Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-84927

A METHODOLOGY TO SUPPORT PRODUCT FAMILY REDESIGN USING A GENETIC ALGORITHM AND COMMONALITY INDICES

Henri J. Thevenot,[1] Jyotirmaya Nanda,[1] and Timothy W. Simpson[2]
The Harold & Inge Marcus Department of Industrial & Manufacturing Engineering
The Pennsylvania State University
University Park, PA 16802 USA

ABSTRACT

Many of today's manufacturing companies are using platform-based product development to realize families of products with sufficient variety to meet customers' demands while keeping costs relatively low. The challenge when designing or redesigning a product family is in resolving the tradeoff between product commonality and distinctiveness. Several methodologies have been proposed to redesign existing product families; however, a problem with most of these methods is that they require a considerable amount of information that is not often readily available, and hence their use has been limited. In this research, we propose a methodology to help designers during product family redesign. This methodology is based on the use of a genetic algorithm and commonality indices - metrics to assess the level of commonality within a product family. Unlike most other research in which the redesign of a product family is the result of many human computations, the proposed methodology reduces human intervention and improves accuracy, repeatability, and robustness of the results. Moreover, it is based on data that is relatively easy to acquire. As an example, a family of computer mice is analyzed using the Product Line Commonality Index. Recommendations are given at the *product family level* (assessment of the overall design of the product family), and at the *component level* (which components to redesign and how to redesign them). The methodology provides a systematic methodology for product family redesign.

Keywords: Product Family Redesign, Commonality Indices, Genetic Algorithm

1. INTRODUCTION

Today's marketplace is highly competitive, global and volatile: customer demands are constantly changing, and they seek wider varieties of products at the same price as mass-produced goods. A common approach for providing variety without losing commonality is to develop product platforms and product families. There are two recognized approaches to product family design [1]. The first is a *top-down (proactive platform) approach*, wherein the company's strategy is to develop a family of products based on a product platform and its derivatives. Examples include Sony Walkmans [2] and Kodak one-time-use cameras [3]. The second is a *bottom-up (reactive design) approach*, wherein a company redesigns and/or consolidates a group of distinct products to standardize components and thus reduce costs. For example, Black & Decker redesigned their products to reduce variety in their motors [4], and Lutron redesigned its product line of lighting control systems around 15-20 standard components that can be configured into more than 100 models specified by the customers [5]. A recent review of several examples of product families can be found in Ref. [6]. In this work, the focus is on supporting a bottom-up approach to platform redesign, starting from an existing product family.

Two common approaches to product family redesign are *commonality* and *modularity*. To assess the degree of *commonality* within a product family, several commonality indices have been developed (see Section 2.3). An extensive comparison between many of these commonality indices and their usefulness for product family design or redesign can be found in Ref. [7]; the work in this paper comes as an extension of this research. *Modularity* arises from the decomposition of a product family into modules. Several studies regarding the measure of product modularity and methods to achieve modularity in product redesign can be found in the literature: a recent overview of modularity and its benefits can be found in Ref. [8], and a comparison of existing measures of product modularity is documented in Ref. [9]. The issue with all of these proposed methods is that they are currently not systematic, i.e, most of the methods requiring a lot of human computation and a considerable amount of information that is not always readily available. There is also a lack of

---

[1] Graduate Research Assistant, Department of Industrial & Manufacturing Engineering
[2] Associate Professor of Mechanical and Industrial Engineering. **Corresponding Author**. Phone/fax: (814) 863-7136/4745. Email: tws8@psu.edu.

methodologies to evaluate the impact of each component within a product family on the degree of commonality within the family or to determine the optimal level of commonality. Consequently, there is a need for less information-intensive measures that are useful during concept development and layout design [9].

In this paper, a methodology for using commonality indices to support product family redesign is introduced. The proposed methodology uses simple data as inputs: a list of components in each product with related information (cost, component connection, manufacturing process, etc.). The list of components is either obtained from a bill of materials, or, if not available, a dissection of the product family is performed. Using this data, commonality indices are evaluated to assess the commonality of the whole family, and a genetic algorithm is then implemented to maximize the value of these commonality indices. The methodology provides recommendations on how to improve the redesign of a product family. In Section 2, the methodology is described and is then applied to a family of computer mice in Section 3. Closing remarks and future work are given in Section 4.

2. SUPPORTING PRODUCT FAMILY REDESIGN USING COMMONALITY INDICES

2.1. Methodology For Redesign

In this paper, the methodology shown in Figure 1 is implemented. Details about the methodology follow.

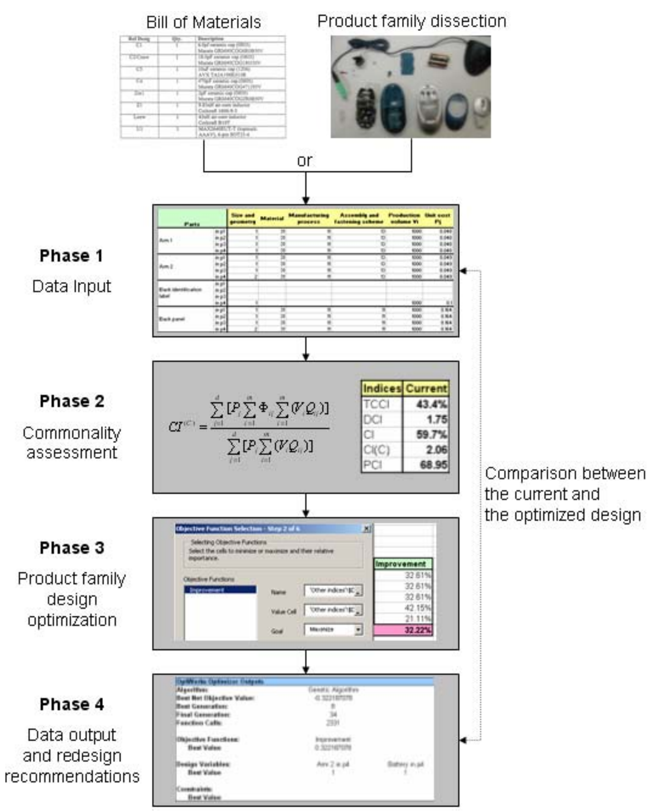

Figure 1. Proposed methodology for product family redesign

The first phase is to ask the user to enter basic information about the product family being studied. Either the information is readily available, or the designer can dissect the products in the family to obtain the necessary data. In the second phase, the assessment of the level of commonality in the family is realized through the computation of commonality indices. The third phase is the use of a genetic algorithm to maximize the level of commonality in the family subject to specific constraints. The fourth phase is the generation of recommendations based on these results.

2.2. Phase 1: Data Input

The first phase in this methodology is to obtain the necessary data for the product family concerned. If the information is already available through a bill of materials, for example, the user enters the appropriate data. If the information is not available, a dissection of the product family is required. To ensure consistency in the dissection, each product within the family is dissected to the lowest level possible, i.e., the parts cannot be further divided into subassemblies. However, some assemblies can be difficult, if not impossible, to dissect to that extent, such as electronic printed circuit boards, which are taken as a single part for analysis. For each part, the data collected are the following:

- *Size and geometry*: this information is used to compare wheter parts are common, variant or unique throughout the product family. A unique part is a part only used by one product in the family. A variant part has the same function between some or all the products in the family, but the design, shape and/or material differ slightly from one product to the next. A common part is the exact same part shared by some or all of the products in the family.
- *Material*: the material of each part is stored.
- *Manufacturing process*: the way the part is produced is also recorded, to see if manufacturing processes can be standardized between the variant parts in a product.
- *Assembly and fastening scheme*: the way the parts are assembled and fastened together is stored.
- *Production volume*: this value enables the designer to “weight” the products in the family, depending on the quantity produced.
- *Unit cost*: the cost of each part aims at giving more weight to the variant parts that are expensive to produce, parts that will first be considered during product family redesign.

For the type of material, the manufacturing process and the assembly scheme, a list of possible choices is given to the designer (see Appendix A for materials, Appendix B for manufacturing processes, and Appendix C for assembly and fastening schemes) based on Ref. [10]. Note that this list is not exhaustive, and the designer can add any data if desired. For the production volume and the unit cost, the data can be either very easy to obtain, but if not available, costs should be estimated using appropriate methods (such as the one found in Ref. [11]).

2.3. Phase 2: Commonality Assessment

To measure the commonality within a product family, several commonality indices have been proposed in the literature. A commonality index is a metric to assess the degree of commonality within a product family. It is based on different parameters such as the number of common components, the component costs, the manufacturing processes, etc. These indices are often the starting point when designing a new family of products or when analyzing an existing family. They are intended to provide valuable information about the degree of commonality achieved within a family and how to improve a product's design to increase commonality in the family and reduce costs. Table 1 gives a list of five indices that could be used in this research, based the data described in Section 2.2; a complete description and detailed comparison of each index can be found in Ref. [12]. These indices are chosen as they are component-based, and they can be easily computed with relatively limited information, such as the parts in the products, their materials, etc. The indices can be computed using the data collected in Phase 1. Ref. [7,12,13] gives more details about these computations.

Table 1. Commonality indices

Name		Developed by	Commonality measure for:	No Commonality	Complete Commonality
DCI	Degree of Commonality Index	Collier [14]	The whole family	1	$\beta = \sum_{j=i+1}^{i+d} \Phi_j$
TCCI	Total Constant Commonality Index	Wacker and Trelevan [15]	The whole family	0	1
PCI	Product Line Commonality Index	Kota, Sethuraman, and Miller [16]	The whole family	0	100
%C	Percent Commonality Index	Siddique, Rosen and Wang [17]	Individual products within a family	0	100
CI	Commonality Index	Martin and Ishii [18,19]	The whole family	0	1
$CI^{(C)}$	Component Part Commonality	Jiao and Tseng [20]	The whole family	1	$\alpha = \sum_{j=1}^{d}\sum_{i=1}^{m} \Phi_{ij}$

In this research, the commonality indices are not categorized; rather, a set of guidelines is provided to the designer to help him/her choose the appropriate indices based on the company's strategy. These guidelines are given in Table 2. This paper focuses on one of the index, namely, the Product Line Commonality Index (PCI). Details on its computation are given in Section 3.2. This index is chosen as extensive study is being conducted on this index to minimize its variability during computation [21,22]. However, the methodology can be easily extended to any of the five indices presented in Table 1. Note that not all of the data proposed in Section 2.2 is used; the component costs and the production volume are not necessary for this index.

Table 2. Criteria for the choice of commonality indices

Commonality indices	DCI	TCCI	CI	PCI	$CI^{(C)}$
Focus on the number of common components	X	X	X		
Focus on the non-differentiating components				X	
Focus on the cost of the components					X

2.4. Phase 3. Product Family Design Optimization using Genetic Algorithm

In this paper, a Genetic Algorithm (GA) is used to maximize the PCI. GAs are adaptive stochastic optimization algorithms involving search and optimization. Instead of working with a single solution at each iteration, a GA works with a number of solutions (collectively known as a population). GAs are based on the notion of the "survival of the fittest", and they operate by searching for and choosing optimal solutions in much the same way that natural selection occurs. GAs only use the objective function while searching for optimized result and not the derivatives, therefore it is a direct search method. GAs work with a coding of the parameter set (set of strings/individual chromosomes), and not the parameters themselves and use probabilistic transition rules [23]. GA method of optimizing product family redesign utilizes the stochastic search nature of genetic algorithms to find combinatorial designs within the search space. GAs appear well suited for solving combinatorial problems typical in product family redesign .

Usually there are only two main components of most GAs that are problem-dependent: (1) the problem encoding and (2) the evaluation function. When the GA is implemented, it is usually done in a manner that involves the following cycle:
- Evaluate the fitness of all of the individuals in the population.
- Create a new population by reproduction. The reproduction process for a pair of chromosomes involves duplicating the two individual chromosomes (the "parents") and then choosing a place (site) on the chromosomes to crossover (or switch) information between them. This results in two new "children" chromosomes in the population, which could have higher fitness values than their "parents". Mutation can also occur when decision variable values in a chromosome are randomly changed.
- The old population is then discarded, and a new iteration is started using the new population.

Every iteration of the GA is referred to as a generation. The exchange of information between chromosomes during crossover allows the algorithm to converge to a global, rather than a local, optimum [23]. Even though the operators are simple, GAs are highly nonlinear, massively multifaceted, stochastic, and complex. In this paper, each attribute of a component is encoded as an integer, which is later converted into a binary representation for the GA. The algorithm maximizes the PCI, subject to the following additional constraints to facilitate the selection of components to be redesigned.

Constraint 1: external/ differentiating parts: the parts that are external on a product usually differentiate the product; these parts should not be modified during redesign. For example, the button shown in Figure 2 should not be modified since it differentiates each mouse.

Figure 2. Example of differentiating parts

Constraint 2: the parts that are unique to one product will not be modified. The unique components are defined as either (1) external and/or differentiating components or (2) for a specific function that is present in only one product. These components are used to keep each product different aesthetically and functionally. Hence, it is desired not to modify these unique components.

Constraint 3: if a part is already common throughout the whole family, the optimizer should not modify the part. We are only looking here at the degree of commonality within a product family. Other parameters, such as the performance of each product, are not considered yet. Hence, the parts that are common through the whole are considered 'best' for the commonality and should not be modified, although the individual performances of each product may not be optimized.

Constraint 4: maximum number of components allowable to change: there is a restriction on the number of parameters to change between the original design and the redesigned family. If this constraint is not added, the optimizer will find the "best" commonality when all the parts are common. By adding this constraint, the designer specifies a maximum number of allowable changes.

Based on these four constraints, the design variables are chosen: only the variant parts are considered. Within this set of parts, four attributes are considered: (1) size and geometry, (2) material, (3) manufacturing process, and (4) materials. For a given part, if an attribute is common between all the products using this part, then this attribute is not considered during optimization.

2.5. Phase 4. Data Output and Redesign Recommendations

Once the optimization is complete, the optimizer proposes a redesign sequence that can be compared to the original redesign. *Note that the optimizer does not currently check the feasibility of the solution into account; rather, the optimizer provides the designer with a ranked list of parameters that most influences the degree of commonality in the product family.* This can be viewed as a reduction of the redesign space, where the designer checks the feasibility of the solution a posteriori in the list of proposed recommendations, rather than checking the feasibility of a redesign solution a priori in a much wider space. Two main types of information are given using the GAs: (1) *at the product family level*, if there exists more than one design for a particular family, then the algorithm assesses each design and classifies them; (2) *at the component level*, a list of components to redesign is proposed to achieve the highest commonality with a minimum number of changes.

Recommendations at the product family level: if the designer wishes to assess more than one design for a product family, the algorithm is also run without the fourth constraint proposed in Section 2.4 (i.e, no limitation on the number of changes in the parameters); hence, once the design is optimized, the "ideal" commonality is reached, i.e., all the parts are common in the product family. An offline analysis of the values obtained after optimization enables the assessment of the different design strategies. To do so, a graph similar to the one shown in Figure 3 is plotted for each design. This graph aims at evaluating different design strategies of the concerned product family, based on how the factors that are changed influence the selected commonality index.

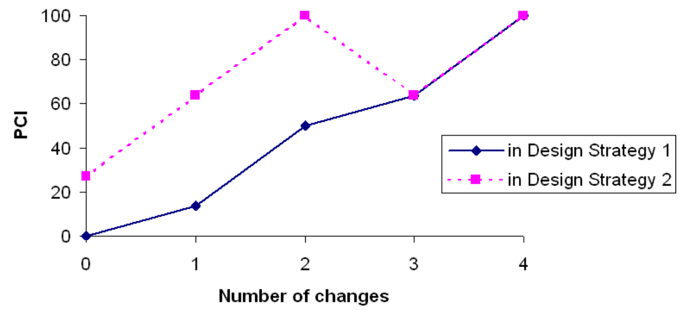

Figure 3. PCI versus number of changes in Design Strategies 1 and 2

The graph is obtained by first categorizing the values obtained for the commonality index based on the number of changes in the parameters. If we consider the example shown in Table 3, we have a product family consisting of three products, each product having two parts. Each part is used in each product. Two different design strategies need to be assessed. In Design Strategy 1 (DS1), the parts are variant in each product (i.e., no commonality). This is represented by attributing three different numbers to each part, one for each product (1, 2 and 3). In Design Strategy 2 (DS2), there are two variants for each part, one variant being used by two products (some level of commonality), represented by having the same number for Part 1 – Product 1 and Part 1 – Product 2, and Part 2 – Product 1 and Part 2 – Product 3. The best design (relative to the concerned commonality indices, in this case the PCI) with the minimum number of changes is achieved through Design Strategy 3 (DS3): the parts are common between all the products in the family (complete commonality; in fact, the three products are identical with regard to these two parts). Depending on the commonality index chosen, this may not always be the best design. For example, if we consider the $CI^{(C)}$, which takes the cost of each component into account, and if component 2 is cheaper to produce than component 1 (provided they both achieve the same function), then the "ideal" design should consist of having only variant 2 of both components. Cost is not captured with the PCI.

Table 3. Three different design strategies for two parts in a product family

		Design Strategy 1	Design Strategy 2	Design Strategy 3
Part 1	in Product 1	1	1	1
	in Product 2	2	1	1
	in Product 3	3	2	1
Part 2	in Product 1	1	1	1
	in Product 2	2	2	1
	in Product 3	3	1	1
Commonality		No commonality → Complete commonality		

By running a GA without constraints on DS1 and DS2, the optimal value of the PCI is the one obtained in Design 3 (complete commonality). This value will be identical for both designs, as shown in Figure 3; however, the minimum number of changes to achieve this complete commonality is different. In DS1, a minimum of 4 changes are necessary to achieve DS3, while only 2 changes are required in DS2, as shown in Figure 3. For any number of changes, the PCI in DS1 is higher or equal to the one in DS2. Hence, we can conclude that DS1 is a "dominated" design relative to the PCI: DS2 achieves higher PCI (hence higher commonality) than DS1, for any given number of changes.

Recommendations at the component level: the algorithm provides a set of possible changes that could be implemented to maximize the commonality of the product family for a given number of changes. The best combination(s) of parts to redesign is proposed; additionally, the algorithm provides a ranked list of possible combinations. For a given number of changes, the designer can then choose the feasible combination of parameters that results in the highest PCI (highest increase in commonality). If we consider the example shown in Table 3 for DS2, with a maximum number of changes set to 2, the algorithm will return the following information (only the first two recommendations are shown):

- First recommendation: {change Part 1 – Product 3 from variant 2 to variant 1, change Part 2 – Product 2 from variant 2 to variant 1}. This results in a PCI of 100.
- Second recommendation: {change Part 1 – Product 1 from variant 1 to variant 2, change Part 1 – Product 2 from variant 1 to variant 2}. This results in a PCI of 63.5.

By giving a list of ranked solutions for possible redesign, the algorithm has reduced the redesign space, which helps focus the designer on the components that influence commonality the most in the family.

3. EXAMPLE REDESIGN: COMPUTER MICE

A complete example of the redesign of a product family is provided, describing the application of each phase step-by-step.

3.1. Phase 1: Data Input

The product family analyzed consists of a set of six computer mice, all from the same manufacturer, as shown in Table 4. The Bills of Materials were not available for these products; hence, a dissection was conducted. More details on the dissection can be found in Ref. [13].

Table 4. The computer mice family

Product Family	Products					
	Optical Mouse	Optical Wheel Mouse	Cordless Optical Mouse	MX300	MX500	MX700
Logitech						
Name	p1	p2	p3	p4	p5	p6

The family is dissected, and the data is stored in an Excel spreadsheet, as shown in Table 5. The first two columns are the name of the parts, and the corresponding product (p1,…,p6), as shown in Table 4. In the next column, Size and Geometry, the designer enters a number indicating if the part is common between different products. For example, for a given part, if two products share the same number, then they share the same component. If the number is different in each column for a given part, then all the products use different variants of the part. If there is no number, then the corresponding product does not contain the corresponding part. In Table 5, the AC adapter is unique, while the Back panel has 5 variants, one being shared between p1 and p2. In the next three columns, the designer enters a number corresponding to the material, the manufacturing process, and the assembly/fastening scheme. These numbers correspond to those found in Appendices A, B and C. Depending on the level of detail desired, the designer can enter either very specific information (e.g., Aluminum 6061) or less detailed information (e.g., non-ferrous alloy). In any case, the designer should be consistent within the different parts, i.e., the level of detail should be the same for all the parts in the family.

Table 5. Example of data entered for the family

Parts		Size and geometry	Material	Manufacturing process	Assembly and fastening scheme
AC Adapter	in p1				
	in p2				
	in p3				
	in p4				
	in p5				
	in p6	1	11	11	13
Back Panel	in p1	1	31	11	12
	in p2	1	31	11	12
	in p3	2	31	11	12
	in p4	3	31	11	12
	in p5	4	31	11	12
	in p6	5	31	11	12

3.2. Phase 2: Commonality Assessment

Using Microsoft Excel, the computation of the PCI is automated; more details on the computation can be found in Refs. [7,12,13]. Contrary to the indices that simply measure the percentage of components that are common across a product family (and hence penalizing families with a broader feature mix), the PCI measures and penalizes the differences that should ideally be common, given the product mix [16]. The PCI is given by:

$$PCI = 100 * \left(\sum_{i=1}^{P} CCI_i - \sum_{i=1}^{P} MinCCI_i\right) / \left(\sum_{i=1}^{P} MaxCCI_i - \sum_{i=1}^{P} MinCCI_i\right) \quad (1)$$

where:

CCI_i = Component Commonality Index for component i.
$= n_i * f_{1i} * f_{2i} * f_{3i}$

$MaxCCI_i$ = Maximum possible Component Commonality Index for component i.
$= N$

$MinCCI_i$ = Minimum possible Component Commonality Index for component i.
$= n_i * 1/n_i * 1/n_i * 1/n_i$
$= 1/n_i^2$

P = Total number of non differentiating components that can potentially be standardized across models.

N = Number of products in the product family.

n_i = Number of products in the product family that have component i.

f_{1i} = Size and shape factor for component i.

= Ratio of the greatest number of models that share component i with identical size and shape to the greatest possible number of models that could have shared component i with identical size and shape (n_i).

f_{2i} = Materials and manufacturing processes factor for component i.

= Ratio of the greatest number of models that share component i with identical materials and manufacturing processes to the greatest possible number of models that could have shared component i with identical materials and manufacturing processes (n_i).

f_{3i} = Assembly and fastening schemes factor for component i.

= Ratio of the greatest number of models that share component i with identical assembly and fastening schemes to the greatest possible number of models that could have shared component i with identical assembly and fastening schemes (n_i).

By substituting the values of CCI_i, $MinCCI_i$, and $MaxCCI_i$, the following formula is obtained for the PCI:

$$PCI = 100 * \left(\sum_{i=1}^{P} n_i * f_{1i} * f_{2i} * f_{3i} - \sum_{i=1}^{P} \frac{1}{n_i^2}\right) / \left(P * N - \sum_{i=1}^{P} \frac{1}{n_i^2}\right) \qquad (2)$$

Equation 3 gives the lower and upper bounds of the PCI.

$$0 \le PCI \le 100 \qquad (3)$$

When PCI = 0, either none of the non-differentiating parts are shared across models, or if they are shared, their size/shapes, materials/manufacturing processes, and assembly processes are all different. When PCI = 100, it indicates that all the non-differentiating parts are shared across models and that they are of identical size and shape, made using the same material and manufacturing process, and the fastening methods are identical.

The PCI value obtained for the family of computer mice is 41.99, on a 0-100 scale. This value provides the baseline for comparison after redesign.

3.3. Phase 3: Product Family Redesign Optimization Using a Genetic Algorithm

Two sets of runs are made, one for the assessment of the product family as a whole (i.e., optimization at the product family-level), and one for the analysis of the effect of the individual components on the commonality of the family (i.e., optimization at the component-level).

Optimization at the product family level: the GA is run to maximize the value of the PCI for the family. Since the parameters (i.e., crossover, mutation, maximum number of generations, population) for the GA are case-dependent, their values that gave the best results are not known a priori. An experimental design is utilized to analyze the effect of the input parameters for the GA on the resulting PCI. Sizing a GA population to ensure maximum computational leverage and accurate sampling has been considered empirically in several studies. Goldberg [23] shows how to set population size in the context of recombinative mixing, disruption, deception, population diversity, and selective pressure to maximize computational leverage. In the current study, we consider a low of 50 and a high of 200 as the population size. Mutation settings obtained from experimental investigation as discussed in the GA literature are shown in Table 6. For our experiment we have considered the lowest (0.001) and highest value (0.01) of the recommended mutation rate, P_m.

Table 6. Commonly used constant settings of the mutation rate P_m in Genetic Algorithms

Pm	Reference
0.001	De Jong [24]
0.01	Grefensette [25]
0.005-0.01	Shaffer et al. [26]

In the GA literature, the crossover probability (Pc) is recommended to start around a value of 0.5. In this paper we have used 0.4 and 0.6 as the low and high value for crossover probability. After choosing the different values for the GA (crossover, mutation, population, maximum number of generations), the implementation is done in Microsoft Excel, using a dedicated plug-in developed by Pi Blue, namely, OptWorks Excel[3]. The crossover method is 2 point: two points are selected on the parent strings. Everything between the two points is swapped between the parents, rendering two children. The problem is formulated choosing the objectives functions and the design variables are defined, as shown in Figure 4 and Figure 5.

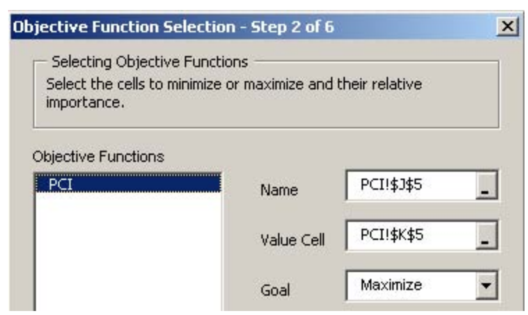

Figure 4. Problem formulation – objective function

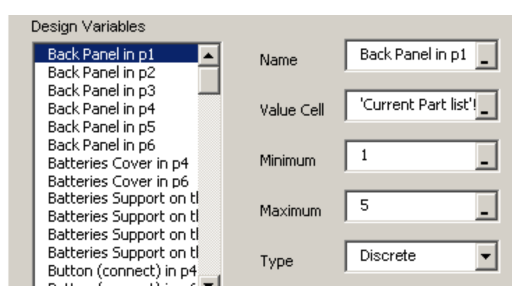

Figure 5. Problem formulation – design variables

[3] http://www.piblue.com/products/optworks_ex.html

For example, if we consider the example shown in Table 7, the part "Back Panel" is present in each product, each having a different variant in size and geometry, but with a common material (plastic), a common manufacturing process (injection molding) and a common assembly and fastening scheme (screwing). Hence, only the "Size and Geometry" factor is considered by the GA and can take any discrete values between 1 and 5 (i.e., there are 5 different variants for "Size and Geometry"), while the other attributes, already common between all the products sharing the part, are not considered. In this paper, this step is done manually by entering the constraints in the software. Future work suggests the automation of this step in order to be applicable to larger-scale problems.

Table 7. Example of part redesign

Parts		Size and geometry	Material	Manufacturing process	Assembly and fastening scheme
Back Panel	in p1	1	31	11	12
	in p2	1	31	11	12
	in p3	2	31	11	12
	in p4	3	31	11	12
	in p5	4	31	11	12
	in p6	5	31	11	12

The objective function is the PCI, and the objective is to maximize it. Note that only the first three constraints proposed in Section 2.4 are taken into account: the results are used to (1) choose the appropriate parameters for the GA and (2) assess the design of the product family. The results for the 16 runs are summarized in Table 8. The best results are obtained in run 14 and run 16, with a PCI value of 63.04, an increase of more than 50% compared to the original value (41.99). This value is far from the "ideal" value of 100, obtained only when all the non-unique parts are used in all the products in the product family, and these parts have the same size and geometry, same material, same manufacturing processes, and same fastening and assembly schemes.

Table 8. Details of experimental runs of GA

Run No.	Cossover Probability Pc	Mutation Probabilty Pm	Population Size Pop	Number of generations Gen	PCI (Best Value)	Improvement
1	0.4	0.001	50	500	**56.551**	34.70%
2	0.6	0.001	50	500	**56.386**	34.30%
3	0.4	0.01	50	500	**54.791**	30.50%
4	0.6	0.01	50	500	**55.066**	31.10%
5	0.4	0.001	200	500	**58.888**	40.20%
6	0.6	0.001	200	500	**63.041**	50.10%
7	0.4	0.01	200	500	**61.006**	45.30%
8	0.6	0.01	200	500	**61.226**	45.80%
9	0.4	0.001	50	5000	**54.543**	29.90%
10	0.6	0.001	50	5000	**57.230**	36.30%
11	0.4	0.01	50	5000	**57.183**	36.20%
12	0.6	0.01	50	5000	**58.036**	38.20%
13	0.4	0.001	200	5000	**62.491**	48.80%
14	0.6	0.001	200	5000	**63.041**	50.10%
15	0.4	0.01	200	5000	**61.556**	46.60%
16	0.6	0.01	200	5000	**63.041**	50.10%

In addition to the PCI value, two other parameters are considered in the comparison of these 16 runs: the number of generations to converge, and the number of function calls. Ideally, we would like to have the highest value for the PCI, while having the number of function calls and the number of generations to converge as low as possible. The comparison of these three parameters (PCI, number of generations to converge and number of function calls) is summarized in Figure 6. The values are standardized between 0 and 100, a higher value indicating a better performance. While run 9 and run 11 converge the fastest, their optimal value is 15% and 11% lower than the one obtained in runs 14 and 16. The most satisfying run is run 14, where the number of function calls and the number of generations to converge is lower than for run 16, although the computational time is higher than in run 9 and 11.

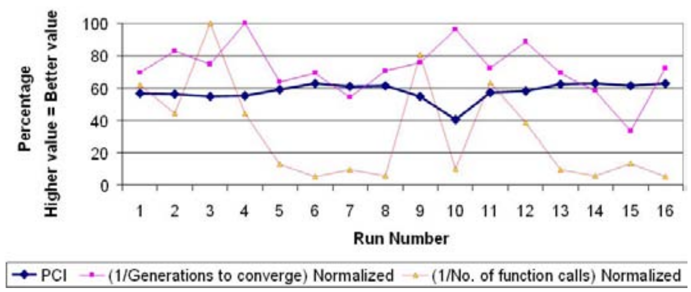

Figure 6. Comparison of the runs

To confirm these results, an analysis of variance is run. The experiment is a 2^4 full-factorial design, and the results are shown in Figure 7. The main factor that influences the PCI is the population size: the larger the population size, the higher the PCI. While the other factors have less influence on the PCI, the main effects plot suggests choosing the crossover value at its high level (0.6), the mutation value at its low level (0.001), and the maximum number of generations set to its high value (5000). This configuration was used in run 14, which confirms that these are the best settings for the GA parameters.

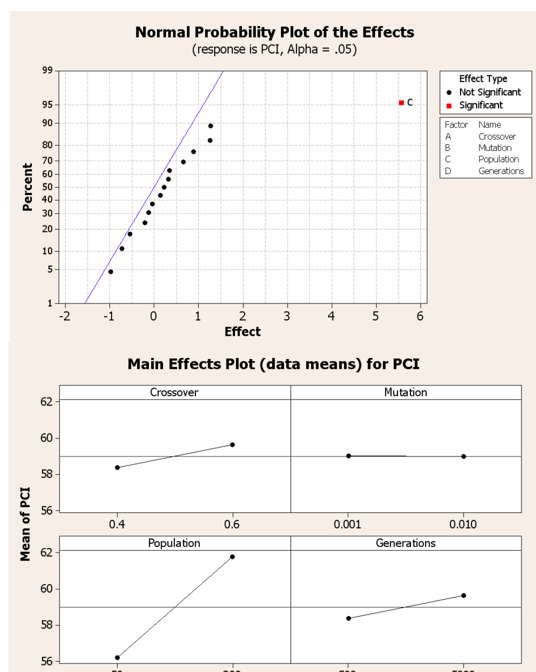

Figure 7. Analysis of variance for the 16 GA runs

We then conducted an offline analysis of the results from run 14. The assessment of the design is done using the graph plotted in Figure 8. On this graph, the minimum number of changes to achieve the best commonality is clearly seen, as well as the rate at which this value is reached. To achieve this ideal commonality, 63 changes are required, which represents 56.8% of the total number of possible parameters (111). Note that the GA only gave PCI values for a high number of changes (50 and over). If the designer is interested in fewer changes, the constraints (on the maximum number of changes) should be added to obtain the specific value. In this research, only one design strategy was considered. Future work suggests the comparison of this current design strategy with the new design of the computer mice, as well as comparison across products families (such as computer mice made by a different manufacturer for example).

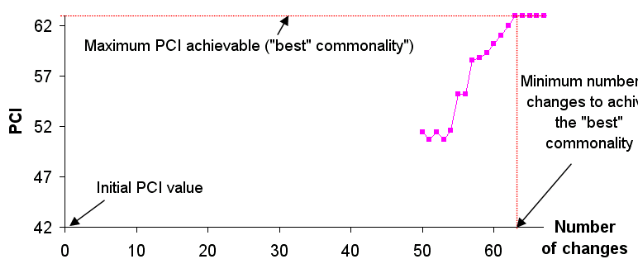

Figure 8. Maximum PCI versus number of changes based on the offline analysis

Optimization at the component-level: the optimization is now run using the four constraints defined in Section 2.4. The values for the genetic algorithm are chosen as the one in run 14. By specifying the maximum number of changes desired, the optimizer gives the best PCI that is achieved with this particular number of changes, as well as the corresponding changes. The feasibility of the proposed solution(s) is not checked as discussed previously, but a ranked list of suggested recommendations is provided, helping designers choose the components that influence commonality the most. The example shown in Table 9 is for a maximum number of changes set to 20. The maximum PCI obtained is 49.57, an increase of 18.05% compared to the original PCI (41.99). For example, it is recommended to make the size and geometry of the back panel in product 1, product 3 and product 6 common, using variant 3 (the original contains three variants for these three products). This results in an increase in PCI of 2.36%. Note that several configurations were obtained for this value of the PCI; only one is represented here as example. While this solution is not necessarily entirely feasible, it helps the designer focus on the components that influence commonality the most.

Table 9. Example of redesign for the product family

Component		Attribute	Original Design	Recommended Design	Delta PCI
Back Panel	in p1	Size and Geometry	1	3	
	in p3	Size and Geometry	2	3	+2.36%
	in p6	Size and Geometry	5	3	
Batteries Support on the PCB	in p4	Size and Geometry	1	2	+1.18%
Button (connect)	in p4	Size and Geometry	1	2	+1.18%
Switch (left-right)	in p2	Size and Geometry	1	3	+1.18%
	in p6	Size and Geometry	4	3	
LED	in p1	Size and Geometry	1	2	+2.36%
	in p4	Size and Geometry	3	2	
Lens	in p1	Size and Geometry	1	2	
	in p2	Size and Geometry	1	2	+2.36%
	in p4	Size and Geometry	3	2	
Mouse Wheel Cover	in p6	Size and Geometry	2	1	+1.18%
Receptor	in p3	Size and Geometry	3	2	
	in p5	Size and Geometry	5	2	+3.54%
	in p6	Size and Geometry	6	2	
Receptor Cover	in p4	Size and Geometry	1	3	+1.18%
	in p6	Size and Geometry	5	3	
Receptor Holder	in p6	Size and Geometry	2	1	+0.35%
RF PCB	in p4	Size and Geometry	1	2	+1.18%

4. CONCLUSIONS

The combined use of genetic algorithms and commonality indices to support product family redesign provides useful information for the redesign of a product family, both at the *product-family level* (assessment of the overall design of a product family) and at the *component-level* (which components to redesign, how to redesign them). The reduction of the redesign space by providing a ranked list of components to modify during product family redesign helps the designer focus on critical components that he/she may not have easily identified without such a systematic approach. This study has limitations however. Future work suggests the use of more detailed commonality indices to assess the commonality within the family. By including other factors such as component costs, manufacturing time, the effect of each component would be even more significant, and can help the designer resolve the balance between too much commonality (lack of individual performance and distinctiveness) and not enough commonality (higher costs). Another research direction is the extension of this research to more product families. Extending the proposed method to more commonality indices and more products will enable the development of a systematic and automatic way of providing recommendations on how to redesign a product family. Finally, the feasibility of the proposed solutions should be considered. Checking the feasibility of the solutions will reduce the redesign space even further by providing the designer with only feasible redesign suggestions.

ACKNOWLEDGMENTS

This work was funded by the National Science Foundation under Grant Nos. IIS-0325402 and DMI-0133923. Any opinions, findings, and conclusions or recommendations presented in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Simpson, T. W., Maier, J. R. A. and Mistree, F., 2001, "Product Platform Design: Method and Application," *Research in Engineering Design,* **13**(1), pp. 2-22.

[2] Sanderson, S. W. and Uzumeri, M., 1997, *Managing Product Families*, Irwin, Chicago, IL.

[3] Wheelwright, S. C. and Clark, K. B., 1995, *Leading Product Development*, Free Press, New York, NY.

[4] Lehnerd, A. P., 1987, "Revitalizing the Manufacture and Design of Mature Global Products," *Technology and Global Industry: Companies and Nations in the World Economy,* B. R. Guile and H. Brooks eds., National Academy Press, Washington, D.C., pp. 49-64.

[5] Pessina, M. W. and Renner, J. R., 1998, "Mass Customization at Lutron Electronics - A Total Company Process," *Agility & Global Competition,* **2**(2), pp. 50-57.

[6] Simpson, T. W., 2003, "Product Platform Design and Optimization: Status and Promise," *ASME Design Engineering Technical Conferences - Design Automation Conference,* Chicago, IL, ASME, Paper No. DETC2003/DAC-48717.

[7] Thevenot, H. J. and Simpson, T. W., 2006, "Commonality Indices for Product Family Design: A Detailed Comparison," *Journal of Engineering Design,* to appear.

[8] Gershenson, J. K., Prasad, G. J. and Zhang, Y., 2003, "Product modularity: Definitions and Benefits," *Journal of Engineering Design,* **14**(3), pp. 295-313.

[9] Gershenson, J. K., Prasad, G. J. and Zhang, Y., 2004, "Product Modularity: Measures and Design Methods," *Journal of Engineering Design,* **15**(1), pp. 33-51.

[10] Swift, K. G. and Booker, J. D., 1997, *Process Selection From Design to Manufacture*, Arnold, London.

[11] Park, J. and Simpson, T. W., 2005, "Development of a Production Cost Estimation Framework to Support Product Family Design," *International Journal of Product Research,* **43**(4), pp. 731-772.

[12] Thevenot, H. J. and Simpson, T. W., 2004, "A Comparison of Commonality Indices for Product family Design," *ASME International Design Engineering Technical Conferences - Design Automation Conference,* Salt Lake City, UT, ASME, Paper No. DETC2004/DAC-57141.

[13] Thevenot, H. J., 2003, "A Comparison of Commonality Indices for Product Family Design," *M.S. Thesis,* Department of Industrial and Manufacturing Engineering, The Pennsylvania State University, University Park, PA.

[14] Collier, D. A., 1981, "The Measurement and Operating Benefits of Component Part Commonality," *Decision Sciences,* **12**(1), pp. 85-96.

[15] Wacker, J. G. and Trelevan, M., 1986, "Component Part Standardization: An Analysis of Commonality Sources and Indices," *Journal of Operations Management,* **6**(2), pp. 219-244.

[16] Kota, S., Sethuraman, K. and Miller, R., 2000, "A Metric for Evaluating Design Commonality in Product Families," *ASME Journal of Mechanical Design,* **122**(4), pp. 403-410.

[17] Siddique, Z., Rosen, D. W. and Wang, N., 1998, "On the Applicability of Product Variety Design Concepts to Automotive Platform Commonality," *ASME Design Engineering Technical Conferences - Design Theory and Methodology,* Atlanta, GA, ASME, Paper No. 1998-DETC/DTM-5661.

[18] Martin, M. and Ishii, K., 1996, "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation," *ASME Design Engineering Technical Conferences - Design Theory and Methodology,* Irvine, CA, ASME, Paper No. 96-DETC/DTM-1610.

[19] Martin, M. V. and Ishii, K., 1997, "Design for Variety: Development of Complexity Indices and Design Charts," *ASME Design Engineering Technical Conferences - Design for Manufacturability,* Sacramento, CA, ASME, Paper No. DETC97/DFM-4359.

[20] Jiao, J. and Tseng, M. M., 2000, "Understanding Product Family for Mass Customization by Developing Commonality Indices," *Journal of Engineering Design,* **11**(3), pp. 225-243.

[21] Simpson, T. W. and Thevenot, H. J., 2005, "Using Product Dissection to Integrate Product Family Design Research into the Classroom and Improve Students' Understanding of Platform Commonality," *International Journal of Engineering Education*, in press.

[22] Simpson, T. W. and Thevenot, H. J., 2005, "Using Product Dissection to Integrate Product Family Design Research into the Classroom and Improve Students' Understanding of Platform Commonality," *2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference,* Long Beach, CA, ASME.

[23] Goldberg, D. E., 1989, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company Inc., Reading, PA.

[24] De Jong, K. A., 1975, "An Analysis of the Behavior of a Class of Genetic Adpative Systems," *Ph.D. Thesis,* Department of Computer and Communication Sciences, University Of Michigan, Ann Arbor, MI.

[25] Grefensette, J. J., 1986, "Optimization of Control Parameters for Genetic Algorithms," *IEEE Transactions on Systems, Man and Cybernetics,* **16**(1), pp. 122-128.

[26] Schaffer, J. D., Caruana, R. A., Eshelman, L. J. and Das, R., 1989, "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization," *3rd International Conference on Genetic Algorithms,* Fairfax, VA, pp. 51-60.

APPENDIX A. LIST OF POSSIBLE MATERIALS

Materials		
1 Ceramics	11 Electronic materials (ferrites/semiconductors...)	
	12 Conctructional ceramics (procelain/stoneware/earthenware)	
	13 Natural ceramics (rocks)	
	14 Glasses (soda/borosilicates/pyroceramics)	
	15 Engineering ceramics (alumina/carbides/nitrides)	
2 Composites (natural/fibre/particulate/dispersion)		
3 Polymers	31 Thermoplastics	311 Partially crystallie (polyamides/acetals/polyathenes)
		312 Amorphous (PVC/polycarbonates/polystrynrens)
	32 Natural polymers (cellulose-based/protein-based)	
	33 Thermosets	331 Rubbers (natural/butyl/silicons/nitrile/styrene)
		332 Epoxies (phenolics/aminos/polyesters/silicones)
4 Metals	41 Ferrous alloys	411 Plain carbon steels (low/medium/high)
		412 Alloy steels (lowalloy/tool/stainless)
		413 Cast irons (grey/white/malleable/nodular)
	42 Non-ferrous alloys	421 Light alloys (zinc, aluminum, magnesieum, titanium)
		422 Heavy alloys (copper/lead/nickel)
		423 Refractory metals (tungsten/tantalum/molybdenum)
		424 Precious metals (gold/silver/plaatinum alloys)

APPENDIX B. LIST OF POSSIBLE MANUFACTURING PROCESSES

Manufacturing processes		
1 Casting	11 Permanent mold	111 Continuous casting
		112 Gravity die casting
		113 Pressure die casting
		114 Squeeze casting
		115 Centrifugal casting
		116 Reaction injection molding
		117 Injection molding
		118 Rotational molding
		119 Compression molding
	12 Permanent pattern	121 Sand casting
		122 Shell molding
	13 Expendable mold and pattern	131 Investment casting
		132 Ceramic/plaster mold casting
		133 Full mold/evaporative pattern
2 Cutting	21 Electromachining	211 Electromechanical machining
		213 Electical discharge machining
	22 Mechanical machining	221 Single point cutting
		222 Multiple point cutting
		223 Grinding/honing/lapping
3 Forming	31 Sheet	311 Sheet metal forming
		312 Vaccum forming
		313 Blow molding
		314 Superlastic forming
	32 Bulk	321 Forging
		322 Rolling
		323 Extrusion
		324 Drawing
	33 Powder processing	331 Slip casting
		332 Pressing and sintering
		333 Isostatic pressing

APPENDIX C. LIST OF POSSIBLE ASSEMBLY AND FASTENING SCHEMES

Assembly schemes			
1 Non-permanent	11 Snapping		
	12 Screwing		
	13 Fit		
2 Permanent	21 Gluing	211 Adhesive bonding	
		212 Brazing	
		213 Soldering	
	22 Riveting		
	23 Welding	231 Fusion welding	2311 Electric arc welding
			2312 Gas welding
			2313 Laser welding
			2314 Electron beam welding
		232 Solid state welding	2321 Forge welding
			2322 Friction welding
			2323 Diffusion bonding