

Chapter 15

Cascading Platforms for Product Family Design

Jiju A. Ninan and Zahed Siddique

Abstract Product family design is a trade-off between distinctiveness of products in the family and commonality between them. Increasing the commonality of components can lead to loss of performance of product variants. Saving in cost comes at the expense of performance of products. Therefore selection of components to be standardized across the family and their configuration is a critical step in the design of product families. A common approach to the product family design is to treat it as a design optimization problem so that trade-off decisions between commonality and performance can be performed. In this chapter we present a scale-based multi-platform optimization approach. The approach uses systematic relaxation to increase leverage among multiple platforms and provide increase in performance for family members supported by the platform. The three stages involved in the approach are (1) single platform, (2) platform evaluation, and (3) platform relaxation. The Black and Decker universal motor family is used to demonstrate the approach.

15.1 Introduction

Product development enterprises normally offer a range of products varying from low cost-low performance to high cost-high performance products to serve different market segments. Traditionally, the product varieties were individually designed and manufactured to suit the requirements of the particular market segment.

J.A. Ninan
Schlumberger, Sugarland, TX, USA

Z. Siddique (✉)
School of Aerospace and Mechanical Engineering, University
of Oklahoma, Norman, OK 73019, USA
e-mail: zsiddique@ou.edu

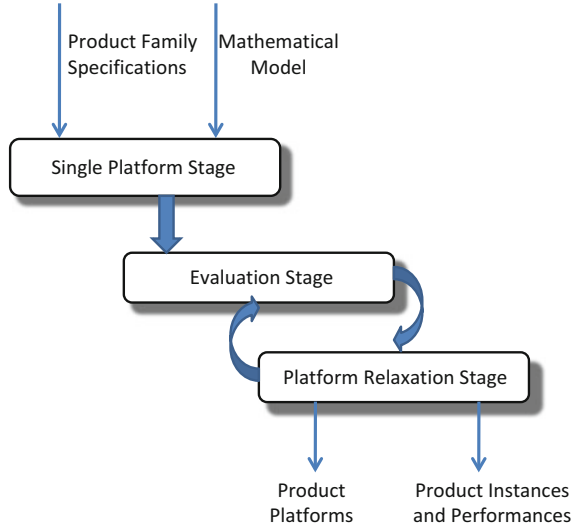
Each product, in a family, had different components, even though they served the same or similar functions. The products in a family lacked commonality among the different products, resulting in high cost in design, manufacturing, and inventory. These costs can be reduced or eliminated by sharing components and parts among the different family members. Companies are moving towards using platforms to support families, as a result of these advantages. Based on the product differentiating factors, product families can be classified as (1) modular product families—wherein product family members are instantiated by adding, substituting, and/or removing one or more functional modules from the product platform and (2) scalable product families—wherein scaling variables are used to “stretch” or “shrink” the product platform in one or more dimensions to obtain the different product variants.

The trade-off between performance and commonality is one of the main concerns in determining a product platform for a set of similar products. Designers need to balance the commonality of the products in the family with the individual distinctiveness of each product in the family. Normally a product family design process includes (1) designing the platform and (2) designing the individual product variants from the platform. Therefore, product family design should focus on the design of the entire family and platform, as well as the individual products. Several researchers have treated the design of product families as a design optimization problem (see Sect. 15.2). The advantage of this methodology is that designers can maintain a balance between commonality and cost. The platform and family members can simultaneously be optimized for performance, cost, and commonality during product design.

Based on the number of stages involved in the design process, product family optimization methods can be categorized as single-stage and multistage design optimization. Single-stage approaches seek to simultaneously optimize the product platform and corresponding products in the family of, while multistage approaches optimize the platform first and then instantiate the individual products from the platform. Single-stage optimization usually requires only one optimization run, but the size of the optimization problem increases tremendously as the number of parameters and number of products in the family increases. Multistage optimization breaks the larger problem into smaller subproblems. They require at least one optimization run for determination of the platform, and “ n ” optimization runs to leverage the “ n ” products in the family from the platform.

A product family can be supported using a single platform, which may cause poor performance of individual product family members. Hence, a single platform might not be sufficient to successfully leverage all of the products in the family (Dai and Scott 2007). In multi-platform, the products are leveraged from two or more platforms so that the loss of performance due to commonalization can be reduced. In multi-platform design, the challenges are to find (1) the minimum number of platforms that can serve the family of products with minimal loss of performance, (2) the platform from which each product is leveraged from, and (3) which parameters constitute the platform parameters for each platform. In a multi-platform approach, one of the drawback is that the platforms can be

Fig. 15.1 Platform relaxation method inputs and outputs



unrelated, increasing the cost in design, manufacturing, assembly, etc. Hence, a multi-platform approach that explicitly tries to ensure some level of commonality among platforms can be more efficient. The approach presented in this chapter combines the single and multi-platform approach to systematically relax a single platform into multi-platforms, which are related through a set of common parameters, to increase performance of the family. The systematic relaxation of parameters allows different platforms to maintain commonality among them.

The inputs to the relaxation formulation are the parametric description of the products, constraints related to performance of products, and underlying mathematical models relating the product parameters to constraints and objectives. The proposed approach consists of three stages (Fig. 15.1):

1. Single platform stage
2. Platform evaluation stage
3. Platform relaxation stage

The starting point for the proposed approach is a product family based on a single platform optimization. The products are designed assuming that a single platform is sufficient to scale all the products in the family. In the second stage, the resulting product family from the first stage is evaluated. The loss of performance due to commonality for each product or the family as a whole is evaluated. Family members, optimized individually for performance, are used as the benchmark for comparison. In the platform relaxation stage, the single platform parameters are systematically relaxed and explored to increase performance of selected family members. The objective of systematic platform parameter relaxation ensures commonality among the first and the succeeding platforms while generating improved products.

15.2 Literature Review

Several researchers have used optimization approaches to design family of products to arrive at a suitable a product platform and also product varieties. Optimization approaches are used to perform trade-offs between commonality (the underlying platform) and the performance of the product variants. Researchers have studied the effect of commonality on individual product performances such as cost, efficiency, strength, and reliability of the product variants. A comprehensive review and classification of product family optimization methods was presented by Simpson (2006). Belloni et al. (2008) compared heuristic optimization approaches for product line design problems. Some of the product platform and family optimization approaches are highlighted next.

Simpson (1998) developed the Product Platform Concept Exploration Method (PPCEM), which is a multistage method for design of scale-based product families. The formulation returns the optimized product platform and the product family instances. Variation-Based Methodology for Product Family Design (VBPDM) (Nayak et al. 2002) is an extension to PPCEM. In VBPDM the platform and scale variables are identified in the optimization formulation. Nelson et al. (2001) presented a multi-criteria optimization model to determine trade-off decisions in product family designs. The authors demonstrated how to generate the Pareto set in case of product family design, where each product family member has different objective functions. Product Family Penalty Function using Physical Programming (PFPF) developed by Messac et al. (2002) uses physical programming (Messac 1996) for product family design. The difference between PFPF and PPCEM is that in PPCEM, the scale variables of the product family needs to be known prior.

Dai and Scott (2007) presented a method for product family design using cluster analysis and sensitivity analysis. They presented a multi-platform design method, where design variables may be shared among variants using any possible combination of subsets. A drawback of the suggested method is that as the problem size increases, the complexity of the problem will also tremendously increase. Simpson and D'Souza (2004) presented a product family design approach using genetic algorithm to simultaneously design the family of products while considering varying levels of commonality within the product family.

Fellini et al. (2006) proposed an approach for obtaining desired level of commonality through problem size reduction in platform selection and then maximizing commonality among variants, while minimizing individual performance deviations. Khire et al. (2006) presented a method to specify the platform configuration and optimize the design of the platform and the individual variants by choosing design variable values while maintaining commonality defined in the platform configuration. Khajavirad and Michalek (2007) presented a decomposed gradient-based approach to jointly determine the optimal selection of components to be shared across product variants and the optimal values for design variables that define those components. Moon et al. (2011) presents a multi-objective particle swarm optimization (MOPSO) approach to select the best platform design strategy

from a set of Pareto-optimal solutions based on commonality and design variation within the product family. Gao et al. (2009) presented a three-step approach for finding the optimal values of platform and non-platform parameters in a modular scale-based platform.

15.3 Platform Relaxation for Multi-platform Design: Overall Approach

The platform relaxation method is a three-stage design process (Fig. 15.1). The first stage is to identify a common platform that can support the entire family of products (single platform stage). The performance of the family members is then evaluated by comparing them with the benchmarks using predetermined criteria. Benchmark products are optimized individually (without commonality) with the same specification as the corresponding family member. Family members that perform inadequately, compared to the benchmark, are segregated and separated out from the current platform. This stage is referred to as the evaluation stage. In the last stage, the parameters of the initial platform are relaxed systematically to arrive at a new platform to support the set of products separated out. Relaxing or cascading the platforms helps to attain commonality between different platforms to achieve higher cost savings. The resulting products, supported by the relaxed platform, are again evaluated and the platform is again cascaded if necessary. The design process is iterative and is continued until all members of the product family have acceptable performance.

The specifications of the product family members and the underlying mathematical model are provided by the designer. The mathematical model usually comprises of bounds and constraints on the design parameters and parametric relation among design parameters and responses. The platform relaxation method returns the parameter values of product family members, configuration of different platforms, platform from which each product is leveraged, performance of product family members, and their performance loss due to commonality. In scale-based product family architecture (Table 15.1), each product instance P_1, P_2, \dots, P_m of the family can be uniquely and completely described by the same set of product parameters x_1, x_2, \dots, x_n . These parameters describe the attributes of the physical components present in the products. If the value of any of the parameters is constant throughout the family (in case of single platform) or a subset of products (in case of multi-platform), the parameter is said to be a platform parameter. The product parameters related to the entire product family are defined as x_{ij} , which indicates the i th parameter for the j th product. The entire family of products is then a set of design parameters (x). The design task is to find the value of the parameters in x that results in a product family with maximum commonality and minimal loss in performance. Y is the set of platform commonality variables corresponding to the product parameters X . PP is the set of product platforms used to leverage the products.

Table 15.1 General steps in platform relaxation method

$PF = \{p_1, p_2, \dots, p_m\}$
 $Y = \{y_1, y_2, \dots, y_n\}$
 $PP = \{pp_k \mid pp_k \text{ is the set of product platforms from which the family is derived}\}$
 $PP = \{pp_1, pp_2, \dots, pp_f\}$
 $P_{ck} = \{pp_{ck} \mid pp_{ck} \text{ is the set of products considered for leveraging from the platform "k"}\}$
 $X_{pk} = \{x_{pk} \mid x_{pk} \text{ is the set of platform parameters for platform "k"}\}$
 C_{ik} is the set of platform parameter values in platform "k" (if $i \in x_{pk}$)
 $N_k = \text{Cardinality of } x_{pk}$

1. Single platform stage

1. Execute "individual optimization formulation"
2. $k = 1$
3. $P_{c1} = PF, P_{c1} = \emptyset$
4. Execute "platform-specified/non-platform-specified formulation"
5. $k = k + 1$

2. Evaluation stage

6. $\Delta_j = f_{\text{benchmark},j}(w_1z_1 + w_2z_2 + \dots + w_nz_n) - f_{\text{family},j}(w_1z_1 + w_2z_2 + \dots + w_nz_n) \forall j \in P_{ck-1}$
 Case 1: All Δ_j values $< \eta$; $P_{ck} = \{ \}$; Goto 9
 Case 2: Some Δ_j values $< \eta$ & other Δ_j values $> \eta$ then (a)/(b)
 (a) Include products with Δ_j values $> \eta$ in P_{ck+1} ; $k = k + 1$; Goto 7
 (b) Include all products in P_{ck} ; Goto 7
 Case 3: No Δ_j values $< \eta$; $P_{ck} = P_{ck-1}$; Goto 7

3. Cascading stage

7. Execute "Platform Relaxation Formulation"
8. Goto 6;
9. End

Initially the number of platforms required is unknown. X_{pk} is the set of variables for each platform. N_k represents the number of platform parameters in each platform. The steps associated with the platform relaxation method and associated optimization formulations are discussed in the following subsections.

15.3.1 Single Platform Stage

The starting point for the platform relaxation method is determining the entire product family using a single platform, where platform parameters have the same value for all the products in the family. There are two possible formulations that have been widely used to determine the single platform (1) platform specified and (2) non-platform specified. In platform-specified approach, the designer specifies the platform parameters. The aim of the optimization formulation is to arrive at an optimum $x (\dot{x})$ which enforces commonality of specified platform parameters throughout the family and also minimizes the loss of performance due to commonality. In the non-platform-specified formulation (Table 15.2), the aim is to explore different levels of commonality and perform trade-off between commonality and

Table 15.2 General formulation for non-platform specified optimization

Indices

j = Product family members, $j \in J, J = \{1,2,3, \dots, m\}$
 t = Product Constraints, $t \in T, T = \{1,2,3, \dots, s\}$
 l = System goals, $l \in L, L = \{1,2,3, \dots, p\}$

Variables

x_{ij} is the parameter ‘i’ in product ‘j’
 y_1, y_2, \dots, y_n are the commonality parameters corresponding to each parameter in I
 G_{1g} is the target goal of objective 1 for product j
 d^+_{1j} is the positive deviation of the first goal for j th product
 d^-_{1j} is the negative deviation of the first goal for j th product
 w_{1j} is the weights for the deviation variables $d_{1j}^{+/-}$ in the objective function
 w_i is the weights for the commonality parameters in the objective function

Objective

$$\sum_{l=1}^p \sum_{j=1}^m f(w_{lj}, d^-_{lj}, d^+_{lj}) - \sum_{i=1}^n w_i y_i$$

Subject to

$x_{ij}^{lower} \leq x_{ij} \leq x_{ij}^{upper}, \forall i \in I$ and $\forall j \in J$ Bounds on the design variable
 $y_i = \begin{cases} 1 & \text{when } x_i \text{ is a platform} \\ 0 & \text{when } x_i \text{ is not a platform} \end{cases}$
 $g_t(x) = 0, t = 1, 2, \dots, s$ Constraint relating to individual products
 $(x_{ij} - x_{ij+1})y_i = 0 \forall i \in I$ and $j \in J, j \neq m$ Commonality constraints
 $y_i^2 - y_i = 0 \forall i \in I$ Constraints for converting y_i to continuous variables
 $A_{lj}(x) + d^-_{lj} + d^+_{lj} = G_{lj} \forall l \in L$ and $j \in J$, Objectives transformed to system goals
 $d^-_{lj}, d^+_{lj} \geq 0, d^-_{lj} * d^+_{lj} = 0 \forall l \in L$ and $j \in J$, Non negativity of deviation variables

loss of performance of family members to arrive at a suitable product platform and leverage the product family members using the platform.

In both instances, platform commonality can be modeled mathematically for the family by

$$x_{ij} = x_{ij+1} j, j \neq m \text{ if } i \in x_p \tag{15.1}$$

where x_p is the set of platform variables. As mentioned earlier to represent sharing of parameters, a set of binary decision variable (0, 1) corresponding to each product parameter are utilized. These platform commonality decision parameters are represented by y_i .

$$y_i = \begin{cases} 1 & \text{when the parameter is a platform parameter} \\ 0 & \text{when the parameter is a scale parameter} \end{cases}$$

y_i parameters can be used to turn on/off the commonality of corresponding parameters. In the platform-specified formulation, y_i values of platform parameters are set to 1 to enforce commonality. In non-platform-specified formulation,

y_i helps to explore levels of commonality by activating or deactivating platform commonality for different parameters. The commonality is maximized through trade-offs between maximum number of platform parameters and performance loss for family members. In this chapter we use the non-platform-specified design formulation. The general form of the non-platform-specified formulation is shown in Table 15.2. $g(X)$ are the constraints that need to be satisfied by each product; l_i and u_i represent the lower and upper bounds of the product parameters x_i . The number of platform counter “ k ” is set to 1 for the single platform stage. In the single platform stage, P_{cI} includes all the products in the family ($P_{cI} = PF, P_{cI} = \emptyset$).

A goal programming model is adopted to consider multiple objectives for product family design. In goal programming the target values are identified for each objective, and the deviation of actual objective value from the targets is captured using deviation variables. Deviation variables d_{ij}^+ and d_{ij}^- are the positive and negative deviation of actual attainment $A_{ij}(x)$ from the target G_{ij} . Both d_{ij}^+ and d_{ij}^- are constrained to have only nonnegative value.

$$\begin{aligned}
 &\text{If } A_{ij}(x) \leq G_{ij}(\text{underachievement}) \text{ then } d_{ij}^- > 0 \text{ and } d_{ij}^+ = 0 \\
 &\text{If } A_{ij}(x) \geq G_{ij}(\text{overachievement}) \text{ then } d_{ij}^+ > 0 \text{ and } d_{ij}^- = 0 \\
 &\text{and if } A_{ij}(x) = G_{ij}(\text{exactly satisfied}) \text{ then } d_{ij}^- = 0 \text{ and } d_{ij}^+ = 0
 \end{aligned}
 \tag{15.2}$$

When values larger than target are undesirable, the positive deviations are minimized in the objective function and vice versa. To keep the actual values close to target, both negative and positive deviations are minimized. The term $\sum_{i=1}^n w_i y_i$ maximizes the number of platform parameters. Different terms in the objective function are weighted so that all of them are given equal priority while optimization is performed.

15.3.2 Evaluation Stage

The benchmarks for the evaluation stage are determined through optimizing the family members individually, subject to design, and performance requirements of corresponding product instance. The formulations (Table 15.3) are run independently, corresponding to each product in the family. The individual optimum corresponds to the best performance that can be achieved subject to requirements of the products.

In the evaluation stage products leveraged from the platform are compared against the benchmark products. Let z_1, z_2, \dots, z_p be the performance measures considered in the objective function; $z_{1j}, z_{2j}, \dots, z_{pj}$ be their value for product “ j ”; and $z_{1j}^*, z_{2j}^*, \dots, z_{pj}^*$ be the value of their corresponding benchmark.

Table 15.3 Individual optimization of product instances

Given:	Mathematical model Product constraints
Minimize	$f(w_1z_1, w_2z_2, \dots, w_nz_n)$
Subject to:	$g_r(X) \leq 0 \forall r \in R$ $l_i \leq x_i \leq u_i \forall i \in I$ (formulation repeated 'm' times for m products)

The performance of the products is $\Delta_j = \pm(N_1z_{1j} - N_1z_{1j}^*) \pm(N_2z_{2j} - N_2z_{2j}^*) \dots \pm(N_pz_{pj} - N_pz_{pj}^*)$. Here N_1, N_2, \dots, N_p are the factors used to normalize the performances for comparison. Depending on the nature of each desired performance measure, the following sign conventions are used for each of the factors in the function:

For positive valued targets: when performance higher than target is desired and the performance measure obtained for product “j” is higher than target, a negative sign is assigned, and when performance is lower than target, a positive sign is assigned.

For negative valued targets: when performance higher than target is desired and the performance measure obtained for product “j” is higher than target, a negative sign is assigned, and if the performance measure obtained for product “j” is lower than target, a negative sign is assigned.

When the performance measure is desired to be exactly equal to the target, a positive sign is assigned. Δ_j values are calculated for each product leveraged from the current platform. After Δ_j is calculated, the following cases represent the possible scenarios:

Case 1—All Δ_j values $< |\eta|$: If all the products have performance within the acceptable limits, $|\eta|$, then further iterations or platforms are not required, and the platform relaxation method is considered to be complete.

Case 2—Some Δ_j values $< |\eta|$: If some of the products satisfy the product performance criteria, while others do not, the designer has two possible options (a) include products with acceptable performance (Δ_j values $< |\eta|$) to be leveraged from the current platform, separate the nonconforming (Δ_j values $< |\eta|$) to be leveraged from relaxed platform. The platform count is now incremented by 1 to $k = k + 1$, and then the platform relaxation formulation is repeated with the nonconforming products. (b) Include both conforming and the nonconforming products and relaxed platform P_{ck-1} further. The choice between options (a) or (b) is dependent on the cost burden of developing another platform and the manufacturing processes involved.

Case 3—No Δ_j values $< |\eta|$; $P_{ck} = P_{ck-j}$: If none of the products are conforming, then the only option is to relax the platform further till conforming products are attained.

Table 15.4 General platform relaxation formulation

Minimize	$\sum_{l=1}^p \sum_{j=1}^m f(w_{lj}, d_{lj}^-, d_{lj}^+)$
Subject to	$x_{ij}^{lower} \leq x_{ij} \leq x_{ij}^{upper}, \forall i \in I \text{ and } \forall j \in p_{ck}$ Bounds on the design variable $y_i = \begin{cases} 1 & \text{when } x_i \text{ is a platform} \\ 0 & \text{when } x_i \text{ is not a platform} \end{cases}$ $g_t(x) = 0, t = 1, 2, \dots, s$ Individual products constraints $(x_{ij} - C_{ik-1})y_i = 0 \forall i \in x_{pk-1} \text{ and } \forall j \in p_{ck}$ Commonality constraints $\sum y_i^3 \leq N_{k-1} - 1, \forall i \in x_{pk-1}$ Relaxation constraints $\sum y_i \geq N_{k-1} - 1, \forall i \in x_{pk-1}$ $A_{lj}(x) + d_{lj}^- + d_{lj}^+ = G_{lj} \forall l \in L \text{ and } j \in J$, Objectives transformed to system goals $d_{lj}^-, d_{lj}^+ \geq 0, d_{lj}^- * d_{lj}^+ = 0 \forall l \in L \text{ and } j \in J$, Non negativity of deviation variables

The threshold value influences the number of products that will be retained in the current platform. Changing the threshold value will change the platform leveraging approach. It is assumed that the designer specifies a reasonable threshold value for loss of performance.

15.3.3 Relaxation Stage

In this stage only the nonconforming products, separated during evaluation stage, is considered. Let p_{ck} be the set of products being considered for leveraging from the platform “ k ” (Table 15.4). Let x_{pk} denote the platform parameters for the current platform pp_k . The idea is to arrive at a new platform pp_{k+1} , which consists of platform parameters x_{pk+1} formed by relaxing one of the platform parameters in x_{pk} to scalable parameter ($x_{pk+1} \subseteq x_{pk}$). The value of platform parameters in x_{pk+1} is held same as that of x_{pk} .

The platform relaxation formulation starts from the previous platform, with all the platform parameters from the previous platform (pp_{k-1}) held initially to the previous platform (C_{ik-1}) values by applying the following constraints:

$$(x_{ij} - C_{ik-1})y_i = 0, \forall i \in x_{pk-1} \text{ and } \forall j = p_{ck} \tag{15.3}$$

Here C_{ik-1} corresponds to the value of platform parameters in the previous platform. Value of y_i parameters is held to 1 for C_{ik-1} . The objective of the formulation is to improve the performance of the products by relaxing the previous platform. The relaxation formulation is used to select a platform parameter that when converted to a scale parameter minimizes the deviation of performance. To achieve this two constraints are introduced:

$$\sum y_i^3 \leq N_{k-1} - 1, \forall i \in x_{pk-1} \tag{15.4}$$

$$\sum y_i \geq N_{k-1} - 1, \forall i \in x_{pk-1} \quad (15.5)$$

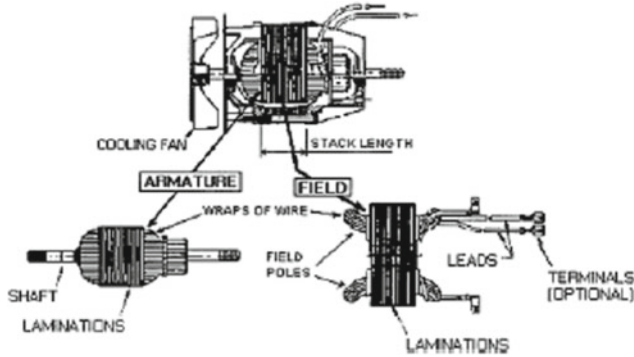
Here N_{k-1} is the number of platform parameters in the previous platform; the formulation selects one of the parameters that can be converted to scale parameters. To satisfy the above constraints, only $(N_{k-1} - 1)$ number of y_i parameters are equal to 1, and the remaining y_i parameters have a value of 0. This constraint also ensures that y_i accepts only discrete, 0 or 1 value and no value in between. The objective function in this case is minimization of deviation parameters.

15.4 Universal Electric Motor Case Study

Universal electric motors are capable of operating on alternating current (AC) and direct current (DC). They deliver more torque for a given current than any other types of AC motors (Chapman 1991). The high-performance characteristics of the universal motor, coupled with their flexibility, have led to a wide variety of applications, especially in household appliances, where they are found in, e.g., electric drills and saws, blenders, vacuum cleaners, and sewing machines (Veinott and Martin 1986).

As shown in Fig. 15.2, a universal electric motor is composed of an armature and a field, which are also referred to as the motor and stator, respectively. The armature consists of a metal shaft and slats (armature poles) around which wire is wrapped longitudinally as many as 1,000 times. The field consists of a hollow cylinder within which the armature rotates. The field also has wire wrapped longitudinally around interior metal slats (field poles) as many as 100 times. In order to reduce cost, size, and weight, the motor that satisfies its performance requirements with the least overall mass and highest efficiency is considered to be the most desirable. The design objective is to design a family of ten universal electric motors that satisfy a variety of torque and power requirements and is supported using platform(s). Different varieties scaled from the platform(s) will meet specific requirements.

The product parameters for the electric motors are (1) number of turns in the armature, (2) number of turns in the field, (3) area of the armature, (4) area of the field wire, (5) radius of the motor, (6) thickness of the stator, (7) current drawn by the motor, and (8) stack length. The current is varied in each motor by using electrical resistors. There is no manufacturing advantage by holding current as a platform. Moreover, varying the current can help to achieve different power requirements without having to vary other parameters that affect the manufacturing process. Torque requirements for individual electric motors are $T = \{0.05, 0.10, 0.125, 0.15, 0.30, 0.25, 0.30, 0.35, 0.40, \text{ and } 0.5\}$. The constraint on magnetizing intensity ensures that the magnetic flux intensity within each motor does not exceed the physical flux carrying capacity of the steel. The constraint on feasible geometry ensures that the thickness of the stator does not exceed the radius of the stator since



<i>Name</i>	<i>Requirement</i>
Torque	$T = \{0.05, 0.10, 0.125, 0.15, 0.30, 0.25, 0.30, 0.35, 0.40, 0.5\}$
Power	$= 300 \text{ W}$
Magnetizing Intensity, H	$< 5000 \text{ A.turns/m}$
Feasible geometry	Radius of motor $>$ thickness of stator
Efficiency of each motor	> 0.15
Mass of each motor	$< 2.0 \text{ Kg}$

Fig. 15.2 Requirements for the universal electric motor product family, adapted from Simpson et al. (2001)

the thickness is measured from the outside of the motor inward. The required output power is taken as 300 W, and the ten torques values range from 0.05 to 0.5.

There are two goals for each motor (1) efficiency and (2) mass, with targets of 79 % and 0.5 kg, respectively. A lower bound of 15 % on efficiency and an upper bound of 2.0 kg for mass are imposed for each product within the product family. The design requirements, range of possible values for product parameters, and the constraints related to the product family as introduced by Simpson et al. (2001) are shown in Table 15.4. In this case study, application of platform relaxation method for designing scale-based product families supported by multiple platforms is demonstrated. The platform relaxation method returns the configuration of the platform(s) from which each motor is leveraged, value of platform parameters, value of scale parameters for each motor, and performance of each motor.

The general steps of platform relaxation method, introduced in the previous section, are used for the design of electric motor product family. The three stages are explained in the following subsections. The general steps in the method are shown in Table 15.1. The product family *PF* consists of ten electric motors $\{P_1, P_2, \dots, P_{10}\}$ with torque requirements of $\{0.05, 0.10, 0.125, 0.15, 0.30, 0.25, 0.30, 0.35, 0.40, \text{ and } 0.5\}$ (Fig. 15.2). There are eight design parameters that describe

each product in the family; hence, there are eight platform commonality parameters in the set Y . These parameters are y_1, y_2, \dots, y_8 corresponding to the product parameters x_1, x_2, \dots, x_8 . The design objective is to find the optimum value of X for the product instances which results in minimum performance loss due to commonality, while maximizing commonality.

15.4.1 Single Platform Stage

In the single platform stage, all the products are considered for leveraging, hence $P_{c1} = PF = P_1, P_2, \dots, P_{10}$. The single platform optimization formulation is used to arrive at a platform that can be used to leverage all the products in the family. The single platform formulation for the universal motor family is shown in Table 15.5.

In this chapter, the universal electric motor case study is treated as a non-platform-specified case. In single platform optimization, a holistic view of the entire product family is adopted to determine a suitable single platform while simultaneously optimizing the platform and the product instances for maximum commonality and minimum loss of performance due to commonality. The objective function consists of minimizing the undesirable negative deviation of efficiency of each motor and positive deviation of mass for each motor and maximizing the number of platform parameters.

The platform commonality constraints are initially introduced as continuous variables ($0 \leq y_i \leq 1$). Integerizing constraints are then used to force the formulation to accept only values of 0 or 1 (binary) for the y_i parameters. This allows the formulation to evaluate the model for values in between while moving to optimum values. This is required for the formulation to be implemented in a gradient-based optimization method. The commonality constraints are used to ensure that platform variables take the same value and scale variables take different value for different products in the family.

The model consists of 128 design variables and 180 constraints. The formulation was implemented in VRAND[®] Visual DOC[®], a commercially available nonlinear optimization tool. Table 15.6 shows the results obtained from single platform optimization formulation. The formulation returned a platform consisting of parameters x_2, x_3, x_4, x_6, x_8 with values of 70, 0, 38, 0.34, 5.91, and 1.62, respectively. Now the number of platform counter is incremented by a value of 1, but first the evaluation of products is performed.

15.4.2 Platform Evaluation Stage

During the single platform stage, all product family members are individual optimized to serve as benchmarks for evaluation. The results for individually optimized universal motors, subject to the requirements and considering no commonality between them, are shown in Table 15.7.

Table 15.5 Single platform formulation applied to universal electric motor family*Variables* x_{ij} = Product parameters in K for each family member j y_j = Platform commonality variables $d_{Eff,j}^-$ = Negative deviation of Goal 1 (efficiency > 0.70) from the target for product j $d_{Eff,j}^+$ = Positive deviation of Goal 1 (Efficiency > 0.70) from the target for product j $d_{Mass,j}^-$ = Negative deviation of Goal 2 (efficiency > 0.50) from the target for product j $d_{Mass,j}^+$ = Positive deviation of Goal 2 (efficiency > 0.50) from the target for product j E_j = Efficiency of motor j M_j = Mass of motor j *Objective:* $z = \sum_{j=1}^{10} d_{Eff,j}^- + \sum_{j=1}^{10} d_{Mass,j}^+ - \sum y_i$ *Subject to:*

(1) Bounds on the design variable

$$100 \leq x_{1j} \leq 1,500 \text{ turns} \quad 1 \leq x_{2j} \leq 500 \text{ turns} \quad 0.01 \leq x_{3j} \leq 1 \text{ mm}^2 \quad 0.01 \leq x_{4j} \leq 1 \text{ mm}^2$$

$$1 \leq x_{5j} \leq 1 \text{ cm} \quad 0.5 \leq x_{6j} \leq 10 \text{ mm} \quad 1 \leq x_{7j} \leq 6 \text{ amps} \quad 0 \leq x_{8j} \leq 10 \text{ cm}$$

$$1 \leq y_j \leq 1 \quad \text{where } \forall j \in J \text{ and } \forall i \in I$$

(2) Magnetic intensity of each motor less than 5,000 A turns/m

$$\frac{\pi(2x_{2j}x_{7j})}{\pi(2x_{5j}+x_{6j})/2+(2(2x_{5j}-x_{6j}-0.007)+2*0.007)} \leq 5000$$

(3) Feasible geometry: $x_{6j} < x_{5j}$

(4) Mass of motor (M) < 2.0 kg

$$\pi d_{steel} x_{7j} (x_{5j}^2 - (x_{5j} - x_{6j})^2) + \pi d_{steel} x_{7j} (x_{5j}^2 - (x_{5j} - x_{6j} - l_{gap})^2) (2x_{7j} + 4(x_{5j} - x_{6j} - l_{gap})) x_{1j} x_{3j} + ((2x_{7j} + 4(x_{5j} - x_{6j})) (2x_{2j} x_{4j})) d_{copper} \leq 2.0 \text{ kg}$$

(5) Efficiency (E) > 0.15

$$\frac{1}{115x_{7j}} \left(113x_{7j} - \left(\frac{\mu_r x_{1j} (2x_{7j} + 4(x_{5j} - x_{6j} - l_{gap}))}{x_{3j}} + \frac{2\mu_r x_{2j} (2x_{7j} + 4(x_{5j} - x_{6j}))}{x_{3j}} \right) x_{7j}^2 \right) \geq 0.15$$

(6) Torque requirement for individual motors

$$\frac{(x_{1j} x_{2j} x_{7j}^2)}{\pi \left(\frac{0.000175}{x_{7j} \mu_0 (x_{5j} - x_{6j} - l_{gap})} + \frac{0.000175}{x_{7j} \mu_0 r} + \frac{\pi(2x_{5j} + x_{6j})}{4\mu_0 \mu_r x_{6j} x_{7j}} \right)} = t \in T$$

$$T = \{0.05, 0.10, 0.125, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.50\}$$

(7) Platform commonality constraints: $(x_{ij} - x_{i,j+1})y_i = 0 \quad \forall i \in I \text{ \& } \forall j \in J, j \neq m$ (8) Integerizing constraints: $y_i^2 - y_i = 0$ (9) Deviation of efficiency from target (70 %): $\frac{E_j}{0.7} + d_{Eff,j}^- + d_{Eff,j}^+ = 1.0 \quad \forall j \in J$ Deviation of actual mass from target mass (0.5 kg): $\frac{M_j}{0.5} + d_{Mass,j}^- + d_{Mass,j}^+ = 1.0 \quad \forall j \in J$ (10) Deviation variables: $d_{Eff,j}^-, d_{Eff,j}^+ \geq 0, d_{Mass,j}^-, d_{Mass,j}^+ \geq 0$

The formulation (Table 15.5) uses a goal programming model to address the multi-objective (target efficiency, target mass) nature of the problem. The deviation of the actual efficiency and mass of the motor is captured using deviation variables $d_1^{+/-}$ and $d_2^{+/-}$, respectively. In the objective function the undesirable negative deviation of efficiency and positive deviation of mass (d_1^- and d_2^+) are minimized.

Here x_1, x_2, \dots, x_8 correspond to the design parameters, which are number of turns in the armature, number of turns in the field, area of the armature, area of the

Table 15.6 Results from the single platform formulation

	Motor									
	1	2	3	4	5	6	7	8	9	10
X ₁	944.96	1,078.61	1,085.41	1,088.65	1,094.58	1,100.44	1,101.90	1,102.57	1,100.56	1,123.78
X ₂	70	70	70	70	70	70	70	70	70	70
X ₃	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
X ₄	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
X ₅	1.72	2.15	2.40	2.62	2.99	3.25	3.36	3.41	3.46	3.32
X ₆	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91
X ₇	3.04	3.28	3.45	3.60	3.97	4.33	4.71	5.08	5.37	6.00
X ₈	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
Efficiency	0.81	0.72	0.70	0.69	0.66	0.63	0.58	0.55	0.49	0.43
Mass	0.35	0.51	0.54	0.57	0.59	0.66	0.70	0.74	0.76	0.77

Table 15.7 Results from individual optimization

	Motor									
	1	2	3	4	5	6	7	8	9	10
X ₁	1,019	1,020	1,021	1,021	1,029	1,011	1,024	1,021	1,020	1,022
X ₂	57	65	69	75	66	57.4	61	54	58	54
X ₃	0.256	0.215	0.214	0.225	0.218	0.201	0.229	0.218	0.239	0.248
X ₄	0.272	0.258	0.255	0.251	0.217	0.201	0.232	0.238	0.234	0.243
X ₅	2.06	2.24	2.24	2.22	2.16	5.49	2.23	2.29	2.37	2.49
X ₆	5.94	5.72	5.71	5.69	5.56	4.84	5.43	5.55	5.56	5.52
X ₇	3.19	3.62	3.72	3.73	4.1	2.38	5.62	5.36	5.13	5.82
X ₈	1.2	1.47	1.65	1.84	2.32	2.3	2.5	2.8	3.12	3.1
Mag intensity	3,543	3,160	4,817	4,981	5,000	5,000	5,000	5,000	5,000	5,000
Efficiency	0.817	0.72	0.705	0.7	0.635	0.59	0.564	0.548	0.508	0.454
Mass	0.33	0.39	0.415	0.45	0.5	0.56	0.63	0.694	0.733	0.78

field wire, radius of the motor, thickness of the stator, current drawn by the motor, and stack length. The individual optima correspond to the best performance that can be achieved for each universal motor in the family. The performances of individually optimized motors serve as benchmarks to determine the loss of performance for the family members supported by the identified common platform.

The benchmark for efficiencies and mass obtained after individual optimization for the product instances are 81.7, 72, 70.5, 70, 63.5, 59.0, 56.4, 54.8, 50.8, and 45.4 % and 0.33, 0.39, 0.415, 0.45, 0.5, 0.56, 0.63, 0.694, 0.733, and 0.78 kg, respectively. The magnetizing intensity for all motors is within the allowable limit of 5,000 A turns/m. After establishing the benchmarks the number of platforms counter “k” is initiated.

The function used to evaluate the performance of products leveraged from the platform is

$$\Delta_j = \pm \left(N_j^1 \times \text{Efficiency}_j^* - N_j^1 \times \text{Efficiency}_j \right) \pm \left(N_j^2 \times \text{Mass}_j^* - N_j^2 \times \text{Mass}_j \right) \tag{15.6}$$

Here N_j^1 and N_j^2 are the corresponding scaling factors that can be used to scale the corresponding benchmark performances to 1. The sign conventions introduced in Sect. 15.3.2 are used to assign positive or negative signs to the value of Δ_j obtained from the equation. Efficiency_j^* , Mass_j^* are the normalized benchmark values for motors, and Efficiency_j , Mass_j are the efficiency and mass of the motors leveraged using the platform. Table 15.8 shows the evaluation of products leveraged from platform 1. The limiting Δ_j value was decided as 0.2. Motors 1, 5, 6, 7, 8, 9, and 10 show loss of performance within acceptable limits. The motors with performance loss due to commonality higher than 0.125 (motors 2, 3, and 4) were separated out to be leveraged from a second platform.

Table 15.8 Evaluation of products leveraged from platform

	Efficiency (normalized)				Weight (normalized)				Δ Total	Feasibility
	Weight	Bench	Family	Δ Eff	Weight	Bench	Family	Δ Mass		
Motor 1	1.2240	1.0000	0.9914	0.0086	3.0303	1.0000	1.0727	0.0727	0.0813	Y
Motor 2	1.3889	1.0000	1.0056	-0.0056	2.5641	1.0000	1.3051	0.3051	0.2996	N
Motor 3	1.4184	1.0000	0.9929	0.0071	2.4096	1.0000	1.3084	0.3084	0.3155	N
Motor 4	1.4286	1.0000	0.9814	0.0186	2.2222	1.0000	1.2622	0.2622	0.2808	N
Motor 5	1.5748	1.0000	1.0409	-0.0409	2.0000	1.0000	1.1840	0.1840	0.1431	Y
Motor 6	1.6949	1.0000	1.0610	-0.0610	1.7857	1.0000	1.1768	0.1768	0.1158	Y
Motor 7	1.7730	1.0000	1.0337	-0.0337	1.5873	1.0000	1.1111	0.1111	0.0774	Y
Motor 8	1.8248	1.0000	1.0018	-0.0018	1.4409	1.0000	1.0620	0.0620	0.0601	Y
Motor 9	1.9685	1.0000	0.9547	0.0453	1.3643	1.0000	1.0355	0.0355	0.0807	Y
Motor 10	2.2026	1.0000	0.9537	0.0463	1.2821	1.0000	0.9910	-0.0090	0.0373	Y

15.4.3 Platform Relaxation Stage

In this stage only the nonconforming products from the platform evaluation stage are considered. The general platform relaxation formulation presented previously is applied to motors 2, 3, and 4; the optimization formulation for the universal motor family is shown in Table 15.9. The objective function in this case consists of minimization of positive deviation in mass from the target and negative deviation of efficiency. The bounds on the design variables are same as single platform formulation.

All y_i parameters associated with the scale parameters in the previous platform are given a value of 0 to hold them as scale parameters. All the platform parameters are initiated as platform parameters and held to the value obtained from the previous platform (constraints 2 and 3). There were five platform parameters in platform 1. The relaxation formulation selects a platform parameter from these five platform parameters and converts it to scale parameters so that motors with acceptable performance can be identified.

The remaining four platform parameters will have the same value as platform 1. This is achieved by using constraint 4 (Table 15.9). The constraint can only be satisfied if four of the y_i parameters have a value of 1 and the remaining one 0. This constraint restricts the continuous y_i parameters to accept only binary values and also helps in selecting the best four platform parameters from the initial five platform parameters. All the remaining constraints are same as single platform formulation, except that they are only applied to the concerned motors 2, 3, and 4.

Table 15.10 shows the values of product parameters and product performances obtained from the platform relaxation formulation. Parameter X_2 was converted from a platform parameter to a scaling parameter.

Significant improvement can be seen in efficiency of the motors but at the expense of mass of the motors (Table 15.11). Efficiency of motors 2, 3, and 4 is higher than the benchmark motors. Since efficiency is a higher target (positive valued in this case), a negative sign is assigned, and since the mass of motors are higher than benchmark, which is undesirable, a positive sign is assigned to combine the values. The combined value, ΔTotal , for the three motors is 0.0581, 0.0941, and 0.1105, which is less than the allowed value of 0.2. Hence, further cascading is not necessary. Table 15.12 shows the combined parameter values and performance of motors derived from platforms 1 and 2.

The platform leveraging strategy for the universal electric motor family is shown in Fig. 15.3, which relates the platform from which each product family member is leveraged and the configuration of each platform in terms of platform parameters and scale parameters.

Table 15.9 Platform relaxation formulation applied to universal electric motor family

$\text{Minimize } \sum_{j=2,3,4} f(d_{Mass,j}^+) + \sum_{j=2,3,4} f(d_{Eff,j}^-)$	
<i>Subject to</i>	
(1) Bounds on the design variables in	$100 \leq x_{1j} \leq 1,500$ turns $1 \leq x_{5j} \leq 1$ cm $1 \leq x_{2j} \leq 500$ turns $0.5 \leq x_{6j} \leq 10$ mm $0.01 \leq x_{3j} \leq 1$ mm ² $1 \leq x_{7j} \leq 6$ amps $0.01 \leq x_{4j} \leq 1$ mm ² $0 \leq x_{8j} \leq 10$ cm
(2) Platform commonality decision variables	
$y_1, y_5, y_7 = 0, 0 \leq y_2, y_3, y_4, y_6, y_8 \leq 1$	
(3) Platform commonality constraints (Cascading)	$(x_{2j} - 70)y_2 = 0$ $(x_{6j} - 5.91)y_5 = 0$ $(x_{3j} - 0.28)y_3 = 0$ $(x_{8j} - 1.62)y_8 = 0$ $(x_{4j} - 0.34)y_4 = 0$
(4) Cascading constraints	
$\sum y_i^3 \geq 4$ and $\sum y_i \leq 4; i = 2, 3, 4, 6, 8$	
(5) Magnetic intensity of each motor less than 5,000 A turns/m	$\frac{\pi(2x_3 + x_6)/2 + (2(2x_3 - x_6) - 0.007) + 2(4.007)}{(2x_2x_7)} \leq 5000$
(6) Feasible geometry: $x_{6j} < x_{5j}; j = 2, 3, 4$	
(7) Mass of motor (M) < 2.0 kg	$\pi d_{steel} x_7 (x_{2j}^2 - (x_{5j} - x_{6j})^2) + \pi d_{steel} x_7 (x_{2j}^2 - (x_{5j} - x_{6j} - l_{gap})^2) (2x_7 + 4(x_{5j} - x_{6j} - l_{gap})) x_{1j} x_{3j} + ((2x_7 + 4(x_{5j} - x_{6j})) (2x_2 x_{4j})) d_{copper} \leq 2.0 \text{ kg}$
(8) Efficiency (E) > 0.15	$\frac{1}{113x_7} (113x_7 \left(\frac{\mu_r x_1 (2x_7 + 4(x_5 - x_6 - l_{gap}))}{x_{xy}} + \frac{2\mu_r x_2 (2x_7 + 4(x_5 - x_6))}{x_{xy}} \right) x_{7j}^2) \geq 0.15$
(9) Torque requirement for individual motors	$\frac{\pi \left(\frac{0.000175}{x_7 \mu_0 (x_5 - x_6 - l_{gap})} + \frac{0.000125}{x_7 \mu_0 \mu_r} + \frac{\pi (2x_3 + x_6)}{4\mu_0 \mu_r x_6 x_7} \right)}{(x_1 x_2 x_7^2)} = \{0.10, 0.125, 0.15, 0.20\}$
(10) Deviation of actual efficiency from target efficiency (70 %)	$\frac{E_j}{0.7} + d_{Eff,j}^- - d_{Eff,j}^+ = 1.0; j = 2, 3, 4$
(11) Deviation of actual mass from target mass (0.5 kg)	$\frac{M_j}{0.5} + d_{Mass,j}^- - d_{Mass,j}^+ = 1.0$

Table 15.10 Optimum design variables and performances obtained from the relaxation formulations

	Motor									
	1	2	3	4	5	6	7	8	9	10
X ₁	–	1,018.00	1,021.00	1,500.00	–	–	–	1,102.57	–	–
X ₂	–	78	86	69	–	–	–	70	–	–
X ₃	–	0.28	0.28	0.28	–	–	–	0.28	–	–
X ₄	–	0.34	0.34	0.34	–	–	–	0.34	–	–
X ₅	–	2.15	2.29	2.00	–	–	–	3.41	–	–
X ₆	–	5.91	5.91	5.91	–	–	–	5.91	–	–
X ₇	–	3.27	3.32	3.64	–	–	–	5.08	–	–
X ₈	–	1.62	1.62	1.62	–	–	–	1.62	–	–
Eff.	–	0.80	0.78	0.72	–	–	–	0.55	–	–
Mass	–	0.46	0.50	0.57	–	–	–	0.74	–	–

15.5 Conclusions

A single platform, in most cases, is insufficient to design a family of products while using platform approach. Single platform approach assumes that when a component or a product parameter is shared, it is shared across all products in the family. As the number of products in the family increases or the portfolio of different products varies considerably, a single platform approach may lead to product family members with inferior performance. In a multi-platform approach, the family members are leveraged from more than one platform so that products with minimal loss of performance can be achieved. Cost efficiency of single platform design may be higher compared to multi-platform design as an increase in number of platforms will lead to increase in cost of the derived product family. In a multi-platform design it is therefore necessary to design the family of products using optimum number of platforms. Also in case of multi-platform design, the combination of products that are leveraged from each platform and the configuration of each platform leading to a family of products with minimal loss of performance need to be determined.

The platform relaxation method is a multi-platform optimization method for the design of scalable product families. The inputs to the formulation are (1) the specification of the product family members, (2) the underlying mathematical model that related the product parameters to performances, and (3) the identification of platform parameters (optional). The platform relaxation method takes a holistic view of the entire product family design process. The mathematical model developed for single platform design is capable of representing both the product platform and the product variants. During the single platform stage of the design process, both the platform and the product variants are simultaneously optimized. Trade-off is performed between the number of platform parameters and the loss of performance due to commonality to arrive at the optimum platform and the optimum product instances. The platform relaxation method converts the binary

Table 15.11 Evaluation of products leveraged from platform 2

	Efficiency (normalized)			Weight (normalized)			Δ Total	Feasibility	
	Weight	Bench	Family	Δ Eff	Weight	Bench			Family
Motor 2	1.3889	1.0000	1.1111	-0.1111	2.5641	1.0000	1.1692	0.1692	Y
Motor 3	1.4184	1.0000	1.1064	-0.1064	2.4096	1.0000	1.2048	0.2048	Y
Motor 4	1.4286	1.0000	1.0229	-0.0229	2.2222	1.0000	1.1333	0.1333	Y

Table 15.12 Combined results for Platforms 1 and 2

	Motor									
	1	2	3	4	5	6	7	8	9	10
X ₁	944.96	1,018.00	1,021.00	1,500.00	1,094.58	1,100.44	1,101.90	1,102.57	1,100.56	1,123.78
X ₂	70	78	86	69	70	70	70	70	70	70
X ₃	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
X ₄	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
X ₅	1.72	2.15	2.29	2.00	2.99	3.25	3.36	3.41	3.46	3.32
X ₆	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91	5.91
X ₇	3.04	3.27	3.32	3.64	3.97	4.33	4.71	5.08	5.37	6.00
X ₈	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62	1.62
Eff.	0.81	0.80	0.78	0.72	0.66	0.63	0.58	0.55	0.49	0.43
Mass	0.35	0.46	0.50	0.51	0.59	0.66	0.70	0.74	0.76	0.77

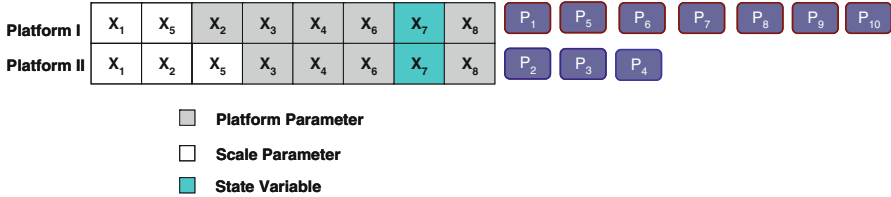


Fig. 15.3 Platform relaxation strategy for universal electric motor case study

platform commonality parameters to continuous parameters to enable the formulation to be implemented in a gradient-based optimization method. The model is constrained mathematically to accept only binary values in the end for the platform commonality parameters. The formulation developed is easy to implement in gradient-based optimization methods.

Relaxation of platform parameters is used to leverage the family when multiple platforms are required. During relaxation, one of the platform parameters is relaxed to a scale parameter so that products with lesser loss of performance can be identified. This reduces the number of platform parameters from the previous platform, which in turn can lead to products with better performances. In the platform relaxation method, the modeling approach is similar to that of single platform formulation. The platform, product instances, and platform commonality are modeled in the relaxation formulation. Both single platform and relaxation formulations initially convert the MINLP to a continuous problem and then constrain the solution to discrete spaces. In case of the relaxation formulation, constraints are simultaneously used to select the platform parameters and also to constraint the model to accept only binary values for commonality parameters.

In the platform relaxation method, the number of platforms required to support the platform is not modeled as part of the formulation. Instead the initial platform is relaxed until all the products with acceptable loss of performance are leveraged. The number of platforms required to support the family depends on the threshold value of the acceptable loss of performance, due to commonality and the path chosen by the designer after the evaluation of products. Since the platform relaxation method uses the single platform and cascades it to generate the subsequent platforms, the approach maintains a relation between the product platforms which can lead to commonality within the different platforms. An evaluation function is used to determine the loss of performance of the product family members due to commonality. If the loss of performance due to commonality for any of the products in the family is greater than a user specified value, a multi-platform approach is used.

Platform relaxation method does not require the identification of platform parameters by the designer; however, it gives the designer the flexibility of being able to specify the platform. The outputs from the method are (1) the different product platforms and the products that are supported by the platforms and (2) the product family instances and their performances. Other secondary information like

loss of performance due to commonality in comparison to benchmarks and the best possible performance of products can be obtained from the formulation. As evident from earlier discussions, the method is comprised of different stages. The method is only applicable towards scalable product families, wherein each product instance in the family can be completely described by the same set of product parameters. Hence, the method will fall under the category of multistage, non-platform-specified, scale-based product family design method.

References

- Belloni A, Freund R, Selove M, Simester D (2008) Optimizing product line designs: efficient methods and comparisons. *Manag Sci* 54(9):1544–1552
- Chapman SJ (1991) *Electric machinery fundamentals*. McGraw-Hill, New York, NY
- Dai Z, Scott M (2007) Product platform design through sensitivity analysis and cluster analysis. *J Intell Manuf* 18(1):97–113
- Fellini R, Kokkolaras M, Papalambros PY (2006) Quantitative platform selection in optimal design of product families, with application to automotive engine design. *J Eng Des* 17(5):429–446
- Gao F, Xiao G, Simpson T (2009) Module-scale-based product platform planning. *Res Eng Des* 20(2):129–141
- Khajavirad A, Michalek JJ (2007) An extension of the commonality index for product family optimization. In: *Proceedings of the ASME IDETC/CIE conference 2007, DETC2007-35605*
- Khire R, Messac A, Simpson TW (2006) Optimal design of product families using selection-integrated optimization (SIO) methodology. In: *11th AIAA/ISSMO symposium on multidisciplinary analysis and optimization*, 6–8 Sept 2006, Portsmouth, VA, AIAA-2006-6924
- Messac A (1996) Physical programming: effective optimization for computational design. *AIAA J* 34(1):149–158
- Messac A, Martinez MP, Simpson TW (2002) A penalty function for product family design using physical programming. *ASME J Mech Des* 124(2):164–172
- Moon SK, Park K, Simpson TW (2011) Platform strategy for product family design using particle swarm optimization. In: *Proceedings of the ASME 2011 international design engineering technical conferences and computers and information in engineering conference IDETC/CIE*, Washington, DC, USA, 28–31 Aug 2011. DETC2011-48060
- Nayak RU, Chen W, Simpson TW (2002) A variation-based method for product family design. *Eng Optim* 34(1):65–81
- Nelson S, Parkinson M, Papalambros P (2001) Multi criteria optimization in product family design. *J Mech Des* 123:199–204
- Simpson TW (1998) A concept exploration method for product family design. Ph.D. Dissertation, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA
- Simpson T (2006) Methods for optimizing product platforms and product families. In: Simpson T, Siddique Z, Jiao JR (eds) *Product platform and product family design*. Springer, New York, NY, pp 133–156
- Simpson TW, D'Souza B (2004) Assessing variable levels of platform commonality within a product family using a multi-objective genetic algorithm. *Concur Eng: Res Appl* 12(2):119–130
- Simpson TW, Maier JRA, Mistree F (2001) Product platform design: method and application. *Res Eng Des* 13(1):2–22
- Veinott CG, Martin JE (1986) *Fractional and subfractional horsepower electric motors*. McGraw-Hill, New York, NY