Optimal Design of Fixture Layouts in Multi-Station Assembly Processes

Pansoo Kim

School of Business Administration, Kyungpook National University, pskim@knu.ac.kr

ABSTRACT

Optimal engineering design is challenging because nonlinear objective functions need to be evaluated in a high-dimensional space. This paper presents a data-mining aided optimal design method. The method is employed in designing an optimal multi-station fixture layout. Its benefit is demonstrated by a comparison with currently available optimization methods.

1. INTRODUCTION

An optimal design problem generally needs to evaluate nonlinear objective functions in a usually high-dimensional space. Nonlinear programming methods usually converge to a solution in a relatively short time. But the quality of the final solution highly depends on the selection of an initial design and these methods are thus known as "local" optimization methods. In order to escape the local optima, one would prefer to use random search based method such as simulated annealing (SA) [1]. Empirical evidence [2] showed that SA is indeed quite effective in escaping local optima but at the expense of considerably slow convergence.

As an example of optimal engineering design, we consider the assembly process of the side frame of a Sport Utility Vehicle (SUV) in Fig 1. The final product, the *inner-panel-complete*, comprises four components: A-pillar, B-pillar, rail roof side panel, and rear quarter panel, which are assembled on three stations (Stations I, II, III). Then, the final assembly is inspected at Station IV (M_1-M_{10} marked in Fig. 1d are key dimensional features). The dimensional quality measured at those key features is mainly determined by the variation input from fixture locators P_1-P_8. The design objective is to find the optimal fixture layout of a multi-station assembly process so that the product dimensional variability (measured at M_1-M_{10}) is insensitive to fixture variation input.

There are eight fixture locators (P_1-P_8) involved in the above-mentioned assembly process. Each part or subassembly is positioned by a pair of locators. For the sake of simplicity, we are only concerned with a 2-dimensional assembly in the X-Z plane, where the position of a locator is determined by its X and Z coordinates. Thus, the design space has 16 parameters and it is continuous, meaning that there are infinite numbers of design alternatives. We can generate a finite candidate design space via discretization, say, using the resolution of 10 millimeter (the size of a locator's diameter) on each panel. This resolution level will result in the number of candidate locations on each panel to be 239, 707, 200, and 3496 respectively. The total number of design alternatives is therefore $C_2^{239}\times C_2^{707}\times C_2^{200}\times\ C_2^{3496}\approx 8.6\times 10^{20}$, where C_a^b is a combinational operator. Apparently, the number of design alternatives is overwhelmingly large and a lot of local optima are embedded in the 16-dimensioned design space. Any local optimization method will hardly be effective and SA random search could be inefficient.

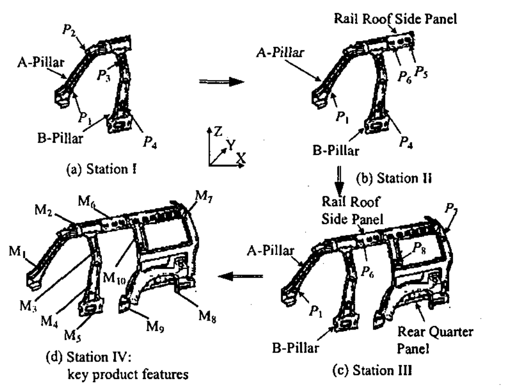

Figure 1. Assembly process of an SUV side frame

Although the general idea could help in discovering valuable design selection guidelines, there is a major obstacle to apply this idea to engineering design problems, especially those with a computationally expensive objective function. The obstacle is that for a new design without enough historical data of designs and operations, generation of design selection rules needs the evaluation of the objective function of all designs in a design library. In order for the design library to be representative for the large volume of design alternatives, one will have to include enough number of designs in the library, which could be too many to be computationally affordable for generating the design selection rules.

Igusa *et al.* [3] proposed a more sophisticated procedure, which circumvents frequent evaluation of an expensive objective function. They employed a much simpler feature function together with a clustering method to reduce the number of designs whose objective function need to be evaluated for the generation of a classification tree.

In this paper, following the general concept proposed by Igusa *et al.* [3], we develop a data-mining aided design optimization method for the aforementioned multi-station fixture layout design. The method includes the following components: (1) a uniform-coverage selection method, which chooses design representatives among original design alternatives for a non-rectangular design space; (2) feature functions of which evaluation is computationally economical as the surrogate of a design objective function; (3) a clustering method, which generates a design library based on the evaluation of feature functions instead of an objective function; (4) a classification method to create the design selection rules. The design effectiveness and efficiency of the proposed method is demonstrated by comparison with SA and local optimization methods.

2. DATA-MINING AIDED OPTIMAL DESIGN METHOD

2.1 Design Objective and Overview of the Method

The goal of fixture layout design is to find an optimal layout so that assembly dimensional variability is insensitive to variation inputs from fixture locators. A linear variation propagation model has been developed in [4] to link the product dimensional deviation (measured at M_1-M_{10}) to fixture locator deviations at P_1-P_8 on three assembly stations. Based on the variation model, a sensitivity index S was developed in [5] as a non-linear function of the coordinates of fixture locators, represented by the 16×1 parameter vector $\boldsymbol{\theta} \equiv [X_1 \ \ Z_1 \ \ \cdots \ \ X_8 \ \ Z_8]^T$, where X_i and Z_i is the pair of coordinates of locator P_i. Using this notation, the fixture layout design is to find a set of $\boldsymbol{\theta}$ that minimizes the sensitivity S while satisfying the geometric constraint $G(\cdot)$, i.e.,

$$\min_{\boldsymbol{\theta}} \ S(\boldsymbol{\theta}) \ \ subject \ \ to \ \ G(\boldsymbol{\theta}) > 0 \cdot \qquad (1)$$

Equation (1) actually captures a general formulation of a non-linear optimization problem. In the above formulation, without loss of generality, we present a minimization problem. A maximization problem can be solved in the same fashion. Generally, the objective function $S(\cdot)$ in an engineering optimal design is complicated. The efficiency of an optimal design algorithm can be loosely determined by how often $S(\cdot)$ is evaluated -- we denote by T the computer time of evaluating $S(\cdot)$ once.

There is virtually no efficient method, allowing us to directly optimize over the huge volume of original design alternatives, such as the as many as 8.6×10^{20}combinations in the fixture layout design. The proposed method will start with extracting *design representatives* from original design alternatives. However, it is often the case that the design representatives, although much less than the original design alternatives, are still too many to be used as the design library. In this paper, we use a clustering method with a set of computationally simple feature functions to facilitate the creation of a design library. This procedure will allow us to eventually have an affordable size of designs as a training dataset in a design library. The overall framework is illustrated in Figure 2.

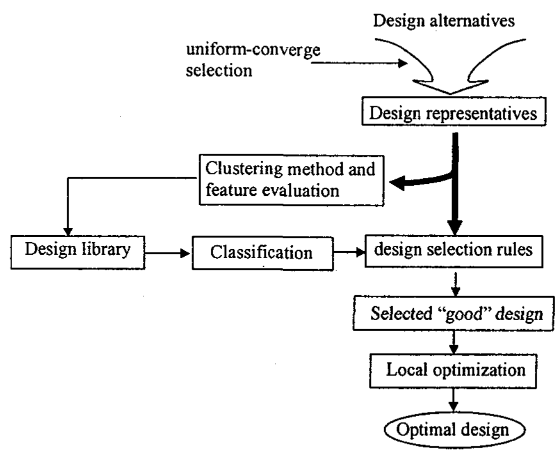

Figure 2. Data-mining aided design optimization

2.2 Uniform Selection of Design Representatives

Unless one has profound knowledge of which part in a design space is preferred in such a selection, a safer way for the selected design being good representatives of the original design set is to select them from a design space as evenly as possible. Igusa *et al.* [3] suggested to randomly select design representatives from the set of design alternatives. The problem of random selection is that probabilistic uniformity does not guarantee an evenly geometric converge in a design space. When the design space is of a high dimension and the sample size is relatively small (e.g., 2,000 chosen from 8.5×10^8 alternatives in Igusa's case), the selected sample will typically cluster in a small area and fail to cover large portions of the design space.

A space-filling design, widely used in computer experiments [6], aims to spread design points evenly throughout a design region and appears to fit well into our purpose of design representative selection. A space-filling design is usually devised using Latin Hypercube Sampling (LHS) [7] or using a uniformity criterion from the Number-Theoretic Method (NTM) [8].

Step 1. Uniformly discretize the candidate design space on each plane using the 10-mm resolution.

Step 2. One each panel, the first locator is sequentially chosen to be at those locations from the discretization process. Once the first locator is selected, the second locator is randomly selected on the same panel among the locations whose distance from the first locator is greater than the half of panel size ($d_0/2$). Denote by $\Omega_j^{(0)}$ the resulting candidate locator set for panel j and by n_j the number of locator pairs included in $\Omega_j^{(0)}$.

Step 3. For i=1 ... max(n_j), randomly select one locator pair from $\Omega_j^{(i-1)}$ for j = 1, 2, 3, 4 *without replacement* and combine these four locator pairs as one design representative for the multi-station assembly. Whenever a $\Omega_j^{(i-1)}$ becomes empty, simply reset $\Omega_j^{(i-1)} = \Omega_j^{(0)}$.

In Step 2, the uniformity of the first locator on each panel is a result of the uniform discretization. However, the uniformity of the second locator is not directly controlled since it is from a simple random sampling. The second locator is chosen to be at least $d_0/2$ away from the first locator because of the aforementioned between-locator distance constraint. After Step 2, the set $\Omega_4^{(0)}$ has the largest number of locator pairs, n_4=3,496. Step 3 actually performs a stratified sampling to generate locator combinations. The stratified sampling will go over $\Omega_4^{(0)}$ once but will have to go over $\Omega_4^{(0)}$ for panel j=1,2,3 multiple times. That is the reason behind the reset of a $\Omega_j^{(i-1)}$ when it is empty. Eventually, a total of n_4=3,496 combinations of locators is generated as design representatives.

2.3 Feature and Feature Function Selection

In order to avoid direct and frequent evaluations of objective function $S(\cdot)$, we use a set of feature functions to characterize the system performance. A feature function maps an engineering system to a feature, which is tied to the design objective. For example, the distance between two locators in the fixture design can be considered as a feature. Generally, any physical quantity that is potentially tied to the design objective can be used as a feature. The set of feature functions is actually a surrogate of the design objective function.

$F_1(\boldsymbol{\theta})$=The largest value of between-locator distances;
$F_2(\boldsymbol{\theta})$=The second largest value of between-locator distances;
$F_3(\boldsymbol{\theta})$=The mean of between-locator distances;
$F_4(\boldsymbol{\theta})$=The second smallest value of between-locator distances;
$F_5(\boldsymbol{\theta})$=The smallest value of between-locator distances.

$F_6(\boldsymbol{\theta})$ = The largest value of distance change ratios;

$F_7(\boldsymbol{\theta})$ = The mean value of distance change ratios;

$F_8(\boldsymbol{\theta})$ = The smallest value of distance change ratios.

Please note that the calculation of the above eight feature functions is very economical and their definitions are also scalable.

2.4 Clustering Method

Using the feature functions as the surrogate of a design objective, the data-mining aided design method will cluster the design representatives into a few groups. Recall that clustering is to segment a heterogeneous population into a number of more homogeneous subgroups [9]. Empirical evidence shows that a clustering method can group the uniformly scattered design representatives so that the resulting clusters are adapted to local optimal areas [3]. Therefore, a group of design representatives after clustering can be loosely considered as a set of designs associated with a local response surface and its center will be around a local optimal point. For this reason, a design library can then be created using a few designs from individual local areas, which constitutes of much less number of designs.

2.5 Classification Method

We will perform classification on the dataset {$\mathbf{F}_i$, S_i} in the design library to generate the design selection rules. Local optimization can be used to evaluate a few designs chosen by the selection rules and yield the final optimal design. In many occasions, as we will see in Section 4, a local optimization method may not be necessary, i.e., a direct comparison among all the selected designs may have given us a satisfactory result.

2.6 Selection of K and J

One issue we left out in Section 2.4 is how to select K (cluster number) and J (seed design number), which are obviously related to both the optimal objective value a design can achieve and the time it consumes.

Unfortunately, a theoretical tie between the clustering result and the behavior of a response surface has not yet been established. Using the multi-station fixture design at hand, we will further investigate this problem through an experimental design approach. Two responses are chosen for a given combination of K and J, namely the smallest sensitivity value it finds (before a local optimization method is applied) and the time it takes. For this data-mining aided optimal design, the overall computation time can be approximately calculated by $T_0+KJ{\cdot}T+N_f T$, where T_0 is the time component independent of the choice of K and J and N_f is the number of designs in the selected "good" design set when the whole design representatives pass through the design selection rule. Component T_0 is also known as the overhead time due to uniform-coverage selection and clustering/classification processes. The second and third components are directly related to the times that the objective function is evaluated. For a given engineering design problem and a choice of K and J, the algorithm computation time will be largely determined by the third component, or equivalently, the value of N_f. For this reason, we use N_f as the second response variable.

3. PERFORMANCE COMPARISON AND DISCUSSION

In this section we compare the algorithm performance of our data-mining aided optimal design (before a local optimization is applied) with other optimization routines. Our design algorithm is implemented with K=9 and J=12, the optimal combination found in Section 2.6.

The performance indices for comparison include the lowest sensitivity value an algorithm can find and the time it consumes. The objective function for the assembly process in Figure 1 is not really an expensive one due to various simplifications we made in variation modeling. The T is only 0.018 seconds on a computer with a 2.20GHz P4 processor. In this study, we purposely use this objective function so that we afford to perform the exploration in Section 2. When a computationally inexpensive function is used, the overhead computing cost T_0 kicks in, which may blind us the benefit of the method for a complicated system with more expensive objective functions. In order to show that aspect, we also include the number of how many times the objective function is evaluated for comparison. When T is large, the time for function evaluation dominates the entire computation cost.

We implemented the above-mentioned optimization algorithms in MATLAB (for simplex search, we used the MATLAB function "*fminsearch*"). They are executed on the same computer. The average performance data of 10 trials are included in Table 1.

Table 1. Comparison of optimization methods

Optimization Methods	S	Time (sec.)	Time for function evaluation
Simplex search	6.825	73.8	3,200 T
SA (k_B =0.9)	3.831	542.8	28,503 T
SA (k_B =0.95)	3.979	259.5	13,606 T
Data-mining aided method	3.894	54.3	283 T

4. CONCLUDING REMARKS

This paper presents a data-mining aided optimal design method. The method is employed to facilitate the optimal design of fixture layout in a four station SUV side panel assembly process. Compared with other available optimization methods, the data-mining aided optimal design demonstrates clear advantages in terms of both the sensitivity value it can find (only 1.6% higher than what a SA found) and the computation time it consumes (shorter than a simplex search and one-tenth of what a SA takes). The benefit could be more obvious for a larger system with a computationally expensive objective function.

REFERENCE

1. Bertsimas, D. and Tsitsiklis, J., (1993), "Simulated Annealing," *Statistical Science*, Vol. 8, pp.10-15.
2. Schwabacher, M., Ellman, T. and Hirsh, H (2001). "Learning to set up numerical optimizations of engineering designs," *Data Mining for Design and Manufacturing, Kluwer Academic Publishers,* pp.87-125.
3. Igusa, T., Liu, H., Schafer, B. and Naiman, D. Q. (2003). "Bayesian classification trees and clustering for rapid generation and selection of design alternatives," *Proceedings of 2003 NSF DMII Grantees and Research Conference*, Birmingham AL.
4. Jin, J. and Shi, J. (1999). "State Space Modeling of Sheet Metal Assembly for Dimensional Control," *ASME Journal of Manufacturing Science & Engineering*, Vol. 121, pp. 756 – 762.
5. Kim, P. and Ding, Y. (2003). "Optimal design of fixture layout in multi-station assembly process," *IEEE Journal of Robotics and Automation*, accepted.
6. Santner, T.J., Williams, B.J., and Notz, W.I. (2003). *Design & Analysis of Computer Experiments*, Springer Verlag, New York.
7. McKay, M.D., Bechman, R.J., Conover, W.J. (1979) "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code." *Technometrics*, Vol. 21, 239-245.