Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2007
September 4-7, 2007, Las Vegas, Nevada, USA

DETC2007-35605

AN EXTENSION OF THE COMMONALITY INDEX FOR PRODUCT FAMILY OPTIMIZATION

Aida Khajavirad
Graduate Student

Jeremy J. Michalek
Assistant Professor

Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT
One critical aim of product family design is to offer distinct variants that attract a variety of market segments while maximizing the number of common parts to reduce manufacturing cost. Several indices have been developed for measuring the degree of commonality in existing product lines to compare product families or assess improvement of a redesign. In the product family optimization literature, commonality metrics are used to define the multi-objective tradeoff between commonality and individual variant performance. These *metrics* for optimization differ from *indices* in the first group: While the optimization metrics provide desirable computational properties, they generally lack the desirable properties of indices intended to act as appropriate proxies for the benefits of commonality, such as reduced tooling and supply chain costs. In this paper, we propose a method for computing the commonality index introduced by Martin and Ishii using the available input data for any product family without predefined configuration. The proposed method for computing the commonality index, which was originally defined for binary formulations (common / not common), is relaxed to the continuous space in order to solve the discrete problem with a series of continuous relaxations, and the effect of relaxation on the metric behavior is investigated. Several relaxation formulations are examined, and a new function with desirable properties is introduced and compared with prior formulations. The new properties of the proposed metric enable development of an efficient and robust single-stage gradient-based optimization of the joint product family platform selection and design problem, which is examined in a companion paper.

KEYWORDS: Product Family, Platform Configuration, Commonality metric, Gradient-based Optimization, relaxation.

NOMENCLATURE
CI: Commonality Index
$\mathbf{f}^i$: Objective function vector for the i^{th} product
$\mathbf{g}^i$: Vector of inequality constraints for the i^{th} product
$\mathbf{h}^i$: Vector of equality constraints for the i^{th} product
m_i : Number of components in the i^{th} product
n_{kr}: Number of k^{th} component in the r^{th} platform
S_{ij}: Platform configuration index set
s_k,: Number of distinct platforms for producing the k^{th} component
u: Total number of distinct components in the product family
$\mathbf{x}^i$: Design variable vector for the i^{th} product
α: Relaxation factor
$\tilde{\lambda}_r^k$: r^{th} eigenvalue of the k^{th} commonality matrix (including zero and nonzero terms)
λ_r^k : r^{th} nonzero eigenvalue of $\mathbf{\Gamma}_k$
η_{pq}^{ij} : Binary commonality decision variables
$\eta_k'^{ij}$: Continuous commonality decision variable
Γ: Commonality objective function for the entire family
$\mathbf{\Gamma}_k$: Commonality matrix for the k^{th} component
$\mathbf{\Gamma}_{kr}$: r^{th} sub-block of $\mathbf{\Gamma}_k$ (discrete form)
$\mathbf{\Gamma}'_{kr}$: r^{th} sub-block of $\mathbf{\Gamma}_k$ (relaxed form)

1. INTRODUCTION
Designing a product family that offers a broad set of variants targeting different market segments and exploits

product platforms to reduce the manufacturing cost is a critical task for many companies. Increasing the number of common components or modules typically has various benefits such as reducing development time, system complexity and manufacturing cost as well as improving the ability to upgrade products. However, increasing the degree of commonality within the family can also reduce the ability of individual product variants to achieve distinct performance targets. Hence, a successful product family should be able to resolve the trade off between commonality and differentiation.

Several indices and metrics have been developed to measure the degree of commonality in a product family. These studies can be classified into two main categories according to their development purpose and application: 1) commonality *indices* for evaluation of existing product lines, and 2) commonality *metrics* for use in product family optimization.

The first group proposes commonality indices for measuring commonality in existing product lines to compare product families or assess improvement of a redesign. Based on the company's focus and perspective, various indices have been developed, each representing a proper measurement with respect to the related criteria[1]. Thevenot and Simpson [12], compare six of these commonality indices from the literature, based on their ease of data collection, repeatability and consistency. According to their study, three of these indices are concerned only with the number of common parts within the family: 1) the Degree of Commonality Index (DCI), introduced by Collier [1], defines commonality as the average number of common parent items per average distinct component part. While this index is easy to compute, its main limitation is the lack of fixed boundaries, making comparison difficult; 2) the Total Constant Commonality Index (TCCI), proposed by Wacker [13], is a modified version of the DCI, in that it is a relative index, and as a result, it has fixed boundaries ranging from 0 to 1. A TCCI value of zero indicates no shared item among the entire family, while a TCCI value of one indicates complete commonality. 3) The Commonality Index (CI), suggested by Martin and Ishii [6, 7], is a measure of the number of unique parts needed to make different varieties. CI ranges from zero to one, and a higher value indicates the entire family was made using fewer unique parts. Other indices consider more specific information about the product line. For example, the Product Line Commonality Index (PCI), introduced by Kota *et al*. [5], measures the difference between the number of parts in a family that are actually common and the number that should be "ideally" common. PCI ranges from 0 to 100, in which a zero value indicates all non-differentiating parts are either not shared or have different size, material, manufacturing and assembly processes. The Percent Commonality Index (%C), (Siddique *et al,* [10],), computes total commonality as weighted-sum of four terms: 1) The percentage of components that are shared among different variants in a platform; 2) the percentage of common connections among the components; 3) the percentage of common assembly sequences; and 4) the percentage of common assembly workstations. This metric range is from 0 to 100, indicating no commonality and complete commonality respectively, and measuring commonality within each platform, rather than across the entire family. Finally, the Component Part Commonality Index (Jiao and Teseng, [3]) considers other factors such as production volume, quantity per operation, and the cost of component parts. It is an extended version of DCI and does not have fixed boundaries.

In order to compare the aforementioned indices, Thevenot and Simpson [12] measured the commonality of eight different product families with each index. They concluded DCI, TCCI and CI are easiest to compute and are repeatable indices (i.e. each has a unique value that can be consistently computed by different people). Moreover, all indices showed a consistent behavior, meaning that all follow the same trend across different product families. Recently, Thevenot and Simpson [18] introduced a comprehensive metric for commonality (CMC), which can be considered as an extension of PCI [5] in that it takes into account other factors such as production volume and component costs. This metric captures more information about each component to assess the impact of each component on the overall level of commonality and diversity in the product family.

In brief, there is no single "true" definition for commonality index, and all aforementioned indices measure the commonality from different viewpoints. Therefore, selection of an index involves consideration of the company focus and viewpoint when designing the product family.

In the second category, commonality metrics are defined within the context of optimization. This group proposed commonality metrics for defining the tradeoff between commonality and the ability to achieve distinct performance targets. Nayak *et al*. [9] proposed a two-stage Variation-Based Platform Design Methodology (VBPDM) for optimizing a family of ten universal electric motors: In the first stage, a compromise decision support problem (DSP) is formulated to maximize commonality by minimizing the normalized standard deviation of the input design variables while satisfying performance constraints. After solving the DSP, if the standard deviation of a design variable is small enough relative to its mean value, it is selected as a platform variable, while others with significant variations are treated as unique variables. After identifying the platform variables and their values in the first stage, the individual products are optimized in the second stage with respect to the non-platform variables satisfying performance objectives and constraints. Messac *et al*. [8] introduced the product family penalty function (PFPF) to find the optimum set of common and scaling parameters for scaled-based product families using physical programming techniques. According to this approach, commonality is maximized by minimizing the percent variation of design variables (i.e.

[1] There are a broad range of indices in the first group. We focus on those that address commonality due to component sharing.

minimizing PFPF) within the product family; therefore, design variables with the highest variations are selected as scaling variables. They use the family of electric motors for demonstrating the proposed approach; however, as mentioned in the paper, increasing the number of products would make the problem intractable and generate many local optima. Simpson and D'Souza [11] proposed a single-stage Genetic Algorithm (GA) based approach for optimizing the product platform and associated product family simultaneously. They used PFPF introduced by Messac *et al*. for measuring the commonality in the family and demonstrated the approach by optimizing a family of three general aviation aircraft.

All aforementioned studies mention that the variation-based metric is a good representation for measuring commonality in that it considers not only the number of common variables but also the degree of similarity among the values of the unique variables relative to one another. However, we argue that this approach does not capture the qualities that companies are looking for when the commonality metric is intended as a proxy for estimating cost reduction due to commonality. That is, considering commonality cost saving due to reduced tooling and supplied chain, manufacturers want to know whether they can use the same component design in multiple variants, or whether a separate component will need to be designed and manufactured for each variant. So, as soon as two components differ sufficiently so that the manufacturer has to provide a distinct set of tooling for each component, the benefit of commonality reduces to zero with respect to cost savings, regardless of how similar the dimensions of unique parts are[2].

Khire and Messac [4] defined a variation penalty function as the sum of the mapped values of the maximum variation of each design variable among all products in the family. They applied mapping by introducing a variable-segregating mapping function (VSMF) for segregating platform variables from scaled variables, in which VSMF is a family of continuous functions that progressively approximate the discontinuous mapping applied in scaled based product families (i.e. design variables are defined as platform variables if their difference falls below a threshold value and as scaled variables otherwise) by using the concept of moving segregation point. Although, this new approach enables the optimization algorithm to define the platform and non-platform variables efficiently within a single stage approach and addresses the unrealistic nature of the PFPF, it considers only *all-or-none* component sharing. That is, a component can either be shared within the entire family or be distinct among all products but cannot be shared among only a subset of the variants. Khajavirad *et al*. [17] applied Martin and Ishii's Commonality Index and proposed a decomposed GA for product family optimization using generalized commonality that allows for multiple "sub-platforms"; however, the approach is discrete in nature and cannot take advantage of efficient gradient-based algorithms. Fellini *et al*. [2] proposed a two stage method that relaxes the commonality metric for use with gradient algorithms. They defined the commonality metric as the summation of all possible pair-wise comparisons within the product family, assigning a binary variable to each pair that is equal to one if two components are the same and zero otherwise. Consequently, the commonality level can be found by summation of all commonality variables within the family. However, while defining the commonality metric as the summation of all possible pairwise comparisons is a better approximation of the indices introduced by the first group, we argue that it still does not provide a practical measurement reflecting the degree of commonality in the family due to its "*double-counting*" property. This shortcoming can be best illustrated with an example: Consider the two alternative commonality configurations for a single component (module) within a family of four variants in Figure 1 (other components are not shown). In the first case, there are two distinct designs for the component of interest: The first is used in both variants 1 and 2, and the second is used in both variants 3 and 4. In the second case, there are also two distinct designs for the component of interest: The first is used in variants 1-3, and the second is used in variant 4. In both cases it is necessary to purchase two sets of tooling, so the two alternatives may be considered equivalent with respect to tooling cost benefits[3]. However, Fellini's commonality metric gives preference to the second case, since there are three common pairs in this set, while there are only two pairs in the first set[4]. Hence, this pairwise comparison-based metric prefers the configurations grouping more components in the same platform and as a result has convergence bias toward product family architectures with all-or-none component sharing.

In brief, while the commonality indices introduced in the first group appear more realistic in measuring the degree of commonality within an existing family according to various perspectives, they have not generally been applied in the optimization context because they require detailed product family structure and platform configuration descriptions prior to computing their value, which is usually not available during the optimization process.

[2] In general, use of flexible manufacturing systems may increase the range of component variation that can be created with the same tooling; however, these systems also typically involve increased capital costs, and we do not consider these cases here.

[3] If more information is known about the production volume of each variant and the life of the tooling, a more accurate prediction can be made; however, commonality metrics are generally applied at a higher level of abstraction so that they do not require excessive data to compute.

[4] If anything, the first alternative would probably be preferred over the second because the sharing appears to be more balanced (again, this depends on production volume).

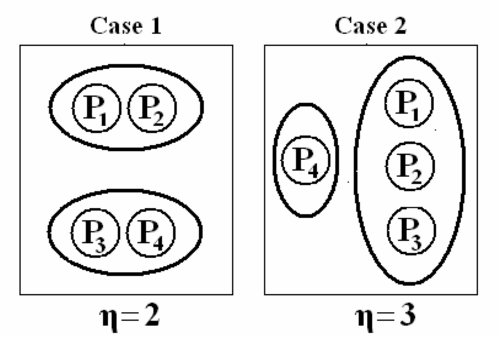

Figure 1. The Double Counting Defect (p_i: i^{th} product, η: Fellini's commonality metric)

In this study, we propose a method for computing the commonality index (CI) introduced by Martin and Ishii [6, 7] using the input data available for any product family without a predefined configuration. Next, the proposed method for computing CI is relaxed to the continuous space to expand suitability for use in gradient-based approaches, and the effect of relaxation on the metric behavior is investigated. As will be shown in the following sections, CI is a function of component pairwise comparisons, which are treated as binary variables (common / not common) in the discrete formulation. When working in a continuous space, these variables should be approximated by a continuous and differentiable function with desirable characteristics. We explore existing approaches, introduce a new function having the desired properties, and compare the new function with prior approaches. The new properties of the proposed metric enable development of an efficient and robust single-stage gradient-based optimization of the joint product family platform selection and design problem, which is examined in a companion paper [14].

2. PROPOSED METHODOLOGY

A product family is defined as a group of related products derived from a number of shared components produced in the same platform. Hence, the basic formulation for optimizing a single product can be extended for optimizing a family of products by considering shared components as equality constraints. Nelson *et al.* [16] proposed the following formulation for optimizing a family of n products with a predefined (*a priori*) platform configuration:

$$
\begin{aligned}
&\textit{Maximize} && \mathbf{f}^i(\mathbf{x}^i) \qquad i = 1, 2, \ldots, n \\
&\textit{with respect to} && \mathbf{x} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n] \\
&\textit{subject to} && \mathbf{g}^i(\mathbf{x}^i) \le \mathbf{0} \\
& && \mathbf{h}^i(\mathbf{x}^i) = \mathbf{0} \\
& && x_p^i = x_q^j \qquad i, j = 1, 2, \ldots, n \;\; , \;\; i < j \\
& && \qquad\qquad (p, q) \in S_{ij}
\end{aligned}
\qquad (1)
$$

In which S_{ij} is a set of index pairs indicating shared component between products i and j. Hence, the platform configuration is defined by a distinct set of individual pairs and imposed by the last equality constraint in Eq.(1). However, in order to find the optimal platform configuration and corresponding set of products simultaneously, the commonality metric and decision variables should be added to the original formulation to resolve the trade off between the commonality and performance objectives. Fellini *et al.*[2] modified Eq.(1) to optimize the joint problem by including the commonality metric in the objective and introducing binary decision variables to find the optimal platform configuration:

$$
\begin{aligned}
&\textit{Maximize} && \left\{\mathbf{f}^i(\mathbf{x}^i), \Gamma(\eta_{pq}^{ij})\right\} \qquad i, j = 1, 2, \ldots, n \;\; , \;\; i < j \\
&\textit{with respect to} && \boldsymbol{\eta}, \mathbf{x} \qquad\qquad\qquad (p, q) \in S_{ij} \\
&\textit{subject to} && \mathbf{g}^i(\mathbf{x}^i) \le \mathbf{0} \\
& && \mathbf{h}^i(\mathbf{x}^i) = \mathbf{0} \\
& && \eta_{pq}^{ij}(x_p^i - x_q^j) = 0 \\
& && \eta_{pq}^{ij} \in \{0, 1\}
\end{aligned}
\qquad (2)
$$

The set S_{ij} in Eq.(2) contains index pairs of components in products i and j that are *candidates* of sharing. Binary commonality decision variables η_{pq}^{ij} are set to one if x^i_p and x^i_q are shared and zero otherwise. As Fellini mentioned, the last equality constraint in Eq.(2) ensures that the commonality variables are consistent with the values of the design variables. However, the formulation permits cases where two design variables are equal and the commonality variable is zero. While the formulation will be consistent at optimal solutions, it is not consistent for all feasible solutions. Hence, in order to ensure consistency for all feasible points, the following constraint[5] can be added to Eq.(2):

$$\eta_{pq}^{ij} + (x_p^i - x_q^j)^2 > 0 \qquad (3)$$

2.1 THE COMMONALITY INDEX

In Eq.(2), $\Gamma(\eta_{pq}^{ij})$ measures commonality within the entire family and is a function of the binary decision variables η_{pq}^{ij}. Defining the proper form for Γ depends on the company's perspective when designing a product family. In this study, we consider the commonality benefit due to tooling cost savings and adopt the commonality index (CI) introduced by Martin and Ishii [6, 7], which is a measure of unique parts: For a product family with a given platform configuration, the commonality level can be calculated as:

[5] This constraint can be relaxed to avoid the strict inequality by introducing a tolerance for commonality deviation, but we need not pursue such a formulation here.

$$\mathrm{CI} = 1 - \frac{u - \max m_i}{\sum_{i=1}^{n} m_i - \max m_i} \quad (4)$$

where u is the total number of distinct components, m_i represents the number of components used in variant i, and n shows the number of variants in the family. CI ranges from 0 to 1, and a higher value indicates fewer unique parts[6]. It should be noted that in computing the CI value from Eq.(4) it is assumed that the product family structure is given. Hence, in order to apply it within an optimization context (i.e. Eq.(2)), it should be reformulated as a function of the binary decision variables. Let us reconsider the commonality representation in Eq.(2). For the k^{th} component in the product family, the commonality matrix $\mathbf{\Gamma}_k$, is defined as follow:

$$\mathbf{\Gamma}_k = \begin{bmatrix} 1 & \eta_k^{12} & . & . & . & \eta_k^{1n} \\ \eta_k^{21} & 1 & & & & \eta_k^{2n} \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & . \\ \eta_k^{n1} & \eta_k^{n2} & . & . & . & 1 \end{bmatrix} \quad (5)$$

As can be found from Eq.(5), $\mathbf{\Gamma}_k$ is always a symmetric matrix with all diagonal elements equal to 1. In Eq.(5), it is assumed that all variants include the k^{th} component, which may or may not be shared with other variants in the family. However, for the general case some variants may not include all components. In this case, the corresponding commonality binary variable in Eq.(5) is set to zero. Imposing the transitivity constraints on commonality decision variables to ensure a consistent matrix (e.g. if $\eta_{12} = 1,\ \eta_{23} = 1 \rightarrow \eta_{13} = 1$), the set of platforms for component i is well-defined, and Eq.(5) can be rearranged to the following block diagonal format:

$$\mathbf{\Gamma}_k = \begin{bmatrix} [1]_{n_{k1}\times n_{k1}} & 0 & \dots & 0 & 0 \\ 0 & . & \dots & 0 & 0 \\ . & & [1]_{n_{kr}\times n_{kr}} & & . \\ & & & . & 0 \\ 0 & 0 & \dots & 0 & [1]_{n_{ks_k}\times n_{ks_k}} \end{bmatrix} \quad (6)$$

where is n_{kr} the number of k^{th} components produced in the r^{th} platform, and $[1]_{n_{kr}\times n_{kr}}$ represents a $n_{kr} \times n_{kr}$ matrix with all elements equal to 1. Furthermore, number of these sub-matrices s_k, also called "blocks", is equal to the number of distinct platforms for producing component k:

$$\sum_{k=1}^{m} \sum_{r=1}^{s_k} n_{kr} = n \times m \quad (7)$$

As we know from linear algebra [15], eigenvalues of a block diagonal matrix are simply those of its blocks. In the case of Eq.(6), the r^{th} block is:

$$\mathbf{\Gamma}_{kr} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & & & 1 \\ . & & & . \\ 1 & 1 & \dots & 1 \end{bmatrix}_{n_{kr}\times n_{kr}} \quad (8)$$

In order to find the eigenvalues of Eq.(8), we should consider two basics from linear algebra: For any given matrix, the number of non-zero eigenvalues is equal to the number of linearly-independent columns or rows, and their summation is equal to the matrix trace. As a result, Eq.(8) has one non-zero eigenvalue equal to n_{kr}. Hence, Eq.(6) has s_k non-zero eigenvalues $\boldsymbol{\lambda}^k$, each equal to the number of components present in the corresponding platform:

$$\lambda_r^k = n_{kr}, \quad r = 1,\dots,s_k \quad (9)$$

For a family consisting of n products each with m components and no commonality, we have $n \times m$ distinct components; however, by using platforms and sharing components among different products, the total number of distinct components can be written as:

$$u = n \times m - \sum_{k=1}^{m} \sum_{r=1}^{s_k} (n_{kr} - 1) \quad (10)$$

Therefore, substituting Eq.(7), Eq.(9) and Eq.(10) into Eq.(4), CI can be reformulated as follows:

$$CI = \frac{\sum_{k=1}^{m} \sum_{r=1}^{s_k} (\lambda_r^k - 1)}{m \times (n-1)} \quad (11)$$

Therefore, using the above formula, CI can be defined as a function of the commonality decision variables n_{kr}, i.e. the form that can be computed in the optimization procedure.

The derived formula for CI finds the total commonality level as a function of binary commonality variables. However, in practice, it is not convenient to solve Eq.(2) in the mixed-integer format. Hence, one alternative way is to omit the independent commonality decision variables and define them according to the individual product design variables as following:

[6] In order to estimate the tooling cost savings more precisely, CI should be reformulated to include cost coefficients representing the cost saving amount due to sharing the corresponding components. However, since this extension has no effect on deriving the appropriate formula for CI, which is the focus of this paper, all cost coefficients are assumed to be equal for simplicity.

$$\eta_k^{ij} = \begin{cases} 1 & \text{If } x_k^i = x_k^j \\ 0 & \text{Otherwise} \end{cases} \tag{12}$$

By approximating Eq.(12) with a continuous function, the total commonality level will be relaxed to the continuous space, and gradient-based methods can be applied for finding the optimal product family. Hence, in the following sections, Eq.(11) will be revised and extended to the continuous space. Next, the proper functional form for approximating Eq.(12) will be investigated.

2.2 RELAXATION OF THE COMMONALITY INDEX

It should be noted that in deriving Eq.(11) using the discrete format, λ_r^k are integer values as well (since they show the number of components in each platform). Hence, by restricting λ_r^k to be the non-zero eigenvalues of the component commonality matrices, the negative terms are omitted from Eq.(11). However, by relaxing CI to the continuous space, this assumption is no longer valid and Eq.(11) should be modified as follows:

$$\mathrm{CI} = \frac{\sum_{k=1}^{m}\sum_{r=1}^{s_k}(\max(\tilde{\lambda}_r^k, 1) - 1)}{m \times (n-1)} \tag{13}$$

In which $\tilde{\lambda}_r^k$ represents the r^{th} eigenvalue of the k^{th} commonality matrix (including zero and nonzero terms). This generalization reduces to Eq.(11) for the discrete case. Eq.(13) introduces a discontinuity in the derivative of CI, however it is possible to eliminate the discontinuity through introduction of a slack variable, and our empirical examples suggest that such reformulation is unnecessary because gradient-based algorithms perform well with the form of Eq.(13).

In order to investigate the effect of relaxation on the total commonality value given by Eq.(13), three basic cases will be considered:

1. Consider an arbitrary platform within the product family with n' components. Using the discrete definition for the commonality metric, the block matrix associated with this platform is as follow:

$$\mathbf{\Gamma}_{kr} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & & & 1 \\ . & & & . \\ 1 & 1 & \dots & 1 \end{bmatrix}_{n_{kr}\times n_{kr}} \tag{14}$$

As we mentioned before, Eq.(14) has one nonzero eigenvalue: $\lambda_r^k = n_{kr}$, showing the number of shared components in that platform and as a result its commonality level. Now, without loss of generality, we perturb the n_{kr}^{th} component, assuming it differs from other components by δ ($0 \le \delta \le 1$). Therefore, the new sub-matrix will become:

$$\mathbf{\Gamma}'_{kr} = \begin{bmatrix} 1 & \dots & 1 & 1-\delta \\ . & & . & . \\ 1 & \dots & 1 & 1-\delta \\ 1-\delta & \dots & 1-\delta & 1 \end{bmatrix}_{n_{kr}\times n_{kr}} \tag{15}$$

As can be seen from Eq.(15), there are only two linearly independent rows (or columns) in the matrix, which is equal to the number of nonzero eigenvalues and their sum is equal to the matrix trace. Furthermore, the summation of all possible a set of eigenvalues multiplication is equal to the i^{th} sum of the a-rowed diagonal minors of the matrix (Jacobson 1974, p. 109). Hence:

$$\begin{aligned} &\lambda_1' + \lambda_2' = n_{kr} \\ &\lambda_1'\lambda_2' = (n_{kr}-1)\,\mathbf{det}\begin{bmatrix} 1 & 1-\delta \\ 1-\delta & 1 \end{bmatrix} \\ &\rightarrow \lambda'^2 - n_i\lambda' + \delta(2-\delta)(n_{kr}-1) = 0 \end{aligned} \tag{16}$$

$$\rightarrow \begin{cases} \lambda_1' = \left(n_{kr} + \sqrt{n_{kr}^2 - 4\delta(2-\delta)(n_{kr}-1)}\right)/2 \\ \lambda_2' = \left(n_{kr} - \sqrt{n_{kr}^2 - 4\delta(2-\delta)(n_{kr}-1)}\right)/2 \end{cases} \tag{17}$$

As can be found from Eq.(17), when δ=0, we have: $\lambda_1' = n_{kr}$ and $\lambda_2' = 0$ (the original case); while for δ=1, we have $\lambda_1' = n_{kr} - 1$ and $\lambda_2' = 1$ (which shows n_{kr}-1 shared components and 1 distinct component). For 0<δ<1, the commonality level for the corresponding platform (i.e. $\lambda_1' - 1$) is sketched in Figure 2 (since λ_2' is always less than 1 for 0<δ<1, it has no effect on the platform commonality level, Eq.(13)).

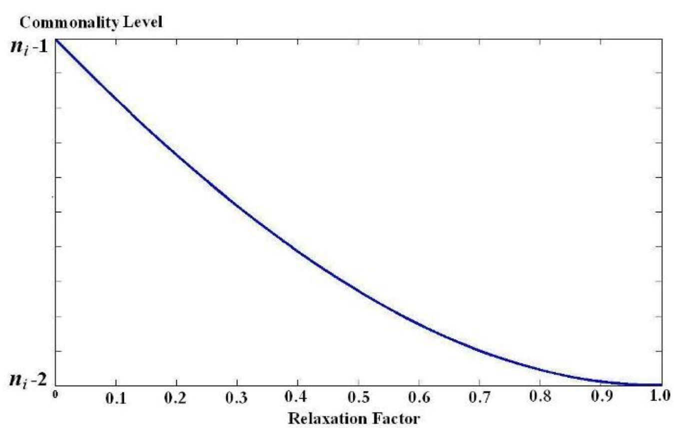

Figure 2. Commonality level change for the case 1

2. Now consider the same platform with n_{kr} shared components, and add an arbitrary component which is initially distinct from this platform. The sub-matrix associated with this augmented platform in discrete format is as follow:

$$\kappa_{kr} = \begin{bmatrix} 1 & ... & 1 & 0 \\ . & & & 0 \\ 1 & ... & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{(n_{kr}+1)\times(n_{kr}+1)} \tag{18}$$

Eq.(18) has two non-zero eigenvalues $\lambda_1 = n_i$ and $\lambda_2 = 1$. Now relaxing the distinct variable to the continuous space, it becomes common to the platform components by the value of γ ($0 \le \gamma \le 1$). Therefore, the new sub-matrix will become:

$$\kappa'_{kr} = \begin{bmatrix} 1 & ... & 1 & \gamma \\ . & & & \gamma \\ 1 & ... & 1 & \gamma \\ \gamma & \gamma & \gamma & 1 \end{bmatrix}_{(n_{kr}+1)\times(n_{kr}+1)} \tag{19}$$

Using the same concepts as the first case, we have the following relations for the eigenvalues of Eq.(19):

$$\left.\begin{aligned} &\lambda'_1 + \lambda'_2 = n_{kr} + 1 \\ &\lambda'_1\lambda'_2 = n_{kr} \,\mathbf{det}\begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix} \end{aligned}\right. \rightarrow \lambda'^2 - (n_{kr}+1)\lambda' + n_{kr}(1-\gamma^2) = 0 \tag{20}$$

$$\rightarrow \begin{cases} \lambda_1' = \left((n_{kr}+1) + \sqrt{(n_{kr}+1)^2 - 4n_{kr}(1-\gamma^2)}\right)/2 \\ \lambda_2' = \left((n_{kr}+1) - \sqrt{(n_{kr}+1)^2 - 4n_{kr}(1-\gamma^2)}\right)/2 \end{cases} \tag{21}$$

As can be found from Eq.(21), when γ=0, we have: $\lambda'_1 = n_{kr}$ and $\lambda'_2 = 1$ (the original case); while for γ=1, we have $\lambda'_1 = n_{kr} + 1$ and $\lambda'_2 = 0$ (which shows n_{kr}+1 shared components and no distinct component). For 0<γ<1, i.e. the continuous case the commonality level for the corresponding platform (i.e. $\lambda'_1 - 1$) is sketched in Figure 3.

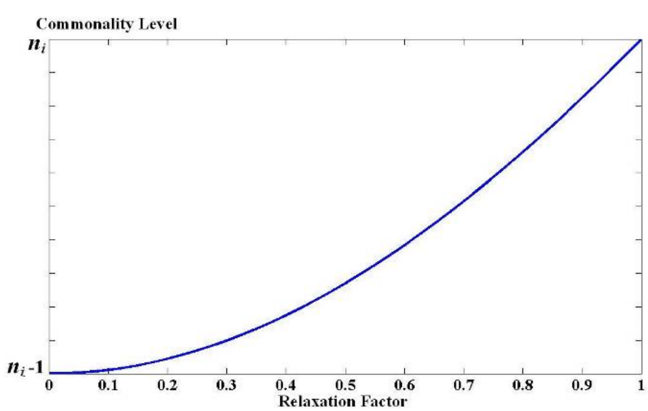

Figure 3. Commonality level change for case 2

3. As a final case, consider two platforms with $n'+1$ and n' components respectively. In this configuration, we are interested to observe how the commonality level changes as the k^{th} component in the first platform differs from the other members and become common with the second platform components simultaneously. The sub-matrix representing the aforementioned platforms has the following form for the discrete representation:

$$\kappa' = \begin{bmatrix} 1 & ... & 1 & 0 & ... & 0 \\ . & & . & . & & . \\ 1 & ... & 1 & 0 & ... & 0 \\ 0 & ... & 0 & 1 & ... & 1 \\ . & & . & . & & . \\ 0 & ... & 0 & 1 & ... & 1 \end{bmatrix}_{(2n'+1)\times(2n'+1)} \tag{22}$$

The above matrix has two non-zero eigenvalues equal to $n'+1$ and n', indicating the number of components in the first and second platforms respectively; hence the total commonality level is equal to $2n'-1$. Without loss of generality, by relaxing the $(n'+1)^{th}$ component in the first platform, the commonality matrix will have the following form:

$$\kappa'' = \begin{bmatrix} 1 & ... & 1 & 1-\alpha & 0 & ... & 0 \\ . & & . & 1-\alpha & . & & . \\ 1 & ... & 1 & 1-\alpha & 0 & ... & 0 \\ 1-\alpha & 1-\alpha & 1-\alpha & 1 & \alpha & \alpha & \alpha \\ 0 & ... & 0 & \alpha & 1 & ... & 1 \\ . & & . & \alpha & . & & . \\ 0 & ... & 0 & \alpha & 1 & ... & 1 \end{bmatrix} \tag{23}$$

in which α ($0 \le \alpha \le 1$) is the relaxation factor. It should be noted that in general, families with multiple platforms, have different factors for each individual platform. In this study, we used the same factor for both platforms for simplifying the analytical equations. However, the conclusions can be extended to the general form. Using the same theories from linear algebra as the previous cases, the following relations for the eigenvalues will be obtained:

$$\begin{aligned} &\lambda'_1 + \lambda'_2 + \lambda'_3 = 2n' + 1 \\ &\lambda'_1\lambda'_2 + \lambda'_2\lambda'_3 + \lambda'_1\lambda'_3 = n'\left(n' - (2\alpha^2 - 2\alpha - 1)\right) \\ &\lambda'_1\lambda'_2\lambda'_3 = 2n'^2\alpha(1-\alpha) \end{aligned} \tag{24}$$

$$\rightarrow \lambda'^3 - (2n'+1)\lambda'^2 + n'\left(n' - (2\alpha^2 - 2\alpha - 1)\right)\lambda' - 2n'^2\alpha(1-\alpha) = 0 \tag{25}$$

Therefore, the total commonality level can be found from the following relation:

$$\eta'' = \left(\frac{3(n'-1) + \sqrt{(n'+1)^2 - 8n'\alpha(1-\alpha)}}{2}\right) \tag{26}$$

As can be found from Eq.(26) for both α=0 and α=1 the commonality level is $2n'-1$ (i.e. the cases in which the component belongs to either first platform or the second one). The metric behavior in the continuous space is sketched in Figure 4. Note that because commonality is being maximized, the non-concavity of this function suggests local maxima.

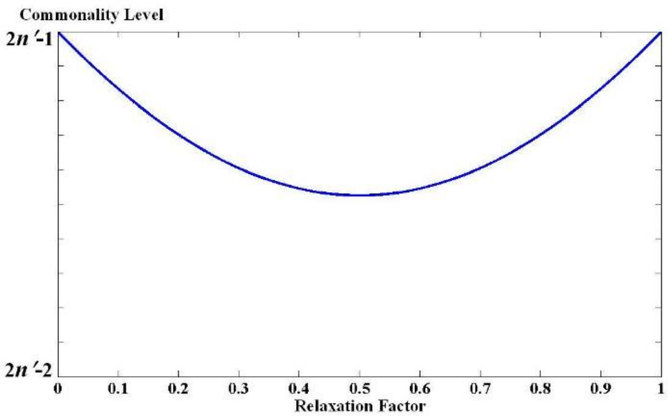

Figure 4. Commonality level change for case 3

The previous three cases can be considered as the basic cases in platform variable selection during optimization in a relaxed continuous space. That is, any change in the product family architecture during the optimization process can be regarded as a combination of these three cases: A distinct component becomes common with an existing platform; a platform component deviates from other members and finally become a distinct component; a platform variable deviates from other members and become common with another platform simultaneously. It should be noted that, since the system is nonlinear (as can be seen from the eigenvalue equations), the superposition principle cannot be applied for a general case. However, the analytical results for these simplified cases along with the numerical justifications for general product families, indicates the validity of the proposed metric for the continuous space.

2.3 CONSISTENCY CONSTRAINTS FOR COMMONALITY VARIABLES AND DESIGN VARIABLES

After defining a commonality index that remains valid in the continuous space, it is necessary to approximate Eq.(12) by a continuous function. As Fellini *et al*. mentioned, Eq.(12) should be approximated by a function that satisfies two requirements: Its range should be [0, 1] with correct values at end points, and it should be continuously differentiable. Using these two criteria, he defined the continuous commonality metric as:

$$\eta_k'^{ij} = \frac{1}{1+\left(\dfrac{x_k^i - x_k^j}{\alpha}\right)^2} \tag{27}$$

where α is a value between 0 and 1 that controls the degree to which the curve approximates the discontinuous step function in Eq.(12) that would describe the discrete nature of commonality: As the α value decreases, the optimal solution of the resulting continuous problem tends toward that of the discrete formulation. However, if α is close to zero, Eq.(27) becomes ill-conditioned, which leads to numerical errors and convergence difficulties. Therefore, the approach is to approximate the discrete formulation by a sequence of continuous optimization problems in which α value is decreased iteratively until variables that are designated as common fall within an acceptable deviation tolerance. However, the convergence properties of the iterative approach depend strongly on the form the approximating function. That is; the function should be selected so that its first derivative with respect to α becomes zero for the acceptable deviation tolerance for α value that the optimization algorithm can handle without numerical errors. We will consider this issue in detail later.

In addition, since Eq.(13) depends on the approximating function, it is desirable to select the function so that CI becomes continuous and differentiable. In this study, we propose two alternatives to Fellini's relaxation formula; in order to find the most well-suited form for Eq.(3), the alternatives will be compared with respect to the aforementioned criteria.

One possible candidate that has been used frequently in probabilistic models for pairwise comparison is the logistic curve. It can be applied as a commonality metric with some slight changes to the standard form, which we call "*half logistic curve*" for distinction:

$$\eta_k'^{ij} = \frac{2e^{-\Delta x_k/\alpha}}{1+e^{-\Delta x_k/\alpha}}, \qquad \Delta x_k = x_k^i - x_k^j \tag{28}$$

Another alternative for approximating Eq.(12) is the *Hubbert curve*, which is the derivative of the logistic function:

$$\eta_k'^{ij} = \frac{4e^{-\Delta x_k/\alpha}}{\left(1+e^{-\Delta x_k/\alpha}\right)^2} \tag{29}$$

First, these three functions will be compared according to their properties with respect to α. Eqs. (27)-(29). Their first derivatives $(\partial\eta'/\partial\alpha)$ are sketched as a function of α for a fixed tolerance (Δx =0.01) in Figures 5 and 6 respectively.

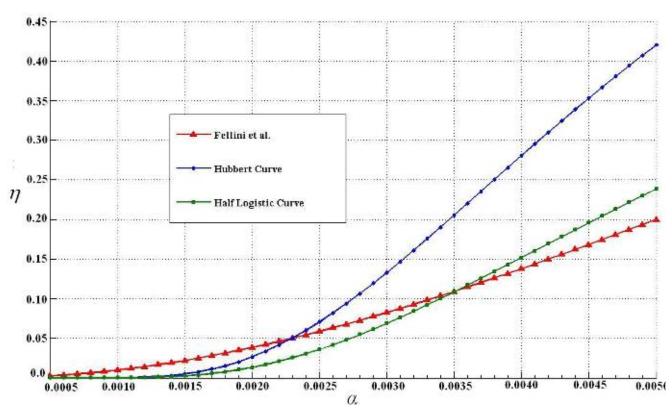

Figure 5. Proposed approximating functions with respect to α.

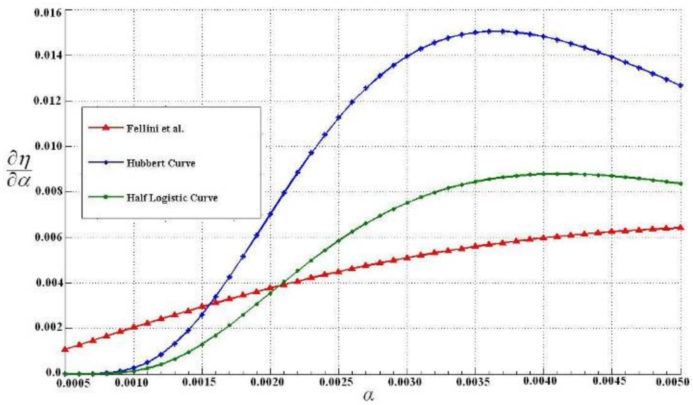

Figure 6. First derivative of the proposed approximating functions with respect to α

Figures 5 and 6 are sketched for $0.0005 \le \alpha \le 0.005$. As can be found from these figures, for both the Half Logistic and Hubbert curves, the first derivative approaches zero for $\alpha \le 0.001$. That is, staring from a larger α and decreasing its value during the iterative procedure, the optimization algorithm will be expected to converge for $\alpha \approx 0.001$ which can be applied in the optimization algorithm without numerical errors. However, for Fellini's proposed function, the curve doesn't approach a constant value even for $\alpha \le 0.0005$, which leads to convergence problems during the optimization process. On the basis of this comparison, the Half Logistic and Hubbert curves should be preferred over Fellini's curve.

Next, in order to investigate the effect of the approximating function on CI characteristics, all proposed functions are plugged in Eq.(11). Here, for simplicity and better visualization, we considered two platforms, each with ten components. We are interested to sketch the commonality change for a sequence of decreasing α values as a distinct component becomes common with the first platform, then deviates from it and becomes common with the second platform, and finally deviates from the second one and becomes distinct again. Results are shown in Figures 7-9 for Half Logistic, Hubbert and Fellini's curves respectively with $x_2 = 1$, $x_3 = 2$, x_1 as the variable on the x-axis and commonality change on the y-axis. As can be seen from these figures, all functions approach the discrete definition as α becomes small (in each case, the "inner" curves that are more sharp at x=1 and x=2 represent cases of smaller α, while the "outer" broader curves represent cases with larger α).

Figure 7 shows that in the case of the Half Logistic curve, CI is not differentiable with respect to Δx at the two extreme points which may cause numerical difficulties for optimization.

However, for both the Hubbert and Fellini's curves, CI has a continuous derivative with respect to Δx and is concave in the neighborhood of the local optima. Hence, with respect to the criteria of differentiability and convexity, Fellini's curve and Hubbert curves are superior to the Half Logistic curve.

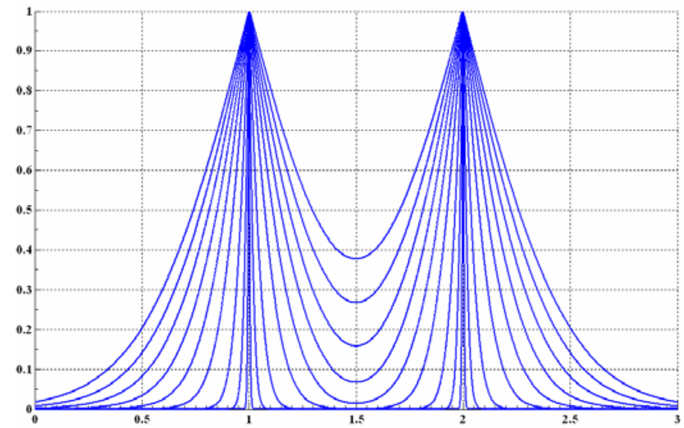

Figure 7. Continuous CI using the Half Logistic curve

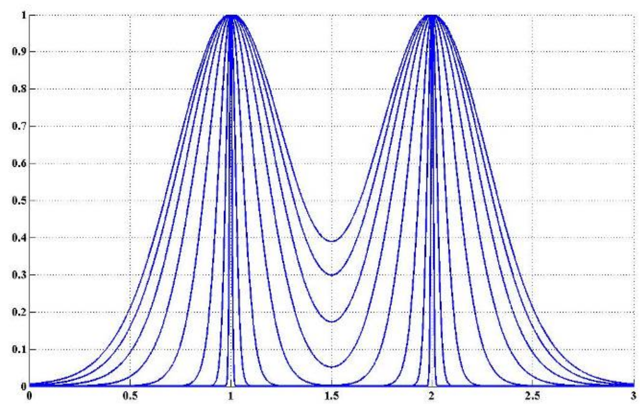

Figure 8. Continuous CI using Hubbert curve

In brief, according to the two aforementioned criteria, the Hubbert curve shows the best characteristics, since it converges to the discrete solution as we decrease α within a number of finite steps and results to a differentiable commonality index.

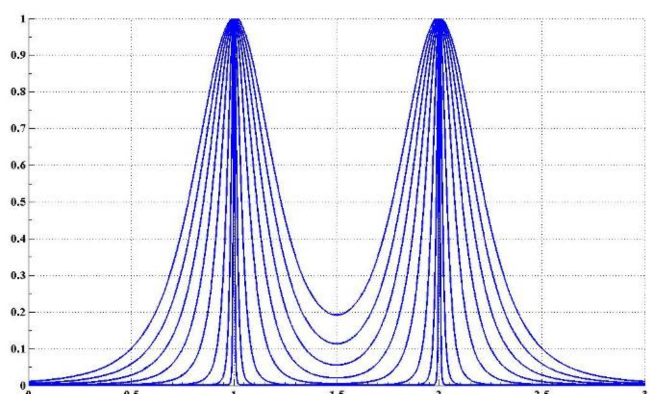

Figure 9. Continuous CI using Fellini's curve

3. CONCLUSIONS

In this study, we proposed a method for computing the commonality index introduced by Martin and Ishii for use in optimization and argued for its improved properties over prior metrics applied in optimization literature. The discrete definition of the commonality index was then relaxed to the continuous space, and properties of the index were examined using base cases. Results show that the proposed metric remains valid in the continuous space, enabling relaxation of the MINLP formulation into a continuous domain which enables use of gradient based approaches as the optimization method. Since, the proposed metric is a function of pairwise comparisons among all possible sets of products present in the family, the function applied for approximating the discrete definition has a considerable effect on the optimization performance. Hence, two important criteria for selecting the approximating function were described, and two proposed alternatives were compared with the prior method. The Hubbert curve showed to be the only alternative possessing both desired characteristics among the available options. Therefore, computing CI using the proposed method along with the Hubbert curve as the approximating function can enable new approaches to solve the product family optimization problem. One such approach using a single-stage gradient-based optimization algorithm with the index and relaxations developed in this paper to solve the joint platform configuration and product family design problem is presented in a companion paper [14].

ACKNOWLEDGEMENTS
This work is supported in part by the Pennsylvania Infrastructure Technology Alliance, a partnership of Carnegie Mellon, Lehigh University, and the Commonwealth of Pennsylvania's Department of Community and Economic Development (DCED).

REFERENCES

1. Collier, D. A., 1981, "The Measurement and Operating Benefits of Component Part Commonality," Decision Sciences, 12(1), pp. 85-96.
2. Fellini, R., Kokkolaras, M., Papalambros, P. and Perez-Duarte, A. Platform Selection under Performance Bounds in Optimal Design of Product Families, Journal of Mechanical Design, July 2005, Vol. 125, No. 4, pp. 524-535.
3. 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.
4. Khire, R. A., Messac, A., 2006, "Optimal Design of Product Families using Selection-Integrated Optimization (SIO) Methodology", 11th AIAA/ISSMO Multidisciplinary Analysis and Optmization Conference, Postmouth, Virginia.
5. 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.
6. Martin, M. and Ishii, K., 1996, August 18-22, "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation," Design Theory and Methodology – DTM'96 (Wood, K., ed.), Irvine, CA, ASME, Paper No. 96-DETC/DTM-1610.
7. Martin, M. V. and Ishii; K., 1997, September 14-17, "Design for Variety: Development of Complexity Indices and Design Charts," Advances in Design Automation (Dutta, D., ed.), Sacramento, CA, USA..
8. Messac, A., Martinez, M. P., and Simpson, T. W., 2002, "Effective Product Family Design Using Physical Programming and the Product Platform Concept Exploration Method," Eng. Optimiz., 34, pp. 245–261.
9. Nayak, R. U., Chen, W., and Simpson, T. W., 2002, "A Variation-Based Methodology for Product Family Design," Eng. Optimiz., 34, pp. 69–81.
10. Siddique, Z., Rosen, D. W. and Wang, N., 1998, September 13-16, "On the Applicability of Product Variety Design Concepts to Automotive Platform Commonality," *Design Theory and Methodology*, Atlanta, GA, USA..
11. Simpson, T. W., and D'Souza, B., 2004, "Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multi-objective Genetic Algorithm," Concurrent Engineering: Research Applications, 12, pp. 119–130.
12. Thevenot, H. J. and Simpson, T. W. (2006), "Commonality Indices for Product Family Design," A Detailed Comparison, Journal of Engineering Design, 17:2 (99-119).
13. Wacker, J. G. and Trelevan, M., 1986, "Component Part Standardization: An Analysis of Commonality Sources and Indices," *Journal of Operations Management*, **6**(2), pp. 219-24
14. Khajavirad, A. and J. Michalek (2007) "A Single-Stage Gradient Based Approach for Solving the Joint Product Family Platform Selection and Design Problem using Decomposition," *ASME International Design Engineering Technical Conferences*, Las Vegas, Nevada, USA.
15. Linear Algebra, with Applications to Differential Equations and Probability TM Apostol, MVC CALCULUS - 1967 - Blaisdell Publishing Company.
16. Nelson, S.A., II, Parkinson, M.B., & Papalambros, P.Y., 2001, "Multicriteria optimization in product platform design". *ASME Journal of Mechanical Design 123(2)*, 199–204.
17. Khajavirad, A., J. Michalek and T. Simpson (2007) "An Efficient Decomposed Genetic Algorithm for Solving the Optimal Joint Product Platform Configuration and Product Family Design Problem," to appear, *Proceedings of the 3rd AIAA Multidisciplinar Design Optimization Specialists Conference*, Honolulu, Hawaii, USA.

18. Thevenot, H. J. and Simpson, T. W. (2006) "A Comprehensive Metric for Evaluating Commonality in a Product Family," *ASME Design Engineering Technical Conferences - Design Automation Conference*, Simpson, T. W., ed., Philadelphia, PA, USA.