A Method for Interactive Decision-Making in Collaborative, Distributed Engineering Design

Gabriel Hernandez
Siemens Westinghouse Power Corporation
4400 Alafaya Trail, MC DV 322
Orlando, FL 32826-2399

Carolyn Conner Seepersad, Janet K. Allen, and Farrokh Mistree
Systems Realization Laboratory, Georgia Institute of Technology,
Atlanta, GA, USA 30332-0405

A decision support method is proposed for integrating distributed, collaborative design activities via mathematical coordination of the individual decisions of designers and other agents involved in the product realization process. This approach involves establishing a domain-independent decision model, expanding the scope of local decisions, and implementing a mathematical coordination mechanism based on game theory and approximate models of the information and strategies of designers. Using this approach, integration of computational models of design, manufacturing, and other portions of the product realization process into a single, large optimization problem is avoided, and the knowledge and expertise of each designer are fully utilized while keeping information transfer and computational requirements at a tractable level.

Keywords: Collaborative Design, Compromise Decision Support Problem, Decision-Based Design, Game Theory

1. A MODEL FOR DECISION SUPPORT IN ENGINEERING DESIGN

Engineering design processes usually involve various individuals who make decisions that affect one another, and effective coordination between these decision-makers is critical. Most of the literature on collaborative design is focused on models for managing design teams (e.g.,[1]), information sharing (e.g.,[2]) or computer-support systems (e.g.,[3]). However, an important issue in collaborative design that has received little attention is negotiation [4]. In a study conducted in the aerospace industry, Crabtree, et al., [5] found that engineers were spending about 10% of their time negotiating, and that this activity was the most frustrating. Recent research in collaborative design involves facilitating collaboration by streamlining dependencies, usually through decomposition and representation methods [6-8]. In this paper, the focus is on decision support, including decision models and game theoretic coordination mechanisms, for multiple designers who explicitly consider multiple, conflicting objectives in the collaborative design process.

In order to develop a consistent decision support approach for negotiation in multi-objective, collaborative design, it is important to begin with a domain-independent model for decision support, as shown in Figure 1. This model is based upon the notions that the principal role of a designer is to make decisions and that these decisions need to be formulated before they can be solved [9].

In this model for decision support, a decision is represented in terms of:

- A set of design variables, $\boldsymbol{x}$.
- A set of goals and constraints and their targets G_i, and
- Models $f_i(\boldsymbol{x})$ that quantify the relationships between the variables and objectives.

The model is domain-independent and facilitates the development of a mathematical approach for collaborative design based on formulating and making interdependent decisions.

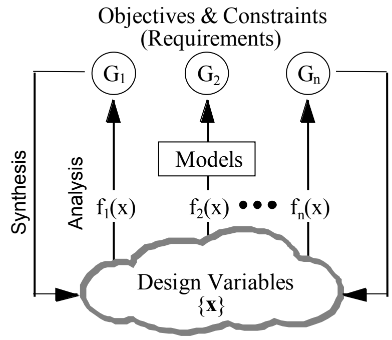

Figure 1 --A Model for Decision Support in Engineering Design [9]

The compromise Decision Support Problem (DSP) embodies the decision support model of Figure 1. The compromise DSP is a mathematical construct that is used to determine the values of design variables that satisfy a set of constraints and achieve a set of conflicting goals as closely as possible [10]. The mathematical form of a compromise DSP is summarized in Figure 2. The system descriptors, namely, system and deviation variables, system constraints, system goals, bounds, and the deviation function are described in detail elsewhere [10]. The concept of a compromise DSP is to minimize the difference between that which is desired (the goals, G_i) and that which can be achieved ($f_i(\boldsymbol{x})$) for multiple goals. This is accomplished by minimizing the deviation function (Z) expressed in terms of deviation variables (d_i^- and d_i^+), which measure the extent to which the goals are achieved. In the compromise DSP, multiple goals are considered by formulating the deviation function either with a weighted sum or preemptively.

The method for collaborative, distributed design presented in this paper is based on formulation of interdependent decisions as compromise DSPs and implementation of a

Given	n, number of design variables		
	p, number of equality constraints		
	q, number of inequality constraints		
	m, number of objectives		
	$f_i(\boldsymbol{x})$, goal achievement function		
	$g_r(\boldsymbol{x})$, constraint function		
	G_i, target values for the objective functions		
	$h_k(d_i)$, function to be minimized at priority level k for preemptive form		
	W_i, weight for Archimedean form		
Find	System variables:	$\boldsymbol{x}=\{x_1,\ldots,x_n\}$	
	Deviation variables:	d_i^-, d_i^+	$i=1,\ldots,m$
Satisfy	Goals:	$f_i(\boldsymbol{x})+d_i^- - d_i^+ = G_i$	$i=1,\ldots,m$
	Constraints:	$h_r(\boldsymbol{x})=0$	$r=1,\ldots,p$
		$g_r(\boldsymbol{x})\geq 0$	$r=1,\ldots,q$
	Bounds	$x_{j,\min} \leq x_j \leq x_{j,\max}$	$j=1,..,n$
	Other	$d_i^- \cdot d_i^+ = 0$	
		$d_i^-, d_i^+ \geq 0$	
Minimize	A deviation function:		
	Preemptive:	$Z=[f_1(d_i^-,d_i^+),\ldots,f_k(d_i^-,d_i^+)]$	
	Weighted Sum:	$Z=\sum_{i=1}^{m} w_i(d_i^- + d_i^+)$	

Figure 2 --Mathematical Form of the Compromise DSP [10]

mathematical coordination mechanism, based on game theory, for representing interactions among multiple designers. The application of game theory in collaborative design is discussed in Section 2, and the proposed decision support method is detailed in Section 3. In Sections 4 and 5, the method is applied to an example, and potential extensions of the approach are suggested.

2. GAME THEORY IN COLLABORATIVE DESIGN

In collaborative design, it is important to consider *multiple decision-makers and multiple objectives*. For complex design organizations, comprehensive, centralized decision-making exceeds the limits of human rationality. There are many advantages of distributed decision-making, including effective use of each designer's expertise and complex, discipline-specific analysis codes and reasonable levels of information exchange across enterprise and geographic boundaries. In addition, mapping the characteristics of a product into a single quantitative business measure (e.g., profit) is difficult during most stages of product development. Instead, it is reasonable to define a set of objectives and associated targets for products and processes, and to develop and identify successful designs that embody a preferable balance among these multiple objectives. If the design process is decentralized, it is reasonable to assume that individual designers have different objectives.

Typically, collaboration among multiple designers with multiple objectives is pursued in *ad hoc* ways such as team meetings, notices, or information exchange (e.g., [11-13,1]. These techniques are effective to some extent in practice, but they do not provide formal support for formulation and integration of the individual decisions that mark the progression of a design. An alternative approach for supporting collaboration is the application of techniques derived from game theory [14-26]. With game theory, mathematical approaches are available for analyzing situations in which the choices made by one decision-maker affect the objectives of other decision-makers, and vice versa. In engineering design, game theory has been employed to facilitate multi-disciplinary design [17-19] and concurrent engineering [15,21,22,25,26] and to extend decision-based design [16,23,24].

Game theory includes two branches: cooperative and non-cooperative game theory. In cooperative game theory, decision-makers form coalitions by agreeing to cooperate with one another. Most research in this branch of game theory involves investigating the stability of these coalitions [27,28]. On the other hand, in non-cooperative game theory the unit of analysis is the *individual* participant in the game who is concerned with doing as well for himself or herself as possible subject to clearly defined rules and possibilities. Individuals may exhibit "cooperative" behavior in 'non-cooperative' games if such behavior is in the best interest of each individual. Since non-cooperative games are focused mainly on the formulation of strategies that 'rational' individuals follow when their actions and objectives are affected by others, its mathematical models and techniques are a suitable foundation for formulating decisions in collaborative design.

Two basic types of formal models are employed in non-cooperative game theory. The first and simpler one is the strategic form or normal form game. It is suitable for modeling interactions between individuals when the timing and sequence of their actions does not affect the outcome. The second type of model is the extensive form game, in which the timing of actions is an important factor in the game.

Once a collaborative design situation is modeled as either a strategic or an extensive game, the next step is to analyze the model, predict how the decision-makers involved should act, and formulate appropriate strategies based on one of two solution techniques: dominance arguments or Nash equilibrium analysis. Dominance arguments have limited applications of practical interest in design; whereas, Nash equilibrium is applicable to a broad class of practical situations.

A Nash equilibrium is a profile of strategies such that each decision-maker's strategy is an optimal response to the other decision-makers' strategies. Each decision-maker can predict a Nash equilibrium, predict that his/her opponents can predict it, and so on. Nash equilibria are the only "consistent" predictions of the outcome of a game because if all decision-makers predict a particular Nash equilibrium, then no decision-maker has an incentive to act differently. A formal definition follows [24]:

Assume a finite player game, where the players are indexed i=1,…,n, and their respective available strategies sets are denoted by X_i. A strategy profile $\overline{x}^* = (\overline{x}_1,...,\overline{x}_{i-1},\overline{x}_i,\overline{x}_{i+1},...,\overline{x}_n)$ is a Nash equilibrium if for each player i and $\overline{x}_i \in X_i$,

$$z_i(\overline{x}^*) \geq z_i(\overline{x}_1,...,\overline{x}_{i-1},\overline{x}_i,\overline{x}_{i+1},...,\overline{x}_j) \text{ for all } \overline{x}_i \in X_i$$

where Z_i is the value of the payoff that player i receives as a result of the strategies chosen by the players in the game.

Collaborating designers may be abstracted as decision-makers in a strategic form game. Each designer, i, controls a set of variables, X_i, and seeks to optimize an objective function, Z_i. The designer's strategy in this game is the value of the variables under his/her control, X_i. Each designer develops a strategy that best achieves his/her objectives, given the strategies of other designers. The result of this game must be a Nash equilibrium, as previously discussed. Despite the terminology, it is possible to achieve cooperative (Pareto efficient) design solutions with non-cooperative models

This abstraction may be illustrated with an example, presented first in [14]. Consider Designers 1 and 2, who wish to minimize the functions $Z_1(x_1,x_2)$ and $Z_2(x_1,x_2)$, respectively:

$$Z_1(x_1,x_2) = x_1x_2 - 3x_1 + x_1^2 \quad (1)$$

$$Z_2(x_1,x_2) = \frac{x_2^2}{2} - x_1x_2 \quad (2)$$

Designer 1 is in control of x_1 and Designer 2 controls x_2, with $x_1 \geq 0$ and $x_2 \geq 0$. How can they decide on the values of these two variables? Suppose that Designer 1 makes the first tentative design at her minimum, $\boldsymbol{X}$=(1.5,0). Designer 2, using $x_1=1.5$, would then choose the value of x_2 that minimizes his own function, i.e., $x_2=1.5$. This value would be passed to Designer 1 who then changes x_1 accordingly until the process converges to a solution—in this case $\boldsymbol{X}$=(1,1), the point of Nash equilibrium. This iterative process is characteristic of decentralized design environments in which, for any choice of decision variable values by one designer, other designers choose their own decision variable values to optimize their own objectives. Every such set of variables is known as a *best reply correspondence (BRC)* [24] or *rational reaction set* [17,18].

A *BRC* is a model that defines the strategy of a decision-maker, k, in a game. It is a function that maps the values of the decision variables of a decision-maker, $\boldsymbol{x}_k$, that optimize his/her objective, $Z_k(\boldsymbol{X})$, as a function of the values of other decision-makers' variables. For example, taking the derivatives of Equations (1) and (2) and setting them equal to zero, the following set of equations are obtained:

$$x_1 = 0.5(3 - x_2) \quad (3)$$

$$x_2 = x_1 \quad (4)$$

Equations (3) and (4) represent the *BRC*'s of Designers 1 and 2, respectively. The Nash equilibrium is the intersection of the *BRC*'s or $\boldsymbol{X}$ = (1,1) in this example.

Analytical models of *BRC's*, such as Equations (3) and (4), are difficult to obtain in practice. To address this challenge, Lewis and Mistree [17] developed a method for approximating *BRC's* using response surfaces, as illustrated in Figure 3. This method requires four steps:

1. Each designer k identifies the system variables of other designers (where the set of other designers is identified as ξ) that affect designer k's performance significantly. This can be done systematically using screening experiments.
2. An experiment is designed in the set of other designers' system variables identified in Step 1.
3. Designer k's compromise DSP is solved repeatedly for each combination of other designers' system variable values included in the experiment designed in Step 2. Each time a compromise DSP is solved, the resulting values of designer k's system variables, $\boldsymbol{x}_k$, that minimize designer k's deviation function are recorded.
4. A response surface is fit to the recorded values of designer k's system variables, $\boldsymbol{x}_k^*$. The BRC_k for player k is the set of response surfaces that approximate the designer's best reply (in terms of his/her system variables that minimize his/her deviation function) to potential variable choices of other designers.

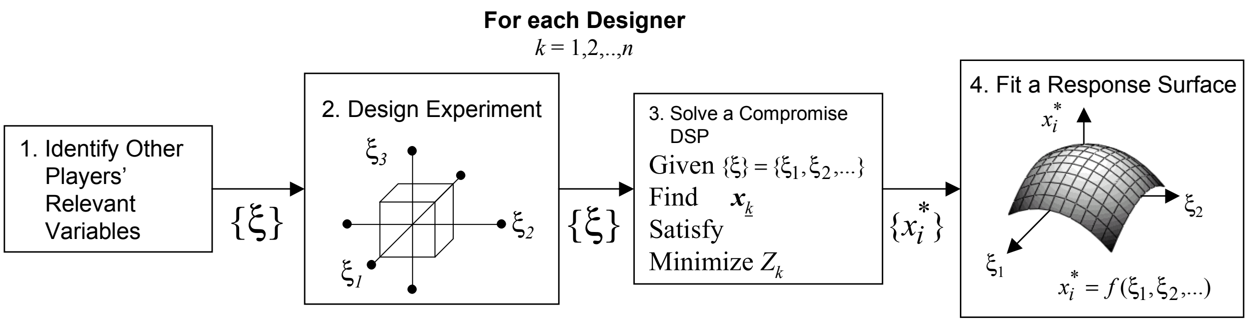

Figure 3 --Developing an Approximation for the *BRC* (adapted from [18])

Unfortunately, the Nash equilibrium solutions obtained with game theory are not *necessarily* Pareto efficient; therefore, in theory, better designs may exist *from a systems perspective*. In the following section, a method for collaborative decision-making is described that addresses this challenge. As suggested by Marston and coauthors [23,24], game theory is employed with the compromise DSP in a decision-based context, and approximations of BRC's are employed as in [18]. However, the method is designed to facilitate *cooperation* among multiple designers. Accordingly, as discussed in the next section, common system-wide objective functions are formulated and employed, and the functional form of the BRC is adapted. The advantages of a method of this type include:

- decentralized decision making with utilization of individual capabilities, knowledge and skills,
- distributed information and computing resources, along with minimization of information and computational burden,
- and improved designs due to efficient mathematical coordination and formulation of decisions.

3. A METHOD FOR MAKING DECISIONS IN COLLABORATIVE ENGINEERING DESIGN

The components of a decision support method for collaborative design include the compromise DSP for mathematically formulating the decisions of each designer and game theory—specifically, the abstraction of designers as decision-makers in a strategic game and the use of approximate *BRC*'s—for coordination of multiple decisions. To facilitate collaboration with these tools, it is important to seek cooperative or Pareto efficient outcomes. Pareto efficient solutions are those solutions $\boldsymbol{X}^*=(x_1^*,x_2^*,\ldots,x_N^*)$ for which there is no other feasible solution $\boldsymbol{X}$ such that $f_k(\boldsymbol{X})\geq f_k(\boldsymbol{X}^*)$ for all k and $f_k(\boldsymbol{X})>f_k(\boldsymbol{X}^*)$ for at least one designer. Nash equilibrium is Pareto efficient when all designers pursue a common objective. A common objective is not required to be a specific design attribute (e.g., mass, cost, etc.) or business measure (e.g., profit); it can be *a combination of the designers' objectives*. However, all of the designers must agree on the formulation of such a common objective.

One approach for formulating a common objective function is to aggregate the individual objective functions into a system deviation function, as modeled in the compromise DSP, using a weighted sum. Given target values (G_{ki}) for the achievement of i individual objectives (f_{ki}) for each designer, k, a series of goals are formulated as follows:

$$f_{ki} + d_{ki}^- - d_{ki}^+ = G_{ki} \tag{5}$$

Then, a common objective function is formulated to be minimized by all N designers:

$$Z = \sum_{k=1}^{N}\sum_{i=1}^{m_k} w_{ki}(d_{ki}^- + d_{ki}^+) \tag{6}$$

where m_k is the number of objectives for designer k and w_{ki} is the weight for objective i of designer k. However, the N designers must agree on a set of weights for Equation 6.

One possibility for facilitating agreement on a set of weights involves utilizing linear physical programming [29]. Instead of specifying weights directly for each objective, the designers agree on five desirability levels for each design objective: (1) ideal (or highly desirable), (2) desirable, (3) tolerable, (4) undesirable, and (5) unacceptable. Then, based on these levels, it is possible to formulate goal functions in a compromise DSP as follows [30]:

$$\frac{f_{ki} - \max(f_{ki} - G_{ki,r+1}, 0)}{G_{ki,r}} + d_{ki,r}^- - d_{ki,r}^+ = 1 \qquad r = 1,\ldots 4 \tag{7}$$

where $r=1,\ldots,4$ are the first four desirability levels. The fifth desirability value, the unacceptable one, becomes an additional constraint:

$$f_{ki} \leq G_{ki,5} \quad \text{(assuming minimization)} \tag{8}$$

Then, a common deviation function is formulated as follows:

$$Z = \sum_{k=1}^{N}\sum_{i=1}^{m}\sum_{r=1}^{4}(w_{ki,r}^- d_{ki,r}^- + w_{ki,r}^+ d_{ki,r}^+) \tag{9}$$

The value of the weights, w_{kir}, are obtained by applying an algorithm described in [30], where details are also provided for formulation of the compromise DSP with a linear physical programming objective function. The formulation of an objective function using "meaningful" degrees of desirability, specified by the designers themselves, makes this approach suitable for collaborative design. It is easier for a team of collaborating designers to agree on desirable or tolerable values for objectives than to agree on arbitrary weights. Minimization of Z in Equation 9 *is Pareto efficient as long as at least one of the deviation variables is greater than zero*. If all deviation variables are zero, then "ideal" target levels have been achieved for all criteria.

Given a common deviation function, Z, that all designers agree to optimize, *how do they proceed to solve the design problem in a coordinated manner?* Each designer acts proactively as a decision-maker and not merely as a function evaluator, but rather than optimizing his/her own objective function, each designer minimizes a portion of the common objective function, Z_k, that is relevant to his/her decision. Generally, Z_k includes the deviation variables and weights associated with designer k (e.g., mass and strength for a structural designer). *BRC*'s are obtained for optimal values of Z_k as functions of *shared* design variables. The subset of designer *k*'s variables that directly affect other designers are the *shared* variables, $\boldsymbol{x}_k^s \subset \boldsymbol{x}_k$, and the set of shared variables for all designers is $\boldsymbol{X}^s = \{\boldsymbol{x}_1^s, \boldsymbol{x}_2^s, ..., \boldsymbol{x}_N^s\}$. The approach includes eight steps:

1. The designers define relevant goals G_{ki} and identify the set of shared variables, $\boldsymbol{X}^s$.
2. The designers agree on five desirability levels for each of the goals defined in Step 1: (1) ideal (or highly desirable), $G_{ki,1}$, (2) desirable, $G_{ki,2}$, (3) tolerable, $G_{ki,3}$, (4) undesirable, $G_{ki,4}$, and (5) unacceptable, $G_{ki,5}$.
3. A system deviation function, Z, (Eq. 9) is formed, and the weights, $w_{ki,r}$, are calculated using the algorithm described in [30].
4. Each designer adopts the minimization of his/her portion of the system deviation function, Z_k, as an objective function.
5. Each designer formulates a compromise DSP, as shown in the top portion of Figure 4, with Z_k as the objective function to minimize.
6. Using this compromise DSP, each designer, k, develops a BRC_k in three steps:
   6.1. An experiment is designed with the set of all shared variables (identified in Step 1) as design factors of the experiment and Z_k (identified in Step 4) as the response variable.
   6.2. The compromise DSP (formulated in Step 5) is solved for each combination of the shared variables included in the experiment and the resulting minimum value of Z_k, (represented here as Z_k^*) is recorded.
   6.3. A response surface is fit to the values of Z_k^* recorded in the previous step. This response surface is the *BRC* of designer k, and represents the minimum achievable value of Z_k as a function of all shared variables: $BRC_k = Z_k^*(\boldsymbol{X}^S)$
7. A system manager (or project leader) uses the *BRC*'s developed by each designer to find the values of the shared variables that minimize the complete objective function Z. A template for the system manager's compromise DSP is included in the bottom portion of Figure 4.

For Every Designer *k (Steps 6 and 8)*

Given The value of shared variables, $\boldsymbol{X}^s$

Find The value of non-shared variables, $\boldsymbol{x}_k^{ns}$

The value of the deviation variables, $d_{ki,r}^+, d_{ki,r}^-$

Satisfy Design Constraints

Design Bounds, $x_{kj,\,min} \le x_{kj} \le x_{kj,max}$,

Other: $d_{ki,r}^-, d_{ki,r}^+ \ge 0$

$$d_{ki,r}^- \cdot d_{ki,r}^+ = 0$$

Minimize
$$Z_k = \sum_{i=1}^{m} \sum_{r=1}^{4} (w_{ki,r}^- d_{ki,r}^- + w_{ki,r}^+ d_{ki,r}^+)$$

For the System Level Manager to Find the Set of Shared Variables *(Step 7)*

Given N, the number of designers

BRC_k a model of the best reply correspondence of each designer, $k=1,...,N$

$$BRC_k = Z_k^*(\boldsymbol{X}^s)$$

Find The set of *shared* variables, $\boldsymbol{X}^s = \{\boldsymbol{x}_1^s, \boldsymbol{x}_2^s, ..., \boldsymbol{x}_N^s\}$

Satisfy $Z_k(\boldsymbol{X}^s) = BRC_k \quad k=1,...,N$

Minimize
$$Z = \sum_{i=1}^{N} Z_k(\boldsymbol{X}^s)$$

Figure 4 – Compromise Decision Support Problems for Collaborative Distributed Design

8. The manager passes the final values of the shared variables, $\boldsymbol{X}^s$, to the individual designers. Then, the individual designers find the values of their non-shared variables, $\boldsymbol{x}_k^{ns}$, by solving their individual compromise DSP's.

This method for collaborative design is applied to an example in the following section.

4. AN ILLUSTRATIVE EXAMPLE

Suppose two designers collaborate to design a simple steel bar and its manufacturing process. The product designer, Designer 1, is interested in designing a solid steel bar to transmit a torque *T* of 100 N·m, as shown in Figure 5. The design variables are the length, *l*, and diameter, *d*, of the bar.

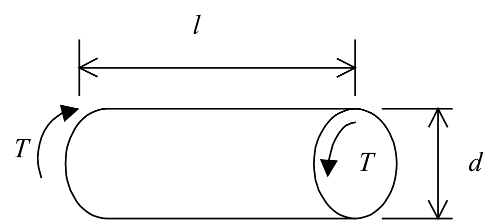

Figure 5 -- A Bar in Torsion

The product designer's objectives are to minimize the mass (*m*) and the shear stress (τ) of the bar. The mass and shear stress are calculated as follows:

$$m = \rho \cdot \frac{\pi}{4} d^2 l = 6157 d^2 l \tag{10}$$

$$\tau = \frac{16T}{\pi d^3} = \frac{510}{d^3} \tag{11}$$

where ρ is the density of steel (7840 kg/m^3). The design is subject to a maximum angular torsion constraint:

$$\theta_{max} = \frac{32\,T\,l}{G \pi d^4} \leq 0.01 \,\text{rad} \tag{12}$$

where *G* is the rigidity modulus of steel (80 GPa). With both *d* and *l* in meters, Equation (12) is rewritten as:

$$g_1 = 785 \times 10^3 d^4 - l \geq 0 \tag{13}$$

Finally, the design is constrained by the following bounds on the design variables:

$$0.01 \leq d \leq 0.05 \;\; [\text{m}] \tag{14}$$

$$0.05 \leq l \leq 0.1 \;\; [\text{m}] \tag{15}$$

The manufacturing engineer, Designer 2, specifies a turning process for the bar. Designer 2 minimizes the turning time, *t*, and the power consumed during the turning process, *P*. The turning time is a function of the length, *l*, determined by the product designer, and the manufacturing engineer's variables—the feed, *f*, the rotational turning speed, Ω, and the number of turning passes, *n* (a positive integer):

$$t = \frac{2nl}{f\,\Omega} \tag{16}$$

The power is estimated as the product of the specific cutting energy for steel (e_s) times the material removal rate (*MRR*):

$$P = e_s MRR \tag{17}$$

where $e_s = 66 \text{x} 10^3$ kW·min/m^3 and the material removal rate (*MRR*) is:

$$MRR = \pi\left(\frac{d+D}{2}\right)\cdot(D-d)\cdot f\cdot\Omega \tag{18}$$

If the available steel bar diameter is 0.055 m, Equation (17) can be rewritten as:

$$P = 210\times10^3\left(\frac{d+0.055}{2}\right)\cdot(0.055-d)\cdot f\cdot\Omega \quad [\text{kW}] \tag{19}$$

The constraints on the process include the depth of cut, *(D-d)/n,* which must be less than or equal to 5 mm:

$$\frac{0.055-d}{n} \le 0.005 \;\; [\text{m}] \tag{23}$$

and the following bounds on the values of *f* and Ω:

$$0.0001 \le f \le 0.025 \;\; [\text{m/rev}] \tag{24}$$

$$60 \le \Omega \le 600 \;\; [\text{rev/min}] \tag{25}$$

Design variables, constraints, and objectives for both the product designer and the manufacturing process designer are summarized in Figure 6 for this example. With this simple example, it is possible to illustrate the solution of a collaborative problem in a distributed manner, i.e., without integrating multiple decisions into a single one.

<table>
<tr><th colspan="3"><u>Product Design</u></th><th colspan="3"><u>Manufacturing</u></th></tr>
<tr><td>Given</td><td></td><td></td><td>Given</td><td>$d,\ l$</td><td></td></tr>
<tr><td>Find</td><td>$d,\ l$</td><td></td><td>Find</td><td>f, Ω, n</td><td></td></tr>
<tr><td>Satisfy</td><td>$g_1 = 785\times10^3 d^4 - l \ge 0$</td><td>Eq. (13)</td><td>Satisfy</td><td>$g_2 = 0.005n - (0.055 - d) \ge 0$</td><td>Eq. (23)</td></tr>
<tr><td></td><td>$0.01 \le d \le 0.05$</td><td>Eq. (14)</td><td></td><td>$0.0001 \le f \le 0.025$</td><td>Eq. (24)</td></tr>
<tr><td></td><td>$0.05 \le l \le 0.1$</td><td>Eq. (15)</td><td></td><td>$60 \le \Omega \le 600$</td><td>Eq. (25)</td></tr>
<tr><td>Minimize</td><td>$m = 6157 d^2 l$</td><td>Eq. (10)</td><td></td><td>$n \in \{1,\ 2, \ldots\}$</td><td></td></tr>
<tr><td></td><td>$\tau = \frac{510}{d^3}$</td><td>Eq. (11)</td><td>Minimize</td><td>$t = \frac{2nl}{f\ \Omega}$</td><td>Eq. (16)</td></tr>
<tr><td></td><td></td><td></td><td></td><td>$P = 210\times10^3\left(\frac{d+0.055}{2}\right)\cdot(0.055-d)\cdot f\cdot\Omega$</td><td>Eq. (19)</td></tr>
</table>

Figure 6 -- Baseline Decisions for Example Problem

Step 1. Identify the goals, G_{ki}, and the set of shared variables, $\boldsymbol{X}^s$. As shown in Figure 6, the relevant objectives for this problem are to minimize *m*, *τ*, *t* and *P*. In this simple problem, the product designer's goals and constraints do not depend on any of the manufacturing engineer's variables; however, the manufacturing engineer requires the values of *d* and *l,* controlled by the product designer. Hence, for this example the set of shared variables is $\boldsymbol{X}^s = \{d, l\}$.

Step 2. For each goal, G_{ki}, define five levels of desirability. Together, the two designers specify five physical programming levels for each of the goals, as shown in Table 1.

Table 1 – Physical Programming Desirability Levels for Mass, Shear Stress, Turning Time, and Power

	Target r				
	1 Ideal	2 Desirable	3 Tolerable	4 Undesirable	5 Unacceptable
m [kg]	0.05	0.15	0.35	0.75	1.5
τ [MPa]	35	50	60	75	100
t [min]	1	1.5	2.25	3.25	5
P [kW]	20	30	50	90	150

For numerical convenience, the target values in Table 1 are normalized by dividing each value by the largest value in the row. For example, the ideal normalized mass is 0.05/1.5 = 0.033. The normalized values are shown in Table 2.

Table 2 -- Desirability Levels for Normalized Mass, Shear Stress, Turning Time, and Power

	Target r				
	1 Ideal	2 Desirable	3 Tolerable	4 Undesirable	5 Unacceptable
$\hat{m}$	0.033	0.1	0.233	0.5	1
$\hat{\tau}$	0.35	0.5	0.6	0.75	1
$\hat{t}$	0.2	0.3	0.45	0.65	1
$\hat{P}$	0.133	0.2	0.333	0.6	1

Step 3. Formulate a deviation function Z and find the various weights. The common objective function to be minimized is:

$$Z = \sum_{r=1}^{4} (w_{1r} d_{1r}^{+} + w_{2r} d_{2r}^{+} + w_{3r} d_{3r}^{+}) \tag{26}$$

where the deviation variables can be obtained from Eqs. (27) to (30):

$$\frac{\hat{m} - \max(\hat{m} - \hat{m}_{\text{Target}_{r+1}}, 0)}{\hat{m}_{\text{Target}_r}} + d_{1r}^{-} - d_{1r}^{+} = 1 \qquad r = 1,..,4 \tag{27}$$

$$\frac{\hat{\tau} - \max(\hat{\tau} - \hat{\tau}_{\text{Target}_{r+1}}, 0)}{\hat{\tau}_{\text{Target}_r}} + d_{2r}^{-} - d_{2r}^{+} = 1 \qquad r = 1,..,4 \tag{28}$$

$$\frac{\hat{t} - \max(\hat{t} - \hat{t}_{\text{Target}_{r+1}}, 0)}{\hat{t}_{\text{Target}_r}} + d_{3r}^{-} - d_{3r}^{+} = 1 \qquad r = 1,..,4 \tag{29}$$

$$\frac{\hat{P} - \max(\hat{P} - \hat{P}_{\text{Target}_{r+1}}, 0)}{\hat{P}_{\text{Target}_r}} + d_{4r}^{-} - d_{4r}^{+} = 1 \qquad r = 1,..,4 \tag{30}$$

and target values for each goal are obtained from Table 2. An algorithm from [30] is exercised to determine the weights for Equation 26, as listed in Table 3.

Table 3 -- Weights for Deviation Function

	r			
	1	2	3	4
w_{1r}	1.50	0.98	1.61	3.10
w_{2r}	0.67	2.63	3.96	7.11
w_{3r}	1.00	1.20	3.25	4.82
w_{4r}	1.50	0.98	1.61	4.90

Step 4. Each designer adopts minimization of his/her portion of the common objective function, Z, as an objective. The product designer is interested in minimizing mass and shear stress. Therefore, she adopts the following objective function:

$$Z_1 = \sum_{r=1}^{4}(w_{1r}d_{1r}^{+} + w_{2r}d_{2r}^{+}) \qquad (31)$$

The manufacturing process designer adopts the turning time and power portions of the objective function:

$$Z_2 = \sum_{r=1}^{4}(w_{3r}d_{3r}^{+} + w_{4r}d_{4r}^{+}) \qquad (32)$$

Step 5. Each designer formulates a compromise DSP. The product designer and the manufacturing process designer separately formulate compromise DSP's using Equations (31) and (32) as objective functions, respectively. In addition, they adopt the unacceptable values in Table 2 as additional constraints. The compromise DSP's are shown in Figure 7.

Product Design

Given $X^s = \{d, l\}$

Find

Deviation variables: d_{ir}^{+}, d_{ir}^{-} $i=1,2;\ r=1,\dots,4$

Satisfy

Constraints: $g_1 = 785\times10^3 d^4 - l \geq 0$ Eq. (13)

$\hat{\tau} \leq 1$ and $\hat{m} \leq 1$

Bounds: $0.01 \leq d \leq 0.05$ Eq. (14)

$0.05 \leq l \leq 0.1$ Eq. (15)

Other: $d_{ir}^{-} \cdot d_{ir}^{+} = 0$

$d_{ir}^{+} \geq 0$ and $d_{ir}^{-} \geq 0$

Minimize

The deviation function:

$$Z_1 = \sum_{r=1}^{4}(w_{1r}d_{1r}^{+} + w_{2r}d_{2r}^{+}) \qquad \text{Eq. (31)}$$

Manufacturing

Given $X^s = \{d, l\}$

Find

Non-shared variables: $x_2^{ns} = \{f, \Omega\}$

Deviation variables: d_{ir}^{+}, d_{ir}^{-} $i=3,4;\ r=1,\dots,4$

Satisfy

Constraints: $g_2 = 0.005n - (D-d) \geq 0$ Eq. (23)

$\hat{t} \leq 1$ and $\hat{P} \leq 1$

Bounds: $0.0001 \leq f \leq 0.025$ Eq. (24)

$60 \leq \Omega \leq 600$ Eq. (25)

$n \in \{1, 2, \dots\}$

Other: $d_{ir}^{-} \cdot d_{ir}^{+} = 0$

$d_{ir}^{+} \geq 0$ and $d_{ir}^{-} \geq 0$

Minimize

The deviation function:

$$Z_2 = \sum_{r=1}^{4}(w_{3r}d_{3r}^{+} + w_{4r}d_{4r}^{+}) \qquad \text{Eq. (32)}$$

Figure 7 -- Compromise DSP's for Product Design and Manufacturing

Step 6. Approximate the BRC_k's with a response surface:
Step 6.1. An experiment is designed with the set of all shared variables (identified in Step 1) as design factors of the experiment and Z_k (identified in Step 4) as the response variable. Here, a central composite experimental design is used to define 9 combinations of the variables *d* and *l*, shown in the first two columns of Table 4. In this table $\hat{d}$ and $\hat{l}$ represent normalized values of *d* and *l* between –2 and 2:

$$\hat{d} = 100d - 3 \tag{33}$$

$$\hat{l} = 80l - 6 \tag{34}$$

Step 6.2. Each compromise DSP formulated in Step 5 is solved for each combination of the shared variables in the experiment, and the resulting minimum objective function value, Z_k^*, is recorded. Note that each compromise DSP is solved independently, i.e., *in a distributed manner*. This is particularly important when the models are too complex to be solved simultaneously, when each problem is solved more effectively using different optimization techniques (e.g., linear or nonlinear programming), when it is important to utilize a designer's expertise with a model and associated analysis codes, and when enterprise or geographic boundaries prohibit widespread information exchange. The resulting values of Z_1 and Z_2 are presented in Table 4.

Table 4 -- Experimental Design for Product Design

Run	*d [m]*	*l [m]*	$\hat{d}$	$\hat{l}$	Z_1^*	Z_2^*
1	0.04825	0.08125	-1	-1	4.085	4.407
2	0.04825	0.19375	-1	1	4.486	5.952
3	0.09475	0.08125	1	-1	5.521	1.476
4	0.09475	0.19375	1	1	6.602	1.933
5	0.025	0.1375	-2	0	4.174	7.082
6	0.118	0.1375	2	0	7.812	0.000
7	0.0715	0.025	0	-2	3.826	2.480
8	0.0715	0.25	0	2	5.238	4.118
9	0.0715	0.1375	0	0	4.602	3.284

Step 6.3. A response surface is fit to the values of Z_k^* recorded in the previous step. The response surfaces are obtained with least squares regression:

$$\mathrm{BRC}_1 = Z_1^* = 4.82+0.902\hat{d}+0.359\hat{l}+0.308\hat{d}^2-0.0577\hat{l}^2+0.17\hat{d}\hat{l} \tag{35}$$

$$\mathrm{BRC}_2 = Z_2^* = 3.36-1.76\hat{d}+0.44\hat{l}+0.0492\hat{d}^2-0.0113\hat{l}^2-0.272\hat{d}\hat{l} \tag{36}$$

Step 7. A system manager (or project leader) uses the BRC's to find the shared variables that minimize the complete objective function Z. The manager solves the following compromise DSP to find the values of *d* and *l*.

Given	BRC_1 (Eq. 35) and BRC_2 (Eq. 36)
Find	$\boldsymbol{X}^s = \{d, l\}$
Satisfy	$Z_1 = BRC_1 \qquad Z_2 = BRC_2$
	Bounds: $0.01 \le d \le 0.05 \qquad 0.05 \le l \le 0.1$
Minimize	$Z = Z_1 + Z_2$

The solution to the above compromise DSP is *d* = 0.039 [m] and *l* = 0.05 [m].

Step 8. Given the values of the shared variables, $\boldsymbol{X}^s$, the designers find the values of their non-shared variables, $\boldsymbol{x}_k^{ns}$, by solving their individual compromise DSP's. Using the values of *d* and *l* from the previous step, the product designer evaluates the resulting mass and shear stress, and the manufacturing engineer solves his compromise DSP again to find the manufacturing design variables. The final values of the design variables are shown in the first row of Table 5. This solution, referred to as "*Dist,*" satisfies all constraints. . In the second row of this table is a solution (labeled "*Seq*") that is obtained by solving the design and manufacturing problems sequentially. In the third row is a solution (labeled "*Int*") that is obtained by integrating the design and manufacturing compromise DSP's and solving a single compromise DSP. The final column shows the normalized (percentage) values of *Z*. As expected for a simple problem like this, the integrated solution yields the best results (i.e., the lowest Z), followed by the distributed collaborative solution obtained with the proposed approach. The difference between the two Z values is 9%. This difference is due to modeling error introduced by approximating the *BRC'*s with 2nd order response surfaces. Observe, however, that the solution obtained by solving the problem in a non-collaborative, sequential process yields a significant increase in the value of *Z*. This loss of performance is evident because the best solution, from a systems perspective, is obtained when the product designer uses a larger diameter, d, thereby "sacrificing" the value of mass in order to reduce the machining time. Even though engineering problems are typically more complex, this simple example illustrates the potential benefits of the mathematical approach.

Table 5 – Solutions for the Collaborative Design Example

	d	l	$f\times10^{-3}$	Ω	n	$\hat{m}$	$\hat{\tau}$	$\hat{t}$	$\hat{P}$	Z	$Z\,\%$
Dist	0.039	0.05	0.55	454	4	0.315	0.085	0.32	0.261	6.49	67
Seq	0.017	0.05	0.55	571	8	0.061	0.999	0.51	0.600	9.62	100
Int	0.045	0.05	1	190	2	0.416	0.056	0.21	0.133	5.61	58

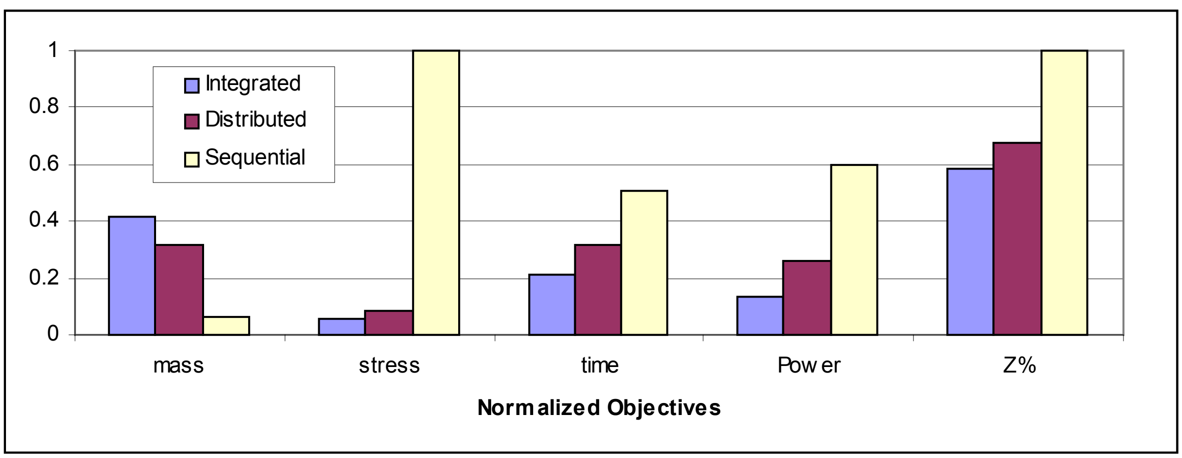

Figure 8 – Comparison of Three Solutions for the Example

5. EXTENSIONS TO THE COLLABORATIVE DESIGN METHOD

An alternative version of the preceding approach can be developed using multi-attribute utility theory. The advantages of using utility theory include its sound, axiomatic basis and the capability of employing expected utility to guide decision-making that is based on the preferences of a designer and valid under conditions of risk or uncertainty.

A modified utility-based method for collaborative design can be established by altering steps 1 through 7 in the preceding method as follows:

Step 2: According to the Arrow theorem [31], a consistent aggregation of group preferences is not theoretically feasible. Any consistent statement of preferences must be made by an individual, rather than by voting or otherwise aggregating group preferences. Thus, a well-informed team leader or project manager establishes single-attribute utility functions for each objective/goal.

Step 3: The team leader then combines the single attribute utility functions into an overall multi-attribute utility function to serve as the system-level objective function.

Step 4: Each of the designers adopts the minimization of their own part of this multi-attribute utility function as their objective function. Generally, this will imply that each designer adopts the single attribute utility function(s) pertaining to his/her part of the design. Since the preferences incorporated into the utility functions are those of a project leader or manager, individual decision-makers work with clear, quantified objectives. However, instead of vesting the project leader with all decision-making ability, each individual designer is empowered to make decisions, using his/her own expertise, to best achieve objectives for his/her portion of the overall system design.

Steps 1 and 5 through 8 remain unchanged.

Utility theory and game theory have been employed together in previous work to explore both cooperative and non-cooperative strategic situations involving two or more designers [24]. Game theory and utility theory are a promising basis for both descriptive and normative foundations for design problems with multiple designers and objectives.

6. CLOSURE

With an appropriate decision support method, engineers can realize effective collaborative design processes by formulating and making decisions in a coordinated manner. The proposed method is based on establishing a common and domain-independent decision model, expanding the scope of local decisions by adopting relevant system performance measures, and implementing a mathematical coordination mechanism based on techniques developed in game theory. There are several advantages of the proposed approach:

- The method is mathematical, and it is based on solid theoretic grounds.
- Designers participate in the decision process, utilizing their knowledge, expertise and skills. The design process remains decentralized.
- The resulting solution is Pareto efficient from a systems perspective.
- Information transfer and computational requirements are minimized. All of the analysis codes and expertise are *not* consolidated into a single overall, computationally expensive model of the system.

The last advantage is particularly attractive if collaboration is to be achieved between designers from different companies working in a common development project, because security and confidentiality could restrict sharing of computational or analysis codes or complete models. Sharing these models is the basis for other collaborative design methods. Our current research is focused on developing computational infrastructure to support this approach and including uncertainty in the process using utility theory

ACKNOWLEDGMENTS

During his doctoral study, Gabriel Hernandez was sponsored by the National Council of Science and Technology of Mexico (CONACYT). Carolyn Conner Seepersad was the recipient of a National Science Foundation Graduate Research Fellowship and is sponsored currently by the Fannie and John Hertz Foundation. Financial support from NSF grant DMI 0085136 is gratefully acknowledged. The cost of computer time was underwritten by the Systems Realization Laboratory of the Georgia Institute of Technology.

REFERENCES

1. Lizotte, S. and B. Chaib-Draa, 1997, "Coordination in CE Systems: An Approach Based on the Management of Dependencies between Activities," *Concurrent Engineering: Research and Applications,* Vol. 5, No. 4, pp. 367-376.
2. Wong, A. and D. Sriram, 1993, "SHARED: An Information Model for Cooperative Product Development," *Research in Engineering Design,* Vol. 5, No. 1, pp. 21-39.
3. Maher, M. L. and J. H. Rutherford, 1997, "A Model for Synchronous Collaborative Design Using CAD and Database Management," *Research in Engineering Design,* Vol. 9, No. 2, pp. 85-98.
4. Bond, A. H. and R. J. Ricci, 1992, "Cooperation in Aircraft Design," *Research in Engineering Design,* Vol. 4, No. 2, pp. 115-130.
5. Crabtree, R. A., M. S. Fox and N. K. Baid, 1997, "Case Studies of Coordination Activities and Problems in Collaborative Design," *Research in Engineering Design,* Vol. 9, No. 2, pp. 70-84.
6. Eppinger, S. D., M. V. Nukala and D. E. Whitney, 1997, "Generalizing Models of Design Iteration Using Signal Flow Graphs," *Research in Engineering Design,* Vol. 9, No. 2, pp. 112-123.
7. Park, H. and M. R. Cutkosky, 1999, "A Framework for Modeling Dependencies in Collaborative Engineering Processes," *Research in Engineering Design,* Vol. 11, No. 2, pp. 84-102.
8. Wang, K. and Y. Jin, 2000, "Managing Dependences for Collaborative Design," *ASME Design Theory and Methodology*, Baltimore, MD. Paper Number DETC2000/DTM-14552.
9. Peplinski, J. D. and F. Mistree, 1997, "A Decision-Based Approach to Enterprise Design and Integration," *Advances in Concurrent Engineering* (S. Ganesan, Ed.), Technomic Publishing, Rochester, MI, pp. 65-74.
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. Liles, D. H. and A. R. Presley, 1992, "Enterprise Modeling within an Enterprise Engineering Framework," *Winter Simulation Conference*, Coronado, CA, IEEE.
12. Kateel, G., M. Kamath and D. Pratt, 1996, "An Overview of CIM Enterprise Modeling Methodologies," *Winter Simulation Conference*, Coronado, CA, IEEE.
13. Browning, T. R., 1997, "Exploring Integrative Mechanisms with a View Towards Design for Integration," *Advances in Concurrent Engineering* (S. Ganesan, Ed.), Technomic Publishing Co., pp. 83-89.
14. Vincent, T. L., 1983, "Game Theory as a Design Tool," *ASME Journal of Mechanisms, Transmissions and Automation in Design,* Vol. 105, No. 2, pp. 165-170.

15. Badhrinath, K. and J. R. J. Rao, 1996, "Modeling for Concurrent Design Using Game Theory Formulations," *Concurrent Engineering: Research and Applications,* Vol. 4, No. 4, pp. 389-399.
16. Hazelrigg, G., 1996, *Systems Engineering: An Approach to Information-Based Design,* Prentice Hall, Upper Saddle River, NJ.
17. Lewis, K. and F. Mistree, 1997, "Modeling Interaction in Multidisciplinary Design: A Game Theoretic Approach," *AIAA Journal,* Vol. 35, No. 8, pp. 1387-1392.
18. Lewis, K. and F. Mistree, 1998, "Collaborative, Sequential and Isolated Decisions in Design," *ASME Journal of Mechanical Design,* Vol. 120, No. 4, pp. 643-652.
19. Lewis, K. and F. Mistree, 1998, "The Other Side of Multidisciplinary Design Optimization: Accommodating a Multiobjective, Uncertain, and Non-Deterministic World," *Engineering Optimization,* Vol. 31, No. 2, pp. 161-189.
20. Cowan, F. S., M. Marston and F. Mistree, 2000, "Design for Excellence: A Microworld, Game-Theoretic Approach," *Engineering Design Conference 2000: Design for Excellence*, (Sivaloganathan, S., and Andrews, P.T.J., eds.), Brunel University, UK, Professional Engineering Publishing Limited, Bury, St. Edmunds, UK.
21. Hernandez, G., C. Conner Seepersad and F. Mistree, 2000, "Commonalizing Subsystem Components to Facilitate Maintenance: A Game Theoretic Approach," *AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA. Paper Number AIAA 2000-4806.
22. Hernandez, G. and F. Mistree, 2000, "Integrating Product Design and Manufacturing: A Game Theoretic Approach," *Engineering Optimization,* Vol. 32, No. 6, pp. 749-775.
23. 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.
24. Marston, M. and F. Mistree, 2000, "Game-Based Design: A Game Theoretic Extension to Decision-Based Design," *ASME Design Theory and Methodology Conference*, Baltimore, MD. Paper Number DETC2000/DTM-14578.
25. Steinhour, W. and K. Krishnamurthy, 2001, "A Game Theoretic Approach to Integrated Product Design," *ASME Advances in Design Automation*, Pittsburgh, PA. Paper Number DETC2001/DAC-21086.
26. Chen, L. and S. Li, 2002, "A Computerized Team Approach for Concurrent Product and Process Design Optimization," *Computer-Aided Design,* Vol. 34, No. 1, pp. 57-69.
27. Kreps, D. M., 1990, *A Course in Microeconomic Theory,* Princeton University Press, Princeton, N.J.
28. Myerson, R. B., 1991, *Game Theory: Analysis of Conflict,* Harvard University Press, Cambridge, MA.
29. Messac, A., S. M. Gupta and B. Akbulut, 1996, "Linear Physical Programming; A New Approach to Multiple Objective Optimization," *Transactions in Operations Research,* Vol. 8, pp. 39-59.
30. 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.
31. Arrow, K. J., 1951, *Social Choice and Individual Values,* John Wiley and Sons, New York.

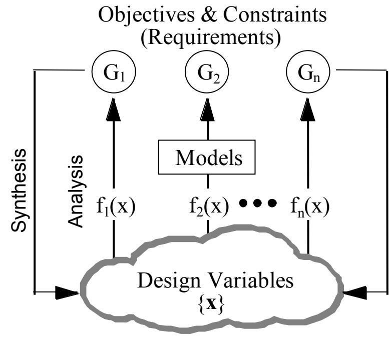

Figure 1 --A Model for Decision Support in Engineering Design [9]

Given n, number of design variables
p, number of equality constraints
q, number of inequality constraints
m, number of objectives
$f_i(\boldsymbol{x})$, goal achievement function
$g_r(\boldsymbol{x})$, constraint function
G_i, target values for the objective functions
$h_k(d_i)$, function to be minimized at priority level k for preemptive form
W_i, weight for Archimedean form

Find System variables: $\boldsymbol{x} = \{x_1, \ldots, x_n\}$

Deviation variables: d_i^-, d_i^+ $\quad i=1,\ldots,m$

Satisfy Goals: $f_i(\boldsymbol{x}) + d_i^- - d_i^+ = G_i$ $\quad i=1,\ldots,m$

Constraints: $h_r(\boldsymbol{x})=0$ $\quad r=1,\ldots,p$

$g_r(\boldsymbol{x}) \geq 0$ $\quad r= 1,\ldots,q$

Bounds $x_{j,\min} \leq x_j \leq x_{j,\max}$ $\quad j=1,..,n$

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

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

Minimize A deviation function:

Preemptive: $Z = [f_1(d_i^-, d_i^+), \ldots, f_k(d_i^-, d_i^+)]$

Weighted Sum: $Z = \sum_{i=1}^{m} w_i (d_i^- + d_i^+)$

Figure 2 --Mathematical Form of the Compromise DSP [10]

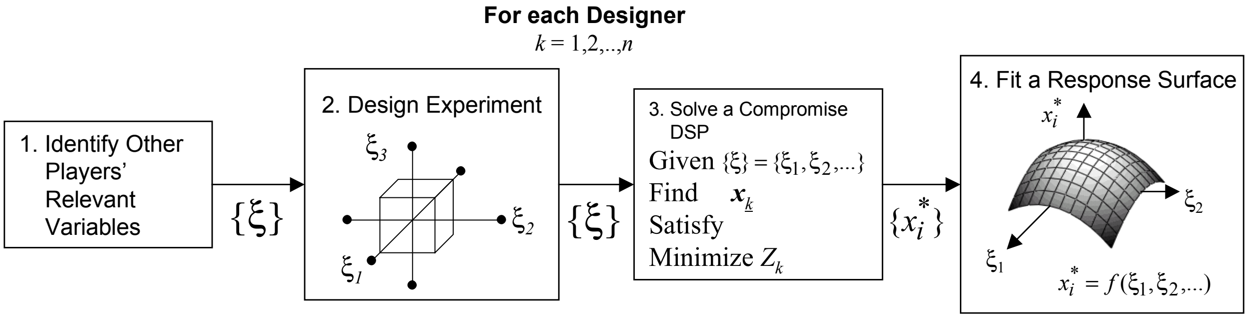

Figure 3 --Developing an Approximation for the *BRC* (adapted from [18])

For Every Designer *k (Steps 6 and 8)*

Given The value of shared variables, $\boldsymbol{X}^s$

Find The value of non-shared variables, $\boldsymbol{x}_k^{ns}$

The value of the deviation variables, $d_{ki,r}^{+}, d_{ki,r}^{-}$

Satisfy Design Constraints

Design Bounds, $x_{kj,\,min} \le x_{kj} \le x_{kj,max},$

Other: $d_{ki,r}^{-}, d_{ki,r}^{+} \ge 0$

$d_{ki,r}^{-} \cdot d_{ki,r}^{+} = 0$

Minimize $Z_k = \sum_{i=1}^{m} \sum_{r=1}^{4} (w_{ki,r}^{-} d_{ki,r}^{-} + w_{ki,r}^{+} d_{ki,r}^{+})$

For the System Level Manager to Find the Set of Shared Variables *(Step 7)*

Given N, the number of designers

BRC_k a model of the best reply correspondence of each designer, $k=1,\ldots,N$

$$BRC_k = Z_k^*(\boldsymbol{X}^s)$$

Find The set of *shared* variables, $\boldsymbol{X}^s = \{\boldsymbol{x}_1^s, \boldsymbol{x}_2^s, \ldots, \boldsymbol{x}_N^s\}$

Satisfy $Z_k(\boldsymbol{X}^s) = BRC_k \quad k=1,\ldots,N$

Minimize $Z = \sum_{i=1}^{N} Z_k(\boldsymbol{X}^s)$

Figure 4 -- Mathematical Formulations for Collaborative Distributed Design

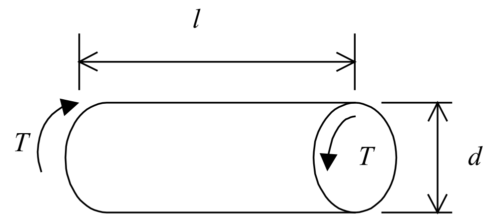

Figure 5 -- A Bar in Torsion

Product Design

Given

Find $d,\ l$

Satisfy $g_1 = 785\times10^3 d^4 - l \geq 0$ Eq. (13)

$0.01 \leq d \leq 0.05$ Eq. (14)

$0.05 \leq l \leq 0.1$ Eq. (15)

Minimize

$m = 6157 d^2 l$ Eq. (10)

$\tau = \dfrac{510}{d^3}$ Eq. (11)

Manufacturing

Given $d,\ l$

Find $f,\ \Omega,\ n$

Satisfy $g_2 = 0.005n - (0.055 - d) \geq 0$ Eq. (23)

$0.0001 \leq f \leq 0.025$ Eq. (24)

$60 \leq \Omega \leq 600$ Eq. (25)

$n \in \{1,\ 2, \ldots\}$

Minimize $t = \dfrac{2nl}{f\ \Omega}$ Eq. (16)

$P = 210\times10^3 \left(\dfrac{d + 0.055}{2}\right)\cdot(0.055 - d)\cdot f \cdot \Omega$ Eq. (19)

Figure 6 -- Baseline Decisions for Example Problem

Product Design	**Manufacturing**
Given $\boldsymbol{X}^s = \{d, l\}$	**Given** $\boldsymbol{X}^s = \{d, l\}$
Find	**Find**
Deviation variables: d_{ir}^{+}, d_{ir}^{-} $i=1,2;\ r=1,\dots,4$	Non-shared variables: $\boldsymbol{x}_2^{ns} = \{f, \Omega\}$
	Deviation variables: d_{ir}^{+}, d_{ir}^{-} $i=3,4;\ r=1,\dots,4$
Satisfy	**Satisfy**
Constraints: $g_1 = 785\times10^3 d^4 - l \geq 0$ Eq. (13)	Constraints: $g_2 = 0.005n - (D-d) \geq 0$ Eq. (23)
$\hat{\tau} \leq 1$ and $\hat{m} \leq 1$	$\hat{t} \leq 1$ and $\hat{P} \leq 1$
Bounds: $0.01 \leq d \leq 0.05$ Eq. (14)	Bounds: $0.0001 \leq f \leq 0.025$ Eq. (24)
$0.05 \leq l \leq 0.1$ Eq. (15)	$60 \leq \Omega \leq 600$ Eq. (25)
	$n \in \{1,\ 2, \dots\}$
Other: $d_{ir}^{-} \cdot d_{ir}^{+} = 0$	Other: $d_{ir}^{-} \cdot d_{ir}^{+} = 0$
$d_{ir}^{+} \geq 0$ and $d_{ir}^{-} \geq 0$	$d_{ir}^{+} \geq 0$ and $d_{ir}^{-} \geq 0$
Minimize	**Minimize**
The deviation function:	The deviation function:
$Z_1 = \sum_{r=1}^{4}(w_{1r}d_{1r}^{+} + w_{2r}d_{2r}^{+})$ Eq. (31)	$Z_2 = \sum_{r=1}^{4}(w_{3r}d_{3r}^{+} + w_{4r}d_{4r}^{+})$ Eq. (32)

Figure 7 -- Compromise DSP's for Product Design and Manufacturing

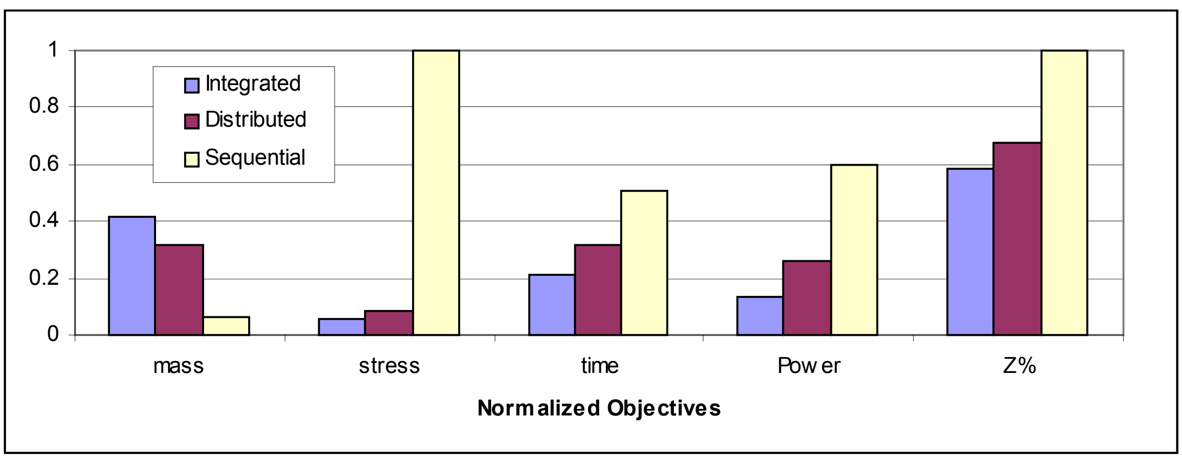

Figure 8 – Comparison of Three Solutions

Table 1 – Physical Programming Desirability Levels for Mass, Shear Stress, Turning Time and Power

	Target r				
	1 Ideal	2 Desirable	3 Tolerable	4 Undesirable	5 Unacceptable
m [kg]	0.05	0.15	0.35	0.75	1.5
τ [MPa]	35	50	60	75	100
t [min]	1	1.5	2.25	3.25	5
P [kW]	20	30	50	90	150

Table 2 -- Desirability Levels for Normalized Mass, Shear Stress, Turning Time, and Power

	Target r				
	1 Ideal	2 Desirable	3 Tolerable	4 Undesirable	5 Unacceptable
$\hat{m}$	0.033	0.1	0.233	0.5	1
$\hat{\tau}$	0.35	0.5	0.6	0.75	1
$\hat{t}$	0.2	0.3	0.45	0.65	1
$\hat{P}$	0.133	0.2	0.333	0.6	1

Table 3 -- Weights for Deviation Function

	r			
	1	2	3	4
w_{1r}	1.50	0.98	1.61	3.10
w_{2r}	0.67	2.63	3.96	7.11
w_{3r}	1.00	1.20	3.25	4.82
w_{4r}	1.50	0.98	1.61	4.90

Table 4 -- Experimental Design for Product Design

Run	d [m]	l [m]	$\hat{d}$	$\hat{l}$	Z_1^*	Z_2^*
1	0.04825	0.08125	-1	-1	4.085	4.407
2	0.04825	0.19375	-1	1	4.486	5.952
3	0.09475	0.08125	1	-1	5.521	1.476
4	0.09475	0.19375	1	1	6.602	1.933
5	0.025	0.1375	-2	0	4.174	7.082
6	0.118	0.1375	2	0	7.812	0.000
7	0.0715	0.025	0	-2	3.826	2.480
8	0.0715	0.25	0	2	5.238	4.118
9	0.0715	0.1375	0	0	4.602	3.284

Table 5 – Solution to the Collaborative Design Problem

	d	l	$f \times 10^{-3}$	Ω	n	$\hat{m}$	$\hat{\tau}$	$\hat{t}$	$\hat{P}$	Z	$Z\%$
Dist	0.039	0.05	0.55	454	4	0.315	0.085	0.32	0.261	6.49	67
Seq	0.017	0.05	0.55	571	8	0.061	0.999	0.51	0.600	9.62	100
Int	0.045	0.05	1	190	2	0.416	0.056	0.21	0.133	5.61	58