UNIVERSITY OF ZIELONA GÓRA
FACULTY OF MECHANICAL ENGINEERING
in association with the Design Society

THIRD INTERNATIONAL SEMINAR AND WORKSHOP
10^{th} - 12^{th} October 2002
Zielona Góra - Łagów
POLAND

ENGINEERING DESIGN IN INTEGRATED PRODUCT DEVELOPMENT
Design Methods that Work

DESIGN OPTIMIZATION PRACTICE IN PRODUCT DEVELOPMENT

Panos Y. Papalambros

University of Michigan
Department of Mechanical Engineering
e-mail: *pyp@umich.edu*

Keywords: design optimization, mathematical programming, simulation-based modeling, decomposition, product families, analytical target cascading, investment modeling.

Abstract: *Design optimization combines mathematical optimization algorithms with engineering analysis models to generate designs with improved performance. In product development this approach is useful for products with a large number of interdependent design decisions or for new products where significant experience has not yet been accumulated. Current efforts are directed primarily towards complex and new technology products, and the augmentation of engineering analysis models with business or "enterprise" performance models, so that optimization results become more meaningful to management. The article reviews these issues adopting an industry application viewpoint.*

INTRODUCTION

The term "optimization" is widely used in a rather loose way to indicate doing something better than the way we are doing it currently. In product development the term is often used in a similar manner to indicate making product decisions that yield a better product. In mathematics optimization in a formal term that describes the process of locating the optimum (minimum or maximum) of a function, possibly subject to several constraints. An optimization problem can be written in a canonical form (here in the so-called negative null form) as follows

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{subject to:} \qquad (1) \\ &\mathbf{h}(\mathbf{x}) \; ? \; \mathbf{0} \\ &\mathbf{g}(\mathbf{x}) \; ? \; \mathbf{0}, \end{aligned}$$

where f is the objective function to be minimized, **h** and **g** are vectors representing the functions in the equality and inequality constraints, respectively, and **x** is the vector of variables that we have to find values for in order to solve the problem of Eq. (1). In product design the objective is a criterion of product performance that should be optimized, the constraints are design requirements that must be satisfied, and the variables are the design quantities that we have the ability to assign values to, as we seek the best design [1].

The ability to represent product decisions in the context of a mathematical model, such as Eq. (1), is clearly limited by our ability to model the entire design situation in the required functional form. It is unlikely to be able to derive proper mathematical models for every design decision or requirement, and so the results from such an optimization will be optimal only with respect to the model used. In this sense the limits to optimization practice are imposed by the limitation of our modeling abilities. This is a general theme throughout this article.

The other issue of immediate concern is the actual solution of the mathematical problem stated in Eq. (1). Such a solution is far from easy when the functions have "bad" mathematical properties, such as nonlinearity, nonconvexity, discontinuities and nondifferentiabilities. Handling these situations can become quite esoteric; some simple ideas that can be helpful in many situations will be discussed.

Design optimization becomes more attractive when the number of decisions to be made becomes too large for relying on intuition and past experience. We often refer to those problems as large-scale or complex problems. The usual approach we take is to partition the original problem into a collection of smaller subproblems, whose solution will yield the solution of the original. Such decomposition strategies become increasingly necessary in practical product design. Their successful implementation requires careful construction of the partitioned problem and of the coordination strategy required to solve all the subproblems in an efficient and compatible manner. Here we will discuss such a strategy, called analytical target cascading, tailored specifically to product development.

The value of the optimization concept becomes often more apparent when facing new product decisions. A common product development strategy is to design a set of product variants tailored to different market segments derived from the same concept and sharing common elements, collectively called a platform. Commonality acts as a constraint on what could be individually optimized variants, so a trade-off exists between maximizing sharing to reduce costs and time, and minimizing the effect of not achieving the ideal variant optima due to the additional constraints. We will look at how design of product families can be stated as a formal optimization problem and the insights that can be gained from its solution.

Design of engineered products can only be done in the context of the producing enterprise and the market in which the product must exist. Traditional design optimization has been limited to design decisions about engineering performance. Product success for both producer and user clearly depends on other requirements, including production requirements, marketing, and investment strategies, collectively referred to as enterprise-wide design. In an effort to bring design optimization into a more central position within the enterprise, and thus increase its value and impact, there is increased effort in augmenting the engineering physics models of performance with models from production, economics, investment science and marketing. We will present ideas of how this augmentation can be accomplished through a product portfolio design problem.

The following sections will provide a limited review of the issues raised above, namely, modeling design responses, some characteristics of the requisite optimization algorithm toolbox, analytical target cascading, optimal product family design, and enterprise-wide optimal product design. More details can be found in cited references.

1. MODELING DESIGN RESPONSES

Optimization as a formal method in the design of products first gained attention during the 1970's starting primarily in the aerospace and chemical industries, followed soon after by the automotive industry. The development and increasingly extensive use of computer-based analysis methods, generally referred to as computer-aided engineering (CAE) tools, has changed the way product development is practiced today. The use of optimization follows closely the availability of CAE tools in the practicing engineering community. For example, the popularity and increased robustness of finite element structural analysis has made structural optimization widely used.

Design optimization use is also more likely to be found in high technology areas, where performance is pushed to extremes and traditional knowledge is either lacking or not codified. For example, structural optimization is extensively used in the aerospace and automotive industries, but is quite limited in the areas of civil and mechanical engineering practice where designs are tightly governed by conservative codes and standards developed over centuries of experience.

The way the response of a given design is computed determines the mathematical and practical nature of the optimization problem. Modern applications will typically include CAE-based analysis tools to compute responses. These tools, often referred to as "simulations", typically involve numerical solution of differential or integral equations, assuming the design as given and computing its responses, namely, the functions f, $\mathbf{h}$ and $\mathbf{g}$ in Eq. (1).

Simulation-based optimization addresses design problems where the objective and/or constraint functions are not expressed with closed-form analytical equations, but with "black box" computer simulations. Typically such functions will be noisy or discontinuous (i.e., non-smooth) and may require long computing time for each function evaluation. For example, Figure 1 shows the fuel economy of an automobile as a function of the final drive ratio. The response is noisy, i.e., there are small perturbations about an underlying trend, and has several discontinuous jumps, i.e., large changes in the response for small changes in the design variable. Numerical optimization with such functions poses several challenges.

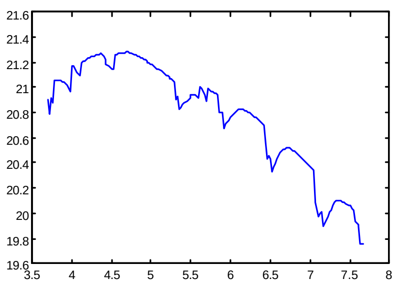

Figure 1 : Example of simulation-based design response

The usual optimization techniques (e.g., Newton's method) employ gradient information from finite differencing to guide a sequential strategy towards the maximum. The noisy nature of the simulation renders finite differencing ineffective at estimating gradients and prevents the algorithm from successfully progressing towards the optimum. Small finite differencing steps will lead the algorithm to a local noise optimum, and large steps will not capture the underlying slope trend accurately.

This difficulty can be dealt with in several ways. A non-gradient optimization algorithm may be used, such as genetic algorithms, but if the function computation cost is too high this approach will not work, since a very large number of function evaluations is needed to achieve reasonable certainty about locating the optimum. Another approach is to adjust empirically both the analysis integration step (which is usually the culprit in noise generation but also determines the analysis solution accuracy) and the differentiation step of the gradient approximation until an acceptable compromise between gradient approximation accuracy and simulation cost is achieved.

Yet another approach is the use of approximations. An approximate model can provide a smooth functional relationship of acceptable fidelity to the true function with the added benefit of small computational cost. The approximate model can be used in conjunction with a gradient-based algorithm or in entirely different ways. How to build and exploit approximations effectively in simulation-based optimization is a thriving research area.

Of particular current interest is the kriging approximation. Kriging is a data interpolation scheme with roots in geostatistics, adapted for data coming from deterministic computer simulations. This form of data collection and approximation is known as Design and Analysis of Computer Experiments (DACE). Kriging models can be used directly with other standard algorithms, or become the basis for a type of non-gradient global optimization algorithm, as we will discuss in the next section.

2. THE OPTIMIZATION TOOLKIT

An often-asked question is "Which is the best optimization algorithm?" The answer comes from the maxim that "The best algorithm is the one you know best!" The truth of this answer derives from the facts that there is really no algorithm that can solve all problems reliably and efficiently, and that any good algorithm will always need some adjustment or "tuning" to achieve solution robustness for any given problem. Experience with a given algorithm *and* with the specific code implementation is a sine qua non for success.

Having said that, we can still provide some basic guidelines for a good optimization toolbox. The workhorse of the toolbox should be a standard sequential quadratic programming (SQP) code with line search options for both exact and inexact line search. When function cost is high and gradients relatively inaccurate, the line search can become very expensive so inexact searches (using only a "sufficient descent" criterion) are better. For the same reason, it is also good to have an SQP variant that handles quadratic subproblem infeasibility with a trust region instead of line search. The SQP algorithm has also the advantage that requires really no tuning. However, its successful use does depend strongly on the scaling of the problem functions, so one should not use SQP without some scaling experimentation. There are many and good code implementations of SQP to choose from.

If one wishes some more options for gradient-based algorithms, good choices would be the generalized reduced gradient (GRG) and Augmented Lagrangian (AL) algorithms. In GRG we get feasibility assurance at the end of every iteration, at a cost of extra computational effort. Handling inequalities relies on an active set strategy that tends to be heuristic. The actual code is conceptually simple but lengthy to implement. Tuning program parameters is often necessary to get reliable performance for large problems. There are some good code implementations and GRG has generally been popular among engineers. AL algorithms have probably not been as extensively used and developed as they may deserve. They can be robust and work well in problems with large numbers of equality constraints.

A second important tool in the box should be a non-gradient-based global search algorithm employing some heuristic search strategy, which will not get hang up on noise, local minima and discreteness. A genetic algorithm (GA) or a Lipschitzian algorithm [2] is a good choice. The simple structure of GAs is very attractive to the non-mathematically minded but computational cost is high and tuning can be excessive. Nevertheless, if one can afford it, when faced with a likely troublesome problem running a GA "just to see what you get" is a good idea.

Another choice for a global search tool is an algorithm that looks for the optimum by creating increasingly better global approximations of the original problem functions, in the vicinity of likely optima. Such algorithms utilize statistical models of the functions to define an infill sampling criterion (ISC). The criterion determines which design points to sample next (the so-called infill samples). The infill samples are then evaluated on the true functions and the models updated. This is a completely different method than algorithms following a search path because the sampling criterion could place the next iterate anywhere in the design space. Eventually, with enough sampling, the approximate model is close enough to the true model so that their corresponding optima will closely match.

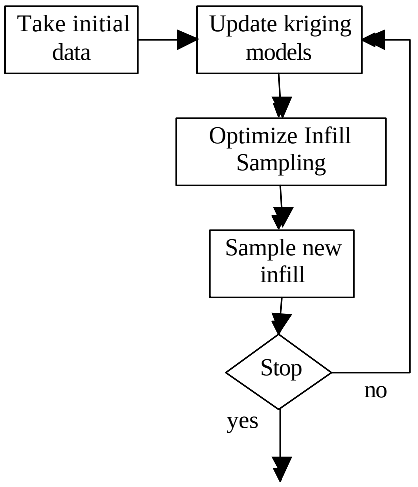

Figure 2: Flowchart of the EGO algorithm

These algorithms have become increasingly attractive for simulation-based optimization and deserve some more discussion here. They are known as Bayesian analysis algorithms [3] because they use statistical models to predict future outcomes.

One particular Bayesian analysis optimization algorithm is known as Efficient Global Optimization (EGO) [4, 5, 6], and employs kriging models as the approximation method. The basic steps in EGO follow (see also Figure 2).

1. Use a space-filling design of experiments to obtain an initial sample of the true functions.
2. Fit a kriging model to each function.
3. Numerically maximize a sampling criterion known as the *expected improvement* function to determine where to sample next.
4. Sample the point(s) of interest on the true functions and update the kriging models.
5. Stop if the expected improvement function has become sufficiently small, otherwise return to 2

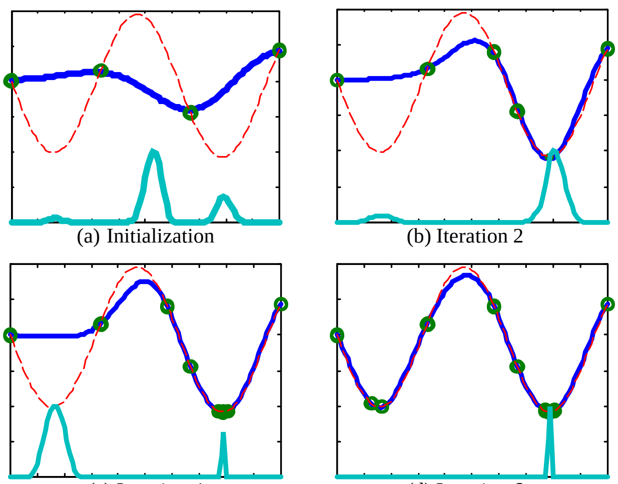

Figure 3: Demonstration of the search strategy of the EGO algorithm.

Looking at a one-dimensional multimodal example in Figure 3 gives an idea of how this works. The w-shaped dashed line is the true objective function we wish to minimize, while the solid line is the kriging approximation conditional to the sample points shown as circles. The plot at the bottom is the sampling criterion, normalized to facilitate comparisons between iterations. The infill sampling criterion is indeed the expected improvement function and it determines where EGO will evaluate the functions. It tends to choose the design points most likely to improve the accuracy of the model and/or have a better function value than the current best point. After the initial sample of four points, the resulting kriging model is a poor fit to the true function. However, the expected improvement function leads the algorithm to sample points where the uncertainty in the model is highest. After two iterations, the model has improved in the region of the local optimum on the right, and the expected improvement function leads EGO to sample another few points in that region where there is a high probability that a better point can be found. By the fourth iteration, the region on the right has been explored, but the uncertainty in the model on the left portion of the model drives EGO to sample points in that region. After six iterations, both local optima have been discovered and the true solution has been found quite accurately.

Because the process of fitting the kriging models and locating the maximum of the infill sampling criterion are optimization problems themselves, the overhead associated with EGO can be significant. Other methods such as genetic algorithms or gradient-based algorithms on the other hand require very little computational effort in determining where to evaluate the functions next. However, they require a large number of function evaluations to converge on a good solution. The benefit of the overhead of Bayesian analysis algorithms is that each iteration uses as much information as possible in determining where to evaluate the functions next, enabling them to locate good solutions with fewer iterations. This makes the Bayesian analysis algorithms best suited to situations where the functions are really expensive.

Finally, the toolbox should have some ability to handle discrete variables. A GA would be quite suitable if the model is purely discrete. Many practical problems are mixed-discrete with unknown (typically nonlinear) mathematical form for the functions. These are truly difficulty problems that can be handled on a case-by-case basis through some type of branch-and-bound or random search strategy, taking any advantage possible from what you know about the physics of the problem and from any possible modeling simplifications. There are no real "standard" toolkit methods for these problems yet.

3. TARGET CASCADING

Target cascading is a key challenge in early product development: How to propagate desirable product characteristics, defined by product's specifications, to the various subsystems and components in a consistent and efficient manner. Consistency means that all parts of the designed system should end up working well together, while efficiency means that the process itself should avoid iterations at later stages, which are costly in time and resources.

Analytical target cascading (ATC) is formalized in a process modeled as a multilevel optimal design problem [7-10]. Design targets are cascaded down to lower levels by partitioning the overall design problem into a hierarchical set of subproblems. For each design subproblem at a given level, a rigorous optimization problem is formulated to minimize deviations from the propagated targets and thus achieve intersystem compatibility. A coordination strategy links all subproblem decisions so that overall product performance targets are met.

ATC requires a hierarchical decomposition of the product and the underlying models into systems, subsystems and components, collectively referred to as elements. An element's design response **R** is a function of the element's own design variables **x** as well as of the responses of (sub)elements making up the element. For each level and for each element in the model hierarchy, a design optimization problem is formulated to match responses dictated by elements above in the hierarchy and to satisfy the element's design constraints. ATC is general enough to account for multiple levels and for interactions between elements at the same level by means of linking design variables.

Now we describe how ATC is applied to the design of an advanced heavy tactical truck [11]. Novel technologies (e.g., series hybrid and electric propulsion systems, in-hub motors, and variable height suspensions) are introduced with the intention of improving both civilian and military design attributes within the framework of a dual-use philosophy. Emphasis is given to fuel economy, ride, and mobility characteristics. A two-level target cascading hierarchy is defined, and five design subproblems are formulated. At the top level, design targets for the truck are matched; at the bottom level, suspension characteristics, cascaded down from the top level, are matched using a detailed model of the suspension system. A schematic of the vehicle configuration is shown in Figure 4. Left and right tires are combined into one tire because of symmetry.

The vehicle is powered by a diesel engine that is connected to the generator through a speed reduction gearbox. The electrical power of the generator, through the power bus, is fed to eight in-hub motors that drive the wheels through a gearbox at each wheel. Additional power may be taken from the battery modules connected to the power bus. Distribution of power and charging and discharging of the batteries are managed by the power control module based on instantaneous power requirements.

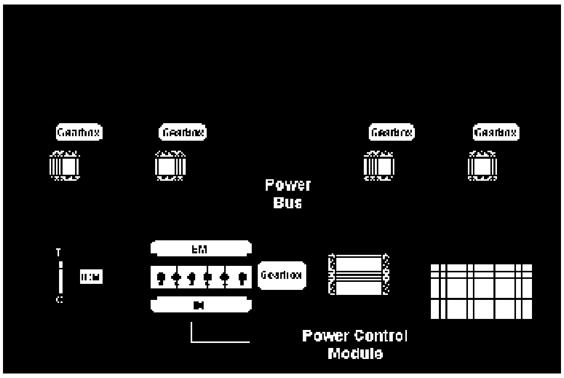

Figure 4: Vehicle model configuration [11]

Models were developed to simulate the transient response of both a series hybrid and an electric-driven truck at the top (vehicle) level, and the response of variable height suspensions at the bottom (system) level. The models at both the vehicle and system levels were tailored to fit the target cascading process. Automated modeling techniques were used to develop vehicle dynamics models that are computationally efficient, accurate, and described by physical parameters. Baseline designs were chosen to be consistent with vehicle concepts of the U.S. Army, whereas vehicle targets were defined to improve on the performance of existing designs.

At the vehicle level, responses $\mathbf{R}_v$ must match desired design specifications $\mathbf{T}_v$. These responses are assumed to be functions of vehicle design variables $\mathbf{x}_v$ and system responses $\mathbf{R}_{s_i}$, for $i = 1,\ldots,n_s$ systems, i.e., $\mathbf{R}_v = \mathbf{r}_v(\mathbf{x}_v, \mathbf{R}_{s_1},\ldots, \mathbf{R}_{s_{ns}})$. To determine target values for system responses $\mathbf{R}_{s_i}$ and vehicle design variables $\mathbf{x}_v$ a minimum deviation optimization problem is formulated as follows:

$$
\begin{aligned}
&\text{Min} \quad \left\| \mathbf{R}_v - \mathbf{T}_v \right\| + \varepsilon_v^R \\
&\text{with respect to} \\
&\mathbf{x}_v, \mathbf{R}_{s_1}, \ldots, \mathbf{R}_{s_{n_s}}, \varepsilon_v^R \\
&\text{subject to} \\
&\sum_{i=1}^{n_s} \left\| \mathbf{R}_{s_i} - \mathbf{R}_{s_i}^L \right\| \le \varepsilon_v^R \\
&\quad \mathbf{g}_v(\mathbf{R}_v, \mathbf{x}_v) \le 0 \\
&\quad \mathbf{h}_v(\mathbf{R}_v, \mathbf{x}_v) = 0
\end{aligned}
\tag{2}
$$

where $\mathbf{x}_v$ is the vector of design variables exclusively associated with the vehicle, $\mathbf{R}_v$ is the vector of vehicle responses, $\mathbf{R}_{s_i}$ is the vector of responses for the *i-th* system making up the vehicle, ε_v^R is a tolerance variable for coordinating system responses, $\mathbf{T}_v$ is the vector of vehicle design targets or specifications,

$\mathbf{R}^L_{s_i}$ is the vector of system response values passed up to the vehicle from the *i-th* system, and $\mathbf{g}_v$ and $\mathbf{h}_v$ are vector functions representing vehicle design constraints.

Once the optimal values of the system level responses $\mathbf{R}_{s_i}$, $i = 1,\ldots,n_s$, are determined by solving the vehicle-level design problem shown above, they are cascaded down to the system level as target values $\mathbf{R}^U_{s_i}$. At the system level, n_s individual minimum deviation optimization problems are formulated to determine the system design variables $\mathbf{x}_{s_i}$. System responses are assumed to be functions of system design variables alone, i.e., $\mathbf{R}_{s_i} = \mathbf{r}_{s_i}(\mathbf{x}_{s_i})$, given the two-level hierarchy assumption. The minimum deviation optimization problems for the $i = 1,\ldots,n_s$ systems are formulated as follows:

$$
\begin{aligned}
&\text{Min} \quad \left\| \mathbf{R}_{s_i} \ ?\ R^U_{s_i} \right\| \\
&\text{with respect to } \mathbf{x}_{s_i} \\
&\text{subject to} \\
&\mathbf{g}_{s_i}(\mathbf{R}_{s_i}, \mathbf{x}_{s_i}) \ ?\ 0 \\
&\mathbf{h}_{s_i}(\mathbf{R}_{s_i}, \mathbf{x}_{s_i}) \ ?\ 0
\end{aligned}
\qquad (3)
$$

The analytical target cascading process iterates between the vehicle- and system-level design problems. At each iteration, values of $\mathbf{R}^L_{s_i}$ and $\mathbf{R}^U_{s_i}$ determined at the system and vehicle levels, respectively, are passed up and down to the other level design problem(s).

For target cascading, the models are decomposed into an integrated vehicle model at the top level and four copies of a suspension system model at the low level, as shown in Figure 5. The top level vehicle model predicts the vehicle responses $\mathbf{R}_v$, whereas the suspension model predicts the system responses $\mathbf{R}_{s_i}$, $i = 1, 2, 3, 4$.

Two sets of targets were used for this concept truck design study. In the first part of the study, the goal was to achieve a fuel efficiency that would be better by at least 50% than the one of the Heavy Expanded Mobility Tactical Truck (HEMTT). Fuel economy was computed at both the Gross Vehicle Weight (GVW, truck weight plus payload, 22,977 kg) and the Gross Combined Vehicle Weight (GCVW, GVW plus trailer, 38,886 kg). Performance metrics were to be maintained at the same levels as those of the HEMTT. In the second part of the study, specific numerical values were not defined as targets; instead, it was attempted to improve all metrics as much as possible.

Attributes of interest for both the series hybrid and the electric drive are compared in Figure 6 for the first set of targets. Results are normalized such that 1.0 represents the target values. The first three targets are achieved. The last three responses are improved compared to the baseline design but do not achieve the targets. The maximum speed

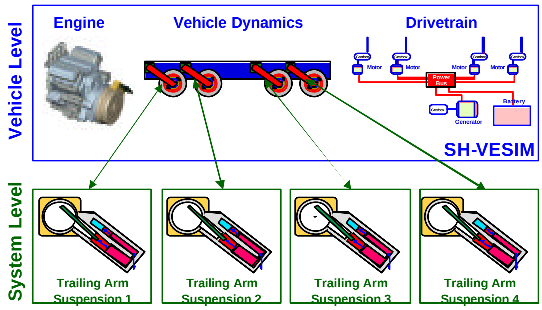

Figure 5: Hierarchy of models for target cascading

degradation for the series hybrid configuration comes very close to the target. Note that responses that achieve or exceed their targets become "neutral" to the optimizer, i.e., they do not contribute to the objective

Thus, responses estimated for GVW achieve their targets, while responses estimated for GCVW do not. One can deduce that performance targets at GCVW were too stringent. The same trend is observed for the second set of targets presented in Figure 7. Results are normalized with respect to the first set of targets for the sake of comparison. In this case the optimizer tries to improve all responses as much as possible without limits. This leads to a dramatic increase of the ride quality. Hybrid vehicle fuel economy shows more modest, but tangible further improvements over the conventional at both GVW and GCVW, the actual fuel economy gains being 17.4 % and 15.6 % compared to the baseline design, respectively. Performance metrics of the hybrid are maintained at about the same level as in the previous study. For the electric drive propulsion system only the ride quality displays significant further improvement compared to the optimization performed for the specific targets, while all other responses remain about the same.

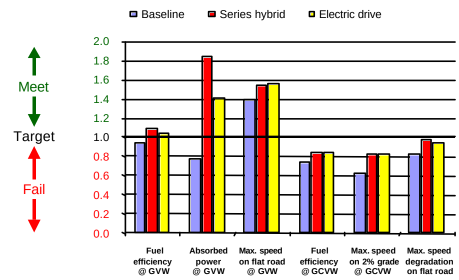

Figure 6: Achievement of specific targets [11]

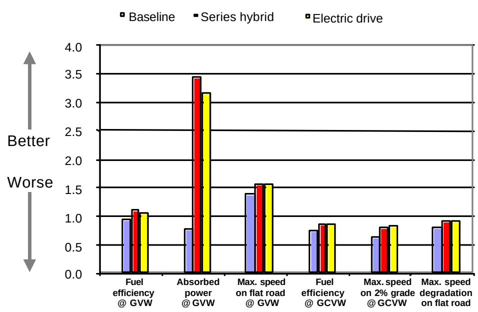

Figure 7: Achievement of maximal improvement targets [11]

4. PRODUCT FAMILIES

Sharing components among products is an effective way to cut costs. Identification of the product platform is a key step in designing a family of products since sharing components usually results in performance losses relative to the individually optimized variants. Here we describe how this problem can be addressed in an optimization framework. More details can be found in the references [12-16].

4.1 Platform-based Design

A product platform is the set of all components, manufacturing processes, and/or assembly steps that are common in a set of products. A product family is the set of products, referred to as variants, built upon a platform. Component sharing, where one or more components are common across a family of products, is illustrated in Figure 8.

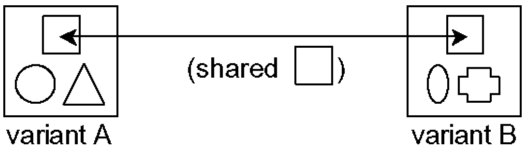

Figure 8: Platform-based products (component sharing).

The following additional definitions are necessary to formulate the variant and family design problems: $P = \{A, B, C, \ldots\}$ is the set of m products; $\mathbf{x}^p$ is the vector of design variables for product p in P; S is the set of indices describing a platform; and $\mathbf{x}^{p,0}$ is the "null" platform optimal design of product p, corresponding to the individually optimized product variants.

The individual and the family optimal design problem for product variant p can be formulated, respectively, as follows:

$$\begin{aligned} \min_{\boldsymbol{x}^p} \quad & f^p(\boldsymbol{x}^p) \\ \text{subject to} \quad & \mathbf{g}^p(\boldsymbol{x}^p) \le 0 \end{aligned} \tag{4}$$

$$\begin{aligned} \min_{\boldsymbol{x}^A, \boldsymbol{x}^B, \ldots} \quad & \{ f^p(\boldsymbol{x}^p) \mid p \in \mathcal{P} \} \\ \text{subject to} \quad & \mathbf{g}^p(\boldsymbol{x}^p) \le 0 \quad p \in \mathcal{P} \\ & x_i^p = x_i^q \quad i \in \mathcal{S}, \; p, q \in \mathcal{P}, \; p < q \end{aligned} \tag{5}$$

Platform selection would then require the following steps: Quantify sharing penalty by considering individual optimal designs and sensitivities of functional requirements; decide which components can be shared (i.e., determine the platform) based on minimal sharing penalty; optimally design the product family around the chosen platform.

When the products in the family contain a large number of components that are candidates for sharing, platform selection entails the solution of a large combinatorial problem.

4.2 Sharing Penalty Vector

The platform problem is deciding which and how many variables to share. The selection of an optimal platform can then be done by minimizing the relative variation of performance of the designs based on any platform with n shared variables - while remaining in the feasible space for the variants. Formally, for two variants A and B this translates to:

$$\begin{aligned} \min_{\mathcal{S} \in \mathbb{S}} \quad & \Delta \\ \text{subject to} \quad & |\mathcal{S}| = n \end{aligned} \tag{6}$$

where $\Delta = \Delta^A + \Delta^B$, and for $p \in \mathcal{P}$:

$$\Delta^p = | f^p(\boldsymbol{x}^p) - f^p(\boldsymbol{x}^{p,0}) | + \sum_{j \in C^p} \max(g_j^p(\boldsymbol{x}^p), 0)$$

The Sharing Penalty Vector (SPV) is computed using first order Tailor series and a heuristic parameter ? in [0, 1], determined by the position of the family solution for a given platform relative to the position of the null platform solutions for the two variants:

$$\begin{aligned} &\mathrm{SPV}_i = \\ &(1-\alpha)\Big(\big| \nabla f^{A,o}{}_i \big| \, \delta_i + \sum_{j \in C^A} \max(\nabla g_j{}^{A,o} \delta_i, 0) \Big) \\ &+ \; \alpha \Big(\big| \nabla f^{B,o}{}_i \big| \, \delta_i + \sum_{j \in C^B} \max(\nabla g_j{}^{B,o} \delta_i, 0) \Big) \end{aligned} \tag{7}$$

The SPV essentially aggregates the sensitivities of the solutions and allows identification of variables, with respect to which the optimal solution is least sensitive. Sharing those will have little impact on individual variant optima. The SPV represents an upper bound on ? ???The design variables can be sorted in order of increasing associated SPV value and a cutoff point for sharing is selected by the designer when SVP values become too large.

The general methodology is as follows:

1. Generate product variants based on design requirements and/or geometry of the model(s) (i.e., no topological changes).
2. Develop appropriate analysis models and identify inputs and outputs.
3. Formulate and solve the optimal design problem for each variant, i.e., find null platform optimal designs.
4. Compare optimal design variable and sensitivity information by selecting ? and computing the sharing penalty vector (SPV) using Eq. (7).
5. Arrange the variables in order of increasing associated SPV value.
6. Using the SPV, decide which components to share or not share.
7. Formulate and solve the family design problem with the chosen platform.
8. Compare family optimal designs to individual variant optimal designs and evaluate performance losses (iterate if necessary).

4.3 Automotive body design application

Two automotive body structure variants are considered with different dimensional properties (lengths) and functional requirements. The structures are modeled using finite elements in MSC-NASTRAN (Figure 9). Modal and static load cases (torsion on the front and rear shock towers, and bending) are considered. The finite element analysis outputs mass *m* and natural frequencies ?, in addition to displacements and stress responses for static load cases of front torsion, rear torsion and bending (denoted d_{ft}, d_{rt}, d_b), as well as corresponding sensitivity information for the design variables, these being the cross-sectional dimensions of the beams (width *b*, height *h*, and thickness *t*) and shell thickness *t* —a total of 66 design variables.

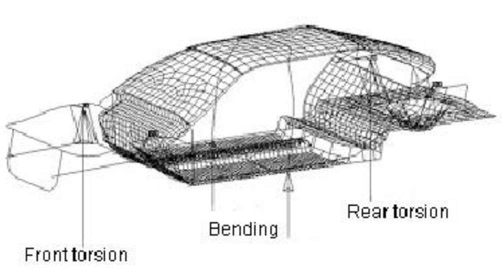

Figure 9: Automotive body structure mode[15].

We consider variants based on the same geometric model (short wheelbase) that are defined by different design objectives and constraints. Two variants with competing objectives are designed, denoted "stiff" and "lightweight", respectively. With a "stiff" objective the designer aims at maximizing the stiffness of the structure to improve ride quality; with a "lightweight" objective the goal is to minimize weight to improve fuel economy.

Flexibility is defined as a weighted sum of the displacements in the three load cases considered (front and rear torsion and bending). The weights approximate the ratios of the expected displacements in each load case, hence flexibility is computed as

$$f = d_{ft} + d_{rt} + 10d_b. \quad (8)$$

Each variant is optimized individually to obtain a null platform design. The individual optimization problems and their corresponding optima are summarized in Table 1.

Table 1: *Optimal design problems and a ssociated null platform optima .*

Variant	Light Weight	Stiff
Obj.	min Mass	min Flexibility
Var.	Beams: b, h and t; Shells t 66 variables	
Constr.	15 Hz $\leq \omega_1$	21 Hz $\leq \omega_1$
	17 Hz $\leq \omega_2$	24 Hz $\leq \omega_2$
	$d_{ft}, d_{rt} \leq 2.9$ mm	mass ≤ 822 kg
	$d_b \leq 0.5$ mm	—
	$\sigma_{max} \leq 25$ MPa	
Opt.	Mass = 691.87 kg	$\varphi = 4.4049$ mm

The SPV for the two variants is computed according to Eq. (7). The family design solution for the total platform represents an upper bound on performance loss. The SPV remains low for the first 50 variables, then begins to increase sharply. We chose a 54-variable platform based on this fact. Table 2 shows the platform-based product solutions, obtained by minimizing the distance between the Pareto set for each platform and the null platform optima. The 54-variable platform shares all but 18% of the variables, and presents a loss in performance of 0.6% for the stiff variant and 1.16% for the lightweight variant. In contrast, the total platform presents a 1.4% loss in performance for the stiff variant and an 18.8% loss for the lightweight variant.

4.4 ATC for product families

Once it has been decided which subsystems, components, etc., will be shared among the products of a family, the statement of the optimal design problem for the entire family fits the formulation of analytical target cascading perfectly. Therefore, the formulation of single-product target cascading has been extended to account for multiple-products. In addition, to account for the presence of both family and individual product specifications the concept of "local" targets was introduced since the previous formulation allowed design targets to be set only at the top level. The extended formulation was applied to a family problem of automotive side frames [16].

Table 2: *Comparison of null platform and 54-variable and total platform.*

Platform	null platform		54-variable platform		total platform	
Variant	Stiff	L. Weight	Stiff	L. Weight	Stiff	L. Weight
mass (kg)	822	691.87	822	699.90	822	822
d_{ft}(mm)	1.581	2.429	1.595	2.270	1.607	1.607
d_{rt} (mm)	1.396	2.148	1.409	1.007	1.419	1.419
d_b (mm)	0.1427	0.2922	0.1429	0.2829	0.1443	0.1443
flexibility(mm)	4.405	7.499	4.433	7.107	4.468	4.468

5. ENTERPRISE MODELS

We now look at how to formulate an "enterprise-wide" decision problem where the net present value of the products under consideration for development is the objective function. The objective depends on both asset allocation variables and product design variables, thus linking product design with financial goals. Constraints are imposed to represent the fiscal perspective of costs and revenues and the technical perspective of product performance. The analysis models used to compute the outcomes of those decisions are computationally intensive simulations based on both investment and engineering science.

5.1 Product Portfolio Design

We demonstrate this idea in a product portfolio valuation problem [18]: Given a production capacity and cost structure, historical data on sales, and two possible products that can be built in the existing production facility, what percent of production capacity should be allocated to each product, and what should be the design specifications for these products? The new idea here is that asset allocation and properties are determined concurrently. We consider decisions to be made by an automotive manufacturing firm that markets premium-compact (PC) and full-size sport utility (SUV) vehicles. This market segmentation adopted in the study follows the J.D. Power classification for vehicles in the United States.

The firm is assumed to operate in a mature industry where complementary assets, such as access to distribution channels, service networks, etc., are assumed available. It is further assumed that the firm's output decision does not affect the product's price. Finally, the decision-maker is assumed to be playing a game against nature: strategy is affected by an exogenously-generated random state, not by competitive interaction. The firm wishes to design new engines and transmissions for both PC and SUV segments. The PC and SUV segments are low and high profit margin segments, respectively. There are *C* units of monthly capacity currently in place for each product (engine and transmission), so *C* is fixed and represented in a capacity constraint. It is assumed that this capacity is not expandable.

The decision-maker faces the following decisions: How should the units of capacity be allocated between the two segments in order to maximize the investment's net present value? What should the performance specifications for engines and transmissions be and how do these specifications affect the resource allocation decision? Consistent with capital budgeting practice, we assume that the firm under consideration evaluates investment opportunities with a single dominant objective: Maximize the market value of the firm's stock. This implies investments with positive net present value (NPV) defined as NPV = E[PV] - K, where E[PV] is the expected present value of future cash flows and K is a constant representing the capital investment needed to develop all designs. Other investment costs are ignored; for example, the cost of building the production facility plant is considered a sunk cost because we assume the plant has already been built.

The enterprise-wide decision problem is formulated as follows.

maximize {*expected net present value*}

with respect to {*engine and final drive ratios for the two vehicles, production capacity allocation for each vehicle*} (9)

subject to {*engineering performance constraints for both vehicles, product demand constraint, capacity constraint, CAFE constraint*}

Monthly profit π is being generated to the end of the life cycle of each product, Eq. (10). The present value of all future monthly cash flows minus the initial cost required for the investment constitutes the objective function of the decision model, Eq. (11).

$$\pi = (\text{Price} \times \text{Profit Margin} - \text{CAFE Cost}) \times (\text{Units sold}) \quad (10)$$

$$\text{NPV} = E\left[\int_0^T (\text{Future Cash Flows}) e^{-(\text{Cost of Capital})} dt\right] - \text{Investment Cost.} \quad (11)$$

Calculation of expected sales is a challenging task. We assume that future cash flows generated by a vehicle's future sales are only imperfectly predictable from current observation. The probability distribution is determined in the present, but the actual path remains uncertain. We consider product demand and the firm's market share as the two main sources of uncertainty. To describe future product demand we assume that the automotive product demand is a Brownian motion. The mean-reverting process is used to model market share uncertainty.

Sales are expressed as the product of the two sources

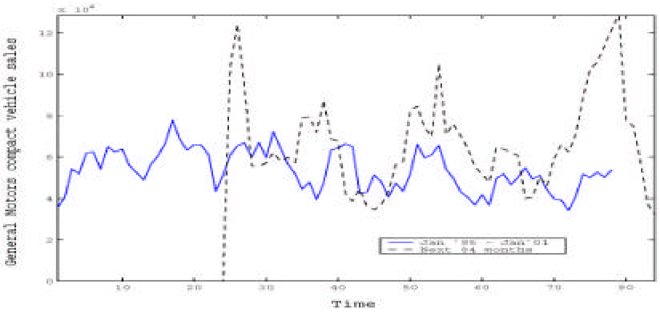

Figure 10: A random walk in the future

of uncertainty defined above, namely, product demand and market share. During the first 24 months of product development and production start-up time we have zero sales.

To estimate the expected present value of future cash flows we generate a big number of random walks; a sample of such a random walk is depicted in Figure 10. Next, we obtain the present values corresponding to the random walks by discounting all future payoffs ? ?across the probability space. The expected present value is computed as the average of these values. Finally, we calculate the NPV by subtracting the necessary capital investment. We assume that the profit margin represents the variable cost structure. The profit margin is set at 1% and 35% for the PC and SUV segments, respectively.

Engineering performance is represented in Eq. (9) by assuming that the price of the product can be expressed as a functional relationship of the vehicle attributes, in this case acceleration. The value curve method for attribute value assessment is used to assess the customer-perceived value of vehicle acceleration. This method is based upon the premise that the value of a performance attribute can be represented by continuous value curves if the performance is itself continuous [19]. Each performance attribute x is assumed to have three specification points (Figure 11): the ideal point where value for the attribute is at its highest level, the baseline point where value is nominal (that is 1), and the critical point where the product becomes valueless independent of the level of other attributes. A heuristic expression can be used to approximate the interpolating value between the critical and ideal point:

$$V(y) ? \left[\frac{(y_C ? y_I)^2 ? (y ? y_I)^2}{(y_C ? y_I)^2 ? (y_0 ? y_I)^2}\right]^{?}, \; y ? \log\left(\frac{1}{x}\right) \quad (12)$$

We will proceed assuming that the only vehicle attribute that influences the customer's purchasing decision is vehicle acceleration. We assume also, that the vehicle price is proportional to the $t_{0\text{-}60}$

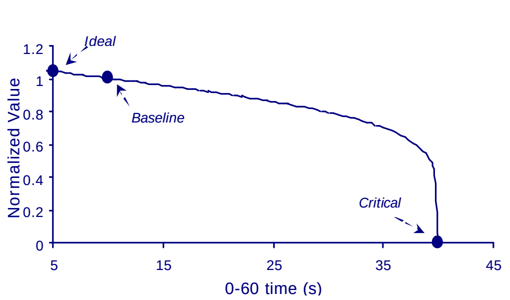

Figure 11: Parabolic approximation for acceleration performance [19]

value. This means that if a design achieves value of 1.1 the firm will price the vehicle 10% higher. We would like to mention here that the focus of the present work is the link between engineering and financial analysis. However, the development of customer valence models is critical. Further development and validation of the proposed design methodology will help define the desired nature of customer valence models and quantify further their role in the decision model.

Bounds on vehicle performance attributes define the constraint set for each segment. The constraints are expressed in terms of the design variables using computer-aided engineering simulation models. The particular model used here is the Advanced Vehicle Simulator (ADVISOR) program [20], which is a powertrain simulation tool. From the ADVISOR model library, the automatic transmission versions of the Chevrolet Cavalier LS Sedan and the rear-wheel drive Tahoe are selected as representative of the PC and SUV products considered here.

For each segment the engineering constraint set is fuel economy, $t_{0\text{-}60}$, $t_{0\text{-}80}$, $t_{40\text{-}60}$, 5-sec distance, maximal acceleration, maximal speed, and maximal grade at 55mph. The constraint bounds are set at values that are 20% beyond the current vehicle nominal performance values to allow for new design possibilities.

Constraints related to investment decisions are as follows. CAFE penalty should be non-positive:

$$? \; Cost_{CAFE} ? 0. \quad (13)$$

The production for each vehicle shall not exceed the total amount that the firm can expect to sell:

$$Production ? Expected\ Sales. \quad (14)$$

To estimate expected sales first we average across the probability space of demand and market share. The average across time yields expected sales. For this example we calculated expected sales for PC and SUV to be 59,000 and 57,300 vehicles per month, respectively.

The firm's fixed monthly production capacity of engines and transmissions must be allocated between the two segments so that

$$Production_{PC} ? Production_{SUV} ? Capacity \quad (15)$$

Problem (9) was solved for two different production capacities. The number of maximum expected sales for SU vehicles is 57,300. The divided rectangles (DIRECT) optimization algorithm was used [2], since it can locate global minima efficiently without derivative information when the number of variables is small. The best feasible solutions and associated net present values are shown in Table 3. Shaded areas emphasize significantly different values.

For a capacity of 100,000 we see that the computed solution matches our intuition. The portfolio aims at producing as many high profit SUVs as the market will bear (57,300). However, when the capacity of the firm equals the market limit, the solution is no longer intuitive. One may conceivably see a choice to produce only SUVs, making them meet the CAFE standard so as not to violate Eq. (13). Yet, forcing the SUVs to meet the CAFE standard reduces their price because they would suffer performance loss. We see that a compromise .between a smaller SUV engine (better fuel economy) and a small percentage of PC vehicle production yields a more profitable portfolio. However, we would not be able to tell exactly how much to compromise and how much smaller the SUV engine should be without including the design variables.

5.2 Analytical Target Setting

In ATC the desired design targets for the product's responses at the top level are assumed given, and the problem is to minimize deviations:

$$\min_{\mathbf{x}} \left\| \mathbf{R} - \mathbf{T} \right\| \qquad (16)$$

Table 3: *Solutions of the enterprise-wide decision problem for different capacities*

Capacity	100,000	57,300
Engine Size (PC)	73kW	75kW
Final Drive Ratio (PC)	3.5	3.4
Capacity Allocated (PC)	42%	17%
Engine Size (SUV)	200kW	162kW
Final Drive Ratio (SUV)	3.7	3.7
Capacity Allocated (SUV)	58%	83%
NPV(over 6 years period)	$15B	$13B

Setting the targets is a critical enterprise decision. The enterprise model developed in the previous section can be used to define the targets analytically:

$$\max_{\mathbf{T}} NPV \; ? \; V(\mathbf{T}), \qquad (17)$$

where the targets are variables. This is defined as the Analytical Target Setting (ATS) problem. Solving the ATS and ATC problems iteratively, we can ensure that the targets set are meaningful from the enterprise viewpoint.

CONCLUDING REMARKS

The versatility and usefulness of a mathematical optimization framework for addressing product design questions are largely limited by our imagination and ability to create the mathematical problem statements. This is a significant challenge but the accomplishments to date are quite rewarding. Although one needs to know one's mathematics to do a good job in optimization, one should not become obsessed with too much rigor. Even the best algorithms are only heuristic solvers for most problems of practical value.

Acceptance by corporate management is also increasing. Linking the purely engineering decisions with those that management worries about will only increase the optimization's appeal and the design engineers' credibility. But this linking must be done with knowledge of the models from other disciplines, such as economics and marketing, which is often hard to acquire at the requisite level of quantification. The added stochastic nature of these models also increases solution complexity.

The alternative, of course, is to make decisions in a less informed manner. Whether to optimize or not is itself an optimization decision: Decide on the tradeoff between the effort needed to optimize and the benefits expected from a successful design optimization study.

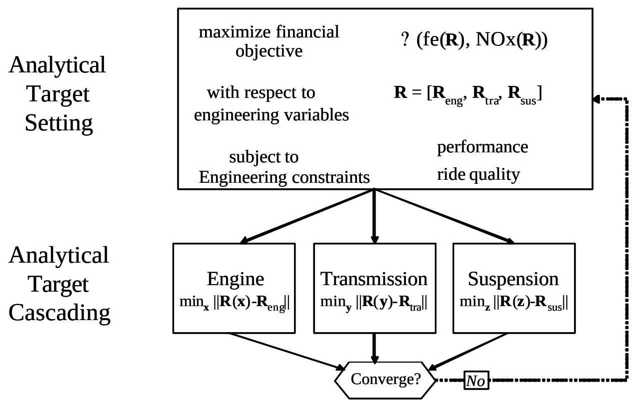

Figure 12: Combining Analytical Target Setting and Analytical Target Cascading.

Acknowledgements

This work has been conducted in collaboration with many colleagues at Michigan and elsewhere too many to enumerate. It has also been supported by several sponsors, most notably, the US Army, General Motors Corporation and Ford Motor Company. This support by colleagues and sponsors is gratefully acknowledged.

References

1. Papalambros, P.Y., and D.J. Wilde, *Principles of Optimal Design: Modeling and Computation*, Cambridge University Press, New York, 2000.
2. Whitehead, J.W., *Design and Performance of Derivative-Free Optimization Algorithms Used with Hybrid Electric Vehicle Simulations*, MS Thesis, Dept. of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA, 2001.
3. Zilinskas, A., ''One-step bayesian method of the search for extremum of an one-dimensional function,'' *Cybernetics*, 1:139–144, 1975
4. Jones, D.R., M. Schonlau, and W.J. Welch. ''Efficient global optimization of expensive black-box functions,'' *Journal of Global Optimization*, 13(4):455–492, 1998.
5. Sasena, M.J., Papalambros, P.Y., and Goovaerts, P., ''Global Optimization of Problems with Disconnected Feasible Regions via Surrogate Modeling,'' *Proc. 9th AIAA /ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2002-5573, Atlanta, Georgia, USA, September 4-6, 2002.
6. Sasena, M.J., *Advancing the Efficiency and Flexibility of Approximation-Based Optimization*, Ph.D. Thesis, Dept. of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA, 2002.
7. Michelena, N., Kim, H.M., and Papalambros, P., "A System Partitioning and Optimization Approach to Target Cascading," *Proc. 12th International Conference on Engineering Design*, Munich, Germany, Aug. 24-26, 1999.
8. Kim, H.M., Michelena, N.F., Papalambros, P.Y., and Jiang, T., ''Target Cascading in Optimal System Design,'' *Proc. 2000 ASME Design Automation Conference*, Paper No. DAC-14265. Baltimore, Maryland, 2000.
9. Papalambros, P.Y., "Analytical Target Cascading in Product Development,'' *Proc. 3rd ASMO UK / ISSMO Conference on Engineering Design Optimization*, Harrogate, North Yorkshire, England, July 9-10, 2001.
10. Kim, H.M., *Target Cascading in Optimal System Design*, Ph.D. Thesis, Dept. of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA, 2001.
11. Michelena, N., Louca, L., Kokkolaras, M., Lin, C.C., Jung, D., Filipi, Z., Assanis, D., Papalambros, P., Peng, H., Stein, J., and Feury, M., "Design of an Advanced Heavy Tactical Truck: A Target Cascading Case Study," *Proc. SAE International Truck and Bus Meeting and Exposition*, F2001-01-2793, Chicago, Illinois, USA, November 12-14, 2001.
12. Wagner, T.C., Papalambros, P.Y, "Selection of Families of Optimal Engine Designs Using Nonlinear Programming and Parametric Sensitivity Analysis," *SAE Paper 971600*, May, 1997.
13. Nelson, S.A., Parkinson, M.B., and Papalambros, P.Y., "Multicriteria Optimization in Product Platform Design," *Proc. 1999 ASME Design Automation Conference*, DAC-8676, Las Vegas, Nevada, USA, September 12-16, 1999. Also, *Transactions ASME, Journal of Mechanical Design*, Vol.123, No.2, pp.199-204, June, 2001
14. Fellini, R., Papalambros, P., and Weber, T., " Application of a Product Platform Design Process to Automotive Powertrains," *Proc. 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2000-4849, Long Beach, California, USA, September 6-8, 2000.
15. Fellini, R., Kokkolaras, M., Michelena, N., Papalambros, P., Saitou, K, Perez-Duarte, A., Fenyes, P., "A Sensitivity-Based Commonality Strategy for Family Products of Mild Variation, with Application to Automotive Body Structures," *Proc. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2002-5610, Atlanta, Georgia, USA, September 4-6, 2002.
16. Fellini, R., Kokkolaras, M., Papalambros, P., Perez-Duarte, A., "Platform Selection under Performance Loss Constraints in Optimal Design of Product Families," *Proc. 2002 ASME Design Automation Conference*, DAC-34099, Montreal, Quebec, Canada, September 29-October 2, 2002.
17. Fellini, R., Kim, H.M., Kokkolaras, M., Michelena, N., and Papalambros, P., "Target Cascading for Design of Product Families," *Proc. 4th Congress on Structural and Multidisciplinary Optimization*, Dalian, China, June 4-8, 2001.
18. Georgiopoulos, P., Fellini, R., Sasena, M., Papalambros, P., "Optimal Design Decisions in Product Portfolio Valuation," *Proc. 2002 ASME Design Automation Conference*, DAC-34097, Montreal, Quebec, Canada, September 29-October 2, 2002.
19. Cook, H.E., *Product Management: Value, quality, cost, price, profits, and optimization*, Chapman and Hall, London, 1997.
20. Brooker, A., Hendricks, T., Johnson, A, Kelly, K., Markel, T., O'Keefe, M., Sprik, S., and Wipke, K.,. ADVISOR 3.0 Documentation. National Renewable Energy Laboratory, 2000.