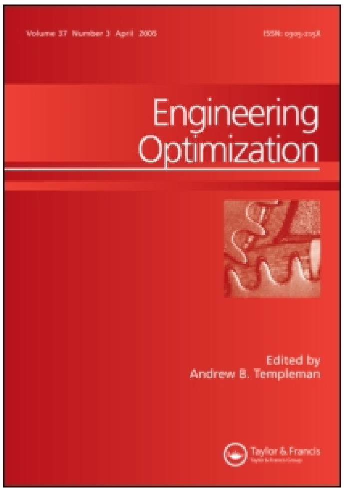


Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713641621

A ReliefF attribute weighting and X-means clustering methodology for top-down product family optimization

Conrad S. Tucker[a]; Harrison M. Kim[a]; Douglas E. Barker[a]; Yuanhui Zhang[a]
[a] University of Illinois, Urbana-Champaign, Illinois

First published on: 08 April 2010

To cite this Article Tucker, Conrad S. , Kim, Harrison M. , Barker, Douglas E. and Zhang, Yuanhui(2010) 'A ReliefF attribute weighting and X-means clustering methodology for top-down product family optimization', Engineering Optimization, 42: 7, 593 — 616, First published on: 08 April 2010 (iFirst)

To link to this Article: DOI: 10.1080/03052150903353328

URL: http://dx.doi.org/10.1080/03052150903353328

Engineering Optimization
Vol. 42, No. 7, July 2010, 593–616

A ReliefF attribute weighting and X-means clustering methodology for top-down product family optimization[1]

Conrad S. Tucker, Harrison M. Kim*, Douglas E. Barker and Yuanhui Zhang

University of Illinois, Urbana-Champaign, Illinois, 61801

(*Received 22 January 2010; final version received 14 September 2009*)

This article proposes a top-down product family design methodology that enables product design engineers to identify the optimal number of product architectures directly from the customer preference data set by employing data mining attribute weighting and clustering techniques. The methodology also presents an efficient component sharing strategy to aid in product family commonality decisions. Two key data mining models are presented in this work to help guide the product design process: (1) the ReliefF attribute weighting technique that identifies and ranks product attributes, and (2) the X-means clustering approach that autonomously identifies the optimal number of candidate products. Product family commonality decisions are guided by once again employing the X-means clustering technique, this time to identify the components across product families that are most similar. A family of prototype aerodynamic air particle separators is used to evaluate the efficiency and validity of the proposed product family design methodology.

Keywords: data mining; X-means clustering; ReliefF; bi-level quasi-separable problem; product architecture; aerodynamic particle separator

Nomenclature

AF	Air flow area.
f_k	Local product design objective function(s), a function of local design variables: $f_k\ (\mathbf{x}_k)$.
$\mathbf{R}^{Eng}$	Engineering design response (feasible/infeasible).
$\mathbf{T}^{Cj}$	Vector of product attributes represented by the cluster centroid in the data mining model.
$\mathbf{y}_{s,k}$	Linking variable at the engineering subsystem level cascaded up to system level.
ε_y	Deviation tolerance between linking variables.
ξ	Particle separation efficiency.
$\mathbf{w}'$	The vector of newly transformed ReliefF weights for target vector $\mathbf{T}^{Cj}$.
$\lVert\cdot\rVert_2^2$	Squared L-2 norm notation measuring the deviation between targets and responses.
k	kth candidate product architecture determined by the results of the X-means clustering.
K	The total number of cluster centroids C_j that exist for the X-means clustering solution.
$Cost_k$	Total product cost represented as the summation of individual component costs.

*Corresponding author. Email: hmkim@illinois.edu

ISSN 0305-215X print/ISSN 1029-0273 online

DOI: 10.1080/03052150903353328
http://www.informaworld.com

1. Introduction

In complex engineering systems that require a wide range of operating conditions, engineers are left with the challenge of designing product portfolios that meet customer preferences. The product family paradigm has been proposed to address the challenges of designing products for mass customization or for highly diversified customer functionality requirements. The term 'product family' is frequently defined in literature as a group of related products that share an underlying product design architecture (Messac *et al.* 2002, Alizon *et al.* 2007, Tucker and Kim 2007). The product family paradigm enables companies to standardize certain aspects of a product and at the same time provide product diversity to customers through product variants. Product cost savings may be realized as a result of product standardization due to 'economies of scale' (*e.g.* cost savings due to a standard manufacturing line for all products, rather than a specialized manufacturing line for each product). However, greater product standardization may also lead to lower product diversity in the market space and diminished product performance (*e.g.* limited customizable features for customers such as product colour, reliability, size, etc.) Therefore, in the product family approach, the level of product standardization versus product variety presents a trade-off scenario as product performance and appeal (from the customer's perspective) may diminish in an attempt to increase product standardization (Messac *et al.* 2002).

The product family design problem has been segmented into two well established domains: the 'Bottom-Up' approach and the 'Top-Down' approach (Alizon *et al.* 2007). In the 'Bottom-Up' approach, companies are more interested in making significant improvements to an existing product portfolio by combining products within the existing product family into a new product architecture. An assumption in the 'Bottom-Up' approach is that the newly redesigned product family will be able to satisfy customer needs through minimal additional technology investments. On the other hand, in the 'Top-Down' approach to product family design, the next generation of products is not based on an existing product family, but instead emerges from a market driven need. This need arises from the evolution of customer preferences far beyond what the current product portfolio can satisfy (Alizon *et al.* 2007). In this work, a 'Top-Down' product family methodology is proposed that analyses large customer preference data sets and identifies candidate product architectures that will be used in the product family design. This product design architecture can represent a group of design components that perform a series of functional processes. Products sharing similar product architecture can satisfy a broader range of customer requirements simply by possessing functionality capabilities that vary beyond the underlying architecture. The sharing of components also has the potential to reduce the time and costs associated with manufacturing diverse products. The challenge of developing a product architecture to be used in a product family presents an interesting design problem as the customer pool and functionality demands increase.

As data storage and information retrieval capabilities become more widely available, there is an emerging trend for companies to acquire and store customer preference data. For example, the physical characteristics (vehicle horsepower, number of doors, colour, etc.) of an automobile purchased by a customer visiting a dealership, along with the customer's demographic information (age, gender, household income, etc.) can enable auto manufacturers to determine emerging trends in the automotive industry and design next generation products accordingly. A great challenge in storing such data for product design purposes, however, is the non-homogeneity of customers, along with their individual preferences. Therefore, as the size of this non-homogeneous data increases, so does the complexity of identifying natural patterns within the data set. The ability to determine suitable product architectures for a particular group of customers becomes a challenge as enterprise decision makers and engineers attempt to extract meaningful patterns within the data set to aid in the product design and development process. Data mining in the context of product development is an emerging area of research that has the potential to significantly impact engineering design and manufacturing efforts (Kusiak 2006, Tucker and Kim 2007).

By identifying patterns within the large data set of customer preferences, engineers can incorporate this knowledge in the product family design process.

The product family design scenario that this article focuses on is described as follows. An enterprise is launching a portfolio of products that potentially share a subset of components. However, there are a few issues that must be resolved *prior to* the design process. First, the data set used in this product portfolio design scenario comprises large-scale non-homogeneous data which indicates that the product (the aerodynamic particle separator) undergoes a wide range of operating/environmental conditions. Second, the enterprise does not have a prior knowledge as to how many product variants should be introduced in the market, although customer preference data is available from survey or market research. Third, the enterprise does not know which components should be shared in case more than one product variant is introduced, although it has the flexibility to accommodate the component sharing decisions.

This article presents a product family design methodology that is driven by data mining capabilities, which resolves the product family design challenges presented above. Often in the engineering design process, quantifying attribute importance from a customer's perspective is challenging. Possessing a mechanism that can identify which performance attributes are more dominantly represented in the preference data set would help the engineering design teams focus resources in a more efficient manner. For this, the ReliefF (Kira and Rendell 1992) attribute weighting algorithm is employed to identify attributes in order of importance in the data set. Then, the X-means clustering algorithm is employed to identify groups of similar operating states within the raw data set. As a result, the number of product variants that should be introduced to reflect preferences (represented in the data) can be identified. Finally, the X-means clustering is applied again to the detailed designs of the initial product variants to identify which components may be shared among them. These sharing design decisions are implemented in a multi-disciplinary design optimization framework where an individual product variant is modelled as an individual subsystem.

The rest of the article is organized as follows. Section 2 provides research background followed by the proposed methodology in Section 3; the methodology is demonstrated in a case study in Section 4 followed by results and discussion in Section 5 and conclusion in Section 6.

2. Research background

A selective literature review on research areas pertaining to the concepts and techniques proposed in this work will be presented. These research areas were reviewed in a selective manner based on their relevance to data mining in product family design/product portfolio optimization.

2.1. *Data mining in product design*

There have been several researchers in the product design community that have incorporated data mining techniques in the product design process. For example, Agard and Kusiak (2004) utilize data mining clustering techniques to address the customer segmentation problem by determining a target market in a new product development process. Association rule mining is then used to discover attribute patterns in the segmented data (Agard and Kusiak 2004). Later works by Kusiak illustrate the benefits of data mining in a wide array of diversified industries such as biotechnology, energy, pharmaceutical, etc. (Kusiak 2006).

Tucker and Kim have incorporated data mining techniques in the product portfolio formulation process for extremely volatile markets (Tucker and Kim 2007, 2008). In such industries, product life cycles are short lived. Therefore, being able to correctly predict a customer's product preferences is paramount to increasing a product portfolio's chances of market success. Tucker and

Kim (2007) approached this design problem by systematically linking a customer's preferences, acquired through predictive data mining techniques, directly with engineering detailed design through multilevel optimization techniques (Kim *et al.* 2002, Kim *et al.* 2003).

Data mining techniques are also employed by Moon *et al.* (2006) in representing the functional requirements of customers. The proposed methodology uses fuzzy clustering techniques to determine the module composition of a product architecture (Moon *et al.* 2006). The work assumes that a product is an amalgam of module-based components with prior knowledge of the functionality capabilities of each module.

The primary contribution of this work is to present a product family design methodology for complex engineering systems that autonomously identifies the number of products to design by extracting weighted product preference information from a customer data set. This work focuses on design problems with large product preference data sets that can be integrated into the product design process. Since it would be impractical and highly expensive (from a cost and logistics standpoint) to design an independent system for each operating scenario, the proposed methodology instead identifies the most similar operating requirements given a large data set of operating conditions/scenarios. This in turn highlights the cost savings associated with product platform design through the concept of component sharing. This is modelled by the shared linking variable in the bi-level quasi-separable problem formulation that attempts to achieve an optimal design solution for each product while concurrently satisfying specific product functionality requirements (Tosserams *et al.* 2007). The term '*quasi-separable'* is used in this work to denote independent sub-problems that share a common design variable/component. In this work, sub-problem simply means a unique product design. The bi-level formulation is used in this work to co-ordinate these sharing decisions among sub-problems. Therefore, the individual sub-problem formulation for the bi-level quasi-separable problem is as follows:
Minimize

$$f_k(\mathbf{y}_{s,k}, \mathbf{x}_k) \tag{1}$$

Subject to:

$$g_k(\mathbf{y}_{s,k}, \mathbf{x}_k) \leq 0 \tag{2}$$

$$h_k(\mathbf{y}_{s,k}, \mathbf{x}_k) = 0 \tag{3}$$

For the quasi-separable formulation, each $\mathbf{x}_k$ represents the vector of local design variables unique to each sub-problem (k), where $k = 1, \ldots, K$ sub-problems. The vector of linking variables $\mathbf{y}_{s,k}$ makes the sub-problems quasi-separable as each sub-problem sharing a linking variable becomes influenced by the solution of other sub-problems sharing the same linking variable. A master problem is used to co-ordinate the linking variable among sub-problems and is explained in more detail in Section 3.2 of this work.

2.2. *Product family optimization*

The product family design paradigm has been investigated extensively throughout the engineering design community. Although there are a wide range of application areas, the underlying focus of product family optimization is to design a group of related products built around a common functional system architecture/platform. The aim is that commonality among product variants will reduce product design and manufacturing costs while still satisfying customer requirements. There have been many proposed methodologies and metrics for evaluating product commonality decisions in product family optimization. For example, the degree of commonality index (DCI) proposed by Collier (1981) measures the ratio of common components existing among products within a product family to the total number of components (Collier 1981). Later proposed commonality strategies such as the total constant commonality index (TCCI) (Wacker and Trelevan 1986), the commonality index (CI) (Martin and Ishii 1996, 1997, Khajavirad and Michalek 2007), component part commonality index ($CI^{(C)}$) (Jiao and Tseng 2000), product line commonality index (PCI) (Kota *et al.* 2000), the percent commonality index (%C) (Siddique *et al.* 1998), the generational variety index (GVI) (Martin and Ishii 2002), the functional similarity index (FSI) (McAdams *et al.* 1999, McAdams and Wood 2002), and the comprehensive metric for commonality (CMC) (Thevenot and Simpson 2007), propose strategies to help improve product commonality decisions by either rewarding or penalizing component sharing decisions. The aforementioned commonality indices are referenced in this work to give the reader a glimpse at the myriad of approaches available to address the issue of commonality in product design and development and how the proposed approach differs from them.

Instead of employing traditional commonality indices such as those listed above, product commonality decisions are investigated by employing the X-means clustering technique during the product family optimization process to identify similar components among product designs, hereby avoiding the need to exhaustively search all possible component sharing possibilities. In the aerodynamic particle separator problem that is investigated, the X-means clustering technique is first used to identify similarities among unique operating requirements. These clusters will form the basis of the individual product platform. For the aerodynamic particle separator problem, commonality decisions will be based primarily on the manufacturing costs associated with each unique design. The costs savings benefits of incorporating commonality decisions in the product family design process will be presented later in this work.

3. Methodology

Figure 1 is a flow diagram visually illustrating the sequence of the proposed product family design methodology. Figure 1 begins with the acquisition of raw customer product preference data and employs data mining attribute weighting and clustering techniques to determine the number of unique products needed for a given data set. One of the novel contributions proposed in this work, to solve the top-down product family research problem, is the ability to identify the optimal number of product architectures based solely on the data set. For products with highly diverse operating conditions, the data set itself may be highly heterogeneous making it quite difficult for engineers to determine the number of products to design in order to satisfy the market space. By employing the data mining ReliefF attribute weighting and X-means clustering techniques to the raw data set (Figure 1), engineers can determine the initial product architectures to design.

Steps 2 and 3 in Figure 1 illustrate the added benefits of component sharing by clustering similar products together in an attempt to reduce product design costs. The X-means clustering technique is employed at the engineering design level to determine which products are similar enough to potentially benefit from component sharing decisions. The details of Figure 1 will now be explained in depth in the following sections.

3.1. *Data mining product preferences*

The data mining of product preferences is the stage where dominant patterns are identified within the raw data set (Fayyad *et al.* 1996). With each unique instance in the data set representing a customer's preferred operating state for the product, the number of operating states can increase rapidly, thereby making it impractical for a single design to exist for each unique state. The engineering design goal is to identify those operating states within the data set that are similar in design requirements (as determined by the data mining algorithm). Since product attributes may vary in terms of design significance, an appropriate attribute weighting technique would help guide the engineering design process. To accomplish these product design challenges, the

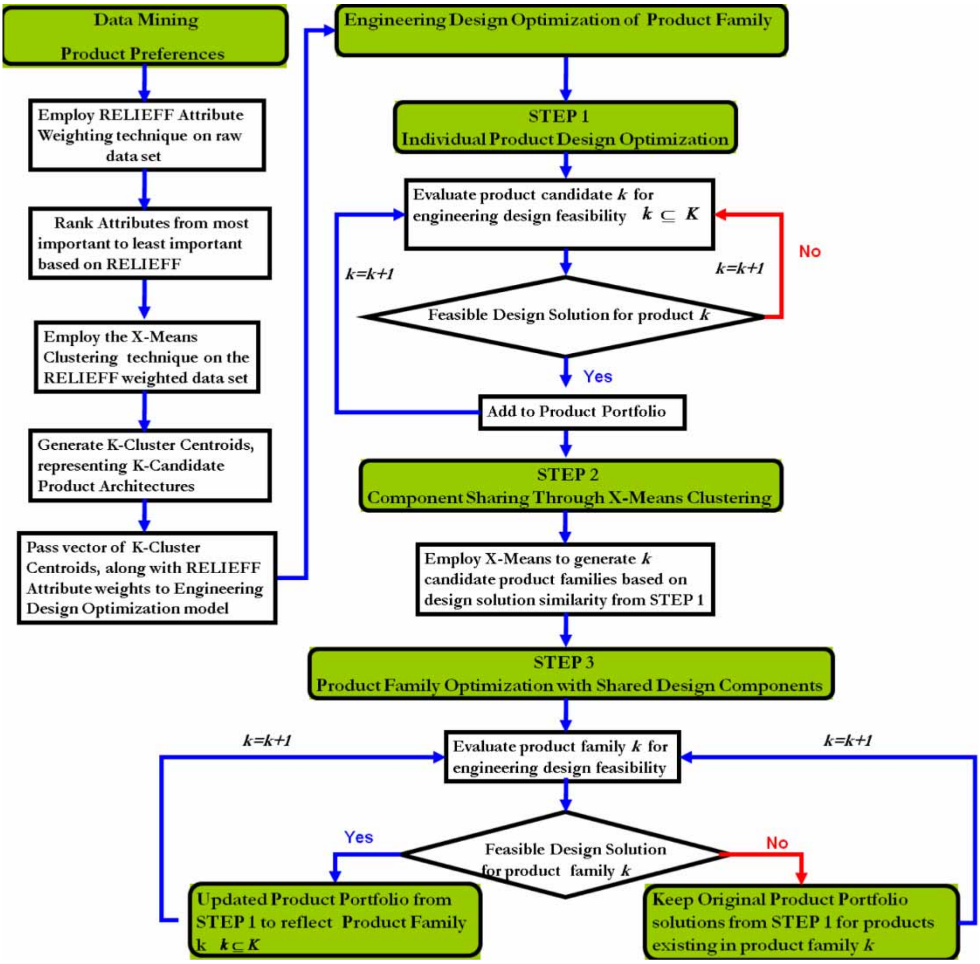


Figure 1. Flow diagram of proposed product family design methodology: From data mining product architecture identification to component sharing through X-means clustering.

ReliefF attribute weighting algorithm is first employed to weight attributes in order of importance in the data set (Kira and Rendell 1992). Then the X-means clustering algorithm is used to identify groups of similar operating states within the raw data set (illustrated in the data mining flow diagram in Figure 1 and visually represented on the left in Figure 2). The weighted attributes will influence both the data clustering process as well as the engineering design model as more valued product attributes will be given more weight in the overall product family design methodology. The X-means clustering algorithm is employed again in the component sharing decisions during the product family optimization stage as similar individual product designs are grouped together by similarity of design (Steps 2 and 3 in Figure 1 and visually represented on the right in Figure 2). Below is an introduction to the ReliefF attribute weighting technique that will later be applied to the raw data set.

3.1.1. *ReliefF product attribute weighting algorithm*

In this work, the enhanced version of the RELIEF algorithm called ReliefF is employed (Kononenko 1994). ReliefF extends the original RELIEF algorithm by enabling it to efficiently

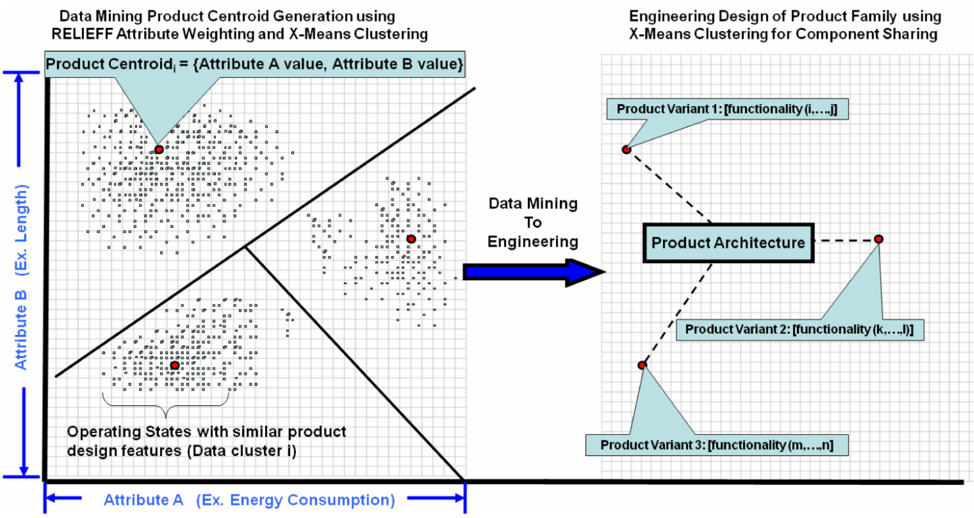


Figure 2. Visual representation of family design based on data mining ReliefF attribute weighting and X-means clustering for product centroid generation and X-means clustering for product family component sharing optimization.

handle multi-class variables and also missing values within the data set (Kononenko 1994). A class variable can be thought of as the response or predictor variable of interest. Examples of class variables in product design data sets may include efficiency, energy consumption, price, etc.

The original RELIEF algorithm proposed by Kira and Rendell (1992) is an attribute evaluation technique that will enable product development engineers to extract the importance of individual product attributes within a raw data set without explicit user provided ranking information (Kira and Rendell 1992). This can prove to be a vital time saving strategy, especially for extremely large data sets containing many attributes. Identifying the order of attribute relevance within a data set can reduce the overall computational complexity and increase the efficiency of data mining algorithms (Kira and Rendell 1992, Kononenko 1994).

Given a raw data set S, m instances are selected to serve as the number of sampled instances where p denotes the total unique attributes within the sample set m (Kira and Rendell 1992). The overall objective of the RELIEF algorithm is to take a random sample, and using a nearest neighbour search, to identify an identical class variable, which is defined as a NEAREST HIT (H), and also a different class variable that is nearest to the sample, defined as a NEAREST MISS (M) (Kira and Rendell 1992). The iterative process of RELIEF estimates attribute weights $W[A_i]$ based on their similarity to a given class, where A_i, represents a unique attribute within the data set. The general form of the algorithm can be represented as follows (Kira and Rendell 1992).

Given m, the desired number of sampled instances, and p, the number of attributes,

1. set all weights $W[A_i]: = 0.0$;
2. for $j: = 1$ to m do begin;
3. randomly select an instance X;
4. find nearest hit H and nearest miss M;
5. for $i: = 1$ to k do begin;
6. $W[A_i] := W[A_i] - \text{diff}(A_i; X; H)/m + diff(A_i; X; M)/m$;
7. end;
8. end;

where the *diff* function above measures the difference between the attribute being evaluated A_i taken from the randomly selected instance X, and the value of that same attribute given the closest hit (H) or closest miss (M). For discrete attributes, if the value of the attribute (A_i) of the randomly selected instance (X) matches that of the nearest hit (H) or nearest miss (M), then the *diff* value is 0 (meaning values are identical), otherwise (1) meaning they are different. For continuous attributes, the actual difference is used and then normalized on a scale of [0, 1].

By its design, the ReliefF attribute weighting technique does not constrain the attributes to non-negative values. Therefore, the weights will first be normalized based on the mini-max normalization (Han and Kamber 2006). For a given vector of attribute weights $[w_1, w_2, \ldots, w_p]$, the weights are normalized using the following formulation:

For weight $i = 1, \ldots, p$

$$w_i' = \frac{w_i - \min}{\max_w - \min_w}(new\_\max_w - new\_\min_w) + new\_\min_w \tag{4}$$

Here,

w_i'	The newly transformed weight (i) of attribute (i).
$\min_w$	The minimum value in the vector of ReliefF attribute weights.
$\max_w$	The maximum value in the vector of ReliefF attribute weights.
new_max_w	The maximum value of the new range.
$new\_\min_w$	The minimum value of the new range.

The new vector of weights $\mathbf{w}'$ determines the level of importance for each target vector $(\mathbf{T}^{Cj})$ at the engineering design level.

The data set with the newly updated weighted attributes will be used to:

- Weight clusters generated by the X-means clustering approach (discussed in the following section).
- Serve as attribute target weights for the engineering product design model.

3.1.2. *X-means clustering*

The X-means clustering algorithm in data mining is an enhancement of the k-means clustering algorithm (Pelleg and Moore 2000). Before investigating the X-means clustering algorithm and its significance in product family optimization, the k-means algorithm will be briefly described.

Given a raw data set of unique customer preferences (operating conditions), the k-means algorithm attempts to partition the original data set into k subsets of the data, where k represents the number of unique subsets or in the appropriate data mining terminology, clusters (Hartigan and Wong 1979, Jain and Dubes 1988). Each cluster contains a centroid, with data points of the cluster associated with this centroid. It is important to note that the number of clusters in the *k*-means algorithm is given *a priori* as a user defined input. In the context of product family design, this would be analogous to the engineering design team specifying the number of product platforms that the customer's product operating requirements must adhere to. Rather than design teams making postulations about the raw data set, a more natural process would be for the inherent patterns of the raw data set to help guide the product platform number (this is one of the contributions of the X-means data mining technique). Although there have been many enhancements to the *k*-means since its conception (Arora *et al.* 1998, Kanungo *et al.* 2002, Tarpey 2007), the basic underlying mathematical formulation can be represented as follows:

$$f = \sum_{j=1}^{K} \sum_{x_i \in S_j} \|x_i - c_j\|^2 \tag{5}$$

Here, S_j is a cluster of data points.
Here, S will be defined as all instances in the raw data set and, therefore, S_j would simply be a subset of this.
c_j is the centroid of a cluster S_j.
x_i is a data point existing within a cluster.
K is the total number of clusters (specified *a priori* by the user).

The iterative process of the k-means algorithm begins by initially selecting the desired number of clusters (S_j) and making an initial guess of the cluster centroid values (c_j) (Hartigan and Wong 1979). The next stage involves assigning a data point to the closest cluster centroid and centroid value (if necessary) by minimizing the error function in Equation (5) until negligible deviation occurs with each iteration.

The X-means clustering algorithm aims to improve on three key areas of the k-means algorithm (Pelleg and Moore 2000).

(1) Eliminating the need for number of clusters to be known as *a priori*.
(2) Improving the computational scalability.
(3) Enhancing the search criteria for updating cluster centroids.

The process by which X-means achieves these improvements is in part based on its selection criterion to determine when to add or replace a specific cluster centroid with child centroids. Child centroids originate from splitting the original solution of a k-means iteration and determining if the child clusters more accurately represent the data points once belonging to the parent centroid (Pelleg and Moore 2000). The posterior probabilities will be used to rank the models $\Pr[M_j|D]$, where D represents the given data set and M_j represents each model with a given cluster size k. The Bayesian information criterion (BIC) is used by X-means to rank which model is a more accurate representation of the original raw data set. Mathematically, the BIC is represented as follows (Kass and Wasserman 1995, Pelleg and Moore 2000):

$$BIC(M_j) = l_j(D) + \frac{p_j}{2} \log R \tag{6}$$

Here,
$l_j(D)$ is the log likelihood of the data taken at the maximum likelihood point.
D represents the given data set.
p_j represents the number of parameters in M_j.
R is the total number of data points of candidate centroids.

3.1.3. *Relevance of X-means to engineering product architecture design*

Engineering design problems involving a wide range of operating states specified by customers, can benefit from X-means clustering by identifying appropriate product functionality criterion for developing a product architecture and subsequent product family. The X-means clustering technique eliminates the need to guess the number of product architectures needed for a particular customer pool by analytically generating the appropriate number of clusters (product architectures) with corresponding product functionality specifications. A user instead specifies a broad range for the number of clusters and X-means will identify the optimal cluster, given the natural patterns within the data set (Pelleg and Moore 2000). This will ensure that the resulting product family will be a true representation of the data set for which the designs are made. Figure 2 illustrates how the cluster centroids of the X-means data mining clustering approach are integrated into the engineering design. The product centroids illustrated in Figure 2 represent the individual vectors of attribute value solutions that best describe *similar* groups of customers within the raw data. Each unique product centroid will form a vector of product preference targets used to guide the product architecture optimization process. The engineering design illustrated in Figure 2 represents the design of individual products based on the X-means cluster centroids where each product will have unique functionality characteristics that aim to satisfy the overall customer preference targets. Section 3.2.3 describes how product variants are then designed based on underlying product architecture under the notion of component sharing.

3.2. *Engineering design optimization of product family*

3.2.1. *Step 1: Individual product design optimization*

The results from the data mining stage provide product design engineers with several vital pieces of information. First, the results from the X-means clustering represent the vector of product attributes that form the product design targets ($\mathbf{T}^{Cj}$) around which a product architecture is designed (Step 1 in Figure 1). Product design targets can range anywhere from physical product dimension targets such as length or width to product performance targets such as efficiency or speed.

The second vital piece of information from the data mining stage is the relevance of each attribute target to the customer as determined by the ReliefF attribute weighting technique. That is, for each attribute target vector ($\mathbf{T}^{Cj}$), there will be an accompanying vector of attribute target weights $\mathbf{w}'$. The engineering product architecture optimization is comprised of the detailed engineering design model and incorporates the results from the data mining stage that help guide the product architecture design. Here, local design variables are used to model the physical dimensions and performance objectives of the product architecture subject to engineering design constraints.

The general mathematical model for the engineering product architecture optimization is as follows:

Note: The deviation is measured by the squared L-2 norm, which will be used throughout the engineering optimization models presented in this work. For example:

$$\|\mathbf{x} - \mathbf{y}\|_2^2 = \sum_i (x_i - y_i)^2.$$

For the kth product architecture,
Minimize

$$F(x)_{Architecture_{(k)}} = f_k + \mathbf{w}' \left\| \mathbf{T}^{C_j} - \mathbf{R}_k^{Eng} \right\|_2^2 \tag{7}$$

Subject to:

$$\mathbf{g}_k(\mathbf{x}_{k,}) \le \mathbf{0}$$
$$\mathbf{h}_k(\mathbf{x}_{k,}) = \mathbf{0}$$

Here,

f_k Local product design objective function (s), a function of local design variables: f_k ($\mathbf{x}_k$).
$\mathbf{T}^{Cj}$ Vector of product attributes represented by the cluster centroid in the data mining model. That is, for cluster centroid $C_j = [A_1, A_2, \ldots, A_p]$, target $\mathbf{T}^{Cj}$ is set as $[A_1, A_2, \ldots, A_p]$ where $A_1, A_2, \ldots, A_p$ represent attribute values for a given centroid C_j.
$\mathbf{w}'$ The vector of newly transformed ReliefF weights for target vector $\mathbf{T}^{Cj}$.

$\mathbf{R}_k^{Eng}$ Vector of engineering responses based on the formulation of the engineering design model. $\mathbf{R}_k^{Eng}$ is a function of local design variables $\mathbf{x}_k$ and is represented by $\mathbf{R}_k^{Eng}$ $(\mathbf{x}_k)$.
$\mathbf{g}_k$ Inequality design constraints bounding the product architecture model.
$\mathbf{h}_k$ Equality design constraints bounding the product architecture model.
K The kth candidate product architecture determined by the results of the X-means clustering.
K The total number of cluster centroids C_j that exist for the X-means clustering solution.

Note: It is important to note that although there may be K candidate product architectures to investigate, there may not always be a feasible design solution for the kth product architecture as generated product preference requirements may be too demanding, given the constraints of the engineering design model. That is, at optimality $k \leq K$.

3.2.2. *Step 2: Component sharing through X-means clustering*

If a feasible design solution exists after Step 1, X-means data mining clustering technique is once again employed, this time to determine the most similar product architecture design solutions within the product portfolio. While the first X-means clustering technique helped identify the similar groups of attributes in the raw data, the X-means clustering employed in Step 2 will help identify the similar groups of design variable values among the feasible product architecture design solutions (Step 2 in Figure 1).

For a given vector of design variables $(\mathbf{x}_k)$ of an optimal product architecture solution, (where the objective function $\mathrm{F}(\mathbf{x}_k)$ of product architecture (k) has been minimized given the external targets $\mathbf{T}^{Cj}$ and the local objective(s) $f_k(\mathbf{x}_k)$), the goal is to determine the similarity among product architecture variable solutions. The notion is that the closer the optimal design solutions are, for example $[(\mathbf{x}_k)$ and $(\mathbf{x}_{k+1})]$, the more likely these product architectures may be able to share certain design components.

3.2.3. *Step 3: Product family optimization with shared design components*

The third and final step in the proposed product family design methodology aims to reduce the product portfolio cost by sharing certain components among product architectures, thereby creating a family of products (Step 3 in Figure 1). Since the component sharing decision is inherently a combinatorial problem, Step 2 of the design methodology eliminates the need to search all possible component sharing combinations by guiding the component sharing decisions based on the optimal solution of each resulting product architecture. Once similar product architectures have been identified by the X-means technique in Step 2, the component variables are identified and modelled as linking variables $(\mathbf{y}_{s,k})$ in the quasi-separable bi-level problem. The model in Step 1 is adapted into a bi-level hierarchical optimization model where *level 1* strictly handles the coordination of the linking variables and *level 2* still remains the product architecture design level, but this time including the linking variable targets as part of the objective function. The bi-level design problem is modelled based on the quasi-separable problem (Kim 2001, Kim *et al.* 2002, Kokkolaras *et al.* 2002, Allison *et al.* 2006). A bi-level model is presented which comprises the component sharing co-ordination model at the upper level and the individual product design model at the lower level. At the component sharing level, updated linking variable values are distributed among product variants in an iterative manner until a feasible solution is achieved that is common among all product variants. If a feasible design solution does not exist for a given sharing scenario (that is, linking variable value $\mathbf{y}_S$ does not converge to a solution shared by all products), the original product design solutions (without shared variables) from Step 1 are kept.

Upper level: Component sharing co-ordination

The upper level (component sharing co-ordination) handles the coordination of linking variables to each of the product variants. In an effort to minimize design costs, certain design intensive and costly product components are shared among the different product variants. Under the quasi-separable formulation, these are represented as the linking variables ($\mathbf{y}_{s,k}$) . The sharing strategy is handled at the component sharing level wherein updated linking variable values from the lower level product architecture design are cascaded up to the component sharing level. Constraint Equation (9) is formulated as an inequality rather than an equality constraint due to numerical difficulties reported in the literature of equality constraint based bi-level formulations that fail to satisfy the standard Karush-Kuhn-Tucker (KKT) conditions for a constrained optimization problem (Alexandrov and Lewis 2002).
Minimize

$$\varepsilon_{\mathbf{y}} \tag{8}$$

Subject to:

$$g1 : \sum_{k\in Q} \left\| \mathbf{y}_s - \mathbf{y}_{s,k}^{Eng} \right\|_2^2 - \varepsilon_{\mathbf{y}} \leq 0 \tag{9}$$

Here,

$\mathbf{y}_s$ Linking variable at the upper level. In essence, $\mathbf{y}_s$ is simply a coordination variable ensuring that at the optimal solution, all of the subsystems attain the same value. Equation (9) is always active in the above formulation so solving for $\mathbf{y}_s$, it can be observed that at each iteration $\mathbf{y}_s$ assumes the average value of the linking variable(s) being shared across the products within the product family.

$\mathbf{y}_{s,k}^{Eng}$ Linking variable value at the lower level cascaded to the upper level. This is constant at each iteration in the above formulation that is subsequently updated at the engineering product architecture optimization level after each iteration.

k The kth candidate product architecture that has been identified for component sharing.

Q The total number of products that exist in a particular candidate product family. This is based on the X-means cluster solutions described in Step 2. The term candidate product family is used because until a feasible design solution can be achieved for the shared component case, these Q products will remain unique products within the product portfolio (note that $Q \leq K$ which simply means that the number of candidate product families cannot exceed the total number of unique products that initially exist).

ε_y Deviation tolerance between linking variables. For each shared variable, another constraint g(i) is added based on a similar formulation as equation (9) and add another tolerance variable in the objective function to represent this additional shared variable.

Lower level: Product family optimization

In the kth sub-problem,
Minimize

$$F(x)_{Architecture_{(k)}} = f_k + \mathbf{w}' \left\| \mathbf{T}^{C_j} - \mathbf{R}_k^{Eng} \right\|_2^2 + \left\| \mathbf{y}_s^U - \mathbf{y}_{s,k} \right\|_2^2 \tag{10}$$

Subject to:

$$\begin{aligned} \mathbf{g}_k(\mathbf{x}_{k,}\ \mathbf{y}_{s,k}) &\leq \mathbf{0} \\ \mathbf{h}_k(\mathbf{x}_{k,}\ \mathbf{y}_{s,k}) &= \mathbf{0} \end{aligned}$$

Here,

f_k Local product design objective function (s).

$\mathbf{T}^{Cj}$ Vector of product attributes represented by the cluster centroid in the data mining model. That is, for cluster centroid $C_j = [A_1, A_2, \ldots, A_p]$, target $\mathbf{T}^{Cj}$ is set as $[A_1, A_2, \ldots, A_p]$ where $A_1, A_2, \ldots, A_p$ represent attribute values for a given centroid C_j.

$\mathbf{w}'$ The vector of newly transformed ReliefF weights for target vector $\mathbf{T}^{Cj}$.

$\mathbf{R}_k^{Eng}$ Vector of engineering responses based on local design variables. $\mathbf{R}_k^{Eng}$ is a function of local design variables $\mathbf{x}_{k,}$ and is represented by $\mathbf{R}_k^{Eng}$ $(\mathbf{x}_{k,}\ \mathbf{y}_{s,k})$.

$\mathbf{g}_k$ Inequality design constraints.

$\mathbf{h}_k$ Equality design constraints.

$\mathbf{y}_s^U$ Linking variable target value cascaded down to the lower level from the upper level; a constant value at each iteration that is subsequently updated with each successful iteration.

$\mathbf{y}_{s,k}$ Linking variable at the lower level. This is local to the kth model and attempts to match the value of $\mathbf{y}_s^U$ at each iteration.

The overall flow of the proposed product family optimization is succinctly described below:

Bi-level product family optimization

Step 1:
Given $\mathbf{w}'$ vector of weights and $\mathbf{T}^{Cj}$ targets, where length($\mathbf{w}'$) = length($\mathbf{T}^{Cj}$) and K cluster centroids:

1. Solve K engineering design problems (**with no** linking variables $\mathbf{y}_{s,k}$) weighting each $\|\mathbf{T}^{C_j} - \mathbf{R}_k^{Eng}\|_2^2$ based on ReliefF;
2. If solution exists for the Individual Product Design Optimization Model (*i.e.*, *optimal* $\|\mathbf{T}^{C_j} - \mathbf{R}_k^{Eng}\|_2^2$ solution while satisfying local objectives and constraints);
3. Optimal solution found for weights $\mathbf{w}'$ and targets $\mathbf{T}^{Cj}$ without sharing components;

Step 2:

4. Employ **X-means** clustering to identify candidate product families based on solution similarities from **Step 1**;

Step 3:

5. Solve bi-level quasi-separable problem (component sharing among products) using the *Upper Level-Lower Level* formulation **with** linking variables $\mathbf{y}_{s,k}$;
6. If feasible solution exists (*i.e.*, *optimal* $\|\mathbf{T}^{C_j} - \mathbf{R}_k^{Eng}\|_2^2$ and $\|\mathbf{y}_s - \mathbf{y}_{s,k}^L\|_2^2$ at the *Lower Level* and also *optimal* ε_y at the *Upper Level*, (ε_y should be close to 0 at the *Upper Level*, indicating a feasible shared component among product variants within a product family));
7. Optimal solution found for weights $\mathbf{w}'$ and targets $\mathbf{T}^{Cj}$ and linking variables $\mathbf{y}_{s,k}$ for each product variant;
8. Else, solution does not exist for linking variable scenario; that is, sharing $\mathbf{y}_{s,k}$ is not feasible for product variants, therefore keep initial solutions found from **Step 1**;
9. end;
10. end;

4. Application: Aerodynamic particle separator case study

Indoor air quality (IAQ) is becoming an increasing concern for human health. Particulate matter is a leading cause of human respiratory illness in addition to degrading the performance of heating ventilation and air conditioning (HVAC) systems (WHO). As a result, abatement technologies for these aerosols are in high demand. Aerodynamic particle separators are filter-less air cleaning devices that can be capable of removing micron size particles with low energy consumption and minimal maintenance (Zhang 2005). Determining the optimal aerodynamic particle separator design for a specific application is challenging when taking into account its unique system requirements and environmental conditions.

4.1. *The engineering design problem*

4.1.1. *Aerodynamic particle separator design*

To demonstrate the effectiveness of the proposed design methodology, the design of a uniflow type particle separator is investigated (illustrated in Figure 3). The basic design of this device can be partitioned into three sections: (1) vane section, (2) straight region and (3) converging region/dust bunker. These sections are defined by eight design variables as shown in Figure 3 and Table 1.

The performance of an aerodynamic particle separator design is strongly dependent on system requirements and environmental conditions. System requirements such as the air cleaning efficiency, pressure drop (thus power consumption), air flow rate and overall device size contribute to the design objectives and directly define the constraints for a given application. Environmental conditions, including the air properties and contaminant particle size distribution can have a significant impact on the performance of a particular design and must also be incorporated into

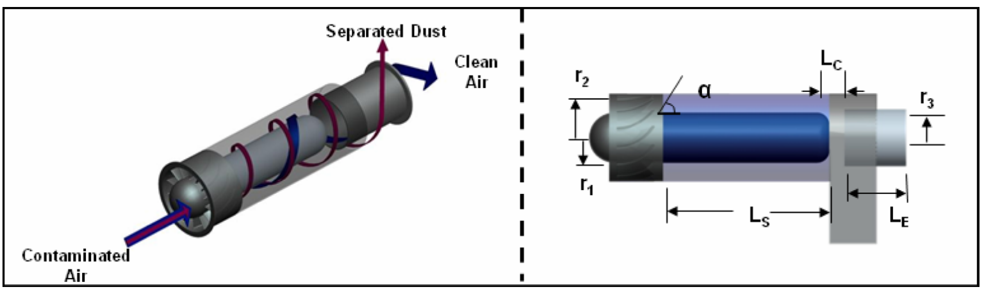


Figure 3. Uniflow type aerodynamic particle separator flow pattern and design variables.

Table 1. Design variable notation for aerodynamic particle separator.

Variable	Units	Description
r_1	Meters (m)	Inner tube radius
r_2	Meters (m)	Outer tube radius
L_S	Meters (m)	Maximum pressure drop
α	Radians (rad)	Vane discharge angle
L_C	Meters (m)	Length of converging gap
r_3	Meters (m)	Radius of exit tube
L_E	Meters (m)	Length of exit tube
N	#	Number of units in parallel

the system model (Barker 2008). Together, these two groups can be used to characterize a given application or operating state. These factors can vary by an order of magnitude between different applications, thereby complicating the design process. The product architecture design objective will, therefore, be to minimize cost while satisfying external product preference targets and local design constraints. The broader enterprise portfolio objective will be to minimize overall product design and development costs by capturing the component sharing opportunities that exist within the product portfolio.

4.2. *Data mining product preferences*

4.2.1. *Raw data set of product operating states*

A data set of 1000 operating states was generated to simulate the large variation in physical requirements and environmental conditions characterizing the broad range of applications in which aerodynamic particle separators are frequently employed (Barker 2008). Table 2 represents a snapshot of the 1000 operating states with distinct product attributes and environmental conditions represented by each column. Section 5 of the article presents the results from both the ReliefF attribute ranking algorithm and the X-means data mining clustering approach and demonstrates how the data mining process influences product family design efforts.

4.2.2. *ReliefF attribute weighting*

The results from the ReliefF attribute ranking approach in Table 3 reveal that the two engineering design targets—efficiency (ζ) and flow area ($AF_{\max}$)—have normalized weights of 0.1687 and 0.0785 respectively. The other attributes in Table 3 are used as design parameters in the engineering design model and, therefore, also play a significant role in the overall optimal solution. The

Table 2. Snapshot of aerodynamic particle separator data set consisting of 1000 states.

	Q (m3/s)	$\Delta_{P\,\max}$ (Pa)	$L_{\max}$ (m)	$A_{F\,\max}$ (m^2)	$N_{\max}$ # units	F(d_p) (%)	ρ_p (kg/m^3)	T_{air} (°C)	P_{air} (kPa)	Efficiency %	Price $
State 1	1.58	250	1	0.5	50	A1	2650	20	101	82	1,200
State 2	1.48	200	0.3	0.1	3	A4	2650	0	99	85	450
State 3	1.27	1500	1.5	1.5	16	Limestone	2700	500	200	90	900

Table 3. Attribute ranking of raw dataset via ReliefF algorithm.

Attribute Rank	Attribute Name	Attribute Weight	Normalized Attribute Weight
Highest	Nmax	0.0881	1.0000
	Efficiency	0.0110	0.1687
	Tair	0.0100	0.1575
	Pair	0.0099	0.1570
	Q	0.0073	0.1287
	Dpmax	0.0036	0.0888
	Afmax	0.0026	0.0785
	Lmax	0.0018	0.0695
	Rhop	−0.0006	0.0433
↓	Operating States	−0.0031	0.0169
Lowest	Fdp	−0.0046	0.0000

efficiency (ζ) and flow area ($AF_{\max}$) are selected based on the type of engineering problem being solved (in other applications, one may choose to set all attributes in the data set as targets for the product architecture design model). This vital information is a data pre-processing step that will help generate product cluster centroids that take into account the weighted attribute preferences of the different operating states given by the raw data set.

Note: ReliefF results were obtained using Weka version 3.5.8 (Witten and Frank 2005) and it took approximately 20 seconds running on a Intel Pentium Duo 2.5 GHz Processor. The normalized attribute weights are based on Equation (4) in Section 3.1.1.

4.2.3. *Data mining X-means clustering results*

The X-means clustering results reveal that a total of five clusters most accurately represents the similarities in the data set of 1000 operating states. The results from Table 4 represent the product design targets and parameters for the product portfolio of aerodynamic particle separators. Initially, each product centroid will be used to design an individual aerodynamic particle separator. Component sharing benefits will then be presented based on the vane section component. [Results attained using Weka version 3.5.8 (Witten and Frank 2005) and Data to Knowledge D2K (McEntire 2003)].

4.3. *Engineering design optimization of product family*

4.3.1. *Step 1: Individual product design optimization*

The X-means clustering algorithm generates $k = 1, \ldots, 5$ clusters, each with unique centroids C_j as illustrated in Table 4. Based on the results from the X-means clustering, and the ReliefF attribute weights accompanying each cluster centroid, engineers can now determine whether an optimal product design solution exists based on the aerodynamic particle separator response model.

The aerodynamic particle separator objective function attempts to match the particle separation efficiency target ($\zeta_k^{C_j}$) and the flow area target (AF^{C_j}) generated from the X-means clustering results while at the same time minimizing product design and manufacturing cost objective. The attributes within a cluster centroid (C_j) will form the design/environmental parameters of the model. The efficiency model selected was initially developed by Zhang (2005) and later augmented

Table 4. Product cluster centroids based on X-means clustering algorithm.

	System requirements					System requirements					
States	Q (m3/s)	$\Delta_{P\,max}$ (Pa)	L_{max} (m)	$A_{F\,max}$ (m^2)	N_{max} # units	$F(d_p)$ (%)	ρ_p (kg/m^3)	T_{air} (°C)	P_{air} (kPa)	Efficiency %	Price $
Product Centroid 1											
285	1.20	463	0.71	0.60	36	Limestone	2226	25	100	78	962
Product Centroid 2											
765	3.39	1507	0.79	0.68	39	A4	2211	71	111	86	1,168
Product Centroid 3											
750	3.36	1623	0.67	0.55	14	Limestone	2389	72	107	85	411
Product Centroid 4											
456	1.94	911	0.72	0.62	24	A1	2278	45	102	80	632
Product Centroid 5											
260	1.29	458	0.69	0.75	11	Limestone	2225	24	100	78	290

by Barker (2008). In this model, the flow is assumed to be fully turbulent and the steady state particle motion results from a balance between the centrifugal force and aerodynamic drag in the Stokes regime (Zhang 2005). The vector **x** contains the eight design variables as described by Table 1 and Figure 3. The cost function was based on the estimated mass of material required and injection moulding costs of the vane section. The material selected is an engineered polymer with a density of 1200 kg/m^3 at a cost of \$3.00 per kilogram. The injection moulding cost is estimated at a fixed cost of \$10,000 per design for the required capital equipment and labour. The efficiency model as a function of variables in **x** and particle size d_{p_i} is shown in Equation (11). The total efficiency for a given particle size distribution is then calculated by Equation (12).

$$\xi(\mathbf{x}, d_{p_i}) = 1 - \exp\left(-\frac{\rho_p d_{p_i}^2 C_c Q \tan(\alpha) L_S}{9\eta(r_2^2 - r_1^2)}\right) \cdot \exp\left(\frac{\rho_p d_{p_i}^2 C_c (V_t^2 G_t(\mathbf{x}) + V_z^2 G_r(\mathbf{x}))}{\eta V_z}\right) \quad (11)$$

$$\xi_T = \sum_{i=1}^{N} \xi(\mathbf{x}, d_{p_i}) \cdot F(d_{p_i}) \quad (12)$$

Here,

Cc	Cunningham slip correction factor.
d_{p_i}	Diameter of particle (i), m.
$F(d_p)$	Particle size distribution.
$G_t(x)$	Efficiency model geometric relationship between design variables, tangential acceleration.
$G_r(x)$	Efficiency model geometric relationship between design variables, radial acceleration.
ρ_p	Particle density, kg/m^3.
η	Air viscosity, Pa·s or kg· m/s.
Q	Air flow rate, m^3/s.
V_t	Tangential velocity of particle mixture.
V_z	Axial velocity of particle mixture.
r_1	Inner tube radius.
r_2	Inner tube radius.
α	Vane discharge angle.
L_S	Maximum pressure drop.

The engineering design model for the aerodynamic particle separator can be mathematically represented as:
kth aerodynamic particle separator
Minimize:

$$F(x)_{Architecture_{(k)}} = w'_\zeta \left\| \zeta_k^{C_j} - \zeta_k^{Eng} \right\|_2^2 + w'_{AF} \left\| AF_k^{C_j} - AF_k^{Eng} \right\|_2^2 + Cost_k \quad (13)$$

Subject to:
Pressure drop constraint (g1):

$$P_T(\mathbf{x}) - P_{\max} \leq 0 \quad (14)$$

Face area constraint (g2):

$$4r_2^2 N - AF_{\max} \leq 0 \quad (15)$$

Product length constraint (g3):

$$L_V + L_S + L_C + L_E - L_{\max} \leq 0 \tag{16}$$

Here,

AF_k	Maximum allowable face area perpendicular to air flow direction.
w'_ζ	Efficiency ReliefF attribute weight.
w'_{AF}	Flow area (AF) ReliefF attribute weight.
$L_{\max}$	Total allowable length of the system.
L_V	Length of vane section.
L_S	Length of straight region.
L_C	Length of converging region.
L_E	Length of exit tube.
$P_T(\mathbf{x})$	Total pressure drop of the system as a function of design variables **x**.
N	Number of aerodynamic particle separator units in one module.
$P_{\max}$	Maximum allowable pressure drop (air flow restriction).
$Cost_k$	Total product cost represented as the summation of individual component costs.

Note: The design model is also bounded by a set of linear inequality constraints $Ax \leq b$ and constraints Equations (14)–(16) that can be further expanded. A more detailed design model can be found in Barker (2008).

4.3.2. *Step 2: Component sharing through X-means clustering*

If an optimal solution exists for the aerodynamic product portfolio based on the X-means clustering targets, the next step is to determine whether additional costs savings can be realized by sharing the most design intensive components among different product architectures. The X-means clustering technique is employed to determine the similarities among the unique aerodynamic particle separator designs based on the solution results after Step 1. A successful sharing solution among products represents a unique product family. The results from the unique aerodynamic particle separator solutions can be seen in Table 5 which is further explained in Section 5.1.

4.3.3. *Step 3: Product family optimization with shared design components*

For the aerodynamic particle separator case study, the vane section is the most design intensive and costly component. The complex curved vanes must be injection moulded, which requires a unique mould to be machined for each vane section design. By employing the X-means clustering technique, product engineers will be able to (1) determine which product architecture designs are similar based on the solutions attained during Step 1 and (2) determine the number of candidate product families to include in the enterprise product portfolio based on the number of X-means cluster centroids generated. The L2 norm distance measure used by X-means will favour those design solutions that are numerically close to one another. This will help guide the sharing decision of the vane section as products with close numerical values for the variables that define the vane section (vane angle α, the inner and outer tube radii r_1 and r_2) will be favoured within a given cluster centroid.

Upper level: Component sharing co-ordination

The upper level (component sharing co-ordination) of the aerodynamic particle separator model will handle the co-ordination of the shared vane section among product families. The component

Table 5. Optimal solutions for individual aerodynamic particle separator designs.

Product	Design Variables r_1 (metres)	r_2 (metres)	Ls (metres)	$\acute{\alpha}$ (rad)	Lc (metres)	r_3 (metres)	AF (metres)	L (metres)	N	Product Effiency ζ(%)	Product Unit Cost $	# Units/ Cluster	Injection Mold Cost $	Total Product Cost $
Particle Separator 1	0.1732	0.1936	0.1098	1.0400	0.0109	0.1163	0.1207	0.6000	4	77.9994	69.01	226	10,000.00	25,595.41
Particle Separator 2	0.0707	0.0842	0.2709	1.0400	0.0088	0.0715	0.27971	0.6799	24	85.9915	171.03	207	10,000.00	45,403.45
Particle Separator 3	0.0248	0.0991	0.3539	1.0400	0.0991	0.0527	0.45301	0.5499	14	84.9963	865.53	173	10,000.00	24,970.02
Particle Separator 4	0.0691	0.0804	0.3696	1.0400	0.0804	0.0683	0.44998	0.6199	24	79.9955	189.70	233	10,000.00	54,199.65
Particle Separator 5	0.1758	0.1936	0.0932	1.0400	0.0095	0.1068	0.10267	0.7500	5	77.9993	85.36	161	10,000.00	23,742.41
Total Product Portfolio cost														173,910.93

sharing objective function will minimize the tolerance deviation variable of each shared variable. There are three variables that define the vane section, including the vane angle α, the inner and outer tube radii r_1 and r_2.

Minimize

$$\varepsilon_\alpha + \varepsilon_{r_1} + \varepsilon_{r_2} \tag{17}$$

Subject to:

$$g1 : \left\|\alpha_s - \alpha_{s,k}^{Eng}\right\|_2^2 - \varepsilon_\alpha \leq 0 \tag{18}$$

$$g2 : \left\|r_{1,s} - r_{1,s,k}^{Eng}\right\|_2^2 - \varepsilon_{r_1} \leq 0 \tag{19}$$

$$g3 : \left\|r_{2s} - r_{2s,k}^{Eng}\right\|_2^2 - \varepsilon_{r_2} \leq 0 \tag{20}$$

Here,

α_s Vane angle linking variable at the component sharing level.
$\alpha_{s,k}^{Eng}$ Value of vane angle linking variable response of engineering design model for product k.
$r_{1,s,k}$ Inner tube radius (r_1) linking variable at the component sharing level.
$r_{1,s,k}^{Eng}$ Value of inner tube radius (r_1) linking variable response of engineering design model for product k.
$r_{2,s,k}$ Outer tube radius (r_2) linking variable at the component sharing level.
$r_{2,s,k}^{Eng}$ Value of outer tube radius (r_2) linking variable response of engineering design model for product k.
ε_α Deviation tolerance variable between vane angle linking variable that is minimized in the objective function.
ε_{r1} Deviation tolerance variable between inner radius linking variable that is minimized in the objective function.
ε_{r2} Deviation tolerance variable between outer radius linking variable that is minimized in the objective function.

To minimize overall product portfolio costs, the number of unique vane section designs will be minimized by sharing this component with products that can attain a feasible design solution given this added objective. Equation (13) is, therefore, reformulated to reflect the candidate product families and also the shared vane components among each of these products within a given product family (represented as linking variables).

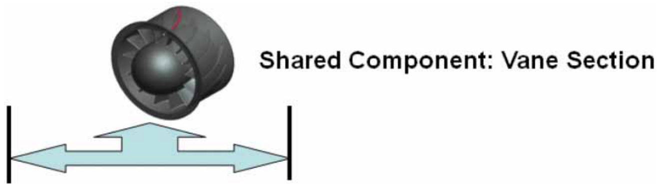
Lower level: Product family optimization

Minimize:

$$F(x)_{Architecture_{(k)}} = w'_\zeta \left\|\zeta_k^{C_j} - \zeta_k^{Eng}\right\|_2^2 + w'_{AF}\left\|AF_k^{C_j} - AF_k^{Eng}\right\|_2^2 + Cost_k + \left\|\alpha_{s,i} - \alpha_{s,k}^{Link}\right\|_2^2 + \left\|r_{1s} - r_{1s,k}^{Link}\right\|_2^2 + \left\|r_{2s} - r_{2s,k}^{Link}\right\|_2^2 \tag{21}$$

Subject to: Constraints as defined in Equations (14), (15) and (16).

Table 6. Optimal solutions for aerodynamic particle separator product families sharing the vane component.*



	Product Design Variables									Product Effiiency	Product Unit Cost	# Units/ Cluster	Injection Mold Cost	Total Product Cost
Product	r_1 (metres)	r_2 (metres)	Ls (metres)	$\acute{\alpha}$ (rad)	Lc (metres)	r_3 (metres)	AF (metres)	L (metres)	N	ζ(%)	$		$	$
Particle Separator Product Family 1														
Variant 1	0.1643	0.1935	0.2771	0.9532	0.0106	0.0884	0.5993	0.2877	4	77.9985	81.54	226	10,000.00	28,428.21
Variant 5	0.1653	0.1935	0.2991	0.9533	0.0097	0.0812	0.7487	0.3088	5	77.9979	105.10	161	Shared	16,921.90
Particle Separator Product Family 2														
Variant 2	0.0908	0.1064	0.5559	0.9446	0.0071	0.0905	0.6797	0.5630	15	95.84444	196.45	207	10,000.00	50,664.52
Variant 3	0.0903	0.1070	0.0509	0.9446	0.0072	0.0686	0.5495	0.0581	12	84.9991	74.77	173	Shared	12,935.74
Particle Separator Product Family 3														
Variant 4	0.069059	0.080357	0.369626	1.04	0.080357	0.068303	0.449983	0.619893	24	79.995506	189.70	233	10,000.00	54,199.65
Total Product Portfolio cost														163,150.02

*Optimal results attained using Matlab® and Tomlab® to solve the mixed integer nonlinear programming problem (Griffiths 2005, Holmstrom *et al.* 2006).

5. Results and discussion

5.1. *Aerodynamic particle separator optimization results*

Given the product design targets from the data mining X-means clustering step, the aerodynamic particle separator model first attempts to identify feasible design solutions for the efficiency (ζ^{Cj}), flow area (AF^{Cj}) targets and given physical and environmental (T_{air}, P_{air}, etc.) parameters for each unique cluster centroid (C_j). The aerodynamic particle separator solutions in Table 5 reveal that a total of five unique products can be designed for the initial five cluster centroids targets generated by the X-means clustering with a total product portfolio cost of \$173,910.

If Step 1 of the product family design methodology is successful, engineers can further investigate the potential costs savings (Steps 2 and 3) that may be realized due to component sharing. The X-means clustering technique performed during Step 2 reveals that out of the five unique aerodynamic particle separator solutions, products 1 and 5 form a feasible unique product family cluster, products 2 and 3 another and finally product 4 cannot be shared with any other product and, therefore, reverts back to the original solution from Step 1. The cost of the injection mould manufacturing process presents an opportunity for the initial product portfolio of five unique products to be redesigned. The vane section of the product which is made through the injection moulding process is shared among similar products existing in the original portfolio. In this case study, the decision to share the vane angle is known *a priori* due to the high cost of designing each individual injection mould for the vane. Step 3 of the product family design methodology employs the X-means clustering algorithm to identify products that have similar vane design solutions. The decision to share the vane angle is an attempt to minimize the overall costs of the enterprise product portfolio by minimizing the number of unique vane sections needed for the five aerodynamic particle separators. Products successfully sharing a vane section will be considered a unique product family and each product existing in this product family, is defined as a variant. However, it must be noted that the cost savings benefits of component sharing using the product family approach to design may be offset by the decrease in the performance capabilities attainable by the newly designed product variants. This trade-off scenario will, therefore, be based on how much cost savings can be realized through component sharing and how much performance deviation can be accommodated by the customer.

The results in Table 6 reveal that sharing products 1 and 5, 2 and 3 (with product 4 being a separate unique design), reduces the total product portfolio cost to \$163,150; a total savings of approximately \$10,760 for this product portfolio design scenario. However, it can be observed that the efficiency of product 2 decreases from 85.99% with the individual optimization model solution (Table 5) to 85.84% with the component sharing product family model solution (Table 6). The level of allowable performance deviation will be dependent on customer expectations and the level of competition within the market space. Although a feasible design may not always exist for every sharing scenario (for example sharing a single vane component for each of the five products returned an infeasible solution), the benefits of investigating sharing strategies through the X-means clustering recommendations may prove beneficial as can be seen from the results in Table 6.

6. Conclusion

In this work, a comprehensive product family design methodology is presented that integrates realistic product operation data with the engineering design of complex products such as the aerodynamic particle separator. The data mining ReliefF algorithm is employed to determine the weights of each attribute. This information is then incorporated into the data mining X-means clustering algorithm in order to generate the number of clusters along with the cluster centroids that are inherent to the data itself. The results of the data mining clustering technique aid in determining the number of unique products to design for a group of highly diverse customers. With this clustering information, a product architecture can be designed that takes into account specific customer product functionality needs that are represented in a large data set. Further cost savings can be realized through a component sharing strategy that is achieved in this work by once again employing the X-means clustering technique to identify similar design solutions. The hope is to expand on the concepts presented in this work by enabling the feasibility of the product architecture optimization step to influence the generation of X-means cluster centroids. That is, local objective functions may be highly sensitive to certain local design variables which can be taken into account during the X-means clustering step.

Acknowledgements

This material is based on work supported by the National Science Foundation under Award No. 0726934. 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.

Note

1. The initial version of this article was presented at the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, British Columbia, Canada.

References

Agard, B. and Kusiak, A., 2004. Data-mining-based methodology for the design of product families. *International Journal of Production Research*, 42 (15), 2955–2969.

Alexandrov, N.M. and Lewis, R.M., 2002. Analytical and computational aspects of collaborative optimization for multidisciplinary design. *AIAA Journal*, 40 (2), 301–309.

Alizon, F., Khadke, K., Thevenot, H.J., Gershenson, J.K., Marion, T.J., Shooter, S.B. and Simpson, T.W., 2007. Frameworks for product family design and development. *Concurrent Engineering*, 15 (2), 187–199.

Allison, J., Walsh, D., Kokkolaras, M., Papalambros, P.Y. and Cartmell, M., 2006. Analytical target cascading in aircraft design. *44th AIAA aerospace sciences meeting and exhibit*, 9–12 January, Reno, Nevada. Reston, VA: AIAA, 16112–16120.

Arora, S., Raghavan, P. and Rao, S., 1998. Approximation schemes for euclidean k-median and related problems. *Proceedings of the 30th annual ACM symposium theory of computing*, 23–26 May, Dallas, TX, New York: ACM Press, 106–113.

Barker, D., 2008. *Development of an optimization design platform for aerodynamic particle separators*. Thesis (Masters). University of Illinois at Urbana-Champaign.

Collier, D., 1981. The measurement and operating benefits of component part commonality. *Decision Sciences*, 12 (1), 85–96.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P., 1996. From data mining to knowledge discovery in databases. AI Magazine, 17, 37–54.

Griffiths, D.F., 2005. *An introduction to matlab*. Nethergate, Dundee, DD1 4HN, Scotland, UK: University of Dundee.

Han, J. and Kamber, M., 2006. *Data mining concepts and techniques*. 2nd edn. San Francisco, CA: Morgan Kaufmann.

Hartigan, J.A. and Wong , M.A., 1979. A k-means clustering algorithm. *Applied Statistics*, 28 (1), 100–108.

Holmstrom, K., Goran, A.O. and Edvall, M.M., 2006. *User's guide for tomlab/minlp*. 855 Beech St 121, San Diego, CA, USA: Tomlab Optimization Inc.

Jain, A.K. and Dubes, R.C., 1988. *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice Hall.

Jiao, J. and Tseng, M.M., 2000. Understanding product family for mass customization by developing commonality indices. *Journal of Engineering Design*, 11 (3), 225–243.

Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R.S. and Wu, A.W., 2002. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 (7), 881–892.

Kass, R.E. and Wasserman, L., 1995. A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association*, 90 (431), 928–934.

Khajavirad, A. and Michalek, J.J., 2007. An extension of the commonality index for product family optimization. *Proceedings of the ASME 2007 IDET and CIE conference*, 4–7 September, Las Vegas, Nevada, USA. Washington D.C.: ASME Press, 1001–1010.

Kim, H.M., 2001. *Target cascading in optimal system design*. Dissertation (PhD). The University of Michigan.

Kim, H.M., Kokkolaras, M., Louca, L.S., Delagrammatikas, G.J., Michelena, N.F., Filipi, Z.S., Papalambros, P.L., Stein, J.L. and Assanis, D.N., 2002. Target cascading in vehicle redesign: A class vi truck study. *International Journal of Vehicle Design*, 3 (3), 199–225.

Kim, H.M., Michelena, N.F. and Papalambros, P.Y., 2003. Target cascading in optimal system design. *Transactions of ASME: Journal of Mechanical Design*, 125 (3), 474–480.

Kira, K. and Rendell, L., 1992. A practical approach to feature selection. *Proceedings of international conference on machine learning*, 1–3 July, Aberdeen. San Francisco: Morgan Kaufmann, 249–256.

Kokkolaras, M., Fellini, R., Kim, H.M., Michelena, N.F. and Papalambros, P.Y., 2002. Extension of the target cascading formulation to the design of product families. *Structural and Multidisciplinary Optimization*, 24 (4), 293–301.

Kononenko, I., 1994. Estimating attributes: Analysis and extensions of relief. *Proceedings of European conference in machine learning*, 6–8 April, Catania, Italy. New York: Springer-Verlag, 171–182.

Kota, S., Sethuraman, K. and Miller, R., 2000. A metric for evaluating design commonality in product families. *Transactions of ASME: Journal of Mechanical Design*, 122 (4), 403–410.

Kusiak, A., 2006. Data mining: Manufacturing and service applications. *International Journal of Production Research*, 44 (19), 4175–4191.

Martin, M. and Ishii, K., 1996. Design for variety: A methodology for understanding the costs of product proliferation. *Proceedings of DETC 1996*, 18–22 August, Irvine, CA. New York: ASME Press.

Martin, M.V. and Ishii, K., 2002. Design for variety: Developing standardized and modularized product platform architectures. *Research in Engineering Design*, 13 (4), 213–235.

Martin, M.V. and Ishii, K., 1997. Design for variety: Development of complexity indices and design charts. *Proceedings of DETC 1997*, 14–17 September, Sacramento, CA. New York: ASME Press.

McAdams, D.A., Stone, R.B. and Wood, K.L., 1999. Functional interdependence and product similarity based on customer needs. *Research in Engineering Design*, 11 (1), 1–19.

McAdams, D.A. and Wood, K.L., 2002. A quantitative similarity metric for design-by-analogy. *Transactions of ASME: Journal of Mechanical Design*, 124 (2), 173–182.

Mcentire, J., 2003. *D2k toolkit user manual*. Urbana, IL: National Center for Supercomputing Applications (NCSA).

Messac, A., Martinez, M.P. and Simpson, T.W., 2002. Effective product family design using physical programming. *Engineering Optimization*, 34 (3), 245–261.

Moon, S.K., Kumara, S.R.T. and Simpson, T.W., 2006. Data mining and fuzzy clustering to support product family design. *Proceedings of the ASME design automation conference*, 10–13 Spetember, Philadelphia, PA. New York: ASME Press.

Pelleg, D. and Moore, A., 2000. X-means: Extending k-means with efficient estimation of the number of clusters. *Proceedings of the 17th international conference on machine learning*, 29 June–2 July, Stanford University, California. San Francisco, CA: Morgan Kaufmann, 727–734.

Siddique, Z., Rosen, D.W. and Wang, N., 1998. On the applicability of product variety design concepts to automotive platform commonality. *Proceedings of DETC 1998*, 13–16 September, Atlanta, GA. Paper no. DETC/DTM-5661.

Tarpey, T., 2007. A parametric k-means algorithm. *Computational Statistics*, 22, 71–89.

Thevenot, H.J. and Simpson, T.W., 2007. A comprehensive metric for evaluating component commonality in a product family. *Journal of Engineering Design*, 18 (6), 577–598.

Tosserams, S., Etman, L.F.P. and Rooda, J.E., 2007. An augmented lagrangian decomposition method for quasi-separable problems in mdo. *Structural and Multidisciplinary Optimization*, 34 (3), 211–227.

Tucker, C.S. and Kim, H.M., 2007. Product family concept generation and validation through predictive decision tree data mining and multi-level optimization. *Proceedings of IDETC/CIE 2007*, 4–7 September, Las Vegas, NV, 971–987.

Tucker, C.S. and Kim, H.M., 2008. Optimal product portfolio formulation by merging predictive data mining with multilevel optimization. *Transactions of ASME: ASME Journal of Mechanical Design*, 130 (4), 0441103-1–15.

Wacker, J.G. and Trelevan, M., 1986. Component part standardization: An analysis of commonality sources and indices. *Journal of Operations Management*, 6 (2), 219–244.

WHO, [online]. Available from: www.who.int/mediacentre/news/releases/2006/pr52/en/index.html [Accessed September 2009].

Witten, I.H. and Frank, E., 2005. *Data mining: Practical machine learning tools and techniques*. 2nd edn. San Francisco, CA: Morgan Kaufmann.

Zhang, Y., 2005. *Indoor air quality engineering*. Boca Raton, FL: CRC Press.