Data-Driven Decision Tree Classification for Product Portfolio Design Optimization

Conrad S. Tucker
e-mail: ctucker4@uiuc.edu

Harrison M. Kim[1]
Assistant Professor
Mem. ASME
e-mail: hmkim@uiuc.edu

Department of Industrial and Enterprise Systems Engineering,
University of Illinois at Urbana-Champaign,
104 S. Mathews Avenue,
Urbana, IL 61801

The formulation of a product portfolio requires extensive knowledge about the product market space and also the technical limitations of a company's engineering design and manufacturing processes. A design methodology is presented that significantly enhances the product portfolio design process by eliminating the need for an exhaustive search of all possible product concepts. This is achieved through a decision tree data mining technique that generates a set of product concepts that are subsequently validated in the engineering design using multilevel optimization techniques. The final optimal product portfolio evaluates products based on the following three criteria: (1) it must satisfy customer price and performance expectations (based on the predictive model) defined here as the feasibility criterion; (2) the feasible set of products/variants validated at the engineering level must generate positive profit that we define as the optimality criterion; (3) the optimal set of products/variants should be a manageable size as defined by the enterprise decision makers and should therefore not exceed the product portfolio limit. The strength of our work is to reveal the tremendous savings in time and resources that exist when decision tree data mining techniques are incorporated into the product portfolio design and selection process. Using data mining tree generation techniques, a customer data set of 40,000 responses with 576 unique attribute combinations (entire set of possible product concepts) is narrowed down to 46 product concepts and then validated through the multilevel engineering design response of feasible products. A cell phone example is presented and an optimal product portfolio solution is achieved that maximizes company profit, without violating customer product performance expectations.
[DOI: 10.1115/1.3243634]

1 Introduction

The emergence of highly competitive markets in the global marketspace has forced companies to reevaluate strategies in ensuring sustainable business endeavors. Attempts to satisfy a wide array of customers quickly and efficiently have led to the concept of product customization, wherein enterprise decision makers strive to better cater to the needs of their customers through a wider array of products to choose from [1]. While this approach is beneficial to the consumer, such design and manufacturing decisions can lead to adverse effects from a manufacturing, distribution, and marketing cost standpoint. Companies continue to place a high premium on the methodologies needed to ensure that mass customization decisions lead to increased or, at the very least, consistent profit margins.

Attempts to mitigate the added costs of mass customization are in part achieved through the product family paradigm [2], wherein products that satisfy the individual product functionality requirements dictated by customer preference are designed around a shared and efficient product architecture. The term *product architecture* is frequently defined as the set of modules/components wherein product variants evolve [3–6]. Commonality among product variants can translate into lower manufacturing costs associated with highly differentiated products through economies of scale [7,8]. The challenges facing enterprise decision makers in the product portfolio development process are multifaceted and include identifying candidate product concepts that have the greatest probability of market success. Attempts to search every possible product concept may be impractical in real life design processes, especially when *first to market* may create tremendous competitive advantages in the market space.

Our approach to product portfolio formulation takes large data sets of customer preference data and extracts meaningful product attribute information to help guide the actual product design and development process. The overall objective of maximizing company profit is realized when a feasible set of product variants is presented in the final solution process. The reduction in resources in this limited and highly efficient narrowing of product concepts will be demonstrated through a cell phone example, where an entire product concept generation space of 576 (exhaustive combination of product attributes) product concepts is narrowed to only 46 through a decision tree data mining approach. The generated product designs are then subsequently tested for engineering feasibility. This is formulated as a multilevel optimization problem, where the generated predictive product concepts are first translated into functional specifications and set as targets at the engineering level for design validation. A feasible product design is therefore defined as one in which all customer preferences are satisfied, without violating engineering design constraints.

This paper is organized as follows. This section provides a brief motivation and background. Section 2 describes previous works closely related to the current research. Section 3 describes the methodology. The methodology is demonstrated in Sec. 4 through a cellular phone portfolio design example. Section 5 presents the results and discussion. Section 6 concludes the paper.

2 Related Work

2.1 Customer Knowledge Acquisition Approaches. There are several well established methods for translating customer requirements into tangible engineering design targets in the product development process. We will briefly review several well known

[1]Corresponding author.

Contributed by the Engineering Informatics Division of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received December 20, 2007; manuscript received February 16, 2009; published online November 2, 2009. Paper presented at the 2007 IDETC, Las Vegas, NV. Assoc. Editor: K. Law.

approaches in Secs. 2.1.1–2.1.3, and in Sec. 2.1.4, we highlight the strengths of data mining as an alternative approach for acquiring customer product requirements.

2.1.1 Quality function deployment. The quality function deployment (QFD) is a design and development methodology that attempts to acquire customer requirements (CRs) otherwise known as the voice of the customer (VOC) and translate them into functional engineering targets [9]. A conventional approach to customer requirement acquisition is through focus group interviews or conducting surveys of a sample of current or future customers [10]. Corresponding weights are assigned to each customer requirement based on an importance rating indicated by a customer [11,12]. A QFD matrix is often used to depict the interdependence between customer requirements and the engineering metrics (EMs) and to aid in brainstorming and designing the optimal product to address customer needs. QFD driven product development methodologies suggest that QFD is well suited for out of the box solutions to customer needs due to the fact that engineering design features are evaluated based on their positive and negative contributions to solving the product design problem [9]. The design of the QFD matrix also makes it easier to benchmark a particular design solution against competing brands.

2.1.2 Conjoint analysis. Conjoint analysis (CA) has been used successfully in marketing to determine how customers value combinations of different product attributes/features [9]. In this approach, a target customer group is identified for the study and presented with a set of attributes (survey format or prop cards [13]), each with different levels (attribute ranges) [14]. *Part-worth utilities* are estimated based on customer importance ranking of individual product attributes. The resulting utility function is used to evaluate customer preferences for different attribute combinations. Although conjoint analysis application areas can range from human psychology to advertising, attempts to directly incorporate it into engineering design optimization and product development have been investigated [15–18]. These conjoint analysis based product development methodologies highlight the ability of the approach to quantify specific product attribute levels in new product development. However, this approach is primarily survey driven and therefore as the attribute space becomes large, preserving the quality of the model becomes a challenge.

2.1.3 Discrete choice analysis. Discrete choice analysis (DCA) is the modeling methodology of consumer choice behavior from a set of mutually exclusive collective exhaustive alternatives [19]. DCA incorporates probabilistic choice theory in determining which product a customer is most likely to choose based on expected utility [20]. In engineering design and development, modeling product demand can therefore be based on a customer utility function model that incorporates unknown parameter estimates and unobservable customer utility components [21–23]. Instances of discrete choice analysis include the probit model and logit models (multinomial, mixed, nested, etc.) to name but a few. In product development, applications are focused on creating consumer choice models either through stated or revealed data [23–26]. Many of these studies reveal the strengths of DCA in quantifying the market share of different brands of products given a set of attributes. The DCA model presents enterprise decision makers and design engineers with relevant probability measures of choosing one product over another based on product characteristics. A challenge of using DCA that is reported in the literature is that of *multicollinearity*, where it becomes difficult to generate a DCA model due to the presence of highly correlated attributes.

2.1.4 Data mining and knowledge extraction in product development. A few fundamental differences between data mining techniques and those discussed in Secs. 2.1.1–2.1.3 are that unlike the QFD and CA techniques that are highly dependent on *stated preference* data acquired through close customer interaction, data mining applications can also deal with *revealed preference* data (real customer purchase behavior that is captured through purchase transactions (in-store, online, etc.) [26]. The absence of the direct customer interaction constraints allows larger data sets to be analyzed through data mining techniques that, in turn, may more accurately reflect the individual preferences of a wider array of customers. In relation to attribute importance, both the QFD and CA extract attribute importance (or relevance) by either requiring customers to rank individual product attributes or rank product concepts as a whole. This added requirement to rank product alternatives or attributes may also limit the size of the data set that can be analyzed or the speed and efficiency by which new products can be designed. In the predictive data mining technique presented in this work, customers are not required to rank product alternatives or attributes. Instead they are only required to select the combination of attributes and price that most closely meet their needs. Product attribute importance is therefore identified during the decision tree model generation by employing the *gain ratio* attribute evaluation metric discussed later in Sec. 3.1.3. Therefore no prior attribute ranking is assumed or required.

The incorporation of data mining techniques in product portfolio development is emerging as a well-founded approach to extracting and analyzing relevant customer information. Kusiak and Smith highlighted several key areas in industrial and manufacturing design processes, where data mining techniques could potentially have great benefits [27]. In the context of product portfolio development, the application of data mining clustering techniques in the design of modular products has also been investigated. Moon et al. used data mining to represent the functional requirements of customers and used fuzzy clustering techniques to determine the module composition of a product architecture [1].

Nanda et al. proposed a product family ontology development methodology (PFODM) that utilizes a formal concept analysis approach in the design of product families [28]. This approach incorporates existing knowledge of the product family in generating a hierarchical conceptual clustering of design components.

Data mining predictive techniques are investigated by Tucker and Kim to extract knowledge from large customer product preference data sets. This approach incorporates customer input at the early stages of the design process by directly integrating customer predictive preferences with engineering design through a data mining Naive Bayes predictive technique [29].

The decision tree generation approach that we adopt in this work presents an enterprise decision maker with product concepts that are determined to be the best indicators for market success [20]. This prediction is based on the C4.5 machine learning algorithm that predicts a certain *class variable* by selecting a particular customer attribute combination that are the *best predictors* (based on the C4.5 algorithm discussed in Sec. 3.1.3 of this particular class variable) of a particular class value [30,31]. The speed and efficiency of the C4.5 algorithm, together with the ease of interpreting the decision tree structure make this data mining approach suitable for the product design scenario that we present in this work.

2.2 The Concept of Novel Previously Unknown Customer Information. The term *product concept* that we define in this work relates to the notion of novel previously unknown customer information that data mining is well known for [32–34]. To illustrate this concept, we present a simple test data set represented in Table 1. The data set contains six customer attributes (columns 1–6) with one predictor variable (Class variable in column 7). Based on the attribute values in Table 1 of Feature, Priority, Type, Connectivity, Battery Life, and Display, there are a total of $3 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 2 = 216$ possible unique combinations (although only 10 out of the 216 combinations exist in the sample data in Table 1). Two fundamental questions arise from our observation.

- How can we determine novel attribute combinations without performing additional data acquisition procedures (customer surveys, focus groups, etc.)?

Table 1 Example data set of customer attributes

Feature	Priority	Type	Connectivity	Battery life	Display	MaxPrice
MP3	Cost	Flip	Bluetooth	5	Screen size	200
MP3	Weight	Flip	Wifi	3	Screen size	160
MP3	Weight	Flip	Bluetooth	3	Resolution	160
MP3	Cost	Shell	Infrared	5	Screen size	80
Camera	Weight	Shell	Wifi	3	Screen size	120
Camera	Weight	Flip	Wifi	3	Resolution	120
Games	Cost	Flip	Bluetooth	5	Resolution	200
Games	Cost	Shell	Wifi	7	Resolution	200
Games	Weight	Flip	Infrared	5	Screen size	160
Games	Cost	Flip	Bluetooth	3	Screen size	160

- How efficiently can we extract these new attribute combinations?

The term *novel* in our work relates to information that is not readily observable or not explicitly defined within the data set but can be quantified through the proposed decision tree induction technique. The following product design question aims to illustrate how novel information can be extracted from a raw data set.

- Given a specific attribute combination not existing within the data set (for example, referring to Table 1 in the paper, we observe that the combination of {Games, Weight, Flip, Bluetooth, 5 h battery, ScreenSize} does not exist within the data set).

1. What price category (MaxPrice) would the above attribute combination fall under?
2. Are all of these attributes needed to predict the price category? That is, if we include only a subset of the attribute space {Games, Weight, and Flip} instead of the entire attribute space {Games, Weight, Flip, Bluetooth, 5 h battery, and ScreenSize}, would it still result in the same price category (MaxPrice) prediction?

The case study example in Sec. 4.1.1 helps address these questions. For example, the decision tree structure in Fig. 2 reveals that for the *Games* phone, as long as the product also includes *bluetooth connectivity* and a *5 h battery life*, we would result in a predicted price of $120. Therefore if design engineers were aiming to design the next generation of *Games* phones to a customer market segment willing to pay $120, then these product attributes would make up the primary product architecture.

Another example of attribute knowledge discovery can be observed in Table 1. If we were to design a camera phone product, we see from rows 5 and 6 that both *Camera* phones, each with slightly different attribute combinations yield a purchase price of $120. However, based on the C4.5 algorithm (explained in Sec. 3.1.3), we observe that no additional attributes are needed to yield a *Camera* phone price of $120 (see the initial partitioning in Table 2), therefore from a product design perspective, no additional resources should be invested in improving additional design features that do not significantly influence the purchase decisions of a customer.

This type of information is not readily observed in the raw data set and will enable design engineers to design the next generation of *Games* and *Camera* phones by including only the relevant attributes along with their predicted attribute levels. Such insights have the potential to save on manufacturing and materials costs, as well as on the time and efficiency of the product design process.

The term *product concepts* used in this work therefore represents attribute combinations within the data set (some of which may not appear in the raw data set) that are generated by the C4.5 predictive model [31,35]. Herein lies one of the fundamental strengths of data mining as opposed to the other customer data collection techniques presented in Sec. 2.1 in that inferences can be made on attribute combinations not readily available in the raw data set without additional customer interactions.

The underlying structure of the C4.5 decision tree algorithm allows us to quantify such hidden patterns within the raw data set. We discuss the theoretical aspects of the C4.5 decision tree algorithm in Sec. 3.1.3 and demonstrate how the classification procedure employed by the algorithm has the potential of classifying novel, previously unknown attribute combinations [31,36].

Note: the data set of 40,000 customer responses used in the case study in Sec. 4 has the same attributes as those found in Table 1; however, since it is the complete data set, all of the attribute values are present in the data set. For example, the Feature at-

Table 2 Test data for decision tree generation

Feature	Priority	Type	Connectivity	Battery life	Display	MaxPrice
			Branch 1			
MP3	Cost	Flip	Bluetooth	5	Screen size	200
MP3	Weight	Flip	Wifi	3	Screen size	160
MP3	Weight	Flip	Bluetooth	3	Resolution	160
MP3	Cost	Shell	Infrared	5	Screen size	80
			Branch 2			
Camera	Weight	Shell	Wifi	3	Screen size	120
Camera	Weight	Flip	Wifi	3	Resolution	120
			Branch 3			
Games	Cost	Flip	Bluetooth	5	Resolution	200
Games	Cost	Shell	Wifi	7	Resolution	200
Games	Weight	Flip	Infrared	5	Screen size	160
Games	Cost	Flip	Bluetooth	3	Screen size	160

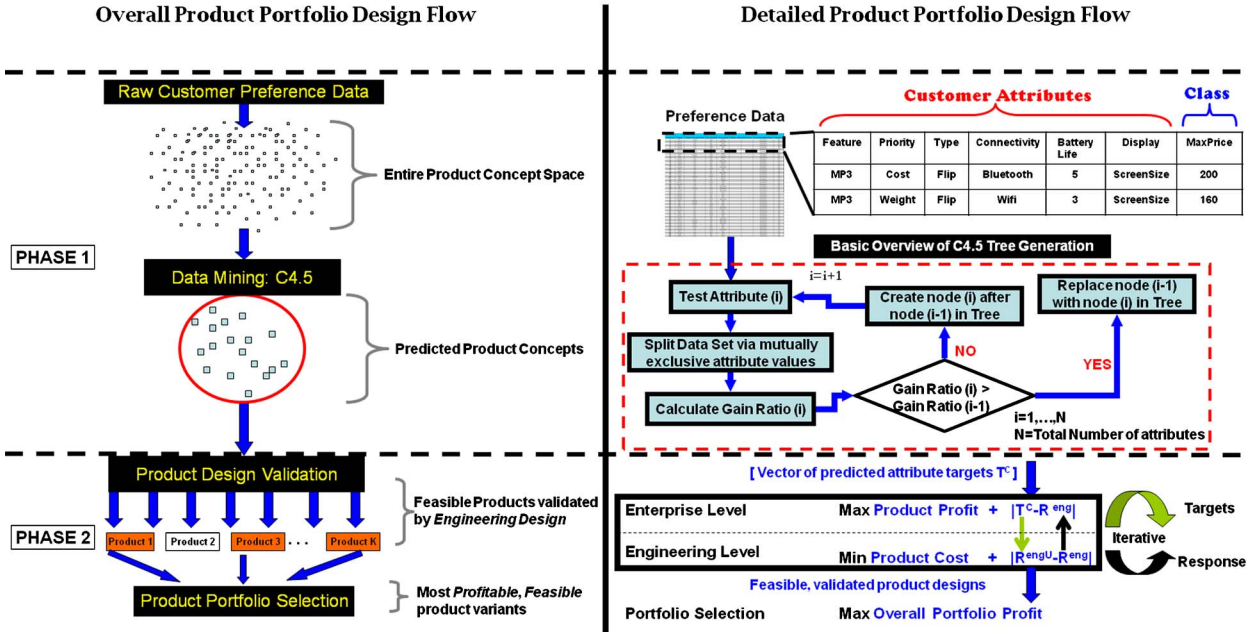

Fig. 1 Overall flow of product portfolio optimization process.

tribute has 6 levels in the data set of 40,000, but since we used a sample in Table 1 for illustrative purposes, all 6 values of the Feature attribute do not show up.

3 Methodology

The entire product portfolio generation process is divided into two phases. *Phase 1* is the customer knowledge discovery process, which entails customer data acquisition, processing, and data mining for feasible set generation. *Phase 2* involves the product concept validation through multilevel optimization and finishes with a product portfolio selection. Figure 1 shows the overall flow of this process (the general flow on the left and the detailed flow on the right of Fig. 1) starting with customer data acquisition and ending with enterprise portfolio selection. The details of the methodology are presented as follows.

3.1 Phase 1: Customer Knowledge Discovery. Knowledge discovery in databases (KDDs) has become known as the nontrivial means of extracting information in large scale databases that were previously too complex for human analysis [37]. Data mining techniques utilize classification algorithms to extract meaningful previously unknown information from large data sets [33]. The concept of data mining can be applied to product portfolio formulation, wherein the exact product specifications and manufacturing quantity (predicted demand information for each individual product concept) can be determined directly from data mining predictions. The process from data extraction to predictive model is as follows.

3.1.1 Data acquisition. The acquisition and storage of data are paramount in the product portfolio formulation process. We first begin by acquiring the raw data set to be used in the data mining

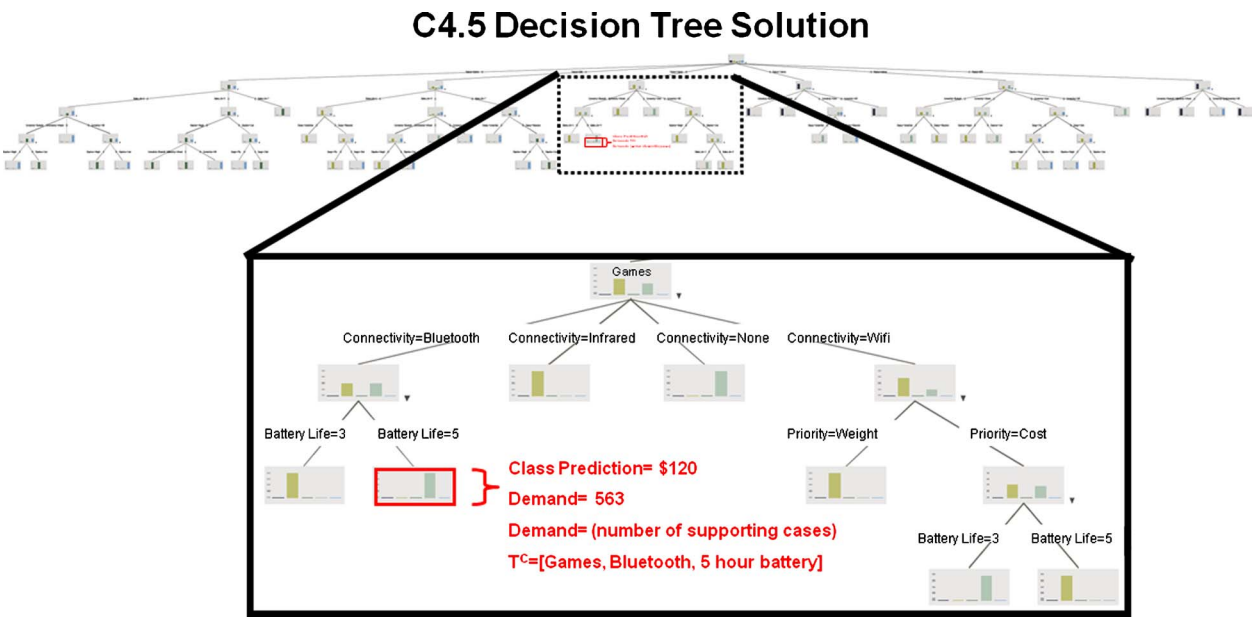

Fig. 2 C4.5 decision tree solution for 40,000 customer data set

sequence. This data can be acquired in several ways. One approach is by conducting a realistic customer survey to capture what customers want and then translating these wants into meaningful engineering design targets [38]. Another approach would be for this data to already exist in a data warehouse, i.e., stored data from past customer purchasing behavior (e.g., *Structured Query Language server* (SQL)) [39]. To increase the predictive capabilities of a classifier, it is often encouraged that the data set be large enough to accurately test the generated model with a portion of the data set.

3.1.2 Data preprocessing: data selection, cleaning, and transformation. The data preprocessing stage is where irrelevant or noisy data are identified and removed, and relevant data are extracted from the raw data [39]. There are many well established approaches that deal with missing attributes or ambiguous responses ranging from the *most common attribute, event covering method, or ignoring the value altogether* [40–42]. When dealing with electronic transactional data (online and in-store), it is then possible to collect, clean, and store these data in a *data warehouse*. A data warehouse is a preprocessing stage that integrates all data into one source (this includes raw data, historical data, summarized data, etc.) [43]. The accuracy of the data mining model will be highly dependent on the data selection and cleaning step, and it is therefore important that considerable time be allocated to preparing high quality data for the pattern discovery step that follows. There are many algorithms that exist in today's data mining analysis tools that are now capable of incorporating this data selection and cleaning process directly with the overall knowledge discovery process [44,45]. The final data preprocessing step involves transforming the data into acceptable forms for the appropriate mining algorithm. Data transformations can include *binning, normalizing, missing value imputation, etc.* [46]. This can either be done manually by the user or automatically by the analysis tool [45].

3.1.3 Pattern discovery. First a particular algorithm is selected and for the predictive analysis for our cell phone architecture design, we have opted to incorporate the C4.5 machine learning algorithm to generate a set of attribute combinations suitable for engineering design evaluation. Each unique attribute combination that predicts a class variable will be considered a candidate product concept. Typically, data mining techniques utilize 2/3 of the raw data to train the machine and the remaining 1/3 to test the model developed. The *N*-fold cross validation technique selects and compares several test models with one another and selects the appropriate model that best predicts the class variable [45]. The C4.5 machine learning algorithm for generating these product concepts is described more in detail below.

3.1.3.1 Product concept generation using C4.5 machine learning algorithm. Our approach to product concept generation adopts the C4.5 data mining tree generation technique first proposed by Quinlan [30]. The algorithm is based on the *divide and conquer* [31,30] technique that decomposes a set of training cases T with class variables $\{C_1, C_2, \ldots, C_N\}$ until the partitioning yields a collection of cases that predicts a single class variable C_i. Each subsequent decomposition of the tree tests a single attribute that has outcomes $\{O_1, \ldots, O_P\}$ that are mutually exclusive to one another [31]. When applied to product portfolio optimization, the class variable can be thought of as the overall performance criteria (determined by the enterprise decision maker) influencing product launch decisions. The class variable selected by the enterprise decision maker can range from a *Price metric* (later to be demonstrated in our cell phone example) to a *Weight* or *Dimensionality* metric, etc.

The manner in which attributes are selected during each stage of tree decomposition is the fundamental strength of the C4.5 algorithm and the primary reason why this data mining technique is so successful when applied to the product portfolio paradigm. The term attribute can be thought of as the quantifiable product requirements of a customer. Examples of attributes may be *minimum fuel economy expectations* (miles per gallon) in the context of automotive design or the *battery life* expectations of a hand held device. The tree termination criterion eliminates the need for an exhaustive search of all possible attribute combinations, and when applied to multilevel optimization formulation in product development, significantly improves on the time and efficiency of developing a portfolio of products (Demonstrated later in our cell phone example).

3.1.3.2 C4.5 gain ratio criterion. To avoid an exhaustive search of all possible attribute combinations, a systematic approach is used to partition the data and to identify what attribute to split in the most efficient manner so as to gain the most information about the class variable. For a given training set T, let us assume that we want to test a particular attribute that has P possible outcomes $\{O_1, \ldots, O_P\}$ [31]. If we define S to be any set of cases (which can either be the entire training set T or a *subset* of T), then the occurrence of a particular class variable C_i can be denoted by

$$\text{freq}(C_i, S) \tag{1}$$

This is simply the number of times a particular class occurs in a given data set. The information gained by splitting a particular attribute i gets its foundation from classical information theory that states: "*The information conveyed by a message depends on its probability and can be measured in bits as minus the logarithm to base 2 of that probability*" [30]. If $\text{freq}(C_i, S)$ determines the number of occurrences of a particular class, then the probability of randomly selecting this class over the entire set of S cases would simply be

$$\frac{\text{freq}(C_i, S)}{|S|} \tag{2}$$

where $|S|$ represents the total number of cases in the data set S.

Following the definition of *information conveyed*, the information that this particular example conveys can be represented as [31]

$$-\log_2\left(\frac{\text{freq}(C_i, S)}{|S|}\right) \text{ bits} \tag{3}$$

It is interesting to note that the range of the class variable C can be set by the enterprise decision maker depending on the desired objective of the company. If customer willingness to pay is the performance metric (Class) to be predicted, then this can be partitioned into $\{C_1, C_2, \ldots, C_N\}$, that is if the data is obtained through a direct customer survey approach.

Later in our cell phone example, the primary criterion for selecting one device over the other is the maximum price a customer is willing to pay for that particular design: MaxPrice, as it is abbreviated in the example, is therefore the class variable to be predicted. To measure the average amount of information needed to identify the class (for example, all values of MaxPrice ranging from [$40 $80 $120 $160 $200]) of a case in a training set, we sum the classes relative to their frequencies in the data set [30]

$$\text{info}(S) = -\sum_{i=1}^{N}\left(\frac{\text{freq}(C_i, S)}{|S|}\right) \cdot \log_2\left(\frac{\text{freq}(C_i, S)}{|S|}\right) \text{ bits} \tag{4}$$

Note: T represents the entire set of training cases while S represents any set of cases within T. Therefore, the above formula can be used to calculate the *information* of subsets of T or the entire data set T. $\text{info}(T)$ therefore measures the average amount of information required to identify the class of a case in T by summing over the product of all the class probabilities and their information, as defined by Eq. (4) [31]. To test the amount of information gain of a particular attribute, we partition this attribute into its

respective mutually exclusive outcomes.

After partitioning T into P possible outcomes for a specified test X (attribute selection), the expected information requirement is the summation of all subsets, as given by [47]

$$\text{info}_x(T) = \sum_{p=1}^{P} \frac{|T_p|}{|T|} \cdot \text{info}(T_p) \tag{5}$$

The *gain* can therefore be defined as the difference in the total average information required to identify a class in the training set minus the information achieved by testing a particular attribute [35]

$$\text{gain}(X) = \text{info}(T) - \text{info}_x(T) \tag{6}$$

The above equation itself is an optimization problem, where the objective is to maximize the information gain, subject to the constraints of the algorithm sequence. Due to the fact that certain attributes may have significantly greater outcomes, this metric alone may not be sufficient as it may skew the predictive capabilities of the algorithm in favor of attributes with greater outcomes. A more accurate predictor of the information that is gained by partitioning T is the *gain ratio criterion* that is defined as [31]

$$\text{gain ratio}(X) = \frac{\text{gain}(X)}{\text{split info}(X)} \tag{7}$$

where

$$\text{split info}(X) = -\sum_{p=1}^{P} \frac{|T_p|}{|T|} \cdot \log_2 \frac{|T_p|}{|T|} \tag{8}$$

The gain ratio represents the proportion of information (i.e., scaled information) generated by the split that is useful in predicting the class variable [31].

The partitioning of a problem into subproblems (i.e., generating concepts) will be terminated when there is only one class in that particular branch [31]. Pruning of subsequent branches can occur if replacing a branch with a leaf will reduce the % error of that node and ultimately the entire branch [36].

3.1.3.3 C4.5 discretization of continuous attributes. The C4.5 algorithm performs discretization and tree induction concurrently and is therefore a function of the information gain metric, rather than a user defined input [47,48]. For the case of a continuous attribute within a given data set (for example, a price or weight variable), a binary split is determined for each attribute based on *minimal entropy* criteria [30,49]. More recent contributions to the C4.5 discretization of continuous attributes employ the minimum description length (MDL) to help minimize the bias that may be inherent in the underlying gain ratio criterion explained above.

Since discretization of continuous attributes is handled during the C4.5 tree generation approach [30,50], the resulting attribute combination represents the most appropriate discretization to predict the class variable. Since C4.5 discretization is limited to the attribute space and does not include the class variable, enterprise decision makers may opt to choose a discrete variable to serve as the class variable. In our cell phone example presented later in the work, the class variable represents pricing information gathered through an online interactive customer survey and therefore is discrete based on the design of the survey. On the other hand if the data set comprises of revealed preference data, such as electronic store purchases or online transactions, the pricing information may be inherently continuous and can therefore either be discretized during the data mining preprocessing step explained in Sec. 3.1.2 or serve as an attribute in the C4.5 formulation (another class variable, such as "purchase phone: *Yes* or *No*," may serve as the class variable in this scenario). One also has the option to employ other data mining techniques that can handle continuous class variables such as M5 Prime [50] or classification and regression trees (CART) [51], which could then be applied to other product design scenarios containing continuous class variables.

3.1.4 Interpretation and evaluation. Phase 1 of the product portfolio formulation process provides us with three critical pieces of information vital to the product concept validation process (Phase 2).

- Set of candidate product concepts: represented as a unique combination of customer attributes.
- Class variable prediction: the predicted performance evaluation for product concept (j). In our example, this is denoted by MaxPrice.
- Aggregated demand for a particular product concept: represented by the total supported cases for a particular predicted class variable (represented as a leaf in the C4.5 decision tree).

3.2 Phase 2: Product Concept Validation Through Multilevel Optimization. The product concepts generated by the C4.5 decision tree data mining technique in Phase 1 need to be validated to ensure that such performance expectations can be realistically designed. In mathematical terms, we model this as a multilevel optimization problem and adopt the analytical target cascading (ATC) [52] multilevel optimization approach (although the methodology is not limited to the ATC only). Phase 2 ends with a portfolio selection decision after *feasible* product concepts have been validated by the interactions between the enterprise level and engineering level.

3.2.1 Enterprise system level. This is where the profit of each individual architecture is calculated. This level includes the set of generated product concepts that are directly incorporated into the engineering product design process. Also included in the enterprise system level is the market demand information predicted for a particular product variant (j). Mathematically, this is represented as follows.

Given

$$\mathbf{T}^C, \text{MaxPrice}_{\{j\}}, d_j, \mathbf{R}^{\text{eng}^L}, \text{cost}^L_{\text{architecture}\{j\}}$$

$$\min -\pi_{\text{architecture}_j} + \|\mathbf{T}^C - \mathbf{R}^{\text{eng}}\|_2^2 + \varepsilon_{\mathbf{R}} + \varepsilon_C \tag{9}$$

with respect to

$$\mathbf{R}^{\text{eng}}, \varepsilon_{\mathbf{R}}, \text{cost}_{\text{architecture}\{j\}}, \varepsilon_C$$

subject to

$$h1: \quad \pi_{\text{architecture}_j}$$

$$- d_j \cdot (\text{MaxPrice}_{\{j\}} - \text{cost}_{\text{architecture}\{j\}}) = 0 \tag{10}$$

$$g1: \quad \|\mathbf{R}^{\text{eng}} - \mathbf{R}^{\text{eng}^L}\|_2^2 - \varepsilon_{\mathbf{R}} \leq 0 \tag{11}$$

$$g2: \quad \|\text{cost}_{\text{architecture}\{j\}} - \text{cost}^L_{\text{architecture}\{j\}}\|_2^2 - \varepsilon_C \leq 0 \tag{12}$$

Enterprise level: variable notation definitions. $\mathbf{T}^C$ represents architecture targets (set of attribute combinations) predicted by C4.5 decision tree model. d_j represents the customer demand for product concept j predicted by the C4.5 data mining tree generation. (Conceptually, this represents the number of cases supporting the final attribute partitioning, yielding a single leaf, i.e., class prediction.) MaxPrice_j is the single class variable predicted by the continual partitioning of the set of training data until a single class is achieved. $\mathbf{R}^{\text{eng}^L}$ is the engineering performance response target from the engineering subsystem level, cascaded up to the enterprise level. $\mathbf{R}^{\text{eng}}$: at iteration 1 of the ATC formulation [53,54] $\mathbf{R}^{\text{eng}}$ represents the enterprise estimation of engineering design capabilities. This will be updated with each iteration to reflect the true design values achievable by the engineering level, i.e., $\mathbf{R}^{\text{eng}^L}$. $\text{cost}^L_{\text{architecture}\{j\}}$ represents the product cost based on the engineering capabilities of meeting predicted customer attributes. At iteration 1, this is estimated by enterprise decision makers and updated

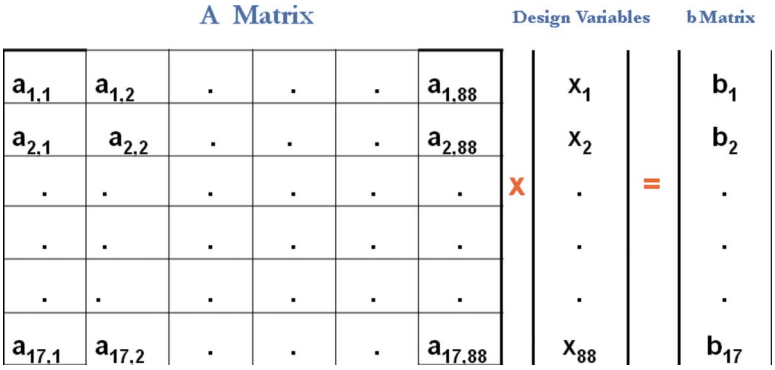

Fig. 3 Set of linear design equations (in matrix form) guiding the product architecture formulation

to reflect the true cost based on engineering response thereafter. $\pi_{\text{architecture}_j}$ is the profit of architecture j, which is a function of price and cost of the product variant j. $\varepsilon_{\mathbf{R}}$ is the deviation tolerance between customer performance and targets and engineering response. ε_C is the deviation tolerance between enterprise product cost estimation and targets and engineering response.

3.2.2 Engineering level. This is where the individual architecture costs are calculated, along with the physical product architecture design. The engineering design level is modeled as a mixed integer nonlinear programming problem [55], with discrete selection variables that govern component choice selections (manufacturer specifications, component design, etc.) and continuous variables that regulate the product dimensions and aesthetic design. The iteration between the *enterprise level* and the *engineering level* determines the feasibility criteria for each product as customer targets are set at the enterprise level and subsequently validated with an engineering subsystem response within a specified tolerance for ε. Mathematically, this is represented as follows.

Given

$$\mathbf{R}^{\text{eng}^U} \tag{13}$$

$$\min \text{cost}_{\text{architecture}_j} + \|\mathbf{R}^{\text{eng}^U} - \mathbf{R}^{\text{eng}}\|_2^2$$

with respect to

$$\mathbf{x}_{\text{eng}}$$

subject to engineering product design equality constraints, production capacity, materials, and supplier constraints

$$\mathbf{h}_{\text{eng}}(\mathbf{x}_{\text{eng}}) = \mathbf{0} \tag{14}$$

$$\mathbf{g}_{\text{eng}}(\mathbf{x}_{\text{eng}}) \leq \mathbf{0} \tag{15}$$

Engineering level: variable notation definitions. $\text{cost}_{\text{architecture}_j}$: the engineering design objective, cost, is the primary performance criterion influencing the product design, while the objective is not limited to the cost. The objective can be any individual product performance objective, such as cost, weight, etc. In our cell phone example problem, the engineering objective is to minimize the cost, as well as to match the attributes targets $\mathbf{R}^{\text{eng}^U}$. $\mathbf{R}^{\text{eng}^U}$ represents the engineering performance response target from the enterprise system level, cascaded down to the engineering level. $\mathbf{R}^{\text{eng}}$ represents the performance response from the engineering design, i.e., $\mathbf{R}^{\text{eng}} = \mathbf{R}^{\text{eng}}(\mathbf{x}_{\text{eng}})$. (The engineering response $\mathbf{R}^{\text{eng}}$ will become $\mathbf{R}^{\text{eng}^L}$ at the enterprise system level.)

The *product architecture* is defined in this work as the engineering design foundation, from which product variants can evolve. The functionality of each product architecture is unique and addresses the fundamental requirements of the product. For example, an MP3 product architecture would be designed such that the MP3 functionality can be easily accessed and controlled by the user. An interactive Game capable (noted as Games in our cell phone example) cell phone would have a product architecture that allows the user to seamlessly switch from game playing mode to phone operation mode. These differences are addressed in the engineering design level, where customer attributes are translated into engineering design functionality through a set of linear constraints (see Figs. 3 and 4).

3.2.3 Enterprise portfolio selection. This is where the overall enterprise portfolio profit is determined by searching through the feasible product space and selecting/deselecting architectures in an attempt to maximize profit by generating an optimal product portfolio. Here, the optimal portfolio is defined as the selected products that maximize the enterprise profit within the product portfolio limit K. The termination of this selection process is determined when either (1) the product portfolio limit is reached in case there exist more profitable product concepts than the limit, or (2) all the profitable product concepts are identified in case the number of profitable product concepts is less than the limit. Mathematically, this is represented as

$$\min - \sum_{j=1}^{k} x_j \cdot \pi_{\text{architecture}(j)} \tag{16}$$

subject to

$$h1: \quad x_j = \{0,1\}, \quad j \in \{1, \ldots, k\} \tag{17}$$

$$g1: \quad \sum_{j=1}^{k} x_j - K \leq 0 \tag{18}$$

Enterprise portfolio selection: variable notation definitions. $\pi_{\text{architecture}(j)}$ represents profit of architecture j. x_j represents the binary variable selecting or deselecting particular architecture ($\pi_{\text{architecture}}$), where $\Sigma_{j=1}^{k} x_j \leq K$. k is the *total feasible product/variants* that can be designed. This numeric value is attained through the engineering design validation process. The value k therefore represents the total number of product/variants that satisfy customer performance and price expectations. K is the *product portfolio limit*. To avoid impractical manufacturing expectations and an oversaturation of products in the market space, the number of products existing in the product portfolio must be constrained. The value set as the maximum portfolio limit may be a function of many externalities including competition, distribution, marketing constraints, etc. In our approach, we have left the product portfolio limit up to the enterprise decision maker. (Note: depending on the number of existing feasible products, this limit may/may not be reached.)

The flow diagram in Fig. 1 represents the overall process from customer preference acquisition via database extraction to the generation of product concepts. The validated product concepts

Linear Constraints: Cell Phone Engineering Design	
A Matrix (Functional Component Selection Process)	**Matrix Element Value**
A(1,[1:3]) Cell Phone 32MB RAM: 3 Manufacturers to choose from	1 or 0
A(2,[4:6]) Cell Phone 64MB RAM: 3 Manufacturers to choose from	1 or 0
A(3,[7:12]) Cell Phone External Memory Storage: 6 Manufacturers to choose from	1 or 0
A(4,[13:18]) Cell Phone Hard Drive Storage: 3 manufacturers of 1 Gig and 3 manufactures of 2 Gig	1 or 0
A(5,[19:25]) Cell Phone Design: 2 Designs (Flip phone or Shell phone) manufactured in-house	1 or 0
A(6,[31:38]) Cell Phone Battery Design: 2 Types (NIMH or LION) manufactured in-house	1 or 0
A(7,[46,47,48]) Cell Phone Connectivity: Bluetooth, Wifi, Infrared from manufacturer	1 or 0
A(8,[49,50]) Cell Phone Microphone selection: 2 Manufacturers to choose from	1 or 0
A(9,[51,52,53]) Cell Phone Earpiece: 3 Manufacturers to choose from	1 or 0
A(10,[54,55]) Cell Phone Audio: Audio Jack component- 2 manufacturers to choose from	1 or 0
A(11,[56,57,58]) Cell Phone External Audio: External Speaker- 3 manufacturers to choose from	1 or 0
A(12,[59:66]) Cell Phone Display Type: 2 Designs (TFT or OLED display) manufactured in-house	1 or 0
A(13,[73,74,75,76]) Cell Phone Camera module: 4 manufacturers to choose from (1Mpixel VS 2 Mpixel)	1 or 0
A(14,[77,78,79,80]) Cell Phone MP3 module: 4 manufacturers to choose from	1 or 0
A(15,[81,82]) Cell Phone Internet module: 2 manufacturers to choose from	1 or 0
A(16,[83,84,85]) Cell Phone processor for Games capability: 3 manufacturers to choose from	1 or 0
A(17,86) Cell Phone module for SMSText capability:1 manufacturer to choose from	1 or 0

Fig. 4 A matrix forming the linear equation set. The matrix is sparse, with active elements signified by a value of 1.

with the highest profit margins will form the product portfolio (subject to the product portfolio limit as determined by enterprise decision makers).

4 Application: Cell Phone Design

4.1 Phase 1: Cell Phone Customer Knowledge Discovery. To validate the proposed decision tree approach in generating a product portfolio, we present a cell phone product portfolio case study. A cell phone survey was designed using the University of Illinois at Urbana-Champaign (UIUC) webtools platform where respondents had the option of selecting a combination of attribute values and the price category that most closely represented their selection [56]. To emphasize the strength of data mining in handling large data sets, additional data were simulated (based on the generated survey questionnaire) using Excel Visual Basic to achieve a total of 40,000 customer responses. The data preprocessing steps explained in Sec. 3.1.2 are handled by the data mining analysis tool [44]. In machine learning techniques, the raw data are partitioned; typically 2/3 is used to train the algorithm, and the remaining 1/3 is used to test the model for predictive accuracy [45]. For demonstration purposes, we have taken a small fraction of the train data T to illustrate the decision tree generation algorithm discussed earlier. A set of ten cases will demonstrate the gain ratio criteria in decision tree decomposition (see Table 2).

Our class variable in Table 2 is MaxPrice and is defined as the maximum price a customer is willing to pay for a particular product. The class variable can be altered, depending on the focus of the enterprise decision makers to reflect the strategic objectives of the company. In our methodology, the primary information we are concerned with in the data mining process is the price sensitivity information predicted by the decision tree with varying attribute combinations.

Each row in Table 2 will be defined as an independent case. (The term case refers to a unique customer response containing certain attribute values along with the associated class value). There are six attributes in our example table represented as {Feature, Priority, Type, Connectivity, Battery Life, and Display}. The class variable MaxPrice is partitioned into five separate mutually exclusive classes [$40, $80, $120, $160, $200].

Since the ten cases in our example do not all belong to the same class, we can implement the *C4.5 divide and conquer* algorithm in an attempt to split the cases into subsets. There are four classes in our cell phone sample train $\boldsymbol{T}$ file (the $40 class of MaxPrice did not occur in this illustration but occurs in larger training sets). $\boldsymbol{T}$ contains three cases belonging to the $200 class, four cases belonging to the $160 class, two cases belonging to the $120 class, and one case belonging to the $80 class for a total of ten cases for our training data in Table 2.

4.1.1 Product concept generation through C4.5 decision tree classification. Step 1: class identification. Following the C4.5 algorithm, the first step is to determine the average information needed to identify a value of MaxPrice in our training data. info(T) (bit) will be defined as

$$\text{info}(T) = -\frac{3}{10}\cdot\log_2\left(\frac{3}{10}\right) - \frac{4}{10}\cdot\log_2\left(\frac{4}{10}\right) - \frac{2}{10}\cdot\log_2\left(\frac{2}{10}\right) - \frac{1}{10}\cdot\log_2\left(\frac{1}{10}\right) = 1.846 \text{ bits} \tag{19}$$

The above info(T) calculation is determined directly from Table 2, where the information needed to identify the three cases of our $200 class out of the total ten cases is represented in Eq. (19) as $-3/10\cdot\log_2(3/10)$ and similarly for each subsequent class identification.

Step 2: attribute selection. The information gained by selecting a particular attribute will determine the sequence of attribute selection and consequently the structure and length of the decision tree or, in product development terms, the number of candidate product concepts that are generated and deemed to be the best predictors of each class of MaxPrice. The tree decomposition process is an iterative approach, substituting one attribute over another if a higher information gain can be realized by selecting this attribute as a node in the tree. Let us now arbitrarily select an attribute to be used as our initial node (root) and calculate the information gained by this selection.

(Attribute test=feature) We then partition the attribute selected into its individual mutually exclusive outcomes (represented by branches in the actual decision tree). We have four cases that are MP3, two cases that are Camera, and four cases that are Games to comprise the ten Feature cases, as illustrated in Fig. 2. We can now determine the expected information requirement of the *Feature* attribute as the weighted sum of the three subsets {MP3, Camera, Games}

$$\begin{aligned}\text{info}_{\{X=\text{feature}\}}(T) = \frac{4}{10}\cdot\left\{-\frac{1}{4}\cdot\log_2\left(\frac{1}{4}\right)-\frac{2}{4}\cdot\log_2\left(\frac{2}{4}\right)\right.\\ \left.-\frac{0}{4}\cdot\log_2\left(\frac{0}{4}\right)-\frac{1}{4}\cdot\log_2\left(\frac{1}{4}\right)\right\}\\ +\frac{2}{10}\cdot\left\{-\frac{0}{2}\cdot\log_2\left(\frac{0}{2}\right)-\frac{0}{2}\cdot\log_2\left(\frac{0}{2}\right)\right.\\ \left.-\frac{2}{2}\cdot\log_2\left(\frac{2}{2}\right)-\frac{0}{2}\cdot\log_2\left(\frac{0}{2}\right)\right\}\\ +\frac{4}{10}\cdot\left\{-\frac{2}{4}\cdot\log_2\left(\frac{2}{4}\right)-\frac{2}{4}\cdot\log_2\left(\frac{2}{4}\right)\right.\\ \left.-\frac{0}{4}\cdot\log_2\left(\frac{0}{4}\right)-\frac{0}{4}\cdot\log_2\left(\frac{0}{4}\right)\right\}=1.00\ \text{bits}\end{aligned} \tag{20}$$

Therefore, the information gained by testing attribute=feature is simply

$$\text{gain}(X)=\text{info}(T)-\text{info}_{\{X=\text{feature}\}}(T)=1.864-1.00=0.864\ \text{bits} \tag{21}$$

In the event that our data set contains one or several attributes with a significantly greater range of outcomes, the split info(X) function can attempt to normalize the attributes.

$$\text{gain ratio}(X)=\frac{\text{gain}(X)}{\text{split info}(X)}=0.57 \tag{22}$$

Each subsequent attribute that is tested on the basis of *gain ratio* criterion is compared to the previous attribute and substituted if a higher gain ratio is achieved. This iterative process is continued until a single class is identified for a given attribute split. Further illustration is given in Phase 1 in Fig. 1, and a visual representation of the generated C4.5 decision tree using the 40,000 raw customer data set is provided in Fig. 2.

Translation of customer attributes to engineering design functionality. Customer predicted attribute information must be translated into meaningful engineering functionality criterion for the product design process. A set of linear equations represented by Fig. 3 indicate which of the product functionality components are included in a particular product architecture. Figure 4 is simply a textual explanation of the A-matrix and indicates which of the engineering components are active.

Depending on the cell phone architecture type and the engineering design objective function, one or several of the elements in each row of the A-matrix will be active (1) or inactive (0). The upper and lower bounds for the linear equations (b-matrix) therefore fluctuate based on the product concept requirements currently being tested. For example, if an MP3 product concept requires a *bluetooth* connectivity feature, the element representing bluetooth connectivity in row 7 of the A-matrix will automatically be active (1) and the lower bound for the connectivity linear constraint (which comprises of three possible connectivity options: Bluetooth, Infrared, or Wifi (see row 7 of Fig. 4)) will immediately be set to 1. That is, b_7 in Fig. 3 will be ≥ 1. Furthermore, the lower bound for the external speaker (Row 11 of the A-matrix in Fig. 4) will be set to 1, indicating that the MP3 cell phone, will come equipped with external audio capability (a functionality translation based on the customer attribute requirement of MP3 music playback). The numbers in closed brackets in each row of the A-matrix in Fig. 4 (i.e., column indices) indicate the number of possible choices for that particular component group.

4.2 Phase 2: Product Concept Validation. *Enterprise level: cell phone design validation and profit calculation.* Once the customer data set of 40,000 cases (with 576 unique attribute combinations) has been narrowed down to 46 generated product concepts (vector of predicted product attribute combinations) via the C4.5 data mining tree generation technique, we must now determine the engineering design feasibility and potential profit margin for each product; mathematically represented as follows.

Given

$$\begin{gathered}T^{\text{battery life}},\quad T^{\text{connectivity}},\quad T^{\text{priority}},\quad T^{\text{display}},\quad T^{\text{type}}\\ \text{MaxPrice}_j,\quad d_j,\quad R^{\text{battery life}^{\text{eng}^L}},\quad R^{\text{connectivity}^{\text{eng}^L}}\\ R^{\text{priority}^{\text{eng}^L}},\quad R^{\text{display}^{\text{eng}^L}},\quad R^{\text{type}^{\text{eng}^L}},\quad \text{cost}_j^L\end{gathered}$$

$$\begin{aligned}\min\ &-\pi_{\text{architecture}_j}+\left\|T^{\text{battery life}}-R^{\text{battery life}^{\text{eng}}}\right\|_2^2\\ &+\left\|T^{\text{connectivity}}-R^{\text{connectivity}^{\text{eng}}}\right\|_2^2+\left\|T^{\text{priority}}-R^{\text{priority}^{\text{eng}}}\right\|_2^2\\ &+\left\|T^{\text{display}}-R^{\text{display}^{\text{eng}}}\right\|_2^2+\left\|T^{\text{type}}-R^{\text{type}^{\text{eng}}}\right\|_2^2\\ &+\varepsilon_{\text{battery life}}+\varepsilon_{\text{connectivity}}+\varepsilon_{\text{priority}}\\ &+\varepsilon_{\text{display}}+\varepsilon_{\text{type}}+\varepsilon_C\end{aligned} \tag{23}$$

with respect to

$$\begin{gathered}R^{\text{battery life}^{\text{eng}}},\quad R^{\text{connectivity}^{\text{eng}}},\quad R^{\text{priority}^{\text{eng}}}\\ R^{\text{display}^{\text{eng}}},\quad R^{\text{type}^{\text{eng}}},\quad \text{cost}_j,\quad \varepsilon_{\text{battery life}},\quad \varepsilon_{\text{connectivity}},\\ \varepsilon_{\text{priority}},\quad \varepsilon_{\text{display}},\quad \varepsilon_{\text{type}},\quad \varepsilon_C\end{gathered}$$

subject to

$$h1:\quad \pi_{\text{architecture}_j}-d_j\cdot(\text{MaxPrice}_j-\text{cost}_j)=0 \tag{24}$$

$$h2:\quad \text{MaxPrice}_j=\{\$40,\$80,\$120,\$160,\$200\} \tag{25}$$

$$g1:\quad \left\|R^{\text{battery life}^{\text{eng}}}-R^{\text{battery life}^{\text{eng}^L}}\right\|_2^2\leq\varepsilon_{\text{battery life}} \tag{26}$$

$$g2:\quad \left\|R^{\text{connectivity}^{\text{eng}}}-R^{\text{connectivity}^{\text{eng}^L}}\right\|_2^2\leq\varepsilon_{\text{connectivity}} \tag{27}$$

$$g3:\quad \left\|R^{\text{priority}^{\text{eng}}}-R^{\text{priority}^{\text{eng}^L}}\right\|_2^2\leq\varepsilon_{\text{priority}} \tag{28}$$

$$g4:\quad \left\|R^{\text{display}^{\text{eng}}}-R^{\text{display}^{\text{eng}^L}}\right\|_2^2\leq\varepsilon_{\text{display}} \tag{29}$$

$$g5:\quad \left\|R^{\text{type}^{\text{eng}}}-R^{\text{type}^{\text{eng}^L}}\right\|_2^2\leq\varepsilon_{\text{type}} \tag{30}$$

$$g6:\quad \left\|\text{cost}_j-\text{cost}_j^L\right\|_2^2\leq\varepsilon_C \tag{31}$$

Here, the attributes are given as product design targets **T**, and the engineering design responses are **R**, for which deviations are defined as ε. Individual product demand is noted d_j with corresponding price MaxPrice_j and cost cost_j. The initial evaluation of the engineering design response is estimated and then subsequently updated with each engineering design response thereafter.

Engineering level: product design validation. After the enterprise profit is calculated for each of the 46 product variant concepts, individual product variants are checked for their feasibility in the engineering design space. Based on the attribute targets, the engineering design team attempts to minimize the cost while meeting the product attribute requirements.

Table 3 Possible shared component variables

Component	Description	Cost range	Design options
Internal memory (RAM)	32 MB RAM discrete choice variable	\$0.15–\$0.35	Manufacturer
Internal memory (RAM)	64 MB RAM discrete choice variable	\$0.41–\$0.51	Manufacturer
External memory	Memory stick pro discrete choice variable	\$1.1–\$1.3	Manufacturer
External memory	Memory stick duo discrete choice variable	\$1.44–\$1.65	Manufacturer
Hard drive	1 GB storage discrete choice variable	\$15.63–\$17.4	Manufacturer
Hard drive	2 GB storage discrete choice variable	\$24.83–\$26.80	Manufacturer
Phone type	Shell phone design variables	$\$2.2\times10^{-4}$/volume	Engineering design
Phone type	Flip phone design variables	$\$1.47\times10^{-4}$/volume	Engineering design
Battery type	Lithium polymer [57] battery design variables	$\$8.03\times10^{-4}$/volume	Engineering design
Battery type	Lithium ion [57] battery design variables	$\$3.79\times10^{-4}$/volume	Engineering design
Connectivity	Bluetooth connection discrete variable	\$5.20–\$5.8	Manufacturer
Connectivity	Wifi discrete choice variable	\$7.0–\$7.3	Manufacturer
Connectivity	Infrared discrete choice variable	\$3.7–\$3.73	Manufacturer
Audio codec	Microphone discrete variable	\$0.81–\$0.84	Manufacturer
Audio codec	Earpiece discrete variable	\$0.1–\$0.14	Manufacturer
Audio codec	Audio jack discrete variable	\$0.6–\$0.8	Manufacturer
Audio codec	External speaker discrete variable	\$1.7–\$3.75	Manufacturer
Display type	TFT LCD [58] discrete variable	$\$5.0\times10^{-3}$/volume	Manufacturer
Display type	OLED [58] discrete variable	$\$8.0\times10^{-3}$/volume	Manufacturer

Given

$$R^{\text{battery life}^U},\quad R^{\text{connectivity}^U},\quad R^{\text{priority}^U},\quad R^{\text{display}^U},\quad R^{\text{type}^U}$$

$$\min \text{cost}_{\text{architecture}_j} + \left\|R^{\text{battery life}^U} - R^{\text{battery life}}\right\|_2^2 + \left\|R^{\text{connectivity}^U} - R^{\text{connectivity}}\right\|_2^2 + \left\|R^{\text{priority}^U} - R^{\text{priority}}\right\|_2^2 + \left\|R^{\text{display}^U} - R^{\text{display}}\right\|_2^2 + \left\|R^{\text{type}^U} - R^{\text{type}}\right\|_2^2 \quad (32)$$

with respect to

$$\mathbf{x}_{\text{eng}}$$

subject to[2] Screen Resolution constraints, Battery Design constraints, Outer Casing Design (Phone Type) constraints, and Design Priority constraints (component cost estimates can be seen in Table 3).

$$\mathbf{g}_{\text{eng}}(\mathbf{x}_{\text{eng}}) \leq \mathbf{0}, \mathbf{h}_{\text{eng}}(\mathbf{x}_{\text{eng}}) = \mathbf{0} \quad (33)$$

Product portfolio selection. Among the feasible product variants (35 out of 46), the final step is to generate product portfolio under the specified limit of 7 total products. For each product variant, the selection variable $\mathbf{x}$ is defined to achieve the final most profitable product portfolio

$$\min -\sum_{j=1}^{k} x_j \cdot \pi_{\text{architecture}(j)} \quad (34)$$

subject to

$$h1: \quad x_j = \{0,1\}, \quad j \in \{1,\ldots,35\} \quad (35)$$

$$g1: \quad \sum_{j=1}^{35} x_j - 7 \leq 0 \quad (36)$$

5 Results and Discussion

Our methodology in formulating an optimal product portfolio presents more than just a set of feasible product concepts, but rather a validated portfolio of product designs that are the best indicators of market success, which ultimately maximize overall enterprise profit. Table 4 presents the final solution achieved in our cell phone case study of 40,000 customer responses that we subsequently narrowed down to 46 predictive product concepts. As can be seen in Table 4, column 10, the multilevel optimization formulation returns a vector of feasible/infeasible product designs based on customer predictive preference targets cascaded down to the engineering level. In our formulation, the term *feasibility* is defined as customer preference targets attained through data mining predictive techniques that are matched within the engineering design response tolerance of $\varepsilon = 0.01$. A product design that fails to satisfy this tolerance is considered to be a suboptimal product variant and is excluded in the optimal product portfolio.

Product feasibility is however not the only measure of product design success. With the incorporation of demand information directly acquired through the C4.5 data mining process, each product variant profit can be calculated based on the unit product cost, the MaxPrice class prediction, and the demand for a particular product concept j. Referring to the results for the Generic Phone architecture in Table 4, we observe that there are 11 product concepts generated through the data mining technique. As the results indicate, generic product variant 11 with a predicted battery life expectation of 7 h and a MaxPrice prediction of \$40, was found to be infeasible in the engineering design formulation. The violated target in this scenario was that of the battery life with a maximum attainable engineering design response of 6.79 h. Using our metric for evaluating feasible designs, this product concept clearly violates our tolerance limit, hence, is excluded as a candidate for our optimal product portfolio. In addition to the feasiblity check, our approach also generates the unit cost for product design with its corresponding profit. For this specific variant, Generic variant 11 in the Table 4, the unit cost is \$53.73; therefore, its corresponding loss is \$15,422. Customers may indicate their preference for this specific variant. However, this concept should not be pursued due to the projected loss, as well as it is outside the maximum portfolio limit size that is described below.

If we observe our Generic architecture results more closely, we can identify several product concepts that are feasible in the engineering design but are omitted in our optimal product portfolio set. As discussed earlier, this is due to the fact that *overall enterprise profit* is the second criterion for evaluating product variants to be included in our optimal product portfolio. There are total of 11 infeasible and/or negative profit generating product variants out of the 46 product concepts predicted by the data mining process, leaving us with 35 candidate products to introduce to the cell phone market. Depending on the enterprise product portfolio limit, and the number of product families that can be managed, all or a few of these products could be considered for market launch.

[2]To enhance the overall flow of the paper, the elaborate constraints governing the engineering design of cell product variants are condensed and represented by only $\mathbf{g}_{\text{eng}}(\mathbf{x}_{\text{eng}})$ and $\mathbf{h}_{\text{eng}}(\mathbf{x}_{\text{eng}})$ above. Refer to the Appendix including Table 3 for detailed cell phone design model.

Table 4 Results of C4.5 data mining product concept generation. The yellow highlighted rows indicate members of the optimal product portfolio.

Product platform	Product variants	Priority	Type	Connectivity	Battery life	Display	Demand	Max Price	Engineering design validation	Product unit cost	Generated profit	Product portfolio member
Generic	1	Weight		Bluetooth	3		293	$80	Feasible	$46.04	$9951	Yes
	2	Cost		Bluetooth	3		297	$40	Feasible	$43.66	−$1087	No
	3			Infrared	3		591	$80	Feasible	$41.59	$22,701	Yes
	4	Weight		Wifi	3		284	$40	Feasible	$47.77	−$2206	No
	5	Cost		Wifi	3		318	$80	Feasible	$45.42	$10,997	Yes
	6	Weight		Bluetooth	5		290	$40	Feasible	$53.69	−$3970	No
	7	Weight		Infrared	5		283	$40	Feasible	$57.08	−$4833	No
	8	Weight		Wifi	5		302	$80	Feasible	$60.65	$5845	Yes
	9	Cost	Flip		5		468	$80	Feasible	$45.04	$16,363	Yes
	10	Cost	Shell		5		413	$40	Feasible	$51.28	−$4659	No
	11				7		1123	$40	Infeasible	$53.73	−$15,422	No
SMS text	1				3	Screen size	907	$160	Feasible	$45.61	$103,752	Yes
	2		Flip		3	Resolution	441	$160	Feasible	$48.44	$49,196	Yes
	3		Shell		3	Resolution	423	$80	Feasible	$47.91	$13,575	Yes
	4	Weight		Bluetooth	5		314	$160	Feasible	$66.64	$29,316	Yes
	5	Cost		Bluetooth	5		331	$80	Feasible	$58.33	$7174	Yes
	6			Infrared	5		578	$120	Feasible	$56.26	$36,844	Yes
	7			Wifi	5		600	$80	Feasible	$59.83	$12,104	Yes
	8				7	Screen size	579	$80	Infeasible	$61.22	$10,872	No
	9	Weight			7	Resolution	296	$80	Infeasible	$64.09	$4710	No
	10	Cost			7	Resolution	294	$40	Infeasible	$64.06	−$7073	No
Games	1			Bluetooth	3		581	$160	Feasible	$55.33	$60,812	Yes
	2			Bluetooth	5		563	$120	Feasible	$68.21	$29,158	Yes
	3			Infrared			1185	$160	Feasible	$49.42	$131,031	Yes
	4			None			1104	$120	Feasible	$45.69	$82,033	Yes
	5	Weight		Wifi			598	$160	Feasible	$56.30	$62,011	Yes
	6	Cost		Wifi	3		304	$120	Feasible	$56.83	$19,203	Yes
	7	Cost		Wifi	5		321	$160	Feasible	$69.71	$28,983	Yes
Camera	1			Bluetooth			1166	$200	Feasible	$87.59	$131,075	Yes
	2			Infrared			1222	$200	Feasible	$88.58	$136,150	Yes
	3			None		Screen size	602	$120	Feasible	$88.59	$18,909	Yes
	4			None		Resolution	580	$80	Feasible	$79.99	$6	No
	5			Wifi			1184	$200	Feasible	$79.98	$142,103	Yes
Internet	1			Bluetooth		Screen size	583	$120	Feasible	$54.67	$38,085	Yes
	2			Bluetooth		Resolution	546	$160	Feasible	$57.51	$55,960	Yes
	3	Weight		Infrared			559	$160	Feasible	$56.59	$57,807	Yes
	4	Cost		Infrared			543	$120	Feasible	$52.60	$36,596	Yes
	5	Weight	Flip	None			295	$160	Feasible	$52.86	$31,607	Yes
	6	Cost	Flip	None			294	$80	Feasible	$48.87	$9151	Yes
	7	Weight	Shell	None			297	$80	Feasible	$51.53	$8455	Yes
	8	Cost	Shell	None			295	$160	Feasible	$48.36	$32,932	Yes
	9			Wifi			1120	$120	Feasible	$56.17	$71,485	Yes
MP3	1	-	-	Bluetooth	-	-	1239	$200	Feasible	$98.48	$125,778	Yes
	2	-	-	Infrared	-	-	1108	$200	Feasible	$95.90	$115,337	Yes
	3	-	-	None	-	-	1124	$80	Feasible	$92.17	−$13,685	No
	4	-	-	Wifi	-	-	1161	$200	Feasible	$99.47	$116,710	Yes

If we assume a maximum portfolio limit size of seven, the enterprise optimal product portfolio, as described in Sec. 3.2, would simply be a selection of the most profitable product variants, subject to the portfolio limit constraint. This is modeled by the selection problem in Eqs. (34)–(36). If we explore our entire feasible product concept space, we will achieve a solution of seven product variants spanning multiple product families. Our final solution yields one product variant from the Games product family, three product variants from the Camera product family (Camera product variants {1,2,5}), and three product variants from the MP3 product family (MP3 product variants {1,2,4}) yielding a total product portfolio sales volume (based on demand information) of 8265 units and an overall enterprise profit of $898,185 (Table 3).

Such powerful insights enable enterprise decision makers to evaluate products/variants based on several dimensions of performance. In our example of 40,000 customers, we observe that each customer does not have to be provided with his/her own unique customizable product, but rather purchasing behaviors can be addressed with the 46 product concepts generated in our data mining predictions. Furthermore, enterprise decision makers can determine which out of these product concepts would be the most successful in an attempt to maximize profit.

6 Conclusion

The volatility of highly competitive consumer markets is the major driving force shaping company strategies in product development. The power to accurately predict and design products before they are launched is a fundamental tool in ensuring a competitive advantage among fierce competition. The major focus of our research is to develop a methodology to predict customer wants and subsequently to design the most profitable products or product variants. We addressed the predictive aspect of product development through data mining and machine learning techniques and generated candidate product concepts along with individual predicted demand information. The validation of these product concepts at the engineering design level increases the

likelihood of the products being market successes if launched. As a result, enterprise decision makers will have several options in formulating an optimal product portfolio. Other metrics, such as level of commonality among product variants, can be used as an additional evaluation metric in deciding product launches. Additional cost savings benefits can be realized through post optimality analysis of shared components. In future works, we plan to present how such commonality analysis techniques may alter the optimal product portfolio solution.

Nomenclature

- K = product portfolio limit (maximum number of existing products at launch)
- $\mathbf{T}^C$ = product variant targets component predicted by decision tree model
- $\mathbf{R}^E$ = engineering design response
- π = projected profit of a feasible product design based on engineering design and predicted demand
- $\varepsilon_{\mathbf{R}}$ = deviation tolerance between customer performance targets and engineering response

Acknowledgment

This material is based on work supported by the National Science Foundation under Award No. 0726934 and the Sandia National Laboratories. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation and the Sandia National Laboratories.

Appendix: Cell Phone, Detailed Design Model

Several variable names are abbreviated (L=length, W=width, T=thickness, Wg=weight, V=volume, cap=capacity, P=power consumption, etc.).

1 Screen Resolution Constraints.

$$h1: \quad (A1 * \mathrm{LCD}_{\mathrm{length}} * \mathrm{LCD}_{\mathrm{width}}) - \mathrm{LCD}_{\mathrm{res}} = 0 \tag{A1}$$

$$h2: \quad (A2 * \mathrm{LCD}_{\mathrm{length}} * \mathrm{LCD}_{\mathrm{width}}) - \mathrm{cost}_{\mathrm{LCD}} = 0 \tag{A2}$$

$$h3: \quad (A3 * \mathrm{LCD}_{\mathrm{length}} * \mathrm{LCD}_{\mathrm{width}}) - \mathrm{weight}_{\mathrm{LCD}} = 0 \tag{A3}$$

$$h4: \quad (A4 * \mathrm{LCD}_{\mathrm{length}} * \mathrm{LCD}_{\mathrm{width}}) - \mathrm{power}_{\mathrm{LCD}} = 0 \tag{A4}$$

$$h5: \quad (A5 * \mathrm{OLED}_{\mathrm{length}} * \mathrm{OLED}_{\mathrm{width}}) - \mathrm{OLED}_{\mathrm{res}} = 0 \tag{A5}$$

$$h6: \quad (A6 * \mathrm{OLED}_{\mathrm{length}} * \mathrm{OLED}_{\mathrm{width}}) - \mathrm{cost}_{\mathrm{OLED}} = 0 \tag{A6}$$

$$h7: \quad (A7 * \mathrm{OLED}_{\mathrm{length}} * \mathrm{OLED}_{\mathrm{width}}) - \mathrm{weight}_{\mathrm{OLED}} = 0 \tag{A7}$$

$$h8: \quad (A8 * \mathrm{OLED}_{\mathrm{length}} * \mathrm{OLED}_{\mathrm{width}}) - \mathrm{power}_{\mathrm{OLED}} = 0 \tag{A8}$$

2 Battery Design Constraints.

$$h9: \quad \mathrm{cap}_{\mathrm{NIMH}} - (NIMH_{\mathrm{const1}} * (V_{\mathrm{NIMH}})) - T_{\mathrm{hours}} * \sum_{i=1}^{N} P_{\mathrm{component}_i} = 0 \tag{A9}$$

$$h10: \quad \mathrm{cap}_{\mathrm{LION}} - (\mathrm{LION}_{\mathrm{const1}} * (V_{\mathrm{LION}})) - T_{\mathrm{hours}} * \sum_{i=1}^{N} P_{\mathrm{component}_i} = 0 \tag{A10}$$

$$h11: \quad ((\mathrm{NIMH}_{\mathrm{const2}} * (L_{\mathrm{NIMH}} * W_{\mathrm{NIMH}} * T_{\mathrm{NIMH}})) - \mathrm{cost}_{\mathrm{NIMH}}) = 0 \tag{A11}$$

$$h12: \quad ((\mathrm{LION}_{\mathrm{const2}} * (L_{\mathrm{LION}} * W_{\mathrm{LION}} * T_{\mathrm{LION}})) - \mathrm{cost}_{\mathrm{LION}}) = 0 \tag{A12}$$

$$h13: \quad ((\mathrm{NIMH}_{\mathrm{const3}} * (L_{\mathrm{NIMH}} * W_{\mathrm{NIMH}} * T_{\mathrm{NIMH}})) - Wg_{\mathrm{NIMH}}) = 0 \tag{A13}$$

$$h14: \quad ((\mathrm{LION}_{\mathrm{const3}} * (L_{\mathrm{LION}} * W_{\mathrm{LION}} * T_{\mathrm{LION}})) - Wg_{\mathrm{LION}}) = 0 \tag{A14}$$

$$h15: \quad \mathrm{battery}_{\mathrm{talk\ time}} - (\mathrm{NIMH} * ((0.0053 * (\mathrm{capacity}_{\mathrm{NIMH}})) + 0.0269)) = 0 \tag{A15}$$

$$h16: \quad \mathrm{battery}_{\mathrm{talk\ time}} + (\mathrm{LION} * ((0.0061 * (\mathrm{capacity}_{\mathrm{LION}})) + 0.1667))) = 0 \tag{A16}$$

$$g1: \quad (\mathrm{NIMH} * L_{\mathrm{NIMH}} + \mathrm{LION} * L_{\mathrm{LION}}) - 0.60 * (\mathrm{SHELL} * L_{\mathrm{SHELL}} + \mathrm{FLIP} * L_{\mathrm{SHELL}}) \leq 0 \tag{A17}$$

$$g2: \quad (\mathrm{NIMH} * W_{\mathrm{NIMH}} + \mathrm{LION} * W_{\mathrm{LION}}) - 0.95 * (\mathrm{SHELL} * W_{\mathrm{SHELL}} + \mathrm{FLIP} * W_{\mathrm{FLIP}}) \leq 0 \tag{A18}$$

$$g3: \quad (\mathrm{NIMH} * T_{\mathrm{NIMH}} + \mathrm{LION} * T_{\mathrm{LION}}) - 0.45 * (\mathrm{SHELL} * T_{\mathrm{SHELL}} + \mathrm{FLIP} * T_{\mathrm{FLIP}}) \leq 0 \tag{A19}$$

3 Design Parameters.

$$\mathrm{A1}, \ldots, \mathrm{A8} = \{14.74, 5 \times 10^{-3}, 4 \times 10^{-2}, 1 \times 10^{-2}, 19.62, 8 \times 10^{-3}, 3 \times 10^{-3}, 3 \times 10^{-3}\}$$

$$\mathrm{NIMH}_{\mathrm{const1,2,3}} = \{21 \times 10^{-2}, 37 \times 10^{-4}, 9.8 \times 10^{-4}\}$$

$$\mathrm{LION}_{\mathrm{const1,2,3}} = \{43 \times 10^{-2}, 8.0 \times 10^{-4}, 8.8 \times 10^{-4}\}$$

$$\mathrm{SHELL}_{\mathrm{const1,2}} = \{2.29 \times 10^{-4}, 5.1 \times 10^{-4}\}$$

$$\mathrm{FLIP}_{\mathrm{const1,2}} = \{1.47 \times 10^{-4}, 4.9 \times 10^{-4}\}$$

4 Cell Phone Outer Casing Design Constraints.

$$h17: \quad (\mathrm{SHELL}_{\mathrm{const1}} * L_{\mathrm{SHELL}} * W_{\mathrm{SHELL}} * T_{\mathrm{SHELL}}) - \mathrm{cost}_{\mathrm{SHELL}} = 0 \tag{A20}$$

$$h18: \quad (\mathrm{FLIP}_{\mathrm{const1}} * L_{\mathrm{FLIP}} * W_{\mathrm{FLIP}} * T_{\mathrm{FLIP}}) - \mathrm{cost}_{\mathrm{FLIP}} = 0 \tag{A21}$$

$$h19: \quad (\mathrm{SHELL}_{\mathrm{const2}} * L_{\mathrm{SHELL}} * W_{\mathrm{SHELL}} * T_{\mathrm{SHELL}}) - Wg_{\mathrm{SHELL}} = 0 \tag{A22}$$

$$h20: \quad (\mathrm{FLIP}_{\mathrm{const2}} * L_{\mathrm{FLIP}} * W_{\mathrm{FLIP}} * T_{\mathrm{FLIP}}) - Wg_{\mathrm{FLIP}} = 0 \tag{A23}$$

$$g4: \quad L_{\mathrm{LCD}} - (0.60 * \mathrm{SHELL} * L_{\mathrm{SHELL}} + 0.60 * \mathrm{FLIP} * L_{\mathrm{FLIP}}) \leq 0 \tag{A24}$$

$$g5: \quad (0.30 * \mathrm{SHELL} * L_{\mathrm{SHELL}} + 0.30 * \mathrm{FLIP} * L_{\mathrm{FLIP}}) - L_{\mathrm{LCD}} \leq 0 \tag{A25}$$

$$g6: \quad W_{\mathrm{LCD}} - 0.90 * (\mathrm{SHELL} * W_{\mathrm{SHELL}} + \mathrm{FLIP} * W_{\mathrm{FLIP}}) \leq 0 \tag{A26}$$

$$g7: \quad 0.7 * (\mathrm{SHELL} * W_{\mathrm{SHELL}} + \mathrm{FLIP} * W_{\mathrm{FLIP}}) - W_{\mathrm{LCD}} \leq 0 \tag{A27}$$

$$g8: \quad L_{\text{OLED}} - (0.60 * \text{SHELL} * L_{\text{SHELL}} + 0.60 * \text{FLIP} * L_{\text{FLIP}}) \leq 0 \tag{A28}$$

$$g9: \quad (0.30 * \text{SHELL} * L_{\text{SHELL}} + 0.30 * \text{FLIP} * L_{\text{FLIP}}) - L_{\text{OLED}} \leq 0 \tag{A29}$$

$$g10: \quad W_{\text{OLED}} - 0.90 * (\text{SHELL} * W_{\text{SHELL}} + \text{FLIP} * W_{\text{FLIP}}) \leq 0 \tag{A30}$$

$$g11: \quad 0.7 * (\text{SHELL} * \text{width}_{\text{SHELL}} + \text{FLIP} * \text{width}_{\text{FLIP}}) - \text{OLED}_{\text{width}} \leq 0 \tag{A31}$$

5 Design Objective Constraints.

$$h20: \quad \text{TotalCost} - \sum_{i=1}^{N} \text{component}(i)_{\text{cost}} = 0 \tag{A32}$$

$$h21: \quad \text{TotalWeight} - \sum_{i=1}^{N} \text{component}(i)_{\text{weight}} = 0 \tag{A33}$$

References

[1] Moon, S. K., Kumara, S. R. T., and Simpson, T. W., 2006, "Data Mining and Fuzzy Clustering to Support Product Family Design," ASME Paper No. DETC2006/DAC-99287.

[2] Desai, P., Kekre, S., Radhakrishnan, S., and Srinivasan, K., 2001, "Product Differentiation and Commonality in Design: Balancing Revenue and Cost Drivers," Manage. Sci., **47**(1), pp. 37–51.

[3] deWeck, O., and Suh, E., 2006, "Flexible Product Platforms: Framework and Case Study," ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Sept.

[4] Farrell, R., and Simpson, T., 2003, "Product Platform Design to Improve Commonality in Custom Products," J. Intell. Manuf., **14**, pp. 541–556.

[5] Fellini, R., Kokkolaras, M., and Papalambros, P., 2006, "Quantitative Platform Selection in Optimal Design of Product Families, With Application to Automotive Engine Design," J. Eng. Design, **17**(5), p. 429–446.

[6] Fixon, S., 2005, "Product Architecture Assessment: A Tool to Link Product, Process, and Supply Chain Design Decisions," J. Operations Manage., **23**(3-4), pp. 345–369.

[7] Berry, S., and Pakes, A., 2007, "The Pure Characteristics Demand Model," Int. Econom. Rev., **48**(4), pp. 1193–1225.

[8] Thevenot, H., and Simpson, T., 2006, "A Comprehensive Metric for Evaluating Component Commonality in a Product Family," ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Sept.

[9] Pullmana, M. E., Mooreb, W. L., and Wardellb, D. G., 2002, "A Comparison of Quality Function Deployment and Conjoint Analysis in New Product Design," J. Prod. Innovation Manage., **19**(1), pp. 354–364.

[10] Cristiano, J. J., Liker, J. K., and White, C. W., III, 2000, "Customer-Driven Product Development Through Quality Function Deployment in the U.S. and Japan," J. Prod. Innovation Manage., **17**(4), pp. 286–308.

[11] Ashihara, K., and Ishii, K., 2005, "Application of Quality Function Deployment for New Business r and d Strategy Development," 2005 ASME International Mechanical Engineering Congress and Exposition, Orlando, FL.

[12] Lowe, A., Ridgway, K., and Atkinson, H., 2000, "QFD in New Production Technology Evaluation," Int. J. Prod. Econ., **67**, pp. 103–112.

[13] Green, P. E., Krieger, A. M., and Wind, Y. J., 2001, "Thirty Years of Conjoint Analysis: Reflections and Prospects," Interfaces, **31**(3), pp. 56–73.

[14] Moore, W. L., Louviere, J. J., and Verma, R., 1999, "Using Conjoint Analysis to Help Design Product Platforms," J. Prod. Innovation Manage., **16**, pp. 27–39.

[15] Grissom, M. D., Belegundu, A. D., Rangaswamy, A., and Koopmann, G. H., 2006, "Conjoint-Analysis-Based Multiattribute Optimization: Application in Acoustical Design," Struct. Multidiscip. Optim., **31**, pp. 8–16.

[16] Li, H., and Azarm, S., 2000, "Product Design Selection Under Uncertainty and With Competitive Advantage," ASME J. Mech. Des., **122**, pp. 411–418.

[17] Michalek, J. J., Feinberg, F. M., and Papalambros, P. Y., 2005, "Linking Marketing and Engineering Product Design Decisions Via Analytical Target Cascading," J. Prod. Innovation Manage., **22**, pp. 42–62.

[18] Olewnik, A. T., and Lewis, K. E., 2007, "Conjoint-HOQ: A Quantitative Methodology for Consumer-Driven Design," *Proceedings of the ASME 2007 IDET Conferences and CIE Conference IDETC/CIE 2007*, ASME, New York.

[19] Ben-Akiva, M., and Lerman, S. R., 1985, *Discrete Choice Analysis: Theory and Application to Travel Demand*, MIT, Cambridge, MA.

[20] Tucker, C. S., and Kim, H. M., 2007, "Product Family Decision Tree Concept Generation and Validation Through Data Mining and Multi-Level Optimization," *Proceedings of the 33rd ASME Design Automation Conference*, Las Vegas, NV, Sept. ASME, New York.

[21] 2006, *Decision Making in Engineering Design*, K. E. Lewis, W. Chen, and L. C. Schmidt, eds., ASME, New York.

[22] Wassenaar, H. J., and Chen, W., 2003, "An Approach to Decision-Based Design With Discrete Choice Analysis for Demand Modeling," ASME J. Mech. Des., **125**, pp. 490–497.

[23] Wassenaar, H. J., Chen, W., Cheng, J., and Sudjianto, A., 2005, "Enhancing Discrete Choice Demand Modeling for Decision-Based Design," ASME J. Mech. Des., **127**, pp. 514–523.

[24] Berry, S. T., 1994, "Estimating Discrete-Choice Models for Product Differentiation," Rand J. Econ., **25**(2), pp. 242–262.

[25] Kim, H. M., Kumar, D., and Chen, W., 2006, "Target Exploration for Disconnected Feasible Regions in Enterprise-Driven Multilevel Product Design," AIAA J., **44**, pp. 67–77.

[26] Kumar, D., 2007, "Demand Modeling for Enterprise-Driven Product Design," Ph.D. thesis, Northwestern University, Evanston, IL.

[27] Kusiak, A., and Smith, M., 2007, "Data Mining in Design of Products and Production Systems," Annu. Rev. Control, **31**, pp. 147–156.

[28] Nanda, J., Simpson, T., Kumara, S. R. T., and Shooter, S. B., 2006, "A Methodology for Product Family Ontology Development Using Formal Concept Analysis and Web Ontology Language," ASME J. Comput. Inf. Sci. Eng., **6**, pp. 103–113

[29] Tucker, C. S., and Kim, H. M., 2008, "Optimal Product Portfolio Formulation by Merging Predictive Data Mining With Multilevel Optimization," ASME J. Mech. Des., **130**, p. 041103.

[30] Quinlan, J., 1986, "Induction of Decision Trees," Mach. Learn., **1**(1), pp. 81–106.

[31] Hunt, B., Marin, J., and Stone, P., 1966, *Experiments in Induction*, Academic, New York.

[32] Agard, B., and Kusiak, A., 2004, "Data-Mining-Based Methodology for the Design of Product Families," Int. J. Prod. Res., **42**(15), pp. 2955–2969.

[33] Braha, D., 2001, *Data Mining for Design and Manufacturing*, Kluwer, Dordrecht, The Netherlands.

[34] Tsang, K. F., Lau, H. C. W., and Kwok, S. K., 2006, "Development of a Data Mining System for Continual Process Quality Improvement," Proc. Inst. Mech. Eng., Part B, **221**(2), pp. 179–193.

[35] Quinlan, J., 1992, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Vol. 1.

[36] Salzberg, S., 1994, "Book Review: C4.5: Programs for Machine Learning by J. Ross Quinlan," Mach. Learn., **16**, pp. 235–240.

[37] Li, X., and Oladdson, S., 2005, "Discovering Dispatching Rules Using Data Mining," J. Sched., **8**(6), pp. 515–527.

[38] Boulicaut, J., Esposito, F., Giannotti, F., and Pedreschi, D., 2004, *Knowledge Discovery in Databases: PKDD 2004*, Springer, New York.

[39] Bruha, I., and Franek, F., 1996, "Comparison of Various Routines for Unknown Attribute Value Processing: The Covering Paradigm," Int. J. Pattern Recognit. Artif. Intell., **10**(8), pp. 939–955.

[40] Grzymala-Busse, J. W., and Hu, M., 2001, "A Comparison of Several Approaches to Missing Attribute Values in Data Mining," LNAI 2005, pp. 378–385.

[41] Ling, J. M., Aughenbaugh, J. M., and Paredis, C. J., 2006, "Managing the Collection of Information Under Uncertainty Using Information Economics," ASME J. Mech. Des., **128**(4), pp. 980–990.

[42] Hand, D. J., 1998, "Data Mining: Statistics and More?," Am. Stat., **52**(2), pp. 112–118.

[43] Kotsiantis, S. B., Kanellopoulos, D., and Pintelas, P. E., 2006, "Data Preprocessing for Supervised Leaning," Int. J. Comput. Sci., **1**(2), pp. 111–117.

[44] McEntire, J., 2003, "D2K Toolkit User Manual," 1st ed., Office of Technology Management, National Center for Supercomputing Applications (NCSA), Urbana, IL, Apr.

[45] Campos, M., Stengard, P., and Milenova, B., 2005, "Data-Centric Automated Data Mining," Fourth International Conference on Machine Learning and Applications.

[46] Amor, N., Benferhat, S., and Elouedi, Z., 2004, "Naive Bayes Vs Decision Trees in Intrusion Detection Systems," 2004 ACM Symposium on Applied Computing.

[47] Grzymala-Busse, J. W., and Stefanowsk, J., 2001, "Three Discretization Methods for Rule Induction," Int. J. Intell. Syst., **16**, pp. 29–38.

[48] Perner, P., and Trautzsch, S., 1998, "Multi-Interval Discretization Methods for Decision Tree Learning," *Advances in Pattern Recognition*, Joint IAPR International Workshops, Springer-Verlag, Sydney, Australia, pp. 475–482.

[49] Quinlan, J., 1996, "Improved Use of Continuous Attributes in C4.5," J. Artif. Intell. Res., **4**, pp. 77–90.

[50] Quinlan, J., 1992, "Learning With Continuous Classes," Proceedings of the Artificial Intelligence, A. Adams and L. Sterling, eds., pp. 343–348.

[51] Breiman, L., Friedman, J., Olshen, R., and Stone, C., 1984, *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA.

[52] Kim, H. M., Michelena, N., Papalambros, P., and Jiang, T., 2003, "Target Cascading in Optimal System Design," ASME J. Mech. Des., **125**(3), pp. 474–480.

[53] Cooper, A. B., Georgiopoulos, P., Kim, H. M., and Papalambros, P. Y., 2006, "Analytical Target Setting: An Enterprise Context in Optimal Product Design," ASME J. Mech. Des., **128**, pp. 4–13.

[54] Kim, H. M., Rideout, D. G., Papalambros, P. Y., and Stein, J. L., 2003, "Analytical Target Cascading in Automotive Vehicle Design," ASME J. Mech. Des., **125**(3), pp. 481–489.

[55] Hidalgo, I. J., and Kim, H. M., 2006, "Multistage System of Systems Model by Analytical Target Cascading," Proceedings of the 11th AIAA/MAO Conference, Portmouth, VA, Sept.
[56] Tucker, C. S., and Kim, H. M., 2006, "Cell Phone Customer Survey," https://webtools.uiuc.edu/survey/Secure?id=5617516, accessed, Oct.
[57] Buchmann, I., 1999, "Battery Mystery Solved: Why Batteries for Digital Cell Phones Fail," Batteries Conference on Applications and Advances, Jan., pp. 359–362.
[58] Klepper, M., Miller, P., and Miller, L., 2003, *Advanced Display Technologies*, Printing Industry Center at Rochester Institute of Technology (RIT), Rochester, NY.