseng (2007, 2009) by presenting an analytical framework of attributes selection and product recommendation. Numerical studies are also conducted to verify the proposed approach.

The remainder of the paper is organized as follows. Related literature will be briefly reviewed in Section 2. In Section 3 we introduce the methodology for attribute selection from coalition game's point of view. A product recommendation approach is elaborated in Section 4. Section 5 presents the detailed procedure for attribute selection. A numerical example is presented in Section 6 to verify the proposed approach. Section 7 concludes the whole paper and points out some further research directions.

## 2. LITERATURE REVIEW

A configurator design can be considered as a reasoning task in its nature. Existing configuration design methodologies can be generally classified into rule-based, model-based, and case-based methodologies, depending on the reasoning techniques used (Sabin & Weigel, 1998).

In a rule-based system, design knowledge is codified as configuration rules or constraints. Most of early configuration systems fall in this category, like R1/XCON, Cossack, BLADES, and MICON. They derive solutions in a forward chaining manner. This kind of systems often suffers from the maintenance issues because of the lack of separation between domain knowledge and control strategy, especially when the configurator system is complex (Yu et al., 1998).

In a model-based system, design knowledge is contained in a system *model*, which consists of decomposable entities and interactions between their elements. The most extensively studied approach is probably constraint-based approach. Mittal and Frayman (1989) first treated configuration tasks as constraint satisfaction problems (CSPs). In this framework, a configuration task is to assign values to all the variables without violating any constraints. Mittal and Falkenhainer also modeled the configuration task as a sequence of dynamic

CSPs to cope with the change of attributes set during product configuration process. Both the parts and design alternatives were considered as variables in a CSP domain (Mittal & Falkenhainer, 1990). Not only compatibility constraints but also activity constraints were introduced into the extension to specify conditions under which a configurator can dynamically include or exclude component based on current selections. Sabin and Freuder (1996, 1998) proposed an idea that represented the configuration task as a new class of nonstandard CSP called composite CSP. It provided a more comprehensive and efficient basis for formulating and solving configuration problems (Sabin & Freuder, 1996, 1998). Gelle and Faltings developed a general framework to handle both continuous and discrete variables in configuration task that is called *mixed* and *conditional CSP* problems (Gelle & Faltings, 2003). A generative CSP framework was also defined that can support resource-balancing constraints (Mailharro et al., 1998; Stumptner et al., 1998). In this framework, component attributes were used to represent the resource demands and supplies.

In a case-based system, the basic idea is to compute the similarity between the input queries and existing product cases. Then the existing configurations that are likely to satisfy the input queries are refined according to customers' particular needs. As pointed by Wielinga and Schreiber (1997), the key issue in case-based configuration is how to retrieve the best configuration from the database and identify aspects that cause violation of constraints or requirements. Different methods have been used to tackle the issue. Rahmer and Voß (1996) used resource-oriented scheme to deal with case adaptation for telecooperation system. Löckenhoff and Messer (1994) presented detailed knowledge engineering based models for case-based configuration. It is a structure-oriented approach where a taxonomical structure of components is mapped onto a graph. The approach is also resource-oriented based on balancing of resource requesting and production model of components. Hüllermeier (1997) recast case-based reasoning task from combination optimization perspective. A combination optimization based approach was applied to solve configuration problems. Critiquing is also a method for case-based reasoning systems by using customers' feedback information (Burke, 2002; Burke et al., 1996, 1997). Customers only need to indicate a directional preference for a feature instead of inputting detailed feature value. Traditional critiquing only copes with single feature. Reilly et al. (2004) extended the technique to multiple features case, which is called compound critiques. They argued that the methods can offer explanatory benefits to help users better understand the structure of the recommendation mechanism and improve the performance of case-based recommendations. Tseng et al. (2005) also used case-based reasoning to construct a new bill of materials to reduce the time and cost of product design.

Table 1 gives an overview of different configuration methods, including the advantages and limitations of each approach. It is worth noting that the proposed method in this paper does not belong to any of the three categories. This paper does not present a holistic configuration design method. It is

**Table 1.** Overview of different configuration approaches

Approach	Advantages	Limitations
Rule-based configurator	<ul style="list-style-type: none"> <li>• Intuitive way of presenting design knowledge and configurations</li> <li>• Good at representing and evaluating heuristic relations</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult to maintain when the product is complex</li> </ul>
Model-based configurator	<ul style="list-style-type: none"> <li>• Good at handling constraints within the configuration rules</li> </ul>	<ul style="list-style-type: none"> <li>• Limited efficiency when solving constraints</li> </ul>
Case-based configurator	<ul style="list-style-type: none"> <li>• More efficient by using existing cases</li> </ul>	<ul style="list-style-type: none"> <li>• The results may be unreliable and inaccurate</li> <li>• Cannot adapt to new product configuration</li> </ul>

mainly concerned with the task of how to present attributes for customers to configure which is a major step for configuration tasks. In this sense, the proposed approach can be applied to any product configuration system.

### 3. ATTRIBUTE SELECTION AS A COALITION GAME

As stated in Section 1, it is crucial for configurators to capture customers' specifications efficiently, because customers are not patient enough to specify a long list of attribute. From a designer's point of view, the configuration design task is to select the attributes and the way of configuring them (Yu et al., 1998). During product configuration process, an attribute is presented for a customer to specify in each configuration round. A well-designed series of attributes could potentially shorten the lengthy iterative information exchange procedure. Therefore attribute selection serves as a critical factor for improving the efficiency of configurators. In this section we introduce a coalition game-based attribute selection criteria to accelerate the product configuration procedure.

#### 3.1. Preliminaries

In this section we will recast the attribute selection problem from game theory point of view, particularly coalition game. In a coalitional game, a set of players have certain payoff functions that represent the benefit achieved by different sub-coalitions in the game. In a formal language, a coalition game is defined by  $(N, v)$  where  $N$  is a set of players and  $v$  is a worth function of any subset of  $N$ , that is, the coalition. The meaning of the worth function is that if  $S$  is a coalition of players who agree to cooperate in a game, then  $v$  is the expected benefit they can get from the cooperation. Here,  $v$  is assumed to be monotone, that is,  $v(S') \geq v(S)$  for  $S \subset S'$  and  $v(\emptyset) = 0$ . Let

$\{x_i\}_{i \in N}$  be a partition of  $v(N)$ , that is,

$$v(N) = \sum_{i \in N} x_i,$$

where  $x_i$  is the benefit that player  $i$  can get from the cooperation. The marginal benefit for player  $i$  with respect to  $S \in N, i \notin S$  is  $\Delta_i(S) = v(S \cup \{i\}) - v(S)$ . Intuitively, the worth function and marginal benefit have the following properties:

- Dummy axiom: if player  $i$  is a dummy player, then  $x_i = 0$ . It means that if a player contributes nothing in the game, he should not receive any benefit.
- Symmetry axiom: if  $i \neq j$  such that  $\Delta_i(S) = \Delta_j(S)$  for all  $i, j \notin S$ , then  $x_i = x_j$ . It means that if two players contribute equally in the game, they should receive the same amount of benefit.
- Linearity axiom: if  $v(S) = v_1(S) + v_2(S)$  where  $v_1$  and  $v_2$  are also nonnegative monotone function satisfying  $v_1(\emptyset) = v_2(\emptyset) = 0$ , then  $x_i = x_i^1 + x_i^2$  where  $x_i^j$  is the cost share for  $v_j$ . This axiom means that two coalition games can be combined.

Then the Shapley value for the  $i$ th player is defined as the expectation  $E(X_i)$  where  $X_i = v((\sigma_1, \sigma_2, \dots, \sigma_i)) - v((\sigma_1, \sigma_2, \dots, \sigma_{i-1}))$  and  $(\sigma_1, \sigma_2, \dots, \sigma_i)$  is a permutation of  $(1, 2, \dots, i)$ , where  $\sigma_j$  can be any number in the set  $(1, 2, \dots, i)$ . For example,  $(\sigma_1, \sigma_2, \sigma_3)$  can be any element in the set of  $\{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$ . Shapley value is the expected marginal worth of a player over all the possible sets of coalitions. The expectation is calculated with respect to all the possible permutations with equal probability. Shapley (1953) proved that Shapley value is the only value that satisfies the three axioms.

### 3.2. Estimating attribute contribution by Shapley value

As stated above, customers are not patient enough to specify a long list of items. In addition, the attributes to be specified differ a lot in terms of the usefulness to reveal customers' needs. We want to estimate the usefulness of each attribute and ask customers to specify the most useful one. In this paper, the Shapley value of each attribute is used to measure the usefulness level.

The calculation of the Shapley value is usually time consuming for the attribute selection problem because it requires summing over all the possible subsets of coalitions and permutation on them. Keinan et al. (2004) presented an unbiased estimator to calculate the Shapley value by uniformly sampling from the permutation over  $N$ . Still, the estimator considers both large and small attributes sets to calculate the contribution values. Cohen et al. (2005) found that in practical problems, the contributions of players in a coalition formed by a subset of  $N$  are not as significant as that in coalition  $N$ , the coalition formed by all the players. They calculated the

contribution value of each player only with respect to  $N$ . Thus, the computational complexity is reduced because coalitions with size smaller than  $N$  are not taken into consideration. In this sense, the Shapley value of the  $i$ th attribute can be approximated by

$$E(X_i) \approx \frac{1}{n} \times \Delta_i(N \setminus \{i\}).$$

Because the parameter  $1/n$  is the same for all the attributes, we only need to consider  $\Delta_i(N \setminus \{i\})$  to select attribute. During each configuration round, the Shapley value of each unconfigured attribute will be calculated and the attribute with the biggest Shapley value is presented to for the customer to configure.

In this paper, we try to eliminate the most uncertainty about a customer's needs in each configuration round. Hence, the amount of uncertainty about customers' needs is adopted as the evaluation criterion. Because one of the key concerns in configurator design is to achieve product configuration quickly and accurately, the most informative attribute should be selected from the remaining unspecified attributes pool. Bearing this in mind, we take  $\Delta_i$  as the form of

$$\Delta_i(C) = \text{entropy}(C) - \text{entropy}(C|i), \tag{1}$$

where  $C$  is the set containing all the end products. Each product has certain probability to be a customer's target. After knowing the value of the  $i$ th attribute, the set  $C$  will be reduced to a subset  $C|i$ ;  $C$  is a discrete random variable with possible states  $1, \dots, n$ . Its entropy is defined as

$$\text{entropy}(C) = - \sum_{k=1}^n p_k \log_2 p_k,$$

where  $p_k$  is the probability that  $C$  is in state  $k$  (Shannon, 1948). The concept of entropy in information theory describes how much uncertainty there is in a signal or random event. Similarly, the entropy of  $C|i$  is

$$\begin{aligned} \text{entropy}(C|i) &= E_i(\text{entropy}(C|i = i_k)) \\ &= \sum_{i_k} P(i = i_k) \times \text{entropy}(C|i = i_k), \end{aligned} \tag{2}$$

where  $i_k$  is the  $k$ th alternative of the  $i$ th attribute.

In summary, from designers' perspective the Shapley value of an attribute is the amount of uncertainty that the attribute can eliminate after getting its value. Shapley value is deployed to select attribute for a customer to configure in each configuration round. The unspecified attribute that can eliminate the most uncertainty will be chosen for the customer to specify.

### 3.3. Estimation of parameters

To calculate the Shapley value of each attribute, we need to know the probability (conditional probability) that each end product meets a customer's needs. Given enough customers' choices data, the probabilities can be estimated from the data.

In this paper, we use the frequency that each end product being selected by customers to approximate the probability that the product will meet customers' needs. It can be proven that it is a maximum likelihood estimator of the probability. The estimation is

$$P(A_i = a_{ij} | A_k = a_{kl}) = \frac{|a_{ij} \cap a_{kl}|}{|a_{kl}|},$$

where  $a_{ij}$  and  $a_{kl}$  are the alternatives of attribute  $A_i$  and  $A_k$ , respectively;  $|a_{ij} \cap a_{kl}|$  is the number of cases with attribute  $A_i$  having value  $a_{ij}$  and in the mean time attribute  $A_k$  having value  $a_{kl}$  in existing configuration data; and  $|a_{kl}|$  is the number of cases with attribute  $A_k$  having value  $a_{kl}$ . To avoid zero probability caused by data sparsity, we apply the widely used smoothing technique by adding constants to both numerator and denominator (Cox, 1972). Then the estimation becomes

$$P(A_i = a_{ij} | A_k = a_{kl}) = \frac{|a_{ij} \cap a_{kl}| + 1}{|a_{kl}| + r}, \tag{3}$$

where  $r$  is the number of alternatives of attribute  $A_i$ . If  $|a_{ij} \cap a_{kl}| = |a_{kl}| = 0$ ,

$$P(A_i = a_{ij} | A_k = a_{kl}) = \frac{1}{r}. \tag{4}$$

It means that each alternative of attribute  $A_i$  is equally likely to be selected by the customer, that is, we assume the prior probability distribution of customers' choices is uniform. When sufficient data are obtained, the knowledge discovered from data will dominate the prior probability, because if  $|a_{ij} \cap a_{kl}|$  is large enough,

$$P(A_i = a_{ij} | A_k = a_{kl}) = \frac{|a_{ij} \cap a_{kl}| + 1}{|a_{kl}| + r} \approx \frac{|a_{ij} \cap a_{kl}|}{|a_{kl}|}. \tag{5}$$

Therefore, this estimation is actually the compromise between the knowledge discovered from the data and the prior belief about the probability distribution of customers' potential likelihood toward different attributes.

Both  $A_i$  and  $A_k$  can be generalized from one single attribute to a set of attributes. The idea is also to use the frequency of each event to approximate the true probability that we are interested in. By law of large number, the estimation will converge to the true probability when the data size increases. In this way, the likelihood of customers' choices dependency among different attributes will be quantified by conditional probabilities. The conditional probabilities will be deployed for the calculation of Shapley value.

#### 4. RECOMMENDATION OF PRODUCT CONFIGURATIONS

The presented product configuration method aims at helping customers find their target products quickly. To further improve the efficiency of product configuration, a product recommendation module is also used to present the most likely ac-

cepted product and thus truncate the product configuration process as early as possible. After a customer configures an attribute, certain number of products will be recommended. If the customer is satisfied with one of them, he can select it and terminate the configuration process ahead of time. Basically a recommendation method is to find the most likely accepted product configuration, that is, the one with the highest probability to meet a customer's requirements, based on the incomplete specifications. In this section, two basic questions regarding product recommendation will be answered, namely, how to calculate the probability of relevance for each product, that is, the product's probability of being a customer's target and in what order to present the recommendations if multiple products are recommended.

Let  $E$  be the set of configured attributes and  $Q$  be the set of attributes that are not specified. Let  $R$  denote the recommendation that is an instantiation of  $Q$  plus the specified attributes set  $E$ . Then we have

$$R = E \cup \left\{ \arg \max_Q P(Q|E) \right\}.$$

By Bayes rule,

$$P(Q|E) = \frac{|Q \cap E|}{|E|},$$

where  $|Q \cap E|$  is the number of configurations with attributes  $Q$  and  $E$  and  $|E|$  is the number of configurations with attributes  $E$ .

If we assume customers' choices among different attributes are independent, the recommendation can be simplified to

$$R = E \cup \left\{ \bigcup_{q_i \in Q} \arg \max_{q_i} P(q_i|E) \right\}, \tag{6}$$

where  $q_i$  is the  $i$ th unspecified attribute and  $P(q_i|E)$  can be estimated by

$$P(q_i|E) = \frac{|q_i \cap E|}{|E|}.$$

This independence assumption is often referred to as "local independence" and has been applied in marketing research (Kamakura & Wedel, 1995).

In this way, each end product's probability of meeting the customer's needs can be calculated. A very natural way to present the recommendations is based on the ranking of the probabilities, which is known as probability ranking principle (PRP; van Rijsbergen, 1979).

In information retrieval literature, expected search length is used to measure the efficiency of a retrieval approach (van Rijsbergen, 1979). It refers to the expected number of items that a customer has screened when he finds a target one. Let  $esl(S)$  represent the expected search length given a set of specifications  $S$ . Then,

$$esl(S) = E[i] = \sum_{i=1}^N p_i \times i,$$

where  $i$  is the search length and  $p_i$  represents the corresponding probability of occurrence of this search length. It is worth noting that  $p_i$  is a function of  $S$ . To simplify the notations, we just use  $p_i$  here.  $N$  is the number of possible search lengths and is bounded by the total number of product configurations. It can be proven that PRP guarantees the smallest expected search length and thus the highest efficiency.

**PROPOSITION 1.** *The PRP results in a minimal expected search length.* ■

**Proof:** Given a set of partial specifications  $S$  from a customer, let  $R_1 = (r_{11}, r_{12}, \dots, r_{1m})$  be the recommendation based on PRP with corresponding probabilities of meeting a customer's needs  $(P_{11}, P_{12}, \dots, P_{1m})$ , which satisfies  $P_{11} \geq P_{12} \geq \dots \geq P_{1m}$ . Let us consider another recommendation approach that proposes the same  $m$  recommendations in another order  $R_2 = (r_{21}, r_{22}, \dots, r_{2m})$  with probabilities of meeting a customer's needs  $(Q_{11}, Q_{12}, \dots, Q_{1m})$ , which is a permutation of  $(P_{11}, P_{12}, \dots, P_{1m})$ . The expected search length can be reformulated as

$$esl(S) = E[i] = \sum_{i=1}^m p_i \times i = \sum_{n=1}^m P(i \geq n),$$

where  $P(i \geq n)$  represents the probability that the first  $n - 1$  recommendations are not the target product (Durrett, 2003). Then we can yield

$$P_{PRP}(i \geq n) = \prod_{k=1}^{n-1} (1 - P_{1k}).$$

Similarly,

$$P_{others}(i \geq n) = \prod_{k=1}^{n-1} (1 - Q_{2k}).$$

Because  $P_{11} \geq P_{12} \geq \dots \geq P_{1m}$  and  $(Q_{11}, Q_{12}, \dots, Q_{1m})$  is a permutation of  $(P_{11}, P_{12}, \dots, P_{1m})$ , we can get

$$\begin{aligned} P_{PRP}(i \geq n) &= \prod_{k=1}^{n-1} (1 - P_{1k}) \leq \prod_{k=1}^{n-1} (1 - Q_{2k}) \\ &= P_{others}(i \geq n) \end{aligned}$$

for any  $n$ . By applying rearrangement inequality, we arrive at

$$\begin{aligned} esl_{PRP}(S) &= \sum_{n=1}^m P_{PRP}(i \geq n) \leq \sum_{n=1}^m P_{others}(i \geq n) \\ &= esl_{others}(S). \end{aligned}$$

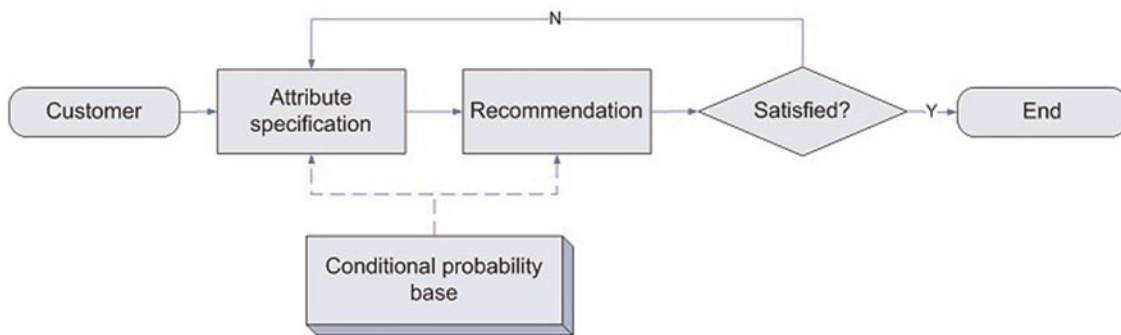
Because  $R_2$  is selected arbitrarily, the PRP results in the minimal expected search length comparing with any other recommendation approaches. ■

It is worth noting that we assume each customer has certain target product(s) in mind that is unknown to designers. However, designers can *guess* which configuration(s) may be the target product(s) based on the customer's partial specifications during product configuring process. This proposition states that the best strategy for designers to present recommendations is based on the probability of relevance. Customers' expected search length can be minimized in this way.

### 5. THE PROCESS OF ADAPTIVE ATTRIBUTE SELECTION FOR CONFIGURATOR DESIGN

The configuring process is an interactive and adaptive communication procedure. A schematic configuration process is shown in Figure 1. The conditional probabilities of customers' choices are estimated offline from existing configuration data. Once the probabilities are estimated, they will serve as the supporting base for product configuration procedure, particularly the attribute selection and recommendation. It sequentially selects the most relevant item from the remaining attributes pool for a customer to configure and recommendations will be presented afterward. The whole operation process can be summarized as follows:

1. Put all of the unspecified attributes into candidate set CS.
2. For each unspecified attributes  $q_i$ , calculate the conditional probability  $P(q_i|E)$  according to Eq. (3), where  $E$  is the set of all the combination of the remaining attributes.
3. For each unspecified attribute  $q_i$ , calculate its Shapley value  $E(X_i)$  based on Eq. (1).
4. Select the attribute with the biggest Shapley value and present it to the customer to configure.
5. Get the customer's specification and remove the attribute from the candidate set CS.



**Fig. 1.** The product configuration process. [A color version of this figure can be viewed online at journals.cambridge.org/aie]

- Calculate the probability that each end product meets the customer’s needs

$$P\left(\bigcup_i \{q_i\} | E\right) = \prod_i P(q_i | E)$$

according to Eq. (6) and present recommendations according to the ranking of the probabilities.

- Get the customer’s feedback toward the recommendations. If the customer is satisfied with one recommendation, end.
- If  $CS = \phi$ , end. Otherwise goes to step 3.

Note that we assume the customer’s target product is in current product family. Because the purpose of this paper is to elicit customer needs and provide customer’s target product efficiently, the case that no target product exists in current product family is not considered here (Fig. 1).

### 6. CASE STUDY

This section uses the configuration process of a simplified personal computer as an example to illustrate the ideas proposed in this paper. The set of components and their alternatives are listed in Table 2. Here we use a sextuple to represent one PC configuration. For example,  $\langle 1, 2, 2, 3, 2, 2 \rangle$  stands for the configuration containing the components A1, B2, C2, D3, E2, and F2. A survey was conducted in an East Asian university and 69 customers’ preferred configurations data were obtained. We want to use them to estimate the conditional probabilities  $P(q_i | E)$  that we need to run our method. However, the sample size is too small for the scale of the configuration task. To handle the data sparsity issue, we generated 1380 configuration data as training data to estimate the conditional probability and 345 testing data by a perturbative bootstrap approach. Bootstrap is a powerful data resampling method in statistics (Efron, 1979). It generates samples from an existing data set, where each sample is obtained by random sampling with replacement from the data set. Considering that customers may be flexible to some choices (Lilien et al., 1992), we add some variants when generating

**Table 2.** List of components and their alternatives for PC

Component	Code	Description
Processor (A)	A1	Intel Core 2 duo 3.16 GHz
	A2	Intel Core 2 duo 2.66 GHz
	A3	Intel Core 2 duo 2.8 GHz
	A4	Intel Pentium Dual-Core 2.6 GHz
	A5	Intel Core 2 quad processor 2.5 GHz
	A6	Intel Core 2 quad processor 2.6 GHz
Memory (B)	B1	2 GB DDR2
	B2	4 GB DDR2
	B3	6 GB DDR2
	B4	8 GB DDR2
Monitor (C)	C1	17-in. LCD
	C2	19-in. LCD
	C3	20-in. LCD
	C4	22-in. LCD or above
Hard disk (D)	D1	160 GB
	D2	250 GB
	D3	500 GB
	D4	750 GB
Disk driver (E)	E1	16X DVD+/-RW*
	E2	Blu-ray disk
	E3	Blu-ray disk + 16X DVD+/-RW*
Display card (F)	F1	Intel GMA 3100
	F2	512-MB NVIDIA GeForce 9800GT
	F3	256-MB ATI Radeon HD 3450 LE
	F4	256-MB ATI Radeon HD 3650
	F5	512-MB ATI Radeon HD 4670

resamples. We call the resampling method perturbative bootstrap.

Before conducting the resampling method, each attribute alternative’s substitutes are identified according to the similarity of performance, price and other characteristics. To make the algorithm more general, let us suppose a product’s attributes set is  $\{A_i : 1 \leq i \leq k\}$ , where  $k$  is the number of attributes. Each attribute  $A_i$  has a set of alternatives  $a_{ij} : 1 \leq i \leq k, 1 \leq j \leq n_i$ , where  $n_i$  is the number of alternatives for the  $i$ th attribute. Each attribute alternative  $a_{ij}$  has a substitute set  $\bar{a}_{ij}$  determined beforehand. The substitute set  $\bar{a}_{ij}$  can be empty if there is no proper substitute for  $a_{ij}$ . The detailed data resampling algorithm is as follows:

```

For  $l = 1$  to  $m // m$  is the data size of the original data set
  For  $i = 1 : k$ 
    For  $j = 1 : n_i$ 
       $u = \text{rand}(0,1)$ ; //generating a random number following uniform(0,1)
      if  $u > h$  and  $\bar{a}_{ij}$  is not empty,  $a_{ij}^*$  equals to one value in  $\bar{a}_{ij}$  with probability  $\frac{1}{|\bar{a}_{ij}|}$ 
        where  $|\bar{a}_{ij}|$  is the cardinal of set  $\bar{a}_{ij}$  and  $h$  is a predetermined threshold;
      else  $a_{ij}^* = a_{ij}$ ;
    end
  end
end
    
```

In this numerical example,  $h$  is set to 0.9. After running the algorithm once, a new set of configuration data are generated. It has the same size as the original data set. We repeat the process 25 times and generate 1380 training data and 345 testing data. Because of the limit of pages, the conditional probabilities estimated from the training data set are omitted here.

**6.1. A product configuration example by applying the proposed approach**

Suppose a new customer’s target configuration is  $\langle 1, 2, 2, 3, 2, 2 \rangle$  that is unknown to designers before the product configuring process.

STEP 1. The probabilities  $P(q_i|E)$  are calculated according to Eq. (3). Because no attributes are specified at the beginning,  $P(q_i)$  is used instead.

STEP 2. The Shapley values of each attribute are calculated according to Eq. (1). For example,

$$\begin{aligned} \Delta_A(S) &= \text{entropy}(S) - \text{entropy}(S|A) \\ &= -\sum_{i=1}^{69} 0.014 \times \log 0.014 \\ &\quad - \left( -0.246 \sum_{i=1}^{17} 0.059 \times \log 0.059 \right. \\ &\quad - 0.159 \times \sum_{i=1}^{11} 0.091 \times \log 0.091 \\ &\quad - 0.246 \times \sum_{i=1}^{17} 0.059 \times \log 0.059 \\ &\quad - 0.087 \times \sum_{i=1}^6 0.167 \times \log 0.167 \\ &\quad - 0.159 \times \sum_{i=1}^{11} 0.091 \times \log 0.091 \\ &\quad \left. - 0.101 \times \sum_{i=1}^7 0.143 \times \log 0.143 \right) \\ &= 2.48. \end{aligned}$$

Similarly, we can get the Shapely values of other attributes.

$$\begin{aligned} \Delta_B(S) &= \text{entropy}(S) - \text{entropy}(S|B) = 1.76; \\ \Delta_C(S) &= \text{entropy}(S) - \text{entropy}(S|C) = 1.75; \\ \Delta_D(S) &= \text{entropy}(S) - \text{entropy}(S|D) = 1.69; \\ \Delta_E(S) &= \text{entropy}(S) - \text{entropy}(S|E) = 1.53; \\ \Delta_F(S) &= \text{entropy}(S) - \text{entropy}(S|F) = 2.26. \end{aligned}$$

As a result, we present the attribute A that has the highest Shapley value to the customer.

STEP 3. After getting the customer’s specification A1, we can present the corresponding recommendation just by

checking the conditional probability table. The independence assumption stated in the last session is used here to present the recommendation. The following recommendation can be yielded:

$$\begin{aligned} \arg \max_B P(B_i|A = A1) &= B2; & \arg \max_C P(C_i|A = A1) &= C3; \\ \arg \max_D P(D_i|A = A1) &= D3; & \arg \max_E P(E_i|A = A1) &= E2; \\ \arg \max_F P(F_i|A = A1) &= F3. \end{aligned}$$

Because the output recommendation  $\langle 1, 2, 3, 3, 2, 3 \rangle$  differs from the target configuration  $\langle 1, 2, 2, 3, 2, 2 \rangle$ , further processing is required.

STEP 4. The Shapley values given that A1 is selected are calculated and result to

$$\begin{aligned} \Delta_{B|A1}(S) &= \text{entropy}(S|A1) - \text{entropy}(S|A1, B) = 1.45; \\ \Delta_{C|A1}(S) &= \text{entropy}(S|A1) - \text{entropy}(S|A1, C) = 1.61; \\ \Delta_{D|A1}(S) &= \text{entropy}(S|A1) - \text{entropy}(S|A1, D) = 1.55; \\ \Delta_{E|A1}(S) &= \text{entropy}(S|A1) - \text{entropy}(S|A1, E) = 1.16; \\ \Delta_{F|A1}(S) &= \text{entropy}(S|A1) - \text{entropy}(S|A1, F) = 2.26. \end{aligned}$$

The highest Shapley value is the one for the attribute F, which we therefore present to the customer to configure.

STEP 5. After getting the customer’s specification F2, the following recommendation can be reached by checking the conditional probability table.

$$\begin{aligned} \arg \max_B P(B_i|A = A1, F2) &= B2; \\ \arg \max_C P(C_i|A = A1, F2) &= C3; \\ \arg \max_D P(D_i|A = A1, F2) &= D3; \\ \arg \max_E P(E_i|A = A1, F2) &= E3. \end{aligned}$$

Thus,  $\langle 1, 2, 3, 3, 3, 2 \rangle$  is recommended.

STEP 6. Because the recommendation is still not satisfactory, the previous attribute selection process should be repeated until the recommendation meets the requirement.

In summary, the configuring process for this customer is shown in Table 3. After three configuration rounds, the customer gets his target PC. Section 6.2 presents the experiment results to show the advantage of the proposed method.

**6.2. Performance comparison**

In this section, we compare the performance of the proposed approach with other attribute selection and recommendation methods. Let “FixSeq + PRP” represent the method using fixed query sequence and PRP based recommendation. Ac-



**Table 3.** The specification defining process for the customer with preference  $\{A1, B2, C2, D3, E2, F2\}$ 

Specification Round	Information Gain	Proposed Item	Recommendation	Target
1	$E_A = 2.48, E_B = 1.76,$ $E_C = 1.75, E_D = 1.69,$ $E_E = 1.53, E_F = 2.26$	CPU (A)	$\langle 1, 2, 3, 3, 2, 3 \rangle$	No
2	$E_B = 1.45, E_C = 1.61,$ $E_D = 1.55, E_E = 1.16,$ $E_F = 2.26$	Display card (F)	$\langle 1, 2, 3, 3, 3, 2 \rangle$	No
3	$E_B = 0.92, E_C = 0.92,$ $E_D = 0, E_E = 0.92$	Monitor (C)	$\langle 1, 2, 2, 3, 2, 2 \rangle$	Yes

ording to the number of attributes, there are  $6! = 720$  possible query sequences for this PC configurator. We calculate the average configuration rounds of all the possible sequences (AverFixSeq + PRP). The results of two arbitrarily selected sequences (“FixSeq1 + PRP” and “FixSeq2 + PRP”) are also presented for comparison. The other approach uses the Shapley value based attributes selection method addressed in this paper but recommends the configuration randomly (Shapley + Rand). The proposed approach is abbreviated as “Shapley + PRP.”

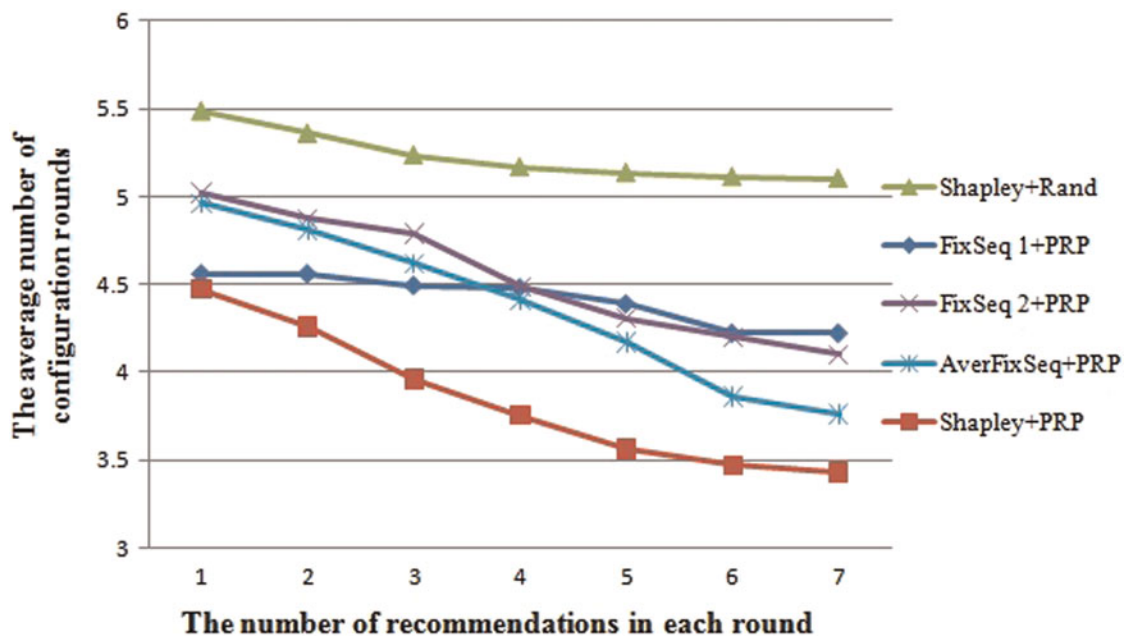
The products in the testing set generated by perturbative bootstrap are used as customers’ targets. In previous section’s example, only one recommendation is provided in each round. In this numerical analysis, multiple products are recommended according to their probabilities of relevance. If a customer’s target PC is in the set of recommendations, the process will end and the corresponding configuration rounds will be recorded. The number of configuration rounds is used as the measure of efficiency. It can be anticipated that if the whole framework performs better, then fewer rounds of com-

munications will occur. Figure 2 shows the experiment results under different approaches.

The  $x$  axis represents the number of recommendations in each round and  $y$  axis is the number of recommendations rounds needed for the customer to find the target product. Because there are six attributes altogether in this example, the worst case requires six configuration rounds. We can see that the “Shapley + PRP” approach proposed in this paper outperforms others. When more recommendations are presented, the configuration rounds are also decreased, because bigger recommendation set is more likely to contain the target product. The experiment results show that our approach provides a promising direction of improving the efficiency of product configuration.

## 7. CONCLUSION

This paper recasts the attribute selection task in configurator design from game theory’s point of view. Shapley values are adopted to measure the usefulness of different attributes.



**Fig. 2.** A comparison of the configuration approaches. [A color version of this figure can be viewed online at [journals.cambridge.org/aie](https://journals.cambridge.org/aie)]

The most relevant attribute is selected for customers to configure during product configuring procedure. The main contributions are as follows:

1. A product configuring process is treated as a sequential decision making procedure. In each configuring round, the most uncertainty about a customer's needs is eliminated. The interactive process runs in an adaptive manner in the sense that different customers will have different query sequences. This offers a brand new perspective for us to understand configurator issues in product customization context.
2. PRP is adopted for product recommendation. It could shield customers from the tedious process of screening and making a selection from a vast number of products and thus overcome the information overload issue. Analytical results show that PRP is optimal with respect to expected search length. The efficiency of matching between demand and supply is thus enhanced.

The presented method also has some limitations and can be enriched along several dimensions. It functions well for customers who have enough expertise and can clearly configure each attribute. For customers who only have vague functional requirements, there are no links between the customers' needs and the detailed attributes in this framework. The configuration task is hard to conduct. How to incorporate customer needs in fuzzy functional requirements form into the configuration task is a future research direction. In addition, this paper assumes the product configuration space is fixed. Innovation and evolution in a product family are not considered. Apparently, it is not profitable to start from scratch again to collect data and implement the approach addressed in this paper. One potential solution is to adapt previous configuration data to the updated product family via some econometric methods. Another direction is to improve the computational complexity of the configuration design task. If the product contains  $m$  attributes and each attribute has  $n$  attribute alternatives, then the computational complexity of the proposed attribute selection task is  $O(m^2n)$ . How to improve the computational complexity remains to be a practical and significant research issue.

## ACKNOWLEDGMENTS

This research is supported by the Hong Kong Research Grants Council (RGC CERG HKUST 620308 and 620609).

## REFERENCES

- Birmingham, W., Brennan, A., & Siewiorek, D. (1988). MICON: a single board computer synthesis tool. *IEEE Circuits and Devices Magazine* 4(1), 37–46.
- Burke, R. (2002). Interactive critiquing for catalog navigation in E-Commerce. *Artificial Intelligence Review* 18(3–4), 245–267.
- Burke, R., Hammond, K., & Young, B. (1996). Knowledge-based navigation of complex information spaces. *Proc. 13th National Conf. Artificial Intelligence*, pp. 462–468. Portland, OR: AAAI Press/MIT Press.
- Burke, R., Hammond, K., & Young, B. (1997). The FindMe approach to assisted browsing. *Journal of IEEE Expert* 12(4), 32–40.
- Cohen, S., Dror, G., & Ruppin, E. (2007). Feature selection via coalitional game theory. *Neural Computation* 19, 1939–1961.
- Cox, D.R. (1970). *The Analysis of Binary Data*. London: Methuen.
- Darr, T., & Birmingham, W. (2000). Part-selection triptych: a representation, problem properties and problem definition, and problem-solving method. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 14(1), 39–51.
- Durrett, R. (2003). *Probability: Theory and Examples*, 3rd ed. New York: Thomson Learning/Brooks–Cole.
- Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics* 7(1), 1–26.
- Elturky, F., & Nordin, R. (1986). BLADES: an expert system for analog circuit design. *Proc. IEEE Symp. Circuit and System*, pp. 552–555, San Jose, CA.
- Enos, L. (2001). Report: five keys for e-tail success. *E-Commerce Times*. Accessed at <http://www.ecommercetimes.com/story/7743.html>
- Frayman, F., & Mittal, S. (1987). Cossack: a constraint based expert system for configuration task. In *Knowledge-Based Expert Systems in Engineering: Planning and Design* (Sriram, D., & Adey, R., Eds.), pp. 143–166. Boston: Computational Mechanics Publication.
- Gelle, E., & Faltings, B. (2003). Solving mixed and conditional constraint satisfaction problems. *Constraint* 8(2), 107–141.
- Hüllermeier, E. (1997). *Case-based search techniques for solving configuration problems*. Accessed at <http://citeseer.ist.psu.edu/old/79018.html>
- Kamakura, W.A., & Wedel, M. (1995). Life-style segmentation with tailored interviewing. *Journal of Marketing Research* 32, 308–317.
- Keinan, A., Sandbank, B., Hilgetag, C., Meilijson, I., & Ruppin, E. (2004). Fair attribution of functional contribution in artificial and biological networks. *Neural Computation* 16(9), 1887–1915.
- Kurniawan, S., Tseng, M., & So, R. (2003). Consumer decision making process in mass customization. *Proc. 2003 World Congress on Mass Customization and Personalization*, p. 38, Munich.
- Lilien, G., Kotler, P., & Moorthy, K. (1992). *Marketing Model*. Englewood Cliffs, NJ: Prentice–Hall.
- Löckenhoff, C., & Messer, T. (1994). Configuration. In *CommonKADS Library for Expertise Modelling* (Breuker, J., & Van de Velde, W., Eds.), pp. 197–212. Amsterdam: IOS Press.
- Mailharro, D. (1998). A classification and constraint-based framework for configuration. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 12(4), 383–397.
- McCutcheon, D., Raturi, A., & Meredith, J. (1994). The customization-responsiveness squeeze. *Sloan Management Review* 35(2), 89–99.
- McDermott, J. (1980). R1: an expert in the computer systems domain. *Proc. 1st Annual National Conf. Artificial Intelligence*, pp. 269–271, Stanford University.
- Mittal, S., & Falkenhainer, B. (1990). Dynamic constraint satisfaction problems. *Proc. American Association for Artificial Intelligence*, pp. 25–32, Boston.
- Mittal, S., & Frayman, F. (1989). Towards a generic model of configuration task. *Proc. Int. Joint Conf. Artificial Intelligence*, pp. 1395–1401. San Mateo, CA: Morgan Kaufmann.
- Piller, F., & Moeslein, K. (2004). Does mass customization pay? An economic approach to evaluate customer integration. *Production Planning & Control* 15(4), 435–444.
- Rahmer, J., & Voß, A. (1996). *Case-based reasoning in the configuration of telecooperation systems*, pp. 93–98, AAAI Technical Report FS-96-03. Menlo Park, CA: AAAI Press.
- Reilly, R., McCarthy, K., McGinty, L., & Smyth, B. (2004). Dynamic critiquing. *Advances in Case-Based Reasoning*, pp. 763–777. Berlin: Springer.
- Sabin, D., & Freuder, E. (1996). Configuration as composite constraint satisfaction. *Proc. 1st Artificial Intelligence and Manufacturing Research Planning Workshop*, pp. 153–161. Menlo Park, CA: AAAI Press.
- Sabin, M., & Freuder, E. (1998). *Detecting and resolving inconsistency and redundancy in conditional constraint satisfaction problems*. Accessed at <http://citeseer.ist.psu.edu/sabin98detecting.html>
- Sabin, D., & Weigel, R. (1998). Product configuration frameworks—a survey. *IEEE Intelligent Systems* 13, 42–49.

- Salvador, F., & Forza, C. (2004). Configuring products to address the customization-responsiveness squeeze: a survey of management issues and opportunities. *International Journal of Production Economics* 91(3), 273–291.
- Schierholt, K. (2001). Process configuration: combining the principles of product configuration and process planning. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 15(5), 411–424.
- Schwartz, B. (2004). *The Paradox of Choice: Why More Is Less*. New York: ECCO.
- Shannon, C.E. (1948). A mathematical theory of communication. *Bell System Technical Journal* 27(7), 379–423.
- Shapley, L. (1953). A value for n-person games. In *Contributions to the Theory of Games*, Vol. 1 (Kuhn, H.W., & Tucker, A.W., Eds.). Princeton, NJ: Princeton University Press.
- Simonson, I. (2005). Determinants of customers' responses to customized offers: conceptual framework and research propositions. *Journal of Marketing* 69(1), 32–45.
- Stumptner, M., Friedrich, G., & Haselböck, A. (1998). Generative constraint-based configuration of large technical systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 12(4), 307–320.
- Tseng, H., Chang, C., & Chang, S. (2005). Applying case-based reasoning for product configuration in mass customization environments. *Expert Systems With Applications* 29(4), 913–925.
- Tseng, M., & Piller, F. (2003). New direction for mass customization. In *The Customer Centric Enterprise* (Piller, F., & Tseng, M., Eds.), pp. 519–535. Berlin: Springer.
- van Rijsbergen, C.J. (1979). *Information Retrieval*. London: Butterworths.
- Wang, Y., & Tseng, M.M. (2008). Incorporating probabilistic model of customers' preferences in concurrent engineering. *Annals of the CIRP* 58(1), 137–140.
- Wang, Y., & Tseng, M.M. (2009). Attribute selection for configurator design based on Shapley value. *Proc. ASME 2009 Int. Design Engineering Technical Conf., Computers and Information in Engineering Conf. (IDETC/CIE 2009)*, Paper No. DETC2009-86904.
- Wielinga, B., & Schreiber, G. (1997). Configuration-design problem solving. *IEEE Expert: Intelligent Systems and Their Applications* 12(2), 49–56.
- Yu, B., Skovgaard, H., & Baan Frond Office Systems. (1998). A configuration tool to increase product competitiveness. *IEEE Intelligent Systems* 13(July/August), 34–41.

---

**Yue Wang** received his PhD from the Department of Industrial Engineering and Logistics Management at the Hong Kong University of Science and Technology and his BS and MS degrees in electronic engineering from Peking University, Beijing. Dr. Wang's research interest is focused on engineering design and manufacturing, artificial intelligence, and applied statistics.

**Mitchell M. Tseng** is a Professor and the Director of the Advanced Manufacturing Institute and Zhejiang Advanced Manufacturing Institute of Hong Kong University of Science and Technology. Prof. Tseng joined Hong Kong University of Science and Technology as the founding department head of industrial engineering in 1993 after holding executive positions at Xerox and DEC. He previously held faculty positions at the University of Illinois at Urbana–Champaign and the Massachusetts Institute of Technology. Dr. Tseng received MS and PhD degrees in industrial engineering from Purdue University and a BS degree from National Tsing Hua University. He is a fellow of the International Academy of Production Engineers (CIRP) and ASME.