An Interval-based Constraint Satisfaction (IBCS) Method for Decentralized, Collaborative Multifunctional Design

Jitesh H. Panchal, Marco Gero Fernández, Christiaan J. J. Paredis, Janet K. Allen and Farrokh Mistree

Concurrent Engineering 2007; 15; 309
DOI: 10.1177/1063293X07083083

The online version of this article can be found at:
http://cer.sagepub.com/cgi/content/abstract/15/3/309
An Interval-based Constraint Satisfaction (IBCS) Method for Decentralized, Collaborative Multifunctional Design

Jitesh H. Panchal, Marco Gero Fernández, Christiaan J. J. Paredis, Janet K. Allen and Farrokh Mistree*

Systems Realization Laboratory, Woodruff School of Mechanical Engineering
Georgia Institute of Technology, Atlanta, GA 30322, USA

Abstract: Set-based design has been proposed as a strategy for multifunctional design problems where stakeholders from different disciplines strive to achieve domain-specific objectives while sharing a set of design variables. This strategy involves communicating information about sets of alternatives in contrast to communicating information about a single alternative at a time. The strategy has been developed for collaborative CAD and for selection among design alternatives during conceptual design, it has not been implemented as a computational method for decentralized collaborative multi-objective design problems. In this article, we address this research gap by presenting an Interval-Based Constraint Satisfaction (IBCS) Method for decentralized, collaborative multifunctional design. The method is based on transforming a decentralized multifunctional design problem into a constraint satisfaction problem by using non-cooperative game theoretic protocols. The resulting constraint satisfaction problem is then solved using interval-based consistency techniques. A non-cooperative game theory protocol is utilized in this method because it reflects the level of information exchange possible in a distributed environment. Central to this protocol is the representation of a Rational Reaction Set (RRS) that encapsulates a designer’s decision-making strategy as a constraint in the design space. An intersection of all designers’ RRSs represents a solution to the overall multifunctional design problem. We use interval-based consistency techniques, specifically box consistency, to sequentially eliminate regions of design space that do not satisfy the individual RRSs, thereby progressively narrowing the design space in order to reduce computational complexity in arriving at a solution. This method stands in marked contrast to the successive consideration of single solution points, as emphasized in existing multifunctional design methods. The key advantages of the proposed method are: (a) gradual reduction of design freedom and (b) non-divergence of solutions. The method is illustrated using two sample scenarios – the solution of a decision problem with quadratic objectives and the design of multifunctional Linear Cellular Alloys (LCAs).

Key Words: decentralized design, game theory, intervals, constraint satisfaction, consistency.

1. Frame of Reference – Collaborative Decision-making Approaches in Design

The key challenge in concurrent engineering is the coordination between the different stakeholders involved in complex product development projects [1,2]. Depending on the nature of the underlying design process, there are two strategies that are commonly employed for effectively synthesizing contributions of interacting designers; these are based on centralized and decentralized decision-making [3–7]. Centralized decision-making requires a single transfer of knowledge from various domain experts to a single decision-maker. It is based on gathering and consolidating information and facilitates the attainment of Pareto-optimal solutions via simultaneous consideration of system level tradeoffs. Hayek [8], on the other hand, advocates decentralized decision-making, pointing out that it is important to delegate responsibility to persons “on the spot” who have intimate knowledge of their respective domains and are (consequently) capable of making any required inferences. Lee and Whang [4] present decentralized decision-making methods in the context of supply chains, whereas Chanron and coauthors [3] offer a decentralized decision-making strategy for the solution of engineering design problems using game theoretic protocols [9]. Androulakis and Reklaitis [5] and Wheeler and Narendra [6] have also developed methods for decentralized decision-making. Decentralized decisions are generally coupled by nature and require a two-way flow of information between decision-makers as well as active involvement of domain experts throughout the decision-making process. In multifunctional design, different designers and domain experts control a common set of design variables and share responsibility for achieving different objectives. It is in this context that different strategies for solving coupled problems are discussed next.

Consider the scenario shown in Figure 1, where Designers A and B are responsible for achieving their
respective design objectives. These objectives are defined in terms of the maximization, minimization, or matching of response variables $Y$. In the given scenario, Designer A controls a set of design variables $X_A$ while Designer B controls a set of design variables $X_B$. This is a general design scenario used to illustrate the concepts. A specific design example, representative of such a scenario, is discussed in section 4. Since the two decisions are coupled, Designer A cannot make a decision about $X_A$ unless the values of $X_B$ are known. Similarly, Designer B cannot make a determination with regard to $X_B$ without knowing the values of $X_A$. Two strategies for solving such a coupled, decentralized problem are point-based iteration and set-based gradual reduction of design space.

In point-based iteration, interacting designers consider a single point within a given design space at a time and iteratively adjust this point until they converge on a solution that satisfies their respective design objectives (each are functions of response variables). Procedurally, one of the designers (say Designer A in the scenario depicted in Figure 1) starts by assuming values of design variables controlled by the other designer ($X_B$) and determines values for his/her design variables ($X_A$) that satisfy his/her objectives ($Y_A$). Using these values of design variables ($X_A$), Designer B can then determine suitable values for his/her design variables ($X_B$) considering his/her own objectives ($Y_B$). This process continues until converging to a single point in the design space ($X_A, X_B$).

One of the primary limitations of point-based methods is that the resulting design processes may be unstable and the results may never converge [7]. Yassine and Braha [10] highlight that convergence is one of the four key challenges associated with successful implementation of concurrent engineering (the remaining three challenges include iteration, overlapping, and decomposition and integration). Klein and coauthors [11] discuss the impact of strong interdependencies between design decisions, making the convergence to a single design solution difficult. Similarly, Ford and Sterman [12] point out that the primary difficulty in concurrent engineering is due to unanticipated iterations. Chanron and coauthors [3,7,13] investigate the underlying dynamics of decentralized processes and corresponding convergence as well as stability criteria using numerical series and linear algebra. The second disadvantage is that design freedom [14] is reduced from the initial ranges of design variables to point values in a single step. Design freedom, here, is defined as the extent to which a system can be adjusted while still meeting the design requirements posed for it [14]. This property severely limits designers in accommodating any changes. In contrast, a more gradual and systematic reduction of the design space and the associated design freedom reduces the premature, unnecessary elimination of potential solutions.

To manage design freedom effectively throughout the design process, a number of researchers have proposed the use of set-based design approaches [15–22]. Instead of communicating information about a single point in the design space at a time, set-based design approaches advocate the transfer of feasible ranges of values for given design variables. The primary reasons for using such set-based design approaches are (1) the communication of sets of possibilities and (2) the subsequent narrowing of these sets, balancing the need to gain more knowledge and progressively reduce uncertainty [18]. The key advantage of such an interaction mechanism is that design freedom remains open for a longer period of time, thereby accommodating changes in the requirements during the execution of the design process and maintaining the autonomy of experts over their respective domains. The reduction in design freedom for point-based methods occurs in a single step, whereas it is more gradual in set-based methods.

An overview of the existing research efforts related to set-based design is presented in Table 1 and discussed next. Liker and coauthors [17] argue the differences between point-based and set-based design approaches. Using product development processes in the United States and Japan, the authors show that set-based approaches facilitate effective communication between stakeholders and reduce rework. Sobek and Ward [18] discuss the general principles of set-based concurrent engineering and illustrate the advantages through examples of design processes adopted at Toyota Motor Corporation. Considering the importance of set-based approaches in collaborative design, there is a need for developing formal computational design methods based on the set-based approach. Recently, various methods have been proposed to implement set-based design using interval constraint solver techniques [23]. For example, Finch et al. [21] use a set-based approach to solve catalog selection problems under uncertainty. Design catalogs are represented as sets of constraints;
the catalog selection problem is addressed as a constraint satisfaction problem. Furthermore, elimination strategies based on arc consistency [24] are relied upon to reduce the design space to a space of feasible alternatives. Another approach for using intervals in selection decisions is presented by Reddy and Mistree [25]. Telerman and coauthors [19] utilize the interval constraint solver approach for distributed collaborative CAD. Nahm and Ishikawa [26] combine set-based approaches with parametric solid modeling techniques to develop a 3D-CAD system for set-based parametric design.

In summary, formal computational implementations of set-based design approaches have been developed using interval constraint solvers for catalog selection problems, and parametric CAD. However, computational methods implementing set-based approaches have not been developed for decentralized multifunctional design problems of the type illustrated in Figure 1 where designers in charge of different functional requirements share a common set of design variables. In this article, we present a method for implementing the set-based design approach for decentralized multifunctional design problems. The method involves (a) transformation of a decentralized multifunctional design problem into an interval constraint solving problem using non-cooperative game theoretic protocols [9,27], and (b) solving the interval constraint solving problem using consistency techniques [28,29]. An overview of non-cooperative game theoretic protocols for modeling interactions between designers is provided in Section 2. Arc and Box Consistency – mathematical tools emanating from Interval Arithmetic – which serve as foundations for the proposed Interval-based Constraint Satisfaction (IBCS) method are also presented. We proceed to outline our method for multifunctional design in Section 3 and illustrate this method with a nonlinear, multifunctional design example in Section 4. Finally, we provide a discussion with regard to current limitations of the proposed method and future opportunities for extension in Section 5.

2. Theoretical Constructs Used in this Article

There are a number of different mechanisms commonly employed for decentralized decision-making in multifunctional design problems. These include applications of multi-disciplinary optimization approaches (e.g., [30]), negotiations (e.g., [31–33]), and game theoretic principles (e.g., [9, 27, 34, 35]). Since game theory has been formalized for both centralized and decentralized decision-making, we build on the underlying protocols to develop the IBCS method. An overview of game theory as applied within the field of engineering design is provided in Section 2.1, with an emphasis on the non-cooperative formulation that appropriately represents coupled, decentralized decision-making. In order to systematically reduce the design space for this problem formulation, we rely on Arc and Box Consistency, two mathematical constructs developed within the area of interval arithmetic. A detailed discussion of Arc and Box Consistency follows in Section 2.2.

2.1 Game Theory Protocols for Collaborative Design

Game theory has been employed as a means of conflict resolution in engineering design, with instantiations varying depending on the nature of the underlying problem addressed. Myerson [36], for example, presents game theory as a method for resolving conflict between multiple decision-makers controlling subsets of design variables and striving to minimize individual cost functions. Other researchers have extended game theory adaptations to solve multi-objective problems [37], for structural optimization [34] and for the integrated design of control structure [38]. Hacker and Lewis [39] develop a robust design method to reduce uncertainty in a non-cooperative system that results from predictions of disciplinary subsystem behavior. Subsequently, Kalsi, Hacker, and Lewis [40] proceed to build upon this framework by solving disciplinary sub-problems

<table>
<thead>
<tr>
<th>Cycle #</th>
<th>Range for thickness (f) (μm)</th>
<th>Range for height (H) (mm)</th>
<th>Range of achievable heat transfer rates (W)</th>
<th>Range of achievable compliance (kJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[4.500 6.500]</td>
<td>[10.00 18.50]</td>
<td>[−6.71 −4.69]</td>
<td>[3.87 5.62]</td>
</tr>
<tr>
<td>1</td>
<td>[5.489 6.185]</td>
<td>[10.25 18.74]</td>
<td>[−6.52 −5.17]</td>
<td>[4.05 4.90]</td>
</tr>
<tr>
<td>2</td>
<td>[5.795 6.046]</td>
<td>[15.43 20.54]</td>
<td>[−5.90 −5.44]</td>
<td>[4.33 4.62]</td>
</tr>
<tr>
<td>3</td>
<td>[5.906 5.995]</td>
<td>[17.51 19.36]</td>
<td>[−5.71 −5.55]</td>
<td>[4.42 4.52]</td>
</tr>
<tr>
<td>4</td>
<td>[5.945 5.977]</td>
<td>[18.30 18.96]</td>
<td>[−5.65 −5.59]</td>
<td>[4.45 4.49]</td>
</tr>
<tr>
<td>5</td>
<td>[5.959 5.970]</td>
<td>[18.58 18.82]</td>
<td>[−5.63 −5.61]</td>
<td>[4.47 4.48]</td>
</tr>
<tr>
<td>6</td>
<td>[5.964 5.968]</td>
<td>[18.69 18.77]</td>
<td>[−5.62 −5.61]</td>
<td>[4.47 4.47]</td>
</tr>
<tr>
<td>7</td>
<td>[5.965 5.967]</td>
<td>[18.72 18.75]</td>
<td>[−5.62 −5.61]</td>
<td>[4.47 4.47]</td>
</tr>
<tr>
<td>8</td>
<td>[5.966 5.967]</td>
<td>[18.74 18.75]</td>
<td>[−5.61 −5.61]</td>
<td>[4.47 4.47]</td>
</tr>
<tr>
<td>9</td>
<td>[5.966 5.967]</td>
<td>[18.74 18.74]</td>
<td>[−5.61 −5.61]</td>
<td>[4.47 4.47]</td>
</tr>
</tbody>
</table>
independently from the rest of the system through the incorporation of ranges. Hernández and coauthors [41, 42] implement game theoretic principles to establish a mathematically supported cooperative framework that enhances the practical, effective, and efficient integration of the enterprise design process. Chen and Li [43] utilize game theory to model concurrent multifunctional product design. Marston and coauthors [27, 44] develop a multi-designer model of engineering design that accounts for uncertainty, cooperation, noncooperation, and coalitions, using the mathematics of decision and game theory. In doing so, he introduces the notion of the set of mathematically complete principles of rational behavior for designers in any design scenario.” [45]

Lewis and Mistree [9] abstract the mathematical foundations of game theory to model complex design processes. They model the strategic relationships among designers sharing a common design space using game theoretic principles and identify Pareto Cooperation, Stackelberg Leader/Follower, and Nash Non-Cooperation as the three game theoretic protocols most representative of the interactions required for decentralized design. Pareto Cooperation is employed to represent centralized decision-making, where all required information is available to every collaborating designer. A Pareto optimal solution is achieved when no single designer can improve his or her performance without negatively affecting that of another. Stackelberg Leader/Follower protocols are implemented to model sequential decision-making processes where the “leader” makes his or her decision, based on the assumption that the “follower” will behave rationally. The follower then makes his or her decision within the constraints emanating from the leader’s choice.

Nash Non-Cooperation refers to decentralized decision processes where designers have to make decisions in isolation due to organizational barriers, time schedules, and geographical constraints. Its mathematical models are suitable for formulating decisions in collaborative design [42]. In this process, decision-makers formulate Rational Reaction Sets (RRS) or Best Reply Correspondences (BRC). A RRS is a mapping (either a mathematical or a fitted function) that relates the values of design variables under a designer’s control to values of design variables controlled by other stakeholders. For example, in a two designer scenario where the first designer controls design variable set \( X_A \) and the second designer controls variable set \( X_B \), the RRS of the first designer is given by \( (X_A)_{\text{RRS}} = f_1(X_B) \) and the RRS of the second designer is given by \( (X_B)_{\text{RRS}} = f_2(X_A) \). The Nash Non-Cooperative solution to the coupled, decentralized decision-making problem is the point of intersection of the RRSs pertaining to the different designers. The resulting Nash equilibrium to the design problem has the characteristic that no designer can unilaterally improve his/her objective function [46]. Assuming two objective functions \( U_A \) and \( U_B \) to be maximized by designers A and B, respectively, a solution \( (X_{AN}, X_{BN}) \) is a Nash solution if \( U_A(X_{AN}, X_{BN}) = \max(X_{AN}, X_{BN}) \) and \( U_B(X_{AN}, X_{BN}) = \max(X_{AN}, X_{BN}) \). The Nash equilibrium thus ensures that each decision-maker’s strategy constitutes an optimal response to the other decision-makers’ strategies.

The approach commonly adopted for solving Nash Non-Cooperative decision-making problems is explicitly calculating the various RRSs and then finding their intersection. In order to calculate the RRS explicitly, a designer assumes the set of values for design variables not within their control and chooses values of his/her own design variables in order to maximize his/her own payoff. Since the evaluation of the RRS is a computationally expensive process, the function is evaluated at discrete points and a response surface model (or similar approximation technique) is employed to derive an explicit functional form of the RRSs. Each RRS is essentially an equation in terms of the design variables. Hence, a set of all designers’ RRSs can be viewed as a set of constraints in the design space. The evaluation of the intersections of RRSs to find the Nash equilibrium is viewed as a constraint solving problem.

The point-based method for solving Nash Non-Cooperative design problems involves making decisions iteratively; one designer begins by assuming values for the other designers’ design variables and makes a decision about his/her own design variables. Other designers use these values in an iterative fashion and determine the values for design variables under their control. This process continues until the solution converges to the Nash equilibrium. As discussed in Section 1, convergence and stability of a point-based method are not guaranteed. In order to overcome these shortcomings of a point-based iteration method, we leverage the interval-based constraint solution methods to determine the Nash equilibrium. Instead of considering a single point in the design space, each designer starts with an interval of design variables and reduces this interval by applying his/her design considerations (RRS)\(^1\) as a set of constraints. Design space reduction is then based on interval-based constraint solution techniques, specifically Arc and Box consistency, the details of which are discussed in Section 2.2.

### 2.2 Arc and Box Consistency

Arc consistency and Box consistency are concepts stemming from interval arithmetic that are focused on checking the consistency of each equation (constraint)
in a set of equations (constraints) in order to eliminate regions in the variable space that do not contain the solution [29,47,48]. In order to understand the concept of Arc consistency, consider an equation of the form \( f(x, y) = 0 \) such that \( x \in X \) and \( y \in Y \), where \( X \) and \( Y \) are sets of values that \( x \) and \( y \) can take. The values of \( x \) and \( y \) are consistent relative to the function \( f \), if for all values of \( x \) in \( X \), there exists \( y \) in \( Y \), and for all values of \( y \) in \( Y \), there exists \( x \) in \( X \), such that the equation \( f(x, y) = 0 \) is satisfied (see ref. [29,47] for a more detailed explanation). This statement can be mathematically represented (where symbols retain their mathematical meaning) as: \( \forall x \in X, \exists y \in Y \) and \( \forall y \in Y, \exists x \in X : f(x, y) = 0 \). The arc consistency principle for a set of equations is illustrated in Figure 2 using two linear equations, \( f_1(x, y) = 0 \) and \( f_2(x, y) = 0 \). The values of \( x \in X \) are consistent with values of \( y \in Y \) with respect to function \( f_1 \) in the figure. Similarly, values of \( x \in X \) are consistent with values of \( y \in Y' \).

In general, Arc consistency is difficult to compute for non-linear constraints [24,49]. Hence the notion of Box consistency is introduced as an approximation of arc consistency where the equations (constraints) are replaced by their interval extensions and the variables are replaced by corresponding intervals. In this context, a variable space is bounded by the lower and upper bounds of the design variables and is referred to as a box. This box is one dimensional if only one design variable is considered, and is multi-dimensional if multiple design variables are considered. It is important to note that if a box represented by the intervals \( X \) and \( Y \) is the solution to the set of equations \( f_1(x, y) = 0 \) and \( f_2(x, y) = 0 \), then the values \( x \in X \) and \( y \in Y \) must be box-consistent with respect to the interval extensions of both functions \( f_1 \) and \( f_2 \). For the set of linear equations shown in Figure 2, the interval that is box-consistent with respect to both functions is thus a single point, specifically the intersection of the two lines making up the system. The same idea is applicable not just to linear functions but to any type of nonlinear function. In order to find the box that is consistent with interval extensions of both \( f_1 \) and \( f_2 \), a sufficiently large box is chosen and its size is reduced systematically by considering one function at a time until Box consistency is achieved for all of the functions considered. Assuming that for a subset \( X_s \) within the interval \( X \), there are no corresponding values in the interval \( Y \) that satisfy the consistency condition, the subset \( X_s \) can be excluded because it does not contain the solution.

Box consistency is used in the proposed method to systematically eliminate regions of design space that do not contain the design solution. We implement the construct of box consistency to successively eliminate those areas of a given design space that do not contain the Nash equilibrium of the system. Box consistency constitutes a systematic means of reducing a shared design space that lends itself to turn-based decision-making, where each designer sequentially and progressively eliminates unacceptable regions of the design space. Since Box consistency also allows us to embody the propagation of ranged sets of specifications among interacting stakeholders, it is quite suitable as a solution algorithm for coupled, decentralized multifunctional decision-making. This strategy of systematic reduction of box size is incorporated into the IBCS method for decentralized design-making presented in this article and forms the basis for the associated systematic reduction of design freedom. This method is presented in detail in Section 3.

### 3. An Interval-Based Constraint Satisfaction (IBCS) Method for Decentralized Multifunctional Design

In the proposed IBCS method, designers start with a design space defined by ranges for each design variable as specified by the domain experts which are assigned control over the specific domains. The interacting decision-makers subsequently proceed to take turns in making decisions about their respective decision variables, thereby progressively eliminating portions of the intervals that do not contain solution acceptable to them. Each designer sequentially reduces the shared design space and passes relevant information regarding the remainder to other designers. This process is continued until either a sufficient degree of convergence is achieved or all design objectives can be satisfied successfully. The steps of the proposed method are listed in Figure 3 and discussed in detail as follows. Each step is explained using a running example where two designers (A and B) are responsible for optimizing responses \( Y_A \) and \( Y_B \), respectively.
design decisions, we represent designers’ decisions in the form of compromise Decision Support Problems (cDSPs). The cDSP is a multi-objective decision model that constitutes a hybrid formulation based on mathematical programming and goal programming [50]. In Figure 4, we present a scenario, where designers A and B formulate their decisions in the form of individual cDSPs.

The structure of the cDSP consists of four key components: “Given”, “Find”, “Satisfy”, and “Minimize”. The “Given” section consists of information available to the designer for decision-making. This includes information about the variables under his/her control (e.g., $V_A = \{v_{A1}, v_{A2}, \ldots, v_{Am}\}$ in the case of designer A), ranges of values for design variables at a given iteration in the cycle (e.g., $[V_{A}]_i$ for designer A), preferences for his/her goals (e.g., $\{U(G_{A1}), U(G_{A2})\ldots\}$ for designer A), and the weights associated with individual goals (e.g., $\{w_{A1}, w_{A2}, \ldots\}$ for designer A). The designers’ preferences are modeled using functions U that can be either complex nonlinear utility functions or simple value functions. The “Find” section captures the outcome of solving the cDSP, i.e., the values of design variables under his/her control (e.g., $[V_{A}]_{i+1}$ for designer A) for specified values of design variables controlled by other designers. The “Satisfy” section consists of information about the design constraints that the designers must satisfy, has the requirement that the updated range of design variables should be a subset of the current range, and comprises the metric for overall goal achievement. In Figure 4, the metric is shown to be a simple weighted combination of preference values for a designer’s goals. However, other combinations of preferences such as in a preemptive formulation, multiplicative preferences, etc. can also be used. All multidisciplinary constraints (e.g., relationships between parameters related to differing domains) should also be captured in the “Find” section of individual cDSPs.

The “Minimize” section of the compromise DSP captures the objective function, which is the deviation from the maximum possible achievement of goals (e.g., $(1 - U_A)$ for designer A).

In the running example, we assume that Designer A is responsible for minimizing his/her objective as given by the response $Y_A$, whereas Designer B is charged with maximizing $Y_B$. The responses are related to the design variables as follows:

$$Y_A = \frac{5}{2}v_{A1}^2 + X_A X_B - 30X_A + \frac{1}{10}v_{B2}^2 - 5X_B \quad \text{(Designer A)}$$

$$Y_B = -5v_{B1}^2 - X_A X_B + 50X_B - \frac{1}{10}v_{A1}^2 + 5X_A \quad \text{(Designer B)}$$

In order to keep this problem simple, we assume that there are no constraints (except for those comprised by the bounds placed on the design variables).
The compromise DSP formulation of design decisions is used in the illustrative problem presented in Section 4.

**Step 4: Identification of active designer** – After each designer has formulated his/her design decision, a sequence of operations in which designers are to make their decisions is determined. The sequence must be such that in a given cycle, each designer receives exactly one turn to reduce the range of his/her design variables. A designer is in the active state if it is his/her turn to make a decision. All other designers are in the inactive state. At any given point in time, only one designer is in the active state. A full cycle of the IBCS method is completed once all of the interacting designers have made a decision, successively reducing the available design freedom. In the running example, suppose designer A is active.

**Step 5: Evaluation of RRS by active designer** – The active designer determines his/her RRS based on the decision formulation from Step 3. The RRSs can be evaluated analytically if the design formulation is sufficiently simple, as in the running example, where the designers’ best response is given by the following equations that also represent the designers’ respective RRSs:

\[
\frac{\partial Y_A}{\partial X_A} = (5X_A + X_B - 30) = 0
\]

\[\Rightarrow X_A = \frac{30 - X_B}{5} \quad \text{(Designer A’s RRS)}\]

\[
\frac{\partial Y_B}{\partial X_B} = (-10X_B - X_A + 50) = 0
\]

\[\Rightarrow X_B = \frac{50 - X_A}{10} \quad \text{(Designer B’s RRS)}\]

If the design formulation is more complicated, the analytical approach is not effective. In such scenarios, the designers should (a) generate points on the RRS by assuming fixed values for design variables controlled by other designers, and (b) fit a response surface through these points. During the initial stages of design process the design space is usually large, forcing designers to develop approximate RRSs. However, as the design process progresses the design space reduces and the designers can develop more accurate RRSs. Although this set of linear equations is quite simple, it is

---

<table>
<thead>
<tr>
<th>Designer A’s cDSP</th>
<th>Information Flow between Designers A, B</th>
<th>Designer B’s cDSP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given</strong></td>
<td>– Designer A’s design variables ( V_A = {v_{A1}, v_{A2}, \ldots, v_{An}} )</td>
<td><strong>Given</strong></td>
</tr>
<tr>
<td></td>
<td>– Range of variables controlled by designer A at cycle ( i ): ( [V_A^i])</td>
<td>– Designer B’s design variables ( V_B = {v_{B1}, v_{B2}, \ldots, v_{Bn}} )</td>
</tr>
<tr>
<td></td>
<td>– Range of variables controlled by designer B: ( [V_B])</td>
<td>– Range of variables controlled by designer B at cycle ( i ): ( [V_B^i])</td>
</tr>
<tr>
<td></td>
<td>– Designer A’s preferences for his/her own goals: ( U(G_{A1}), U(G_{A2}), \ldots, )</td>
<td>– Range of variables controlled by designer A: ( [V_A^i]_{+1})</td>
</tr>
<tr>
<td></td>
<td>( \text{where } G_{A1}, G_{A2}, \text{ etc. are goals for designer A, and } U(G_{Ak}) ) is preference function for goal ( G_{Ak} )</td>
<td>– Designer B’s preferences for his/her own goals: ( U(G_{B1}), U(G_{B2}), \ldots, )</td>
</tr>
<tr>
<td></td>
<td>– Weights for preference functions ( {w_{A1}, w_{A2}, \ldots} )</td>
<td>( \text{where } G_{B1}, G_{B2}, \text{ etc. are goals for designer B, and } U(G_{Bk}) ) is preference function for goal ( G_{Bk} )</td>
</tr>
<tr>
<td><strong>Find</strong></td>
<td>– The values of designer A’s design variables at cycle ( i+1 ): ( V_{A,i+1} ) for fixed values of variables under B’s control.</td>
<td><strong>Find</strong></td>
</tr>
<tr>
<td></td>
<td>– The values of designer B’s design variables at cycle ( i+1 ): ( V_{B,i+1} ) for fixed values of variables under A’s control.</td>
<td>– Designer B’s design constraints</td>
</tr>
<tr>
<td><strong>Satisfy</strong></td>
<td>– Designer A’s design constraints</td>
<td>– Values of design variables should be inside the lower and upper bounds, i.e., ( [V_A^i]_{+1} \subseteq [V_A^i] )</td>
</tr>
<tr>
<td></td>
<td>( \text{Values of design variables should be inside the lower and upper bounds, i.e., } ) ( [V_B^i]_{+1} \subseteq [V_B^i] )</td>
<td>– Metric for overall achievement of goals: ( U_A = \Sigma w_A U(G_{Ak}) )</td>
</tr>
<tr>
<td></td>
<td>( \text{Metric for overall achievement of goals: } U_A = \Sigma w_A U(G_{Ak}) )</td>
<td>– Metric for overall achievement of goals: ( U_B = \Sigma w_B U(G_{Ak}) )</td>
</tr>
<tr>
<td><strong>Minimize</strong></td>
<td>– Deviation from the maximum possible overall achievement of goals: ( (1-U_A) )</td>
<td><strong>Minimize</strong></td>
</tr>
<tr>
<td></td>
<td>– Deviation from the maximum possible overall achievement of goals: ( (1-U_B) )</td>
<td>– Deviation from the maximum possible overall achievement of goals: ( (1-U_B) )</td>
</tr>
</tbody>
</table>

Figure 4. The compromise DSP word formulation of the decision made by each designer in the interval-based method.
Step 6: Reduction of the interval of design variables controlled by the active designer – During his/her turn, the active designer is presented with a range of values for the design variables controlled by other (inactive) designers (Figure 4). This range represents a set of values within which inactive designers have the freedom (and responsibility) to select any value they choose. Given this range and the RRS, the active designer determines the range for his/her design variables that will satisfy his/her objectives, regardless of what values other designers determine for the decision variables under their control. This is equivalent to determining the range of design variables that satisfy consistency with respect to the RRS of the active designer (see Section 2.2 for details on consistency). The RRS is converted into an interval constraint and Box consistency is applied to eliminate ranges of the active designer’s design variables. Various algorithms have been proposed in the interval constraint solver literature. One such algorithm for evaluating Box consistency is the Branch and Prune algorithm [24,49]. It is important to note that consistency is checked only using the active designer’s RRS. This approach is analogous to interval constraint solver techniques.

Referring back to the running example, Designer A is provided with the range of values for $X_B$ within which, Designer B has the freedom to select any value. Based on this range of $X_B$, Designer A determines a range of values for $X_A$ such that for any value of $X_A$ within the specified range, a value for $X_A$ can be chosen that will satisfy his/her response ($Y_A$). The starting ranges for the two design variables are $X_A = [0, 10]$ and $X_B = [0, 10]$. Considering the range of $X_B = [0, 10]$, Designer A determines the range of $X_A$ that minimizes his/her objective $Y_A$. This range is evaluated to be $X_A = [4.0, 6.0]$ by elimination of intervals that do not satisfy Box Consistency [47]. It can also be seen by plotting the RRS that all values of $X_A < 4.0$ and $X_A > 6.0$ can be excluded from the initial range of $X_A$, because these values do not lead to Box Consistency with respect to $X_A = 30 - X_B / 5$.

Step 7: Convergence check – The process of reducing the ranges of design variables is followed cyclically until the solution converges to a point. Often this degree of convergence is attained at a single point – the Nash equilibrium. In special cases, there may be multiple Nash equilibria, where a slight modification in the solution process is required. This special case is discussed in Section 4.2. The solution is said to converge to a single point in the design space if each of the dimensions of the box falls below a pre-defined tolerance limit for the corresponding design variables. Since the current range of design variables in the running example is $X_A = [4.0, 6.0]$ and $X_B = [0, 10]$, the process is continued.

Step 8: Communication of reduced intervals to the next active designer – The ranges of values of design variables from the active designer are passed on in sequence to an inactive designer who then becomes active during his/her turn. Steps 5 through 7 are repeated for each active designer.

Using the range determined by Designer A, Designer B is able to eliminate those values of $X_B$ from his/her starting range that do not result in Box Consistency with respect to $X_B = 50 - X_A / 10$. The resulting range for $X_B$ is $[4.40, 4.60]$. This concludes the first cycle.

The sequential range reductions continue until the ranges of $X_A$ and $X_B$ converge to a point. The ranges of design variables resulting from successive cycles are as follows:

- Cycle 0: $X_A = [0.000, 10.000]$, $X_B = [0.000, 10.000]$
- Cycle 1: $X_A = [4.000, 6.000]$, $X_B = [4.400, 4.600]$

The solution converges to $X_A = 5.102$ and $X_B = 4.489$, a result one might expect based upon the intersection of the designers’ respective RRSs. A prerequisite initial condition for application of this method is that the starting ranges for variables controlled by both designs are such that it is possible for the active designer to find a value for his/her design variables (satisfying his/her objectives) for all values of design variables controlled by inactive designers. For example in the two-designer scenario, for any value in the range of $X_B$, Designer A should be able to select a value of $X_A$ that satisfies $Y_A$. Similarly, for any value in the range of $X_A$, Designer B should be able to select a value of $X_B$ that satisfies $Y_B$. If the initial condition (mentioned earlier) is met, the process described in Figure 3 can be initiated. If the conditions are not met, however, the active designer cannot reduce the design space further. Designers can overcome such situations by either (a) changing the assignment of design variable control or (b) redefining the design space.

Although the method is presented in this article using examples involving two designers, it is equally applicable to design scenarios involving more than two designers. The method is also applicable in the scenarios where each designer handles more than one design variable. In such a scenario, the design variable interval reduction step (Step 6) involves reducing the ranges of multiple design variables under the control of the active designer.
Furthermore, since the decision formulation in the running example remains the same during each cycle, the RRS for each designer also remains the same. In such a scenario, Step 3 needs to be executed only once by each designer. However, the method allows the designers to update their decision formulation based on new information available after completing any given cycle. This is particularly important as designers improve their understanding of a design by gathering more information along the design process. This is one of the main advantages of a set-based approach to design. Having described the method, we proceed to illustrate it with a multifunctional design problem involving the design of Linear Cellular Alloys in Section 4.

4. Illustrative Example – Linear Cellular Alloy Design

In this section, we present an example centered on the multifunctional design of Linear Cellular Alloys (LCA) [51,52] in order to demonstrate the applicability of the IBCS method to complex non-linear problems, where designers aim to achieve target values of their respective objectives. This stands in contrast to the example presented in Section 3, where each designer was interested in maximizing/minimizing their objectives. Using this example, we illustrate how a multifunctional design problem is formulated when the design problem is more complicated. In the LCA problem, the designers’ preferences are characterized by more complicated utility functions, and the behavior of LCAs is evaluated using complex finite element models. Additionally, handling situations with multiple Nash equilibria is illustrated.

Linear cellular alloys are honeycomb materials (Figure 5) that are produced via the extrusion of ceramic slurry through a multistage die. The slurry is composed of a binder mixed with metal oxide powders. The structure resulting from the extrusion is first dried and reduced into the metallic phase in a hydrogen rich environment. It is then sintered to produce nearly fully dense metal composites. A wide range of cell sizes and shapes including functionally graded structures can be achieved using this manufacturing process. The resulting materials are especially suitable for multifunctional applications that require both strength and heat transfer capabilities [53]. Applications of these materials include heat sinks for microprocessors and combustor liners for aircraft engines. One of the main advantages of these LCAs is that desired structural and thermal properties can be obtained by designing the cell shape, cell arrangement, and cell wall thicknesses, as well as dimensioning the overall LCA structure.

Consider a scenario where a multifunctional LCA is to be designed with both thermal and structural requirements. The design is to be carried out by two designers – Designers A and B. The design problem involves evaluation of design variable values that will satisfy the functional requirements as closely as possible. In this case, the following two geometric parameters of the LCA can be controlled by the designers: overall height of LCA (H) and wall thickness (t). All other parameters in the LCA geometry are fixed. Designer A controls overall height (H) and is responsible for achieving the targeted total heat transfer rate (Q). Designer B is responsible for compliance and controls the wall thickness of the rectangular LCA.

The design requirements for thermal and structural performance of the LCA are given by Designers A and B.

<table>
<thead>
<tr>
<th>Designer A’s cDSP</th>
<th>Information Flow between Designers A, B</th>
<th>Designer B’s cDSP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given</strong></td>
<td></td>
<td><strong>Given</strong></td>
</tr>
<tr>
<td>Designer A’s design variable: Height of the LCA (H)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range of Height at cycle i: [H]_i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range of thickness: [t]_i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designer A’s preferences for heat transfer rate: U(Q) (see Figure 6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Find</strong></td>
<td></td>
<td><strong>Find</strong></td>
</tr>
<tr>
<td>Height H_{i+1} for a prespecified value of thickness by designer B</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Satisfy</strong></td>
<td></td>
<td><strong>Satisfy</strong></td>
</tr>
<tr>
<td>[H]_{i+1} = [H]_i</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U_A = U(Q)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Minimize</strong></td>
<td></td>
<td><strong>Minimize</strong></td>
</tr>
<tr>
<td>(1−U_A)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Ranges of values for [t]_i |
| Ranges of values for [H]_{i+1} |

Figure 5. cDSP decision formulations for LCA example.
and B, respectively, in terms of target matching preference functions as shown in Figure 6. The preference function varies from 0 to 1, with higher numbers indicating preference. Both designers formulate their decisions in terms of the cDSPs shown in Figure 5. Notice that since each designer controls only one design goal, weights for goals are not required.

4.1 Application of the IBCS Method for LCA Example

The steps presented in Section 3 are followed for the LCA design example. The results obtained by applying the proposed IBCS method (Figure 3) are presented in Table 1. In this table, the ranges of design variables (overall height and wall thickness) after each successive cycle are presented. The gradual reduction of the design space along the design process for the case where Designer A makes the decision before Designer B is plotted in Figure 6(a). This example shows that the proposed method can be applied to nonlinear problems. It is important to note that after each cycle, achievable target values for compliance and overall heat transfer remain ranges.

The reduction of design space for the case where Designer B makes a decision before Designer A is shown in Figure 6(b). Note that there is a change in the rate of convergence of the solution as the order (precedence) of decision-making is changed. The rate of convergence directly affects the number of cycles required for convergence to the Nash equilibrium. The effect of decision-making sequence on the convergence rate for IBCS requires detailed analysis, and is not studied in this article.

4.2 Handling multiple Nash equilibria

A special case in the IBCS method is when the RRSs intersect more than once in a given design space (i.e., multiple Nash equilibria exist). In the case of multiple Nash equilibria, convergence to a point solution may be impeded. This means that the size of the box may remain constant from one cycle to the next. The procedure presented in Section 3 needs to be augmented to handle such cases. Consider a hypothetical case of two designers with RRSs intersecting more than once, as shown in Figure 7(a). After several reductions of the design space using the proposed IBCS method, the region containing possible solutions is reduced to rectangle ABCD. Clearly, subsequent cycles will not reduce the design space further. A possible solution to this problem is to partition the design space into subsets (e.g., rectangles AEFD and EBCF in Figure 7(b)) to which the IBCS method is then applied in parallel. The result of one post partition cycle is shown in Figure 7(c).

Multiple Nash equilibria occur in the LCA design scenario when the designers’ preferences are such that the targets for heat transfer and compliance in Figure 6 are $-5.0$ and $5.2$, respectively. In this case there are two Nash equilibria, and the design space cannot be reduced after $t = [4.914, 5.099] \mu m$ and $H = [20.93, 29.32] mm$. At this stage, the design space is split into two subspaces by dividing the range of height values into the following two subsets: $H_1 = [20.93, 25.00]$ and $H_2 = [25.00, 29.32]$, and the IBCS is carried out in parallel for both these subspaces. The resulting solutions from both subspaces converge to the corresponding Nash equilibria as shown in Table 2. The two Nash equilibria
are at: \( t = 4.914 \, \mu m, \ H = 20.93 \, mm \) and \( t = 5.085 \, \mu m, \ H = 28.19 \, mm \), respectively.

**Closure**

Set-based design is a conceptual framework for concurrent engineering [18]. Its computational implementations have been presented for selection decisions and collaborative CAD but methods for decentralized multifunctional design based on set-based concepts are not available. In this article, we address this gap by presenting a computational implementation of a set-based design approach in the form of an Interval-Based Constraint Satisfaction (IBCS) method. The method is based on two foundational constructs – (a) noncooperative game theory, and (b) interval-based consistency. Non-cooperative game theory is used to transform a multifunctional design problem into a constraint satisfaction problem via the evaluation of Rational Reaction Sets. Principles from interval-based consistency are used to solve the resulting set of constraints in a cyclic manner through the propagation of ranges of design variables. The method is illustrated using two examples. Specifically, we solve a noncooperative game, centered on the intersection of linear

---

**Table 2. Convergence of solution in the case of multiple Nash equilibria.**

<table>
<thead>
<tr>
<th>Cycle #</th>
<th>Subspace 1</th>
<th>Subspace 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Range for thickness ((t)) ((\mu m))</td>
<td>Range for height ((H)) (mm)</td>
</tr>
<tr>
<td>0</td>
<td>[4.914, 5.099]</td>
<td>[20.93, 25.00]</td>
</tr>
<tr>
<td>1</td>
<td>[4.914, 5.014]</td>
<td>[20.93, 24.43]</td>
</tr>
<tr>
<td>2</td>
<td>[4.914, 4.961]</td>
<td>[20.93, 22.43]</td>
</tr>
<tr>
<td>3</td>
<td>[4.914, 4.935]</td>
<td>[20.93, 21.58]</td>
</tr>
<tr>
<td>4</td>
<td>[4.914, 4.924]</td>
<td>[20.93, 21.21]</td>
</tr>
<tr>
<td>5</td>
<td>[4.914, 4.918]</td>
<td>[20.93, 20.99]</td>
</tr>
<tr>
<td>6</td>
<td>[4.914, 4.916]</td>
<td>[20.93, 20.94]</td>
</tr>
<tr>
<td>7</td>
<td>[4.914, 4.915]</td>
<td>[20.93, 20.94]</td>
</tr>
<tr>
<td>8</td>
<td>[4.914, 4.915]</td>
<td>[20.93, 20.94]</td>
</tr>
</tbody>
</table>
RRSs, in the first example (a set of quadratic equations). Application of the method to cases where RRSs are non-linear is demonstrated in the second example (LCA design).

The key advantages inherent to the proposed method include (a) nondivergence of solutions, and (b) gradual reduction of design freedom, prolonging the adaptability to changes in requirements (during the design process). Addressing convergence and unanticipated iteration problems through the proposed method is a significant benefit to concurrent engineering in complex product development. It is noted that non-divergence is not equivalent to convergence. There may be situations where designers are unable to reduce a given design space further by simply following the IBCS method. Using this method, the designers are able to identify such situations in a single cycle. Design efficiency is improved; resources are conserved. Point-based methods on the other hand, often result in divergent iterations.

A gradual reduction of a design space means that there is a range of responses that can be satisfied after any given cycle. Hence, limited changes in design objectives can be accommodated without re-executing the design process in its entirety (i.e., without repeating the preceding design cycles). This remains true as long as the updated design objectives can be satisfied by a point in the design space corresponding to the current design cycle. Notice that keeping design freedom open may not provide a significant advantage if (a) the complete process is automated on a network of computers using simulation models, and (b) the likelihood of changes in requirements during the execution of the process is small. However, it is advantageous in situations where the process takes sufficiently long, the decisions involve human decision-makers, and there is a possibility of changes in design requirements throughout the execution of design process. The proposed method allows designers to update their decision formulations by including the information gathered along the design process.

Further development of this method is centered on investigating cases where – (a) designers have additional, local design variables that are not shared, but depend on the values of shared parameters, and (b) design variables are defined on discontinuous or piece-wise defined intervals. Other considerations include scalability of the method to cases characterized by more than two designers, each in charge of multiple design variables. Different partitioning schemes may not only lead to different answers but may also change the convergence characteristics underlying a design problem. Convergence criteria may also be used as a basis for the assignment of design variable control to different designers for intelligent sequencing and convergence rate optimization.

Acknowledgements

We gratefully acknowledge the Air Force Office of Scientific Research (AFOSR) Multi-University Research Initiative Grant (1606U81) and National Science Foundation grants DMI-0085136 and 0407627. We thank Dr Carolyn Conner Seepersad for helping with the Linear Cellular Alloy design example.

References


Dr Jitesh Panchal

Dr Jitesh Panchal is a Visiting Assistant Professor in the Woodruff School of Mechanical Engineering at Georgia Institute of Technology, Savannah. He received his Bachelor of Technology degree from Indian Institute of Technology at Guwahati, and MS and PhD in Mechanical Engineering from Georgia Institute of Technology. His research interests are in the field of Engineering Systems Design. Specifically, his current research focus is on simulation-based distributed design methodologies and technologies for multiscale systems. Dr Panchal is a member of American Society of Mechanical Engineers (ASME) and American Institute of Aeronautics and Astronautics (AIAA).

Dr Marco Gero Fernández

Dr Marco Gero Fernández is a Senior Consultant with Capgemini Business Consulting Services. He received his BSE from Duke University and earned MS, PhD, and MBA degrees from the Georgia Institute of Technology. He was a National Science Foundation Graduate Research Fellow, GE Faculty of the Future Award Recipient, NSF IGERT Fellow, and Woodruff Teaching Fellow. Dr Fernández’s research interests include distributed collaborative design and manufacture, communications protocols, negotiations, resource allocation, and trade-off management. Applications are focused on intricacies inherent in comprehensive design challenges such as those associated with materials design, including the topological design of cellular materials, and complex systems engineering.

Dr Janet K. Allen

Dr Janet K. Allen is an Associate Professor of Mechanical Engineering at Georgia Tech, Savannah. She received PhD in Biophysics from the University of California at Berkeley and her SB in Life Science from the Massachusetts Institute of Technology in 1973 and 1967, respectively. In 1992 she became a Senior Research Scientist in the Systems Realization Laboratory in The George W. Woodruff Department of Mechanical Engineering at Georgia Institute of Technology. She is a Fellow of the American Heart Association, Council on Atherosclerosis (1985) and has published more than 60 refereed papers.
Dr Christiaan J. J. Paredis

Dr Christiaan J. J. Paredis is an Assistant Professor in the Woodruff School of Mechanical Engineering at Georgia Tech. He received his MS degree in Mechanical Engineering from the Catholic University of Leuven (Belgium) in 1988, and his MS and PhD in Electrical and Computer Engineering from Carnegie Mellon University in 1990 and 1996, respectively. From 1996 to 2002, he was a Research Scientist at the Institute for Complex Engineered Systems at Carnegie Mellon University. Dr Paredis has a broad, multidisciplinary background. In his research, he combines aspects of information technology, simulation, and systems theory to support the design of mechatronic systems, focusing in particular on decision making under uncertainty in conceptual design. Dr Paredis received the 2007 CETL/BP Junior Faculty Teaching Excellence Award and the 2007 SAE Ralph R. Teetor Educational Award. He is the vice-chair of the ASME Computers and Information in Engineering (CIE) Division, and is the Chair for the 2007 CIE annual conference.

Dr Farrokh Mistree

Dr Farrokh Mistree is a Professor of Mechanical Engineering, Associate Chair of the Woodruff School of Mechanical Engineering and Associate Director of Georgia Tech, Savannah. His design experience spans mechanical, aeronautical, structural, and industrial engineering. His teaching experience spans courses in engineering design, naval architecture, solid mechanics, operations research and computer science. His current research focus is on learning how to manage design freedom in multi-scale design (from molecular to reduced-order models) to facilitate the integrated design of materials, product and design process chains. He is committed to developing a design pedagogy that is rooted in Decision-Based Design and adaptive action learning. It is in this context that he enjoys experimenting with ways in which design can be learned and taught. Dr Mistree has co-authored two textbooks and over 300 technical publications. Dr Mistree was recognized for his research and teaching in 1999 and 2001, respectively: ASME Design Automation Committee’s 1999 Design Automation Award and 2001 Jack M. Zeigler Woodruff School Outstanding Educator Award. He is the Secretary-Treasurer of Pi Tau Sigma Mechanical Engineering Honor Society and served as a reviewer for ABET. He is a Fellow of ASME, an Associate Fellow of the AIAA, a Member of ASEE and the Society of Naval Architects and Marine Engineers. And what does Farrokh Mistree want to do in the future? He wants to have fun—fun in defining the emerging science-based discipline of design. Fun in providing an opportunity for highly motivated and talented people to learn how to achieve their dreams.