

DESIGNING EVOLVING FAMILIES OF PRODUCTS USING THE UTILITY-BASED COMPROMISE DECISION SUPPORT PROBLEM

Carolyn Conner Seepersad
Mechanical Engineering Department
The University of Texas at Austin
Austin, TX 78712-0292

Farrokh Mistree * and Janet K. Allen
Systems Realization Laboratory
The George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0405

ABSTRACT

By designing families of products based on product platforms, many companies are providing product variety efficiently and effectively and facilitating the development of an overall product development strategy. An important factor in product development is the evolution of a family of products in response to changing technologies, market demands, competitive pressures, and other factors. In this paper, we present a method for designing product platforms for an *evolving* product family that meets changing performance requirements with an expanding portfolio of products. The method is based on the utility-based compromise Decision Support Problem—a multi-objective decision support model with an objective function derived from utility theory. With this method, a designer can balance multiple, non-deterministic objectives in product platform design and explicitly consider the evolution of platform-based families of products in response to shifting performance targets. We apply this method to an example study of product platform design for an expanding family of absorption chillers.

* Corresponding Author. Email: farrokh.mistree@me.gatech.edu. Phone/fax: 404-894-8412/9342

KEYWORDS:

Product platforms, product families, compromise Decision Support Problem, utility theory, utility-based compromise Decision Support Problem

NOMENCLATURE

A	Type of absorber tube
A_i	System goal
$COST$	Cost goal
CT_i	Expected cycle time for i^{th} product
d^+, d^-	Deviation variables in a compromise DSP
ΔP	Pressure drop of the chilled water in the evaporator
DSP	Decision Support Problem
E	Type of evaporator tube
$E[]$	Expected value
G_i	Goal target value
k	Scaling constant in a utility function
L	Tube Length
T_{SOA}	Temperature of LiBr Soln. Leaving Absorber
T_{WOE}	Temperature of Chilled Water Leaving Evap.
u	Single attribute utility function
U	Multi-attribute utility function
X_{SOA}	Concentration of LiBr/Water Soln. Leaving Abs.
Z	Deviation function

1. FRAME OF REFERENCE

The paradox facing modern enterprises is the increasing demand for customization combined with an apparently conflicting need to preserve economies of scale in design and manufacturing. Many companies and designers are addressing this paradox by strategically creating families of products that *provide* variety in the marketplace and simultaneously *reduce* variety in the enterprise [1,2]. Product families are groups of products that share similar form, function and technology base and can be easily adapted to satisfy a variety of customer requirements or target specific market niches. This is often accomplished by designing a standard *product platform*, or common core of the family—including form, components, modules, interfaces, and technology

base—from which derivative products can be generated through modification, augmentation, and renewal.

[INSERT FIGURE 1.]

The design of product platforms and derivative families of products involves seeking a compromise among multiple, conflicting criteria. Increasing commonality within a *family* of products often yields economic and efficiency advantages, but at the same time, *individual* products often deviate from desired performance specifications or targets. Clearly, there are enterprise- or family-level benefits from designing platform-based families of products such as lower inventory, product development, and manufacturing costs and shortened manufacturing and product development cycle times [1,3]. From a customer’s perspective, however, platform-based families of products are expected to satisfy a diverse range of requirements, such as different tiers of price, performance, and functionality across the major market niches in the market segmentation grid developed by Meyer and Lehnerd [1,4] and illustrated in Figure 1. Unfortunately, as noted by Nelson and coauthors [5], the performance of platform-based products typically degrades relative to products designed individually or independently for each market niche because platform-based products are forced to share common platform modules, components, and/or parameter values with other products in the family.

In light of these conflicting requirements, several authors have recognized the importance of multiple objectives in product family design and modeled central portions of the associated design activities as multiobjective optimization problems or multiobjective decisions. A variety of approaches have been proposed such as physical programming [6], multicriteria optimization with Pareto set generation [5], fuzzy analysis in conjunction with the analytic hierarchy process (AHP) [7], customer value functions captured with conjoint analysis [8], and a multiobjective

formulation based on weighted sums [9]. We have used the compromise Decision Support Problem (DSP)—a multiobjective decision model that is a hybrid formulation based on mathematical programming and goal programming [10]—for designing product platforms that are scaled into families of products (e.g. [11-13]).

Deterministic multiobjective decision-making approaches may not be appropriate for product family design decisions because they do not accommodate the uncertainty surrounding such forward-looking decisions. Uncertainty could be associated with changes in technologies, system operating environments, or resources associated with fabricating and operating a system [14,15]. Also, uncertainty could be associated with variations in the market such as customer tastes, needs, or requirements for a family of products. As a result, performance requirements and/or consumer demand for derivative products may change, creating a need for robust product platforms that serve as common building blocks for not just a single product or an initial family of products but a continuous stream of derivative products that target old and new market niches (c.f., [1,16]). According to Meyer and Lehnerd [1,4], an important factor in the success of any product platform is its potential for successfully generating a continuous stream of new products over its lifetime.

[INSERT FIGURE 2.]

As shown in Figure 2, product platforms are designed typically for an initial portfolio of products, represented by the first generation of products in the figure. However, the portfolio of products associated with a product platform is rarely static. For example, the product portfolio may be required to expand in response to shifting targets for performance objectives (such as output power or mass). In Figure 2, a potential expansion is indicated with partially shaded blocks that represent new products in the second generation product portfolio. The success of a

product platform(s) is determined largely by whether it can be leveraged efficiently and effectively to support these expansions and changes. Unfortunately, the changes that initiate the *evolution* of a platform-based family of products, via addition of new derivative products or modification of existing ones, are difficult to predict when a product platform is initially designed. Usually, there are many potential derivative products, each of which has only a *probability* of being introduced into the market. The risk is that a product platform may not be designed to support new derivative products effectively if those products are not considered during the design process. Therefore, product platform design involves determining product platform specifications that yield preferred *expected* outcomes over time, and one of the central challenges in product family design is multiobjective decision-making *in the context of risk and uncertainty*.

Several approaches have been suggested for making product platform design decisions with multiple, non-deterministic objectives. The approaches include expected value and real options [15], net present value [17], and a fuzzy ranking methodology combined with the analytic hierarchy process [18]. In this paper, we present the utility-based compromise Decision Support Problem (DSP)—a quantitative, decision-based approach for designing product platforms that can be leveraged efficiently and effectively to support expanding and evolving product family portfolios [19]. By merging utility theory and the compromise DSP to create the utility-based compromise DSP, we leverage: (a) the strength of utility theory as a mathematically rigorous approach for rational, preference-consistent, multi-objective decision-making in the context of risk and uncertainty and (b) the strength of the compromise DSP as a mathematical construct for supporting decisions that involve *both* compromise among a set of conflicting goals *and* satisfaction of a set of constraints and bounds derived from informed engineering judgment.

In the following section, the utility-based compromise DSP is introduced and described in the context of a product family design process. In Section 3, application of the utility-based compromise DSP is illustrated by designing a platform-based family of products that evolves in response to changing performance targets.

2. THE UTILITY-BASED COMPROMISE DECISION SUPPORT PROBLEM FOR NON-DETERMINISTIC, MULTIOBJECTIVE PRODUCT PLATFORM DESIGN

Our method for designing product platforms and derivative products is illustrated in Figure 3. It includes the following phases: (1) identification of requirements for the family of products, (2) determination of product platform architecture and product platform design strategy, and (3) formulation and solution of a utility-based compromise DSP, supported by (4) search and (5) analysis activities. The outcome is a set of design specifications for the product platform(s) and the family of products based on the platform(s). First, in Section 2.1, we provide a detailed description of the utility-based compromise DSP, our primary contribution in this paper to the product platform/family design process. In Section 2.2, we discuss the role of the utility-based compromise DSP in the broader context of a product platform/family design process.

[INSERT FIGURE 3.]

2.1 The Utility-Based Compromise DSP

The utility-based compromise DSP is based on the compromise DSP with an objective function formulated using multi-attribute utility theory. The compromise DSP is a mathematical construct through which the conflicting goals in product family design are modeled. It is a multi-objective decision model that is a hybrid formulation based on mathematical programming and goal programming [10]. It is used to determine the values of design variables that satisfy a set of constraints while achieving a set of conflicting goals as well as possible. The system descriptors, namely, system and deviation variables, system constraints, system goals, bounds

and the deviation function are described in detail in [10] and are not discussed further here. In the compromise DSP, multiple goals can be combined in the objective or deviation function with Archimedean weightings, preemptively (lexicographically) [10], or more recently, with linear physical programming [20].

[INSERT FIGURE 4.]

In the *utility-based* compromise DSP, illustrated in mathematical form in Figure 4 [19], the system goals and objective function are formulated using utility theory. A single attribute utility function is assessed for each goal according to the procedure provided by Keeney and Raiffa [21]. Accordingly, each goal is formulated for the utility-based compromise DSP as follows:

$$E[u_i(A_i(x))] + d_i^- - d_i^+ = G_i \quad (1)$$

by derivation from the original compromise DSP goal formulation:

$$A_i(x) + d_i^- - d_i^+ = G_i \quad (2)$$

where $E[]$ is the expectation operator, u is a single attribute utility function, and d_i , A_i , and G_i are deviation variables, goal values, and goal target values, respectively, for goal i as described in [10]. The major modification is replacement of the goal achievement values, $A_i(x)$, in Equation (2) with the expected utility of those values, $E[u_i(A_i(x))]$ in Equation (1). The target value, G_i , in Equation (1) corresponds to an ideal utility value, which generally assumes a value of one. The deviation variables, d_i^- , d_i^+ , in Equation (1) measure the extent to which the expected utility of the goal underachieves or overachieves its target value. If uncertainty is present in the problem formulation, the deviation variables are deterministic in the utility-based compromise DSP goal formulation of Equation (1), whereas the deviation variables would be probabilistic in the standard compromise DSP goal formulation in Equation (2), and satisfaction of constraints on them would no longer be guaranteed [22].

To accommodate multiple goals in the utility-based compromise DSP, we formulate the deviation or objective function that is minimized during the design process. A multi-attribute utility function is assessed to combine the single attribute utility functions. If certain assumptions about the independence of a decision-maker's preferences are appropriate, the multi-attribute utility function may assume a simplified additive form [21], as follows:

$$U(\mathbf{X}) = \sum_{k=1}^m k_i u_i(A_i(\mathbf{X})) \quad (3)$$

where scaling constants, k , are obtained with preference assessment procedures described by Keeney and Raiffa [21]. The deviation or objective function in the utility-based compromise DSP is formulated to maximize expected overall utility, which is equivalent to minimizing deviation from the ideal target value for expected overall utility (i.e., 1). According to convention in goal programming and the compromise DSP, the latter formulation is chosen for the utility-based compromise DSP:

$$Z = 1 - E[U(\mathbf{X})] \quad (4)$$

By assigning a goal target value of *one* in Equation (1), solving Equation (1) for expected utility, and substituting the result into Equations (3) and (4), we derive the additive deviation function:

$$Z = \sum_{i=1}^m k_i (d_i^- + d_i^+) \quad (5)$$

More complex formulations such as multiplicative multi-attribute utility functions and corresponding deviation functions can also be accommodated, and details are provided in [19].

The utility-based compromise DSP is a theoretical and pragmatic extension of both utility theory and the compromise DSP, combining the strengths of both constructs to support multiobjective decision-making in the context of risk and uncertainty. Utility theory has been shown to be a mathematically rigorous, domain independent approach for multiobjective

decision-making [21,23]. A decision-maker's preferences are assessed explicitly and modeled as utility functions that are valid for conditions of risk and uncertainty as well as tradeoffs among multiple attributes. As long as a decision-maker's preferences obey a set of axioms, it can be proven mathematically that his/her preferred alternative—and therefore the rational choice—is the one with the highest expected utility [23].

Utility theory has been applied to a few engineering design problems (e.g., [24-28]), but some fundamental challenges have been encountered, as mentioned by Thurston [29] and Simon [30]. For example, a decision-maker's preferences must satisfy a set of axioms (e.g., the von Neumann-Morgenstern axioms [23] or the Savage axioms [31]). The utility assessment process can be time-consuming and tedious, and the decision-maker must be consistent in his/her preferences and know what he/she wants—no 'fuzzy' preferences are permitted. Furthermore, in strict applications of utility theory, constraints and bounds on the problem are not permitted [23], and designers are required to investigate and evaluate every alternative except those prohibited by nature. Whereas all of these challenges may be restrictive for an engineering designer, the latter challenge is critical. For a designer with limited computational and physical resources, it is not practical to evaluate the properties or consequences of an infinite or unbounded range of alternatives for a complex problem.

The challenge of limited resources is alleviated by using utility theory within the decision support construct of the compromise DSP. By specifying constraints and bounds on the design variables in the compromise DSP, a designer can bound the design or alternative space, thereby explicitly limiting the range of alternatives or design variable values to be considered. While generating and evaluating *all* possible alternatives is an intractable task for all but the simplest designs, a designer who uses the compromise DSP can create a bounded, artificial design space

in which utility theory may be applied to find the rational, most preferred alternative (or set of design variable values) *with respect to the bounded space*, thereby facilitating the search for *satisficing*¹ solutions.

Within the utility-based compromise DSP, the compromise DSP supports the use of engineering judgment for formulating and bounding complex decisions involving multiple, conflicting goals in the utility-based compromise DSP whereas utility theory provides a mathematically rigorous approach for representing designer preferences and identifying preferred alternatives in the context of risk or uncertainty. The resulting utility-based compromise DSP provides support for forward-looking product platform design decisions that involve multiple goals, tradeoffs, and uncertainty associated with changing market demand, competitive pressures, technological innovations, and other factors, as discussed in Section 1.² In the next section, we explore the role of the utility-based compromise DSP in a typical product platform/family design process.

2.2 An Overview of a Product Platform/Family Design Process

As illustrated in Figure 3, a product platform/family design process begins with **identification of requirements** for the family of products. Requirements, including goals and constraints, are established as precisely and quantitatively as possible for a comprehensive set of criteria representing diverse perspectives such as manufacturing, engineering, management, and marketing. Since a *family* of products is designed, many of the requirements are specified as ranges or sets (e.g., power output ranging from 100 to 500 kW or matching a set of targets, $\zeta = \{100, 200, 400, 500\}$ kW). Furthermore, since product families evolve, it is important to

¹ Satisficing decision methods facilitate the search for good or satisfactory solutions to complex problems instead of illusive optimal ones. Designers use satisficing methods not because they prefer satisfactory solutions to optimal ones but because resource limitations typically leave no alternative. Solutions obtained with satisficing methods may not be optimal with respect to the 'real' world without the constraints and bounds imposed by a designer. If a designer has used good judgment in constraining the design problem, however, the solutions should be satisfactory (as well as rational and preference-consistent with respect to the bounded design space, when utility theory is used).

² A utility-based selection Decision Support Problem is also available [32].

define requirements for both initial product offerings *and* to anticipate potential expansions or changes in the product portfolios along with the likelihood of such expansions. Several techniques have been proposed for formulating and visualizing requirements for families of products (e.g., market segmentation grids [4], Variety Voice of the Customer graphs [33], Product Variety Tradeoff Charts [34], Product Family Maps [35]).

After the requirements for a product family have been identified, the next step in Figure 3 is to **determine the product platform/family design strategy**. Generally, design strategies are used to determine the components, interfaces, processes, technologies, and/or design variables to be standardized in the product platform and those to be modified independently to realize each member of a family of products. Several design strategies are suggested in the literature. Uzumeri and Sanderson [36] emphasize *standardization* for enhancing product flexibility, as do McDermott and Stock [37], Collier [38], and Kota and Sethuraman [39]. Chen and coauthors [40] suggest designing flexible products that can be readily adapted in response to large changes in customer requirements by changing a small number of components or modules. Rosen [41], Siddique and Rosen [42], Dahmus and coauthors [43], Fujita [44], and Gonzalez-Zugasti and Otto [45] advocate *modular* product architecture as a means of achieving product variety. Rothwell and Gardiner [16] advocate *robust designs* as a means of improving system flexibility. Simpson and coauthors [11,12,34,46,47] build upon this work with a method that facilitates the synthesis and exploration of a common product platform concept that can be *scaled* into a family of products as demonstrated in the example in Section 3. With this method, the product platform consists of a set of design variables and parameters with standardized values for the product family while one or more design variables are identified as *scale factors* that can be used to scale the product platform to meet a range of performance requirements. Other authors provide

qualitative guides and frameworks for a product family design strategy (e.g., [7,48-50]). By adopting a product platform/family design strategy and applying it to a specific family of products, a designer implicitly identifies options for product platform components, interfaces, parameters, and/or variables (i.e., *platform design specifications*) that may be standardized for the derivative products associated with the product platform. A designer also identifies the components, interfaces, parameters, and/or variables (i.e., *product design specifications*) that may be modified from product to product to achieve product variety.

The next major activity in Figure 3 involves embodying the product platforms and derivative products via multiobjective decision-making, search, and analysis. In our method, we **formulate and solve a utility-based compromise DSP**, as described in Section 2.1, to facilitate the search for platform and product designs that offer desirable compromises or tradeoffs among competing objectives in the context of risk or uncertainty. Formulating a product platform design decision is a crucial initial step, but *solving* it can be challenging, as well. Several authors have focused on optimization techniques for searching a product platform design space (e.g., [44,51-53]). Search techniques are usually coupled with analyses or simulations to evaluate the technical and economic characteristics of candidate designs. The outcome of the design process is a set of specifications for the common product platform(s) and the specifications that are unique for each product. Using the utility-based compromise DSP, product platform specifications are designed to yield preferred expected performance for an evolving or expanding product family portfolio; thus, expected performance estimates for the family of products are an additional outcome of the design process.

In the following section, the product platform/family design process, including the utility-based compromise DSP, is used to design product platforms and derivative products for an evolving family of industrial absorption chillers.

3. PRODUCT PLATFORM DESIGN FOR AN EVOLVING FAMILY OF INDUSTRIAL ABSORPTION CHILLERS

Absorption chillers are refrigeration systems used for cooling large spaces, industrial processes, and similar applications. An absorption chiller includes four basic components: (1) the evaporator, (2) the absorber, (3) the generator, and (4) the condenser, as shown in Figure 5a. Usually, the absorber and evaporator are combined in a single shell (as illustrated in Figure 5b) as part of a thermal compressor that fulfills the role of a mechanical vapor compressor in a mechanical chiller. Our focus is on the preliminary design of a family of absorber-evaporator modules.

[INSERT FIGURE 5.]

Suppose a manufacturer intends to offer a family of absorption chillers with refrigeration capacities ranging from 200 to 1300 tons. Depending on market demand, the manufacturer may offer subsets of the following capacities: 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, and 1300 tons. In conventional design processes, the absorber-evaporator module is designed uniquely for each capacity to minimize the cost of material while providing the specified cooling capacity. This creates a large variety of components, specifically different combinations of tube lengths and types³ for both the absorber and the evaporator for each capacity. If the manufacturer could reduce the variety of tubes and create one or more product

³ A tube type is associated with a standard material, thickness, and number of fins per inch.

platforms based on the types and lengths of evaporator and absorber tubes, physical inventory could be reduced along with purchasing, material handling, storage, and production costs.

In previous work, we have shown that there are considerable benefits from designing the chillers around a common product platform in which the sizes and types of absorber and evaporator tubes are standardized [11]. However, we assumed that the set of manufactured cooling capacities (and therefore the portfolio of products in the product family) and the demand for each capacity were static and predetermined. Here, using the method described in Section 2, we design multiple product platforms for an *evolving* product portfolio. Product platforms are designed to yield satisfactory performance for an *initial* portfolio of products and satisfactory *expected* performance for several potential changes in the product portfolio over time. In this section, we discuss the application of the product platform/family design process illustrated in Figure 3 for the design of the evolving family of absorption chillers.

3.1 Step 1: Identify Requirements for the Family of Products

The absorption chiller manufacturer intends to span the market from 200 to 1300 ton cooling capacities as shown in Figure 6. The objective is to design one or more product platforms to serve as a foundation for the family of products. As indicated by the dashed white lines in the product platform block in Figure 6, it may be desirable to create multiple product platforms for the family. During the design process, the number of platforms is investigated along with the specifications for the platforms. As shown in Figure 6, suppose that market research indicates that current market demand is substantial for 200, 600, 1000, and 1300 ton capacities; therefore, the four products are included in the first generation of products to be leveraged from the product platform(s). Also, suppose that market research indicates that three different product portfolio expansions are possible over time and that the probability of each expansion is approximately

25%.⁴ The possible expansions are labeled A, B, and C in Figure 6 and represent additions of 300-500 ton products, 700-900 ton products, and 1200-1300 ton products, respectively, to the initial product portfolio. Yearly demand is estimated at 80 products per year, and it is assumed to be distributed evenly among the products. The objective is to design one or more product platforms that support the first generation of products *and* yield satisfactory expected performance if one or more of the potential expansions are implemented in the second generation of platform-based products.

[INSERT FIGURE 6.]

3.2 Step 2: Determine the Product Platform/Family Design Strategy

Our product platform design strategy is to specify each of the different platforms by determining a length and selection of tubes for the absorber-evaporator module that can be used as common standard components for developing the products based upon the platform. Each individual product is realized by *scaling* the number of standardized tubes in the absorber-evaporator module. Accordingly, we design combinations of evaporator and absorber tubes that would be suitable for each platform and compare the results. For this case study, we investigate a range of product platform options that include 1, 2, 3, or 4 platforms for the initial portfolio of products.

3.3 Steps 3-5: Formulate and Solve a Utility-Based Compromise DSP

Before a utility-based compromise DSP can be formulated for absorption chiller design, analysis techniques must be established to evaluate product and manufacturing process performance.

⁴ The portfolio expansions and associated probabilities are simplified for this example. Selecting additional or alternative product portfolio expansions (including expansions with different numbers of products) and associated probabilities would affect the numerical results of this example but would not affect the *approach*.

Product Analysis

The design of the absorber/evaporator module is carried out for the *standard* operating conditions described in [13]. The commercial alternatives for selection of tubes for the evaporator and the absorber are shown in Tables 1 and 2, respectively.

The classification of parameters that define the synthesis of the *individual* products is shown in Tables 3 and 4. The design variables are tube length, evaporator tube type, and absorber tube type. The goals include expected cycle or production time for each type of product (i.e., for each tonnage) and the average cost per unit produced. Constraints on absorber-evaporator module design are listed in Table 4.

[INSERT TABLE 1.]

[INSERT TABLE 2.]

[INSERT TABLE 3.]

[INSERT TABLE 4.]

Production System Analysis

The models for estimating cost and cycle time are described in detail in [11] and are not repeated here. Briefly, cycle times per product are estimated using response surface models of the effective mean processing time at a series of assembly stations. The response surface models are assembled using queuing networks [54], modified for multiproduct networks [55]. Expected cycle time is determined for each type of product (i.e., each tonnage). Average cost per unit is the sum of average labor costs, material costs, and inventory costs per unit. Inventory costs are estimated using a basic (Q,r) model [56]. Models are implemented as described in [11]; the only

augmentation is a set of material discounts applied to the cost model to account for large bulk orders.⁵

Formulating the Utility-Based Compromise DSP

[INSERT FIGURE 7.]

The utility-based compromise DSP for product platform design is presented in Figure 7. As shown, there are three design variables: tube length, L , type of evaporator tube, E , and type of absorber tube, A , *for each platform*. The compromise DSP can be exercised to design multiple product platforms simultaneously by including a set of design variables for each platform. To customize the compromise DSP for a platform design scenario, the designer indicates the products in production, the number of platforms, and the assignment of products to platforms (i.e., the extent of each platform). Product platform design constraints include the four constraints outlined in Table 4 as well as constraints on acceptable goal values, insuring that goal values have positive utilities. Depending on the number of products included in the product platforms, there are up to thirteen goals for product platform design: maximize the expected value of average cost per product (COST) and maximize the expected value of expected cycle time for the subset of the twelve products in production at a specific time (CT200, CT300, ..., CT1300).

Goals may correspond to either system (i.e., family) level attributes or individual product attributes for the family of products. Individual product attributes are those for which acceptable *system* or overall family performance is not adequate; each individual product must possess acceptable levels of these attributes to meet requirements. In this example, COST is a system level attribute, assuming that production costs can be aggregated across products and platforms.

⁵ If only one absorber (evaporator) platform (i.e., length and type of tube) is utilized, the material cost for absorber (evaporator) tubes is discounted 3%. If two platforms are utilized, the discount is 1%. These discounts are *representative* of bulk discounts; actual material discounts vary with the type and quantity of material.

On the other hand, cycle time (CT) is an individual product attribute since customers are concerned with lead time for *each* individual absorption chiller order. Average cycle time across the family is not a useful assessment of product family performance if some products experience unacceptably long cycle times; thus, cycle time goals are considered separately for each product.

[INSERT TABLE 5.]

Inclusion of utility theory impacts the compromise DSP formulation and solution in several ways. The *Given* information includes assessed single attribute utility functions for each goal as well as constants for combining the single attribute utility functions into a multi-attribute utility function. Specifications for each of the single attribute utility functions, assessed according to the procedure suggested in [21], are included in Table 5. Utility functions for cycle time are provided for each product. They are valid regardless of the number and extent of product platforms. The utility function for cost is based on average cost per unit of chillers produced. Cost is averaged across products (and platforms). In this case study, an additive form of the multi-attribute utility function is utilized. This choice is justified by verifying the independence conditions suggested in [21]. Constants for combining the single attribute utility functions into an additive multi-attribute utility function are listed in Table 6 for two types of product platform scenarios. The columns indicate scaling constants for designing any four or seven representative products. Finally, as discussed in Section 2, the goals are formulated in terms of expected utility. The deviation function in this example is based on an additive version of a multi-attribute utility function.

[INSERT TABLE 6.]

Solving the Utility-Based Compromise DSP for Absorption Chiller Product Platform Design

The utility-based compromise DSP for product platform design is solved using a Simulated Annealing algorithm in the commercial software OptdesX[®] [57] for the specified demand distributions in Figure 6. Simulated annealing algorithms are based on the physical phenomenon of annealing in solids and are particularly useful for discrete or mixed continuous/discrete problems (see, eg., [58,59]). The parameters used in the solution process are 500 optimization cycles, an initial probability of accepting a worse design of 0.5 and a final probability of accepting a worse design of 1.0e-06. Two alternative starting points for optimization are indicated in Table 7. The starting points yield identical or nearly identical design variable values for all trials. The results for the upper starting point are reported here. In addition, empirical results have been validated with convergence plots and monitoring of active and inactive constraints, but those details are omitted here for brevity.

[INSERT TABLE 7.]

A single platform is designed by constraining the length, absorber tube type, and evaporator tube type to be equivalent for each of the products based on the product platform. Two platforms are designed by permitting the length and type of tubes for products in the first platform to vary independently of the length and type of tubes in the second platform (i.e., two sets of design variables are specified in the compromise DSP), and the two platforms are designed simultaneously. Four platforms are designed similarly. The *number* of evaporator tubes and the *number* of absorber tubes in each chiller are dependent variables that are permitted to vary (up to 500) to accommodate the performance range (in tons) covered by the platform—thereby achieving vertical product platform leveraging (see Figure 1). The solution of the compromise

DSP is a separate set of design variables (E, A, and L) for each product platform as well as average cost per chiller and expected cycle time for each product with positive market demand.

The utility-based compromise DSP in Figure 7 is customized for each product platform and demand scenario to be investigated (e.g., 1 platform for the first generation of products in Figure 6; 4 platforms for the second generation of products). It is solved *once* for each scenario; if multiple product platforms are designed, they are designed simultaneously. Individual product specifications (i.e., the number of tubes needed for the required tonnage, given a product platform design) are obtained simultaneously, as well. This information is needed to calculate the cycle time for each product.

3.4 Results and Discussion

Product platforms are designed for the absorption chillers in a series of three experiments.

In Experiment 1, product platforms are designed for the initial product portfolio in Figure 6.

When the utility-based compromise DSP is formulated and solved, four product platforms are permitted—one for each of the products in the first generation product portfolio. It is possible for each of the products to have different tube types and sizes, resulting in four separate product platforms, but it is also possible for one or more of the products to have identical or nearly identical tube types and sizes, resulting in one, two, or three product platforms. To confirm the resulting specifications for multiple product platforms, the results are compared with platform designs that have been restricted to one or two platforms. Also, the results are compared with a benchmark family of individually designed products to verify that it is preferable to design a platform-based family of products rather than independent products.

In Experiment 2, product platforms are designed for the first generation of products combined individually with each of the potential second generation expansions of the product portfolio in Figure 6. This experiment confirms that different product platform specifications are

desirable when the portfolio of products changes. This result leads to the challenge of selecting a set of platform specifications that yield both satisfactory performance for the initial product platform and preferred *expected* performance for the potential product portfolio expansions.

In Experiment 3, product platforms are designed for non-deterministic evolution of the product portfolio. Specifically, product platforms are designed to yield preferred *expected* performance for the three possible second generation expansions of the initial product portfolio. Expected utility is utilized within the compromise DSP to design the product platforms. The expected performance of the resulting platform design is compared to the expected performance that could be obtained by utilizing one of the platform designs from the second set of experiments.

Results of the three experiments are reported in Tables 8, 9, and 10. All costs and cycle times are *averages* per chiller. Costs and cycle times in this example reflect trends for the products but have been disguised to protect the manufacturer. Before interpreting the results, it should be noted that costs and cycle times can be compared between alternative product platform designs for a fixed set of products but *not* between different sets of products. Goal values such as average cost estimates are expected to be higher for Scenario C in Figure 6, for example, than for Scenario A because the products are larger and more expensive to produce, on average.

[INSERT TABLE 8.]

[INSERT TABLE 9.]

[INSERT TABLE 10.]

It is important to note that the *numerical results* presented here may be sensitive to the assumptions detailed in Step 1 (e.g., potential product portfolio expansions and associated probabilities and alternative product platform extents). However, our emphasis is on illustrating

the *approach* rather than the results, *per se*, and for this purpose, the example is independent of the details of the assumptions in Step 1.

Results of Experiment 1: Designing Product Platforms for the Initial Product Portfolio

One, two, and four product platforms are designed for the initial product portfolio of Figure 6, and the results are reported in the first three columns of Tables 8 and 9. As can be observed from the data, it is preferable to have multiple platforms for this product portfolio. Average cost for *each* of the 80 total products, for example, is reduced from \$177,763 to \$170,582 for single platform versus four platform families, and cycle times remain relatively unchanged. The cost decrease corresponds to lower material costs associated with customizing absorber and evaporator tube types for *each* product in the four platform design strategy.

The results in Table 8 have been compared to a benchmark family of individually designed products. If the four platform design is utilized to design all twelve potential products (200 to 1300 tons), average cost *per unit* is approximately \$1000 less than the average cost for a benchmark family of individually designed products. Cycle times for a benchmark family are worse than any of the cycle times indicated in Table 8. Thus, it is preferable to design (multiple) product platforms for this family of products.

Results of Experiment 2: Designing Product Platforms for Possible Portfolio Expansions

Four platforms are designed for each of the product portfolio expansions in Figure 6, and the results are reported in the middle columns of Tables 8 and 9. The purpose is to demonstrate that different product platform designs are appropriate for different product portfolios. As shown in Table 9, the product platform for each of the expanded product portfolios is different, and these product platform designs are different from the four platforms designed for the initial product portfolio in Experiment 1. The platform design specifications are different because they are

tailored for the different portfolios of chiller capacities manufactured for each scenario. As an aside, it is important to note that the dramatic cost difference between Expansions A and C is due to the presence of smaller products in Expansion A. Recall that cost is average cost *per chiller* in the product portfolio.

Results of Experiment 3: Designing Product Platforms for Non-deterministic Evolution of the Product Portfolio

Finally, a single set of product platforms is designed to accommodate the possible product portfolio expansions illustrated in the second generation in Figure 6. The resulting product platform design is included in Table 9 and the expected goal values are included in the last column of Table 8. To obtain the reported design, a 25% probability is assigned to each of the possible expansions—A, B, and C in Figure 6—and to the continuation of the initial product portfolio without expansion. The objective is to design a product platform that is satisfactory for the initial product portfolio and yields preferred *expected* performance with respect to potential portfolio expansions. This is a challenge because different product platforms are appropriate for different product portfolios, as demonstrated previously. Expected values are considered in the goal and deviation function formulations of the compromise DSP. For example, if a particular set of design variable values yields costs of $Cost_A$, $Cost_B$, $Cost_C$, and $Cost_{INIT}$ for expansions A, B, and C and the initial product portfolio, respectively, the expected utility for the cost goal would be:

$$E[u(COST)] = 0.25 * u(COST_A) + 0.25 * u(COST_B) + 0.25 * u(COST_C) + 0.25 * u(COST_{INIT}) \quad (4)$$

The performance of the robust product platform design is at least as good and in some cases significantly better in terms of expected costs and cycle times than the scenario-specific product platform designs of the second experiment. If the platform designs in Tables 8 and 9 for

expansions A, B, or C of Experiment 2 were utilized for the evolving product portfolio rather than the design of Experiment 3, expected costs and cycle times would rise substantially. This is illustrated in Table 10. In Table 10, the results (in terms of expected cost and expected average cycle time per unit) are reported for using the designs of Experiment 2 (expansions A, B, and C) for the evolving product portfolio of Figure 6. It is illustrative to compare these values to the results from Experiment 3, in which the potential product portfolio expansions are considered explicitly. For example, if the product platform design for expansion A in Experiment 2 were utilized, expected average costs per chiller would be approximately \$12,000 higher. If 80 products are produced per year, the design from Experiment 3 offers the manufacturer expected savings of \$960,000 per year over the design for expansion A in Experiment 2. Since the design for Expansion B in Experiment 2 is identical to the Experiment 3 design, it offers equivalent expected performance, but the design for Expansion C in Experiment 2 provides slightly worse expected performance and lower expected utility. By explicitly considering potential expansions of the product portfolio during the product platform design process, we obtain product platform designs that generally offer better expected performance than scenario-specific product platform designs.

4. CLOSURE

We have introduced the utility-based compromise DSP for formulating and supporting the non-deterministic, multiobjective decisions that are centrally important in a product family and product platform design process. The utility-based compromise DSP is a *decision support* construct that facilitates bounding and modeling decisions that involve seeking balance or compromise among a set of conflicting and non-deterministic goals, such as costs, performance, and manufacturing considerations for product platform design. Within the utility-based compromise DSP, utility theory permits mathematically rigorous modeling of designer

preferences so that decisions can be guided by expected utility in the context of risk or uncertainty associated with potential outcomes.

We have shown that the utility-based compromise DSP is effective for designing static or evolving families of products. With this approach, a designer can develop multiple product platforms in the context of *uncertainty* with respect to evolution of a product portfolio over time. It is possible for a designer to consider *explicitly* both initial and expansive product platform leveraging strategies as shown in Figure 2. As illustrated with the example in Section 3, the benefits of explicitly considering and modeling potential product portfolio expansions can be substantial, and product platform designs may change significantly, as a result. Although it is not demonstrated in this paper, the utility-based compromise DSP can be used to accommodate additional sources of variability in product platform design, such as unexpected changes in technology, operating environments, or resources associated with fabrication of a family of products. Together, these capabilities facilitate the synthesis of an overall product development strategy for a design/manufacturing enterprise in which multiple product platforms are planned, designed, and embodied with substantially fewer resources for an evolving group of related products.

ACKNOWLEDGMENTS

During her graduate work at the Georgia Institute of Technology, Carolyn Conner Seepersad was supported by fellowships from the National Science Foundation and the Fannie and John Hertz Foundation. We gratefully acknowledge Gabriel Hernandez for development of the absorption chiller example and NSF DMI-0085136 for financial support. The cost of computer time was underwritten by the Systems Realization Laboratory of the Georgia Institute of Technology.

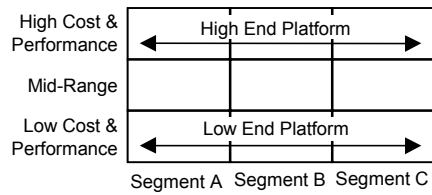
REFERENCES

1. Meyer, M. H. and A. P. Lehnerd, 1997, *The Power of Product Platforms: Building Value and Cost Leadership*, Free Press, NY.
2. Goldhar, J. and M. Jelinek, 1983, "Plan for Economies of Scope," *Harvard Business Review*, Vol. 61, No. 6, pp. 141-148.
3. Robertson, D. and K. Ulrich, 1998, "Planning for Product Platforms," *Sloan Management Review*, Vol. 39, No. 4, pp. 19-31.
4. Meyer, M. H., 1997, "Revitalize Your Product Lines through Continuous Platform Renewal," *Research Technology Management*, Vol. 40, No. 2, pp. 17-28.
5. Nelson, S. A., M. B. Parkinson and P. Y. Papalambros, 2001, "Multicriteria Optimization in Product Platform Design," *ASME Journal of Mechanical Design*, Vol. 123, No. 2, pp. 199-204.
6. Messac, A., M. P. Martinez and T. W. Simpson, 2002, "Effective Product Family Design Using Physical Programming," *Engineering Optimization*, Vol. 34, No. 3, pp. 245-261.
7. Tseng, M. M. and J. Jiao, 1999, "Methodology of Developing Product Family Architecture for Mass Customization," *Journal of Intelligent Manufacturing*, Vol. 10, No. 1, pp. 3-20.
8. Du, X. and M. M. Tseng, 1999, "Characterizing Customer Value for Product Customization," *ASME Design for Manufacturing Conference*, Las Vegas, NV. Paper Number: DETC99/DFM-8916.
9. Gonzalez-Zugasti, J. P., K. N. Otto and J. D. Baker, 2000, "A Method for Architecting Product Platforms," *Research in Engineering Design*, Vol. 12, No. 2, pp. 61-72.
10. Mistree, F., O. F. Hughes and B. A. Bras, 1993, "The Compromise Decision Support Problem and the Adaptive Linear Programming Algorithm," *Structural Optimization: Status and Promise* (M. P. Kamat, Ed.), AIAA, Washington, D.C., pp. 247-286.
11. Hernandez, G., T. W. Simpson, J. K. Allen, E. Bascaran, L. F. Avila and F. Salinas, 2001, "Robust Design of Product Families with Production Modeling and Evaluation," *Journal of Mechanical Design*, Vol. 123, No. 2, pp. 183-190.
12. Simpson, T. W., J. R. A. Maier and F. Mistree, 2001, "Product Platform Design: Method and Application," *Research in Engineering Design*, Vol. 13, No. 1, pp. 2-22.
13. Seepersad, C. C., G. Hernandez and J. K. Allen, 2000, "A Quantitative Approach to Determining Product Platform Extent," *ASME Advances in Design Automation Conference*, Baltimore, MD. Paper Number: DETC2000/DAC-14288.
14. Simpson, T. W., U. Lautenschlager and F. Mistree, 1998, "Towards Mass Customization in the Age of Information: The Case for Open Engineering Systems," *The Information Revolution: Present and Future* (W. Read and A. Porter, Eds.), Ablex Publications, Greenwich, CT, pp. 49-71.
15. Gonzalez-Zugasti, J. P., K. N. Otto and J. D. Baker, 2001, "Assessing Value in Platformed Product Family Design," *Research in Engineering Design*, Vol. 13, No. 1, pp. 30-41.
16. Rothwell, R. and P. Gardiner, 1990, "Robustness and Product Design Families," *Design Management: A Handbook of Issues and Methods* (M. Oakley, Ed.), Basil Blackwell Inc., Cambridge, MA, pp. 279-292.
17. Georgiopoulos, P., R. Fellini, M. Sasena and P. Y. Papalambros, 2002, "Optimal Design Decisions in Product Portfolio Valuation," *ASME Advances in Design Automation Conference*, Montreal, Canada. Paper Number: DETC2002/DAC-34097.

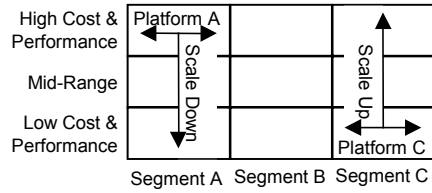
18. Tseng, M. and J. Jiao, 1998, "Design for Mass Customization by Developing Product Family Architecture," *ASME Design for Manufacturing Conference*, Atlanta, GA. Paper Number: DETC98/DFM-5717.
19. Seepersad, C. C., 2001, "A Utility-Based Compromise Decision Support Problem with Applications in Product Platform Design," *M.S. Thesis*, G.W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA.
20. Hernandez, G., J. K. Allen and F. Mistree, 2001, "The Compromise Decision Support Problem: Modeling the Deviation Function as in Physical Programming," *Engineering Optimization*, Vol. 33, No. 4, pp. 445-471.
21. Keeney, R. L. and H. Raiffa, 1976, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley and Sons, New York.
22. Marston, M., J. K. Allen and F. Mistree, 2000, "The Decision Support Problem Technique: Integrating Descriptive and Normative Approaches in Decision Based Design," *Journal of Engineering Valuation and Cost Analysis*, Vol. 3, No. 2, pp. 107-129.
23. von Neumann, J. and O. Morgenstern, 1947, *The Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ.
24. Thurston, D. L., 1991, "A Formal Method for Subjective Design Evaluation with Multiple Attributes," *Research in Engineering Design*, Vol. 3, No. 2, pp. 105-122.
25. Thurston, D. L., J.V. Carnahan, T. Liu, 1994, "Optimization of Design Utility," *ASME Journal of Mechanical Design*, Vol. 116, No. 3, pp. 801-808.
26. Hazelrigg, G. A., 1998, "A Framework for Decision-Based Design," *ASME Journal of Mechanical Design*, Vol. 120, No. 4, pp. 653-658.
27. Wan, J. and S. Krishnamurty, 1998, "Towards a Consistent Preference Representation in Engineering Design," *ASME Design Theory and Methodology Conference*, Atlanta, GA. Paper Number: DETC98/DTM-5675.
28. Callaghan, A. R. and K. E. Lewis, 2000, "A 2-Phase Aspiration-Level and Utility Theory Approach to Large Scale Design," *ASME Design Theory and Methodology Conference*, Baltimore, MD. ASME, DETC2000/DTM-14569.
29. Thurston, D. L., 1999, "Real and Perceived Limitations to Decision Based Design," *ASME Design Theory and Methodology Conference*, Las Vegas, NV. Paper Number: DETC99/DTM-8750.
30. Simon, H. A., 1983, *Reason in Human Affairs*, Stanford University Press, Stanford, CA.
31. Savage, L. J., 1954, *The Foundations of Statistics*, Wiley, New York.
32. Fernandez, M. G., C. C. Seepersad, D. W. Rosen, J. K. Allen and F. Mistree, 2005, "Decision Support in Concurrent Engineering--The Utility-Based Selection Decision Support Problem," *Concurrent Engineering: Research and Applications*, Vol. 13, No. 1, pp. 13-28.
33. Martin, M. V. and K. Ishii, 1997, "Design for Variety: Development of Complexity Indices and Design Charts," *ASME Design for Manufacturing Conference*, Sacramento, CA. Paper Number: DETC97/DFM-4359.
34. Simpson, T. W., C. C. Seepersad and F. Mistree, 2001, "Balancing Commonality and Performance within the Concurrent Design of Multiple Products in a Product Family," *Concurrent Engineering: Research and Applications (CERA)*, Vol. 9, No. 3, pp. 177-190.
35. Meyer, M. H. and J. M. Utterback, 1993, "The Product Family and the Dynamics of Core Capability," *Sloan Management Review*, Vol. 34, No. 3, pp. 29-47.

36. Uzumeri, M. and S. Sanderson, 1995, "A Framework for Model and Product Family Competition," *Research Policy*, Vol. 24, No. 4, pp. 583-607.
37. McDermott, C. M. and G. N. Stock, 1994, "The Use of Common Parts and Designs in High-Tech Industries: A Strategic Approach," *Production and Inventory Management Journal*, Vol. 35, No. 3, pp. 65-68.
38. Collier, D. A., 1981, "The Measurement and Operating Benefits of Component Part Commonality," *Decision Sciences*, Vol. 12, No. 1, pp. 85-96.
39. Kota, S., K. Sethuraman and R. Miller, 2000, "A Metric for Evaluating Design Commonality in Product Families," *ASME Journal of Mechanical Design*, Vol. 122, No. 4, pp. 403-410.
40. Chen, W., D. W. Rosen, J. K. Allen and F. Mistree, 1994, "Modularity and the Independence of Functional Requirements in Designing Complex Systems," *Concurrent Product Design* (R. Gadh, Ed.), ASME, Vol. 74, pp. 31-38.
41. Rosen, D. W., 1996, "Design of Modular Product Architectures in Discrete Design Spaces Subject to Life Cycle Issues," *ASME Advances in Design Automation Conference*, Irvine, CA. Paper Number: DETC96/DAC-1485.
42. Siddique, Z. and D. W. Rosen, 2000, "Product Family Configuration Reasoning Using Discrete Design Spaces," *ASME Design Theory and Methodology Conference*, Baltimore, MD. Paper Number: DETC2000/DTM-14666.
43. Dahmus, J. B., J. P. Gonzalez-Zugasti and K. N. Otto, 2000, "Modular Product Architecture," *ASME Design Theory and Methodology Conference*, Baltimore, MD. Paper Number: DETC2000/DTM-14565.
44. Fujita, K., 2002, "Product Variety Optimization Under Modular Architecture," *Computer Aided Design*, Vol. 34, No. 12, pp. 953-965.
45. Gonzalez-Zugasti, J. P. and K. N. Otto, 2000, "Modular Platform-Based Product Family Design," *ASME Advances in Design Automation Conference*, Baltimore, MD. Paper Number: DETC2000/DAC-14238.
46. Simpson, T. W., W. Chen, J. K. Allen and F. Mistree, 1996, "Conceptual Design of a Family of Products Through the Use of the Robust Concept Exploration Method," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Bellevue, WA. AIAA Paper Number 96-4161.
47. Farrell, R. S. and T. W. Simpson, 2003, "Product Platform Design to Improve Commonality in Custom Products," *Journal of Intelligent Manufacturing*, Vol. 14, No. 6, pp. 541-556.
48. Martin, M. and K. Ishii, 1996, "Design for Variety: A Methodology for Understanding the Costs of Product Proliferation," *ASME Design Theory and Methodology Conference*, Irvine, CA. Paper Number: DETC96/DTM-1610.
49. Ishii, K., C. Juengel and C. F. Eubanks, 1995, "Design for Product Variety: Key to Product Line Structuring," *ASME Design Theory and Methodology Conference*, Boston, MA, Vol. 83-2, 499-506.
50. Fujita, K. and K. Ishii, 1997, "Task Structuring Toward Computational Approaches to Product Variety Design," *ASME Advances in Design Automation Conference*, Sacramento, CA. Paper Number: DETC97/DAC-3766.
51. Fujita, K., T. Yoneda, S. Akagi and M. Ishakawa, 1998, "Simultaneous Optimization of Product Family Sharing System Structure and Configuration," *ASME Advances in Design Automation Conference*, Atlanta, GA. Paper Number: DETC98/DAC-5608.

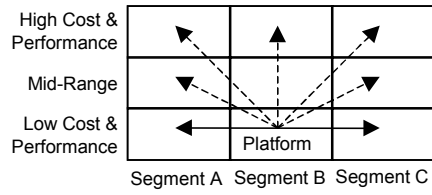
52. D'Souza, B. S. and T. W. Simpson, 2002, "A Genetic Algorithm Based Method for Product Family Design Optimization," *ASME Advances in Design Automation Conference*, Montreal, Canada. Paper Number: DETC2002/DAC-34106.
53. Kokkolaras, M., R. Fellini, H. M. Kim, N. F. Michelena and P. Y. Papalambros, 2002, "Extension of the Target Cascading Formulation to the Design of Product Families," *Structural and Multidisciplinary Optimization*, Vol. 24, No. 4, pp. 293-301.
54. Whitt, W., 1983, "The Queueing Network Analyzer," *Bell System Tech. J.*, Vol. 62, No. 9, pp. 2779-2815.
55. Bitran, G. and D. Tirupati, 1988, "Multiple Product Queueing Networks with Deterministic Routing: Decomposition Approach and the Notion of Inference," *Management Science*, Vol. 34, No. 1, pp. 75-100.
56. Hopp, W. J. and M. L. Spearman, 1996, *Factory Physics*, 1st Ed., Irwin, Boston, MA.
57. Balling, R., L. Borup, B. Busaker, T. Chambers, D. Davidson, G. Gritton, J. Free, A. Parkinson, J. Talbert and D. Warren, 1998, *OptdesX: A Software System for Optimal Engineering Design, Users Manual, Release 2.0.4.*, Design Synthesis, Inc., Provo, UT.
58. Bohachevsky, I., M. Johnson and M. Stein, 1986, "Generalized Simulated Annealing for Function Optimization," *Technometrics*, Vol. 28, No. 3, pp. 209-217.
59. Aarts, E. and J. Korst, 1989, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, Wiley, Chichester.



(a) Horizontal Leveraging



(b) Vertical Leveraging



(c) Beachhead Approach

Figure 1 – Market Segmentation Grid and Potential Platform Leveraging Strategies [1]

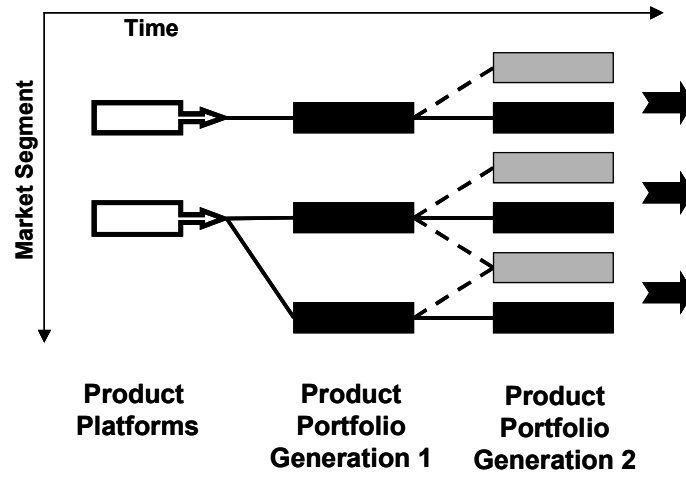


Figure 2 – Product Platforms Support Expansion and Evolution of a Family of Products

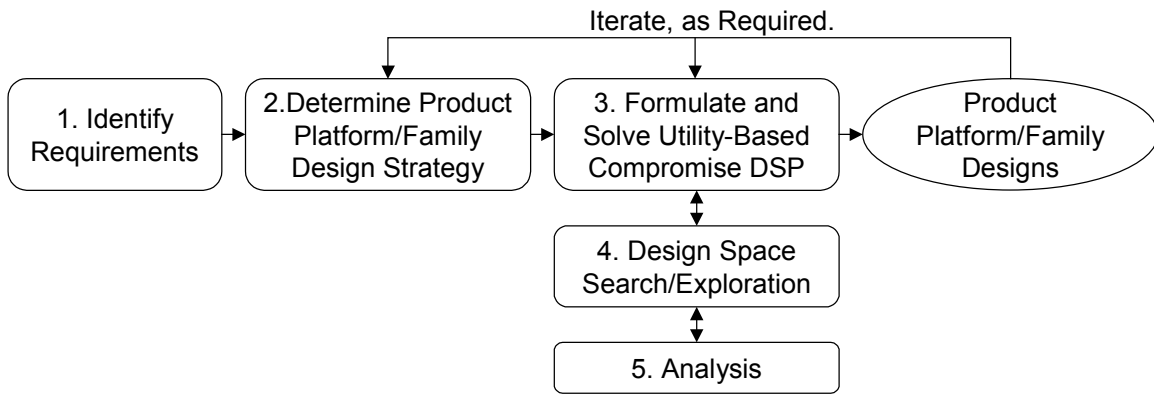


Figure 3 – A Product Platform/Family Design Process

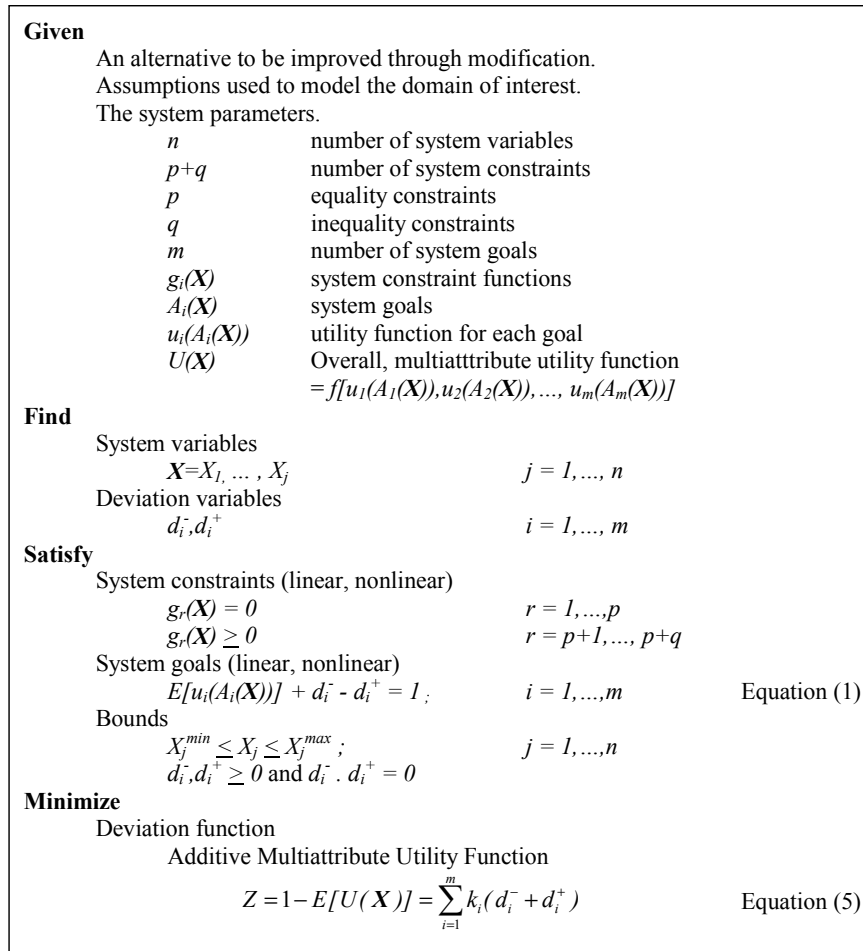
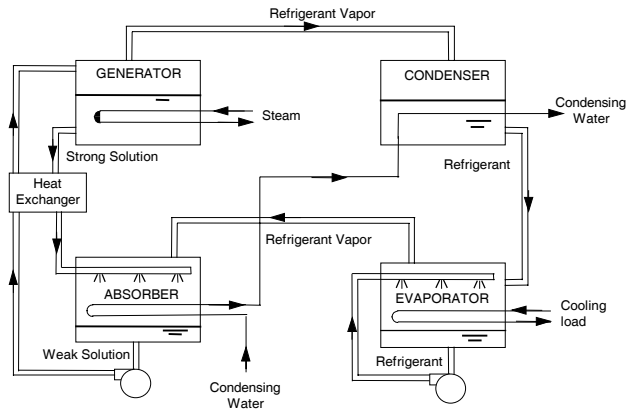
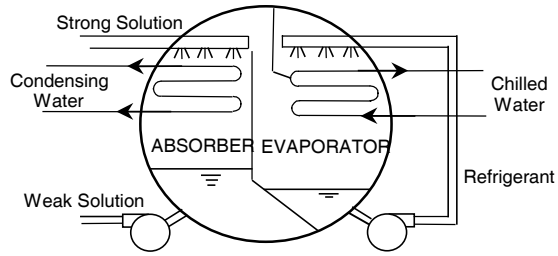


Figure 4 – Mathematical Formulation of the Utility-Based Compromise Decision Support Problem [19]



(a) Components of the Absorption Refrigeration Cycle



(b) The Absorber-Evaporator Module

Figure 5 – Layout of an Absorption Chiller

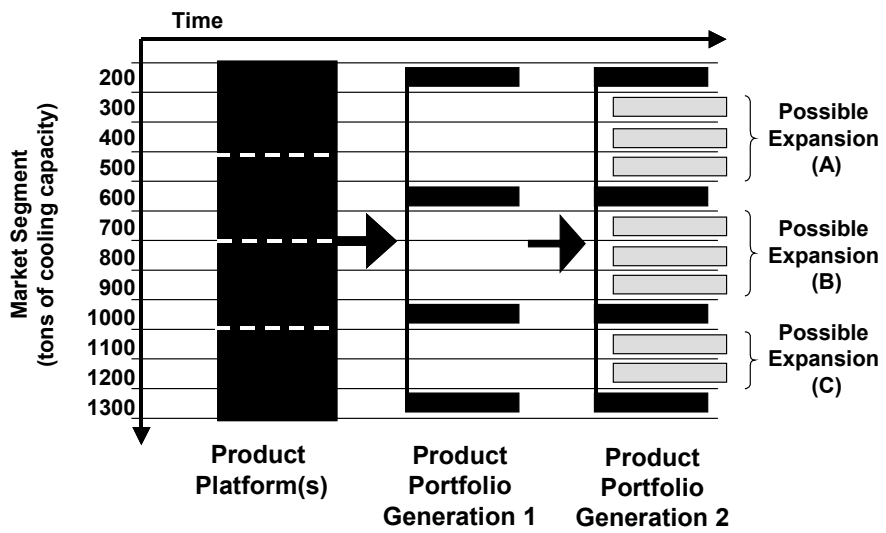


Figure 6 – Initial Product Portfolio and Possible Product Portfolio Expansions for the Family of Absorption Chillers

Given

- Heat-transfer models of the absorber-evaporator
- Production system models of cost and cycle-time
- Commercial tube alternatives
- Production Plan: chiller capacities to be produced and production quantities for each capacity
- Product Platform Configuration: number of product platforms to be designed and the products to be based on each platform
- Assessed utility functions for each goal and constants for combining the functions into a multi-attribute utility function (Tables 5 and 6)

Find

For each platform:

L , Length of tubes of the absorber-evaporator module per platform

E , Selection of tube for the evaporator per platform

A , Selection of tube for the absorber per platform

For each goal:

$$d_i^-, d_i^+ \quad i = 1, \dots, \# \text{ goals}$$

Satisfy

System Constraint

For each product:

$$TWOE \leq 7^\circ \text{ C} \quad (\text{Table 4})$$

$$TSOA \leq 42.4^\circ \text{ C} \quad (\text{Table 4})$$

$$XSOA \leq 59.5\% \quad (\text{Table 4})$$

$$\Delta P \leq 7 \text{ (ft of water)} \quad (\text{Table 4})$$

$$COST \leq COST_{utility=0}$$

$$CT_R \leq CT_{R, utility=0} \quad (R=200 \text{ to } 1300)$$

System Goals

$$E[u(COST)] + d_i^- - d_i^+ = 1 \quad \text{Equation (1)}$$

$$E[u(CT_R)] + d_i^- - d_i^+ = 1; R=200 \text{ to } 1300 \quad \text{Equation (1)}$$

Bounds

$$5.18 \text{ m} \leq L \leq 7.32 \text{ m}$$

$$E \in \{1,2,3,4\}$$

$$A \in \{1,2,3,4\}$$

$$d_i^+, d_i^- \geq 0$$

$$d_i^+ \cdot d_i^- = 0$$

Minimize

Additive Multiattribute Utility Function

$$Z = 1 - E[U(\mathbf{X})] = \sum_{i=1}^n k_i (d_i^- + d_i^+) \quad \text{Equation (5)}$$

Figure 7 – Utility-Based Compromise DSP for Absorption Chiller Product Platform Design

TUBE	E1	E2	E3	E4
Outer Diameter-cm	1.589	1.91	2.22	2.54
Wall Thickness-cm	0.064	0.064	0.064	0.064
Fins/cm	11.8	11.8	11.8	11.8
Fin Height-cm	0.13	0.13	0.13	0.13
Fin Thickness-cm	0.05	0.05	0.05	0.05
Material	Copper	Copper	Copper	Copper
Cost (\$/m)	19.69	21.33	22.97	24.61

TUBE	A1	A2	A3	A4
Outer Diameter-cm	1.59	1.91	2.22	2.54
Wall Thickness-cm	0.064	0.064	0.064	0.064
Fins/cm	smooth	smooth	smooth	smooth
Material	Copper	Copper	Copper	Copper
Cost (\$/m)	10.66	11.48	12.30	13.12

Name	Category	Range	Units
Tube Length	Control factor	5.18 – 7.32	m
Evap. Tube Type	Control factor	E1 – E4	
Absorb. Tube Type	Control factor	A1 – A4	
# of Evap. Tubes	Control factor	< 500	
# of Absorb. Tubes	Control factor	< 500	
Avg. Cost/unit	Goal		\$
Exp. Cycle Time	Goal		hours

Constraint	Value
Max Pressure Drop of Chiller Water (ΔP)	7 (m of water)
Max Temp of Chilled Water Leaving Evaporator (TWOE)	7° C
Max Temp of LiBr/Water Solution Leaving Absorber (TSOA)	42.4° C
Max Concentration of LiBr/Water Solution Leaving Absorber (XSOA)	59.5%
Range of Length/Diameter Ratio for Absorber/Evaporator Module	$3 \leq L/D \leq 15$

Goal	a	b	c (10^{-12})	d
Cost	2.6675	0.3205	1165.5	6.7968
CT200	2.2507	1.1832	1.3866	39.796
CT300	2.2692	1.1682	1.4138	38.865
CT400	2.2876	1.1537	1.4008	38.0186
CT500	2.3058	1.1397	1.3548	37.2444
CT600	2.3239	1.1261	1.2834	36.5329
CT700	2.3418	1.1130	1.1943	35.8760
CT800	2.3595	1.1002	1.0943	35.2672
CT900	2.3771	1.0878	0.9893	34.7008
CT1000	2.3946	1.0758	0.8837	34.1723
CT1100	2.4120	1.0641	0.7812	33.6776
CT1200	2.4292	1.0528	0.6841	33.2134
CT1300	2.4463	1.0417	0.5942	32.7766

Table 6– Scaling Constants (k's) for Multi-attribute Utility Function

$$\text{Additive Form: } U(\mathbf{X}) = \sum_{i=1}^m k_i u_i(A_i(\mathbf{X}))$$

Goal	Scaling Constant (k _i)	Scaling Constant (k _i)
Cost	0.3846	0.36
CT200	0.1538	0.09
CT300		0.09
CT400		0.09
CT500		0.09
CT600	0.1538	0.09
CT700		
CT800		
CT900		
CT1000	0.1538	0.09
CT1100		
CT1200		
CT1300	0.1538	0.09

Table 7 – Starting Points for Optimization Algorithm		
Design Variable	Lower Starting Point	Upper Starting Point
L	6.86	7.32
E	3	4
A	3	4

Table 8 -- Final Goal Values for Experiments 1, 2, and 3							
	Experiment 1			Experiment 2			Experiment 3
Max # Platforms	1	2	4	4	4	4	4
Reference to Figure 6	Generation 1	Generation 1	Generation 1	Generation 1 + Expansion A	Generation 1 + Expansion B	Generation 1 + Expansion C	25% Prob of Expansion A,B,C, or No Expansion
COST (\$)	177,763	174,030	170,582	159,393	174,449	191,817	172,484*
CT200 (Cycle Time--hrs)	302	302	302	278	298	321	300*
CT300				295			296*
CT400				313			313*
CT500				334			335*
CT600	378	380	378	353	373	396	375*
CT700					394		394*
CT800					416		417*
CT900					438	461	449*
CT1000	464	466	468	440	460	483	463*
CT1100						506	506*
CT1200						541	541*
CT1300	543	549	544	517	543	566	545*

- Expected Values

Table 9 -- Design Variable Values for Experiments 1, 2, and 3

		Experiment 1			Experiment 2			Experiment 3
# Platforms		1	2	4	4	4	4	4
Reference to Figure 6		Generation 1	Generation 1	Generation 1	Generation 1 + Expansion A	Generation 1 + Expansion B	Generation 1 + Expansion C	25% Prob of Expansion A,B,C, or No Expansion
Tonnage	Variable							
200-400	L (m)	7.32	7.24	7.32	7.32	7.32	7.32	7.32
	E	4	4	4	4	4	4	4
	A	1	1	4	3	3	4	3
500-700	L	"	"	7.32	7.24	7.24	7.24	7.24
	E	"	"	3	3	3	3	3
	A	"	"	1	1	1	1	1
800-1000	L	"	7.24	7.32	7.32	7.24	7.24	7.24
	E	"	3	2	4	3	3	3
	A	"	1	1	1	1	1	1
1100-1300	L	"	"	7.32	7.24	7.24	7.24	7.24
	E	"	"	3	4	3	3	3
	A	"	"	1	4	1	1	1

Table 10 -- Comparing Expected Performance of Product Platforms				
Experiment (Scenario)	3	2 (A)	2 (B)	2 (C)
Expected Cost	172,484	184,852	172,484	172,977
Expected Average Cycle Time per Unit	418	419	418	419