Design Knowledge Extraction in Multi-objective Optimization Problems

Sunith Bandaru
sunithb@iitk.ac.in

Kalyanmoy Deb
deb@iitk.ac.in

Kanpur Genetic Algorithms Laboratory, Department of Mechanical Engineering
Indian Institute of Technology Kanpur
Kanpur 208016, U.P., India

ABSTRACT

This work concerns the post-optimal analysis of the trade-off front of a multi-objective optimization problem to extract useful design knowledge pertaining to these high-performing solutions. The expected knowledge basically consists of statistically significant relationships between the objective functions and decision variables. These relationships are represented in an intuitive and easy-to-use mathematical form. Since a number of such relationships may exist, for efficiency it is desirable that they are obtained in a single knowledge extraction step. Further, problem knowledge can be explored at two levels: lower and higher. At the lower-level, our interest is in finding a subset of the trade-off solutions to which the obtained relationships are exclusive. The higher-level knowledge addresses the effect of varying the problem parameters (that are kept constant in one run) on the trade-off front and therefore on the relationships. These concepts are explained through different examples.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*

General Terms

Algorithms

Keywords

knowledge extraction, innovization, design principles

1. INTRODUCTION

Knowledge discovery is a concept which applies to a multitude of data points. A single optimal solution to a design problem which involves only one objective is insufficient to yield any design specific knowledge. On the other hand, if the same design involves multiple *conflicting* objectives, a single solution is unlikely to exist. Methodologies exist which can generate a set of designs that are optimal in the Pareto sense. Useful problem knowledge can be gained by analyzing these designs post-optimally. Of prime importance are the features that are common to all or most of these designs. They are referred to as 'design rules' [5] or 'design principles' [4]. If these commonalities can be automatically, reliably and efficiently extracted for a wide range of problems it will be possible to identify certain fundamental characteristics that the design should (or should not) possess in order for it to be Pareto-optimal. Various ways of usefully employing this knowledge (design rules or principles) have been proposed:

1. The most significant of the design rules can be stored in the knowledge base of an expert system for future reference in design cases that are structurally similar to the original one used for knowledge extraction [7].
2. The knowledge can be used as an investigative tool for analyzing existing designs with desirable qualities and proposing changes which can push the design towards Pareto-optimality [5].
3. Certain representations of the knowledge can be used to determine the sensitivity of the Pareto-optimal designs to the decision variables [4].
4. Crucial design decisions can be governed by the extracted design principles.
5. Suitable forms of the knowledge can be integrated into an optimization algorithm to give a heuristics-based local search algorithm for improving near Pareto-optimal solutions.

In the following section, we discuss different design knowledge representations available in literature and give a short description of a new design methodology called *innovization*. Thereafter in Sec. 2, we describe the automated innovization algorithm developed as part of this work. Results with this algorithm are illustrated in Sec. 3.1 on a well-studied truss design problem. Next an architectural design problem is briefly considered in Sec. 3.2 to show how the proposed knowledge extraction scheme can be beneficial even when the objectives are vague. Finally in Sec. 4, we very briefly show some preliminary results from a higher-level innovization study on a practical friction-stir welding problem.

1.1 Knowledge Representation

Knowledge can be implicit or explicit. Implicit knowledge can be very difficult to employ in the ways described above.

GECCO'11, July 12–16, 2011, Dublin, Ireland.

Explicit knowledge, on the other hand, requires a *language*, e.g. mathematical, visual, etc. Broadly speaking, the way the extracted knowledge is represented mainly depends on the problem. For example, an association rule based representation may be well-suited for designs with class variables but may lose efficacy in, say, an engineering design problem. As a result, different representations have been proposed in literature. Most notable are rule-based [8], self-organizing maps [6] and dendograms [10].

Innovization (*innov*ation through optim*ization*) [4] proposes the use of monomials (where negative exponents are allowed) for knowledge representation. Recall that a design principle is a feature that is common to all or a majority of the Pareto-optimal solutions. In the innovization framework, this means

Definition 1. $\psi(\mathbf{x}) \equiv \phi_1^{b_1}(\mathbf{x})\phi_2^{b_2}(\mathbf{x})\phi_3^{b_3}(\mathbf{x})\ldots = c$ is a design principle applicable to $A \subseteq P$, the set of Pareto-optimal solutions, if there exist problem specific functions ϕ_j and exponents $b_j \in \mathbb{R}$: c evaluates to the same value for all decision vectors $\mathbf{x}$ in A.

Here, ϕ_j's are user-defined functions and usually the variables and objectives of the problem are the primary choice. This mathematical form of knowledge representation for the design principles can be very beneficial to a designer interested in knowing, for example, how changes in one variable will effect other variable(s) if the design has to retain its Pareto-optimal nature. Also, this representation easily lends itself to the various uses that the extracted knowledge can be put to.

For illustration, let us consider the simple two-bar truss-structure shown in Figure 1. The total volume (V) and the maximum stress (S) induced in either of the bars have to be minimized simultaneously. The cross-sectional areas x_1, x_2 and the dimension y are the design variables ($\mathbf{x}$) of the multi-objective problem which is formulated as:

$$\begin{array}{ll} \text{Minimize} & V = x_1\sqrt{16+y^2} + x_2\sqrt{1+y^2}, \\ \text{Minimize} & S = \max(\sigma_{AC}, \sigma_{BC}), \\ \text{Subject to} & \max(\sigma_{AC}, \sigma_{BC}) \le 10^5 \text{ kPa}, \\ & 0 \le x_1, x_2 \le 0.01 \text{ m}^2, \\ & 1 \le y \le 3 \text{ m}. \end{array} \quad (1)$$

This simple design problem is solved using NSGA-II. A Pareto-optimal front with 1000 points is obtained. Manual innovization requires that we plot different variable/objective combinations and perform a regression if significant correlations are found. Figure 2(a) shows two significant design principles. They tell the designer that a majority of Pareto-optimality occurs when V and x_2 are directly proportional to x_1. Since these principles are applicable to the same set of points, a third design principle can also be generated.

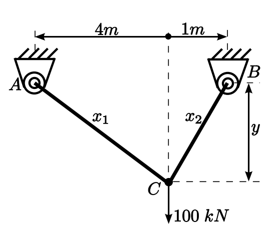
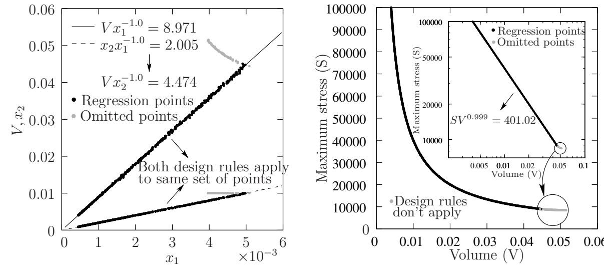


Figure 1: Two-bar truss configuration.



(a) Design principles involving V, x_1 and x_2.

(b) Design principle involving objectives S and V.

Figure 2: Manual innovization for truss design.

Figure 2(b) shows another design principle involving the objectives S and V. Here a logarithmic transformation of the points is required before regression. Note that all these relationships fit into the mathematical structure in Definition 1. Though in this case, the obtained relationships are exact[1], this need not be true for generic problems. All that should be expected is empirical interdependencies between the ϕ functions (referred to as 'basis functions' in the context of automated innovization).

2. AUTOMATED INNOVIZATION

Manual innovization cannot be used to discover interactions between more than two or three basis functions due to the cognitive inability of humans to perceive in higher dimensions. Furthermore, the process is tedious and involves data transformations and an additional task of regression. The human-error factor can also not be overruled. Here, we propose a clustering based methodology to extract the design principles automatically. The user is required to only input a set of N basis functions. Within the automated innovization framework, the i-th design principle is represented as,

$$\psi_i(\mathbf{x}) \equiv \Pi_{j=1}^{N} \phi_j(\mathbf{x})^{a_{ij}b_{ij}} = c_i. \quad (2)$$

The manual choosing of various basis function combinations is automatically accomplished through the Boolean variables a_{ij}'s which represent the presence (1) or absence (0) of a basis function. For all such choices, we quantify the significance S_i of $\psi_i(\mathbf{x})$ using the percentage of the trade-off solutions that it applies to. According to Definition 1, these trade-off solutions form the subset A. Since the c_i-values are expected to be equal within this subset, a clustering algorithm can be used to identify A. We propose the following grid-based clustering methodology [2]:

1. Sort $\{c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(m)}\}$ evaluated for the m trade-off solutions using $\psi(\mathbf{x})$.
2. Divide the range $[c_{i,min}, c_{i,max}]$ into d_i divisions. n_{d_i} be the number of c_i-values falling in the d_i-th division.
3. Label the divisions with $n_{d_i} \ge \lfloor \frac{m}{d_i} \rfloor$ (the average number of c_i-values per division) as *sub-clusters*.
4. Label the trade-off points corresponding to the c_i-values in the remaining divisions as *unclustered*.

[1]Eq. (1) can be analytically solved using $x_1\sqrt{16+y^2} = x_2\sqrt{1+y^2}$ and $\sigma_{AC} = \sigma_{BC}$ (identical resource allocation).

5. Merge adjacent sub-clusters to form $\mathcal{C}_i$ clusters.

6. Count the number of unclustered trade-off points $\mathcal{U}_i$.

Here, d_i is a parameter of the clustering algorithm which can take any value in the range $[1, m] \; \forall \; i$. A suitable value for it can be obtained by minimizing the number of clusters and unclustered points. Unclustered trade-off points will have c_i-values which are very different from each other and also from those in the clusters. Thus they are the points that do not follow the i-th design principle. By definition, the significance can now be obtained as,

$$S_i = \frac{m - \mathcal{U}_i}{m} \times 100\%. \tag{3}$$

2.1 Converging to a Design Principle

The near Pareto-optimal nature of the solutions obtained from a numerical algorithm like NSGA-II prevents the c_i-values in the subset A from being exactly equal. Therefore in the present work, the coefficient of variance ($c_v = \sigma/\mu$) is used to as a measure of the spread of c_i-values. For converging iteratively to a design principle and hence improving its significance, the c_v's have to be minimized in all $\mathcal{C}_i$ clusters. The exponents a_{ij}'s and b_{ij}'s become variables in the resulting optimization problem. The parameter d_i of the clustering algorithm is also treated as a variable, and its value is optimized for minimum number of clusters. As there can be many clusters, the c_v's from different clusters are summed. It was found [2] that the percentage coefficient of variance has about the same order of magnitude as the number of clusters. Hence the following weighted objective function is used for finding optimal a_{ij}, b_{ij} and d_i,

$$\text{Minimize} \quad \mathcal{C}_i + \sum_{k=1}^{\mathcal{C}_i} c_v^{(k)} \times 100\%. \tag{4}$$

2.2 Preserving Multiple Design Principles

A straightforward minimization of the above objective function will only yield the most significant design principle present in the trade-off dataset. However, innovization is concerned with finding all principles which meet a threshold significance. Previous manual [4] and automated innovization [2] studies achieved this through multiple trials with different sets of basis functions. A recent study [1] exploits the population based nature of GAs to evolve multiple design principles simultaneously using a niching strategy. In a GA each population member will represent a possible design principle with an associated significance S_i. Two population members, u and v, which use different sets of basis functions i.e. $\exists \; j : a_{uj} \neq a_{vj}$, can evolve into two different design principles and hence are not compared during the selection operation of GA. Thus different 'species' of design principles can be made to coexist in the population. Since a_{ij}'s are Boolean, they can be encoded as a binary variable of string length N. The exponents b_{ij} can theoretically take any real value. However, their range can be restricted to $[-1, 1]$ by simply dividing each of them with the exponent having maximum absolute value in every generation of the GA. For example, $\phi_1^{2.5}\phi_2^{1.0}\phi_3^{-4.0} = c$ becomes $\phi_1^{-0.625}\phi_2^{-0.25}\phi_3^{1.0} = c'$. Note that the design principle is not effected in the process.

In order to prevent trivial solutions (when $a_{ij} = 0 \; \forall \; j$) and complex design principles (when many basis function are involved), the constraint $1 \leq \sum_j a_{ij} \leq \mathcal{N}$ is introduced in the problem formulation, whose complete form (for the i-th design principle) is:

$$\begin{array}{lll} \text{Minimize} & \mathcal{C}_i + \sum_{k=1}^{\mathcal{C}_i} c_v^{(k)} \times 100\%, & \\ \text{Subject to} & S_i \geq S_{reqd}, & \\ & 1 \leq \sum_j a_{ij} \leq \mathcal{N}, & 1 \text{ binary variable} \\ & -1.0 \leq b_{ij} \leq 1.0, & N \text{ real variables} \\ & 1 \leq d_i \leq m, & 1 \text{ integer variable.} \end{array} \tag{5}$$

where S_{reqd} and $\mathcal{N}$ are user-supplied values indicating respectively, the minimum threshold significance for the design principles and the maximum number of basis functions that can participate in forming that design principle. The mixed variable nature of the problem also justifies our use of a GA to solve Eq. (5). The constraints are handled using Deb's penalty-parameter-less approach [3].

Table 1: Reduced design rules for the truss problem

i	$a_{i1}^*b_{i1}^*$	$a_{i2}^*b_{i2}^*$	$a_{i3}^*b_{i3}^*$	$a_{i4}^*b_{i4}^*$	$a_{i5}^*b_{i5}^*$	Rule
1	1.0000	0.0	0.0	-1.0006	0.0	$V/x_2 = c$
2	0.0	1.0000	0.0	1.0005	0.0	$Sx_2 = c$
3	0.0	0.0	1.0000	-1.0009	0.0	$x_1/x_2 = c$
4	0.0	0.0	0.0	0.0	1.0000	$y = c$

3. RESULTS

3.1 Row Echelon Reduction

The $m = 1000$ trade-off solutions obtained earlier for the truss design problem are used to form the data-set for $N = 5$ basis functions, namely:

$$\phi_1 = V, \quad \phi_2 = S, \quad \phi_3 = x_1, \quad \phi_4 = x_2, \quad \phi_5 = y.$$

With $S_{reqd} = 80\%$ and $\mathcal{N} = 5$, [1] reports the complete set of 20 design principles that were obtained from a single knowledge extraction step. However, many of them are redundant. For example, the third design principle derived form the other two in Figure 2(a) is extraneous information for the designer who would prefer having all the design rules in a compact form. We now use the row reduced Echelon form of the exponents to condense the 20 design rules and present them in a usable form. A tolerance is used during row operations to accommodate for the fact that the original data-set need not exactly represent the Pareto-optimal front. Table 1 shows the reduced design rules for the truss design problem. These are same as those in Figure 2.

3.2 Architectural Design Case

Decision-making tasks sometimes involve soft objectives in the sense that they contain uncertainty, imprecision or vagueness. The present example shows how design principles obtained automatically in such a design case can aid decision-making. The problem is the optimal placement of a number of residential units in an urban design, so that two vague objectives are maximized: south/west garden area for every house and visual privacy on the south facades. Figure 3 shows the plot with three existing houses, and the decision variables associated with seven independent houses (H) and two rows of houses (G).

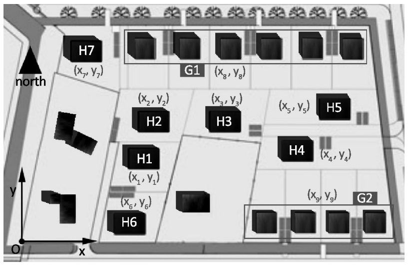


Figure 3: Schematic of the plot showing the coordinate decision variables.

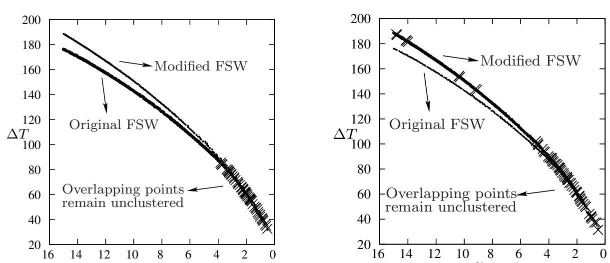
One of the 302 design principles obtained using the automated innovization approach is,

$$y_3^{0.3509}x_4^{0.5718}x_5^{1.0000}x_9^{0.3809} = constant. \quad (6)$$

This relationship applies to 134 out of 135 points of the Pareto-optimal front making it a significant one. It gives the interdependency between four different variables. For instance, most notably the principle tells the decision-maker that if house H5 were moved to the west, then H4 should be moved to the east, so that Pareto-optimality is ensured. This principle aims to compensate the loss in west garden area and decrease in privacy of H5.

4. HIGHER-LEVEL INNOVIZATION

As discussed earlier, higher-level knowledge addresses the effect of parameters that are usually kept constant in a single optimization run. Higher-level innovization is simply the extraction of this higher knowledge through the automated innovization technique. The present bi-objective problem concerns process optimization of friction stir welding (FSW) operation with respect to minimization of temperature gradients (which effect weld quality) and maximization of weld speed (for high production rate). The optimization formulation can be found in [9]. The thickness of the weld which is a parameter for the original FSW problem is taken as a variable in the modified FSW problem. The result is a change in the trade-off front as shown in Figures 4(a) and 4(b). The $\times$ represent the unclustered solutions in both the figures.



(a) Clustering corresponding to $u_{weld}^{-0.9892}T_{average} = c$.

(b) Clustering corresponding to $u_{weld}^{0.2142}T_{ahead} = c$.

Figure 4: Higher-level knowledge in FSW problems.

5. CONCLUSIONS

In this paper, we have described an automated methodology for extracting hidden knowledge in the form of mathematical relationships from the Pareto-optimal solutions of a multi-objective problem. Since many such relationships showing the interactions between various problem entities are possible, a niching strategy is implemented in the algorithm to obtain all design principles simultaneously. We demonstrated the algorithm for the simple truss design problem. The condensed rules are found to be exactly the same as the analytically/manually obtained ones thus validating the algorithm. Next we used the automated innovization methodology in an architectural design problem involving 20 basis functions. Many relationships are obtained in the process and the significance in decision-making is briefly demonstrated. We also presented some preliminary results from higher-level innovization. The lower-level knowledge discovery, and development of a hybrid innovization-optimization algorithm is considered part of the future work.

6. REFERENCES

[1] S. Bandaru and K. Deb. Automated Innovization for Simultaneous Discovery of Multiple Rules in Bi-objective Problems. In *Evolutionary Multi-Criterion Optimization*, pages 1–15. Springer, 2011.

[2] S. Bandaru and K. Deb. Towards automating the discovery of certain innovative design principles through a clustering based optimization technique. *Engineering optimization*, 2011.

[3] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2–4):311–338, 2000.

[4] K. Deb and A. Srinivasan. Innovization: Innovating design principles through optimization. In *GECCO '06 - Proceedings of the 8th annual conference on genetic and evolutionary computation*, pages 1629–1636. New York: ACM, 2006.

[5] C. Mackenzie and J. Gero. Learning design rules from decisions and performances. *Artificial Intelligence in Engineering*, 2(1):2–10, 1987.

[6] S. Obayashi and D. Sasaki. Visualization and data mining of Pareto solutions using self-organizing map. In *Evolutionary Multi-Criterion Optimization*, pages 796–809. Springer, 2003.

[7] A. Radford, P. Hung, and J. Gero. New rules of thumb from computer-aided structural design: Acquiring knowledge for expert systems. In *CAD84*, pages 558–566. Butterworths, 1984.

[8] K. Sugimura, S. Jeong, S. Obayashi, and T. Kimura. Kriging-model-based multi-objective robust optimization and trade-off-rule mining using association rule with aspiration vector. In *2009 IEEE world congress on computational intelligence*, pages 522–529. IEEE Press, 2009.

[9] C. Tutum, K. Deb, and J. Hattel. Hybrid search for faster production and safer process conditions in friction stir welding. In *Proceedings of the 8th international conference on Simulated evolution and learning*, SEAL'10, pages 603–612. Springer-Verlag, 2010.

[10] T. Ulrich, D. Brockhoff, and E. Zitzler. Pattern identification in Pareto-set approximations. In *GECCO '08 - Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 737–744. ACM, 2008.