NEAR OPTIMAL DESIGN OF FIXTURE LAYOUTS IN MULTI-STATION ASSEMBLY PROCESSES

A Dissertation

by

PANSOO KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2004

Major Subject: Industrial Engineering

NEAR OPTIMAL DESIGN OF FIXTURE LAYOUTS IN MULTI-STATION ASSEMBLY PROCESSES

A Dissertation

by

PANSOO KIM

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Yu Ding (Chair of Committee)	
Guy L. Curry (Member)	Amarnath Banerjee (Member)
Jyhwen Wang (Member)	Mark L. Spearman (Head of Department)

August 2004

Major Subject: Industrial Engineering

ABSTRACT

Near Optimal Design of Fixture Layouts in Multi-station Assembly Processes.

(August 2004)

Pansoo Kim, B.S., Pusan National University;

M.S., Pusan National University

Chair of Advisory Committee: Dr. Yu Ding

This dissertation presents a methodology for the near optimal design of fixture layouts in multi-station assembly processes. An optimal fixture layout improves the robustness of a fixture system, reduces product variability and leads to manufacturing cost reduction. Three key aspects of the multi-station fixture layout design are addressed: a multi-station variation propagation model, a quantitative measure of fixture design, and an effective and efficient optimization algorithm. Multi-station design may have high dimensions of design space, which can contain a lot of local optima. In this dissertation, I investigated two algorithms for optimal fixture layout designs. The first algorithm is an exchange algorithm, which was originally developed in the research of optimal experimental designs. I revised the exchange routine so that it can remarkably reduce the computing time without sacrificing the optimal values. The second algorithm uses data-mining methods such as clustering and classification. It appears that the data-mining method can find valuable design selection rules that can in turn help to locate the optimal design efficiently. Compared with other non-linear optimization algorithms such as the simplex search method, simulated annealing, genetic algorithm, the data-mining method performs the best and the revised exchange algorithm performs comparably to simulated annealing, but better than the others. A four-station assembly process for a sport utility vehicle (SUV) side frame is used throughout the dissertation to illustrate the relevant concepts

and the resulting methodology.

DEDICATION

TO MY PARENTS

ACKNOWLEDGMENTS

I would like to express my very sincere gratitude to Dr.Yu Ding, the chairman of my advisory committee, for his guidance, support, and advice during my research work. This work could not have been completed without his generosity and patience as well as his profound knowledge. I would also like to express my appreciation to Dr. Guy L. Curry, Dr. Amarnath Banerjee, and Dr. Jyhwen Wang for serving on my committee.

I would like to show my deepest appreciation to my parents, Mr. Bong-Gil Kim and Mrs. Kyung-Nam Kim, for their endless love and support and also to my wife, Jeongwon Yang and daughter Sangwan Kim for their endurance and support during my study.

NOMENCLATURE

P_i	locating pin i
NC_i	NC block i
M_i	key dimensional feature i
n_{PLP}	total number of fixture locators
$\boldsymbol{\theta}$	location of fixture locators, fixture layout
$\boldsymbol{\theta}_0$	initial or reference fixture layout
S	intuitive sensitivity index
$S(\cdot)$	sensitivity function, upper bound of sensitivity
$G(\cdot)$	geometric constraints
X_i, Z_i	coordinates of locator P_i
$\mathbf{x}_{i,k}$	product dimensional state of part i on station k
$\mathbf{u}_{j,k}$	random deviation of j^{th} fixture pair on station k
$\mathbf{x}_{\underline{k}}$	state of product on station k
$\mathbf{u}_k$	fixture deviation vector on station k
$\mathbf{y}_k$	product measurements at station k
$\mathbf{w}_k$	un-modeled error or higher order term
$\mathbf{v}_k$	observation error
δ	perturbation operator
α	orientation angle
$\mathbf{A}_k$	dynamic matrix, change of fixture layout from station k to station k+1

$\mathbf{B}_k$	input matrix of station k
$\mathbf{C}_k$	observation matrix of station k
$\mathbf{\Phi}$	state transition matrix, $\mathbf{\Phi}_{k,i} \equiv \mathbf{A}_{k-1}\mathbf{A}_{k-2}\cdots\mathbf{A}_i$, $k > i$ and $\mathbf{\Phi}_{i,i} \equiv \mathbf{I}$
$\mathbf{D}$	fixture design information, $\mathbf{D} \equiv [\mathbf{C}_N\mathbf{\Phi}_{N,1}\mathbf{B}_1 \quad \mathbf{C}_N\mathbf{\Phi}_{N,2}\mathbf{B}_2 \quad \cdots \quad \mathbf{C}_N\mathbf{B}_N]$
$\hat{\mathbf{y}}$	fixture-induced production deviation
$\mathbf{u}$	variation input, $\mathbf{u}^T \equiv [\mathbf{u}_1^T \quad \cdots \quad \mathbf{u}_N^T]$
$\{\lambda_i\}_{i=1}^p$	eigenvalues of $\mathbf{D}^T\mathbf{D}$, where p is the column number of $\mathbf{D}$
tr(·)	trace of a matrix
det(·)	determinant of a matrix
N_c	total number of candidate locations
Δ	improvement in the $S(\boldsymbol{\theta})$
d_0	the radius of each panel
Ω_i	locator pair set for panel i
$F(\cdot)$	feature function
$\mathbf{F}$	vector of its feature functions
L_m	between locator distances for m locator pairs
N_k	the number of elements in cluster k
$C(\cdot)$	cluster which it belongs
$\mathbf{m}_k$	cluster center
$\|\cdot\|$	vector 2-norm
K	the number of cluster

J	the number of designs selected from each cluster
N_f	the number of designs in the selected good design set
T_0	overhead time
N_t	total number of function evaluation
N_r	the number of designs in design representatives
k_{B}	cooling ratio, the Boltzmann's constant
M	population size
P_c	recombination rate
P_m	permutation rate

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION*

Dimensional quality control is one of the major challenges in discrete-part manufacturing. In the automotive and aviation industries, for instance, dimensional problems contribute to about two-thirds of all quality-related problems during a new product launch (Shalon, Gossard, Ulrich and Fitzpatrick, 1992; Ceglarek and Shi, 1995). Automotive and aircraft assembly processes are typical multi-station panel assembly processes in which fixtures are used extensively to provide physical support and to coordinate references to parts and subassemblies. As a result, fixture design greatly affects the dimensional accuracy of the final products.

Figure 1 illustrates a typical 3-2-1 fixture used in panel assembly processes. It consists of two locating pins, P_{4way} and P_{2way}, and three NC blocks, $NC_{1\text{-}3}$. The two locating pins constrain three degrees of freedom in the X-Z plane, where the 4-way pin controls part motion in both X- and Z- directions and the 2-way pin controls part motion in the Z-direction. Three NC blocks constrain the other degrees of freedom of the workpiece. When a workpiece is non-rigid, more than three NC blocks may be needed in order to reduce part deformation. An *n*-2-1 fixture layout, denoted by

---

This dissertation follows the style of *Technometrics*.

$\{P_{4way}, P_{2way}, NC_i, i = 1,2, \ldots ,n\}$, is a more generic setting in panel assembly processes.

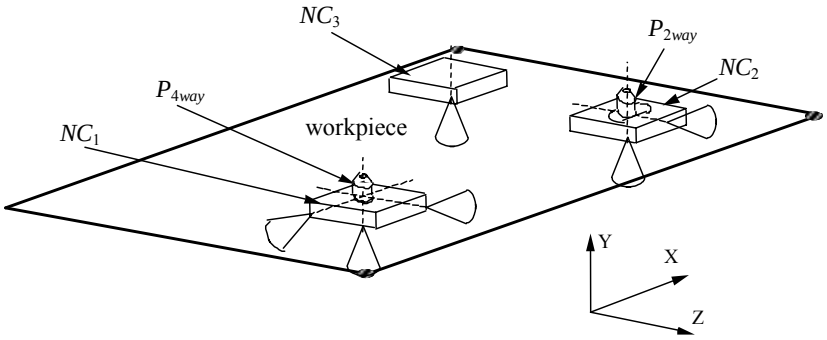


Figure 1 Illustration of a 3-2-1 fixture

Product dimensional variations resulting from locating pins and NC blocks are generally different: variation from locating pins causes a (global) rigid-body motion of a workpiece while variation from NC blocks can cause (local) deformations. In this study, we are more interested in the global variation phenomena related to locating pins. Hence, we use $\{P_{4way}, P_{2way}\}$ as a simplified representation of an *n*-2-1 fixture layout.

A real panel assembly process always consists of multiple assembly stations. For example, an assembly line in an automotive body-shop could involve up to 80 stations to assemble 150 to 250 sheet-metal parts into the structure of a vehicle. Consider the assembly process of the side frame of a sport utility vehicle (SUV) in Figure 2. The final product, the *inner-panel-complete*, is comprised of four

components: A-pillar, B-pillar, rail roof side panel, and rear quarter panel, which are assembled on three stations (Stations I, II, III). The final assembly is then inspected at Station IV (M_1-M_{10} marked in Figure 2(d) are key dimensional features).

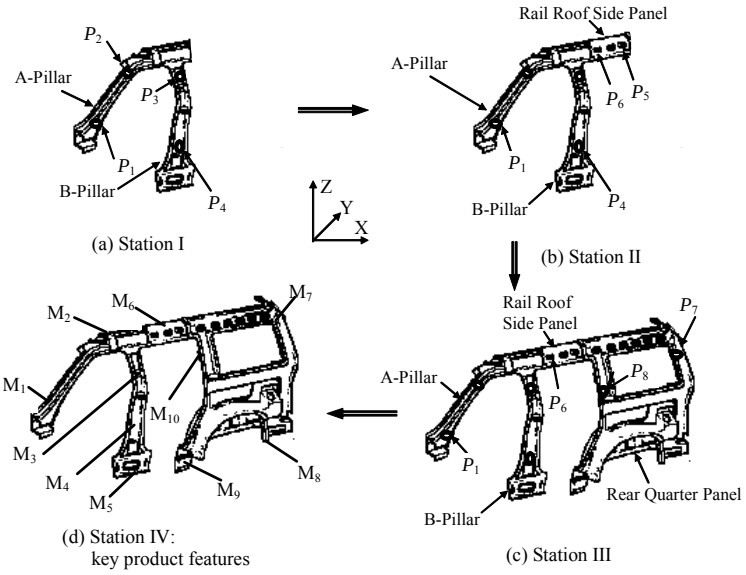


Figure 2 Assembly process of a SUV side frame

In such a multi-station process, the aforementioned 3-2-1 fixture is used on every station to ensure product dimensional accuracy. In Figure 2, P_1–P_8 are the so-called principal locating points (PLP), which are the pinholes used to position a part on an assembly station. However, the same symbols are also used interchangeably to represent locating pins. Thus, a fixture layout in a multi-station process can be represented using these PLPs as follows

$$\{\{P_1, P_2\}, \{P_3, P_4\}\}_{\mathrm{I}} \rightarrow \{\{P_1, P_4\}, \{P_5, P_6\}\}_{\mathrm{II}} \rightarrow \{\{P_1, P_6\}, \{P_7, P_8\}\}_{\mathrm{III}} \rightarrow \{\{P_1, P_8\}\}_{\mathrm{IV}},$$

where the assembly process starts from Station I (indicated by the subscript) and the arrow represents a transition from one station to the next. As an example, $\{\{P_1, P_4\}, \{P_5, P_6\}\}_{\mathrm{II}}$ means that at Station II the first workpiece, the sub-assembly "A-pillar+B-pillar," is located by P_1 and P_4 and the second workpiece, the rail roof side panel, is located by P_5 and P_6.

I.1 Problem definition

In a multi-station assembly process, dimensional variations could originate from fixture elements on every station, propagate along the production line, and accumulate on the final assembly. The dimensional quality of the final assembly depends on: (i) input variation level, and (ii) process sensitivity to variation inputs. The former issue is usually addressed by tolerance design. This dissertation focuses on the second issue, i.e. an optimal design of fixture layouts in a multi-station assembly process, so that the process is insensitive to variation input.

Fixture layout design in a multi-station process determines the locations of fixtures on every assembly station. Since the problem is equivalent to the determination of PLP locations on an assembly product, three aspects should be addressed: (1) a variation propagation model that links fixture variation inputs on every station to product dimensional variations; (2) a quantitative design measure that benchmarks the sensitivity of different fixture layouts, and (3) optimization algorithms

that find the optimal fixture layouts.

Research efforts have been made for the first aspect of fixture layout design, which is to establish a linear variation propagation model that links the product dimensional deviation (measured at M_1-M_{10}) to fixture locator deviations at P_1-P_8 on three assembly stations (Jin and Shi 1999; Ding, Ceglarek and Shi 2000; Camelio, Hu and Ceglarek 2001). This dissertation focuses on the second and third aspects of fixture layout design. Based on the variation model, a sensitivity index S will be developed as a non-linear function of the coordinates of the fixture locators. The optimization algorithm will use this sensitivity index for determining a robust fixture system in a multi-station panel-assembly process.

The design parameters are the locations of fixture locators, denoted as $\boldsymbol{\theta} = [X_1 \quad Z_1 \quad \cdots \quad X_{n_{PLP}} \quad Z_{n_{PLP}}]^T$, where n_{PLP} is the total number of fixture locators, e.g. $n_{PLP} = 8$ for the process in Figure 2 and X_i and Z_i are the coordinates of locator P_i. Using this notation, the optimal fixture layout design problem attempts to find a set of $\boldsymbol{\theta}$ that minimizes sensitivity S while satisfying the geometric constraint $G(\cdot)$:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \quad S(\boldsymbol{\theta}) \\ &\textit{subject to} \quad G(\boldsymbol{\theta}) \geq 0. \end{aligned} \tag{1}$$

I.2 Prior work on fixture layout design problem

Earlier research on fixture design employed kinematical and mechanical analysis to explore accessibility, detachability, and location uniqueness of a fixture, aiming at the automatic generation of fixture layouts (Asada and By, 1985). Heuristic

algorithms were developed for the automatic generation of fixture configurations (Chou, Chandru and Barash, 1989; Ferreira, Kochar, Liu and Chandru, 1985). Trappery and Liu (1990) summarized the research before 1990 on fixture-design automation and a more recent summary can be found in Chapter 1 of Cai, Hu and Yuan's (1997) work.

These fixture designs methods are considered *deterministic* approaches because they consider neither random manufacturing errors of fixture elements nor workpiece positioning errors induced by fixturing operations. Since a workpiece or a fixture element is unavoidably subject to manufacturing errors, researchers studied the problem of robust fixture design in a stochastic environment.

One branch of robust fixture design aims at finding optimal fixture positions that minimize the deflection of a compliant workpiece under a working load. This research usually does not consider the manufacturing errors of fixture elements. However, fixture-related local deformation and micro-slippage are considered error sources (DeMeter, 1995; Melkote, 1995).

Another branch of robust fixture design is known as the *variational* approach because it considers fixture error or workpiece surface error and tries to find an optimal fixture layout that makes the positioning accuracy of a workpiece insensitive to input errors (Cai et al., 1997; Wang, 2000; Wang and Pelinescu, 2001; Soderberg and Carlson, 1999). Variational fixture design often starts with developing a sensitivity measure that characterizes the robustness of a fixture system; this sensitivity measure is determined by fixture layout and is independent of fixture error

input. Essentially, the smaller the sensitivity, the more robust the fixture system is. For example, Wang (2000) maximized the determinant of the information matrix (D-optimality), which is the inverse of the sensitivity matrix, and Cai et al. (1997) minimized the Euclidean norm of their sensitivity matrix. Meanwhile, heuristic or rule-based methods have also been developed for designing robust fixture layout (Soderberg and Carlson, 1999). Research work by Rong, Li and Bai (1995), Choudhuri and DeMeter (1999), Ding, Ceglarek and Shi (2002a) and Carlson (2001) is also relevant in the sense that it provides variation/tolerance analysis of a fixture system while the difference is that the issue of fixture synthesis is not addressed.

Past variational fixture designs were conducted mainly at the single-machine level rather than at the multi-station system level with the fixture layout optimization being limited to a single workstation. Based on our description of the 3-2-1 fixture used in panel assembly processes, it is apparent that a station-wise optimization of fixture layouts is different from a system-wide optimization. Suppose that one had optimized the positions of P_1, P_2, P_3, and P_4 on Station I. (Note that P_1 and P_4, the PLPs on A-pillar and B-pillar, respectively, will be reused on Station II.) Thus, when a station-wise optimization is carried out on Station II, one could choose to optimize all fixtures on Station II as if P_1 and P_4 were not optimized on Station I or he could keep the optimized positions of P_1 and P_4 and only optimize the fixture layout (P_5 and P_6) that supports the newly added part. Obviously, neither approach will lead to an overall optimal fixture-layout in a multi-station process.

Research on multi-station fixture optimization is limited because of the

inherent difficulty resulting from multi-station variation modeling, development of design criteria, as well as the choices of efficient optimization methods. Recent research addresses the issue of multi-station variation modeling using either a station-indexed state space model (Jin and Shi, 1999; Ding et al., 2000; Camelio et al., 2001; Zhou, Huang and Shi, 2003) or a datum-machining surface relationship graph (DMG) (Rong and Bai, 1996). Xiong, Rong, Koganti, Zaluzec and Wang (2002) further studied non-linear fixturing models for variation prediction in multi-station aluminum welded assemblies. Based on linear variation models developed for panel assembly processes (Jin and Shi, 1999; Ding et al., 2000; Camelio et al., 2001), this dissertation will continue the development of design criteria and optimization algorithms for multi-station fixture design.

One more note is on fixture *diagnosis* (Ceglarek and Shi, 1996; Chang and Gossard, 1998; Carlson, Lindkvist and Soderberg, 2000; Ding, Ceglarek and Shi, 2002b), which is to pinpoint malfunctioning fixtures based on in-line measurements from Optical CMMs. It is apparent that fixture diagnosis is an *in-line* technique that complements the *off-line* fixture design method. It is not surprising that both types of research share the common theoretical background of variation modeling and analysis. Overall, the methodologies reviewed in this chapter are summarized in Table 1.

Table 1 Comparison of fixture-design methodologies

Problem Domain			Methodologies	
Fixture Design	Deterministic		Asada and By (1985), Chou et al. (1989), Ferreira et al. (1985), Trappey and Liu (1990)	
	Robust Design for Minimal Deflection		Menasa and DeVries (1991), Rearick et al. (1993), DeMeter (1995), Melkote (1995), Hockenberger and DeMeter (1996), Cai and Hu (1996), Huang and Hoshi (1999)	
	Variational Robust Design	Single Station	Cai et al. (1997), Wang (2000), Wang and Pelinescu (2001), Soderberg and Carlson (1999), Rong et al. (1995), Choudhuri and DeMeter (1999), Carlson (2001)	
		Multi-Station	Modeling & Analysis	Rong and Bai (1996), Jin and Shi (1999), Camelio et al.(2001), Ding et al. (2000,2002), Zhou et al. (2003), Xiong et al. (2002)
			Fixture Optimization	*To be presented in this research*

I.3 Outline of the dissertation

Figure 3 shows an outline of this dissertation. Following this introduction, Chapter II presents the variation propagation model and explains the major variation phenomena in a panel assembly process. Chapter II also presents the selection of design measures. The revised exchange algorithms, illustrated by solving the fixture-layout in the SUV side-frame assembly process, are presented in Chapter III. Chapter IV presents the data-mining method that can help to find an optimal design with higher efficiently. These two algorithms and a few existing non-linear optimization algorithms, such as the genetic algorithm and simulated annealing, are compared in Chapter V. Finally, we conclude this dissertation in Chapter VI.

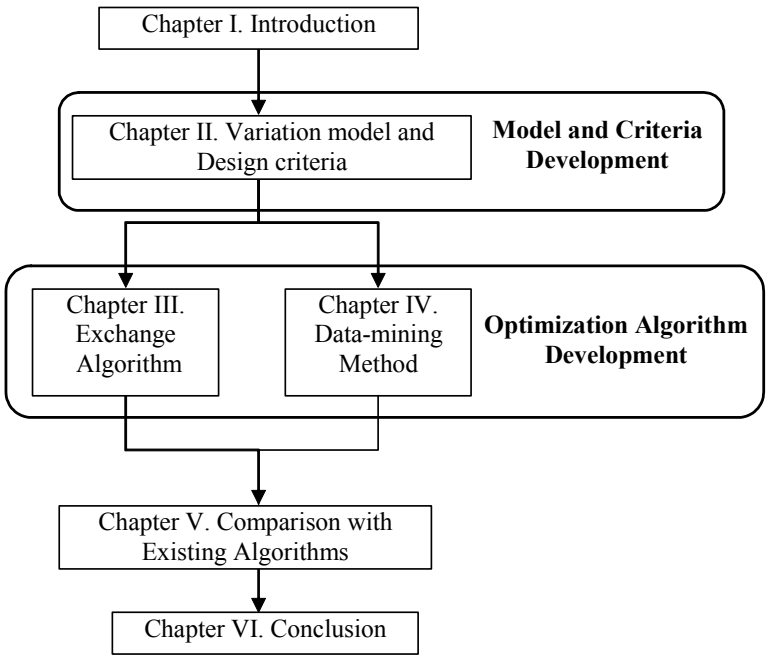


Figure 3 Outline of the dissertation

CHAPTER II

VARIATION MODEL AND DESIGN CRITERIA

Dimensional variation models that link fixture variation to dimensional measurements have been developed using standard kinematics analysis (Paul, 1981). A few variation propagation models were recently developed for multi-station assembly processes using a state-space representation (Jin and Shi, 1999; Ding et al., 2000; Camelio et al., 2001). Since this model is an integral part of multi-station fixture design, we briefly explain key elements in the modeling procedure and then present a general model structure. A 2D assembly process is modeled in this chapter and based on the variation model, a sensitivity index S will be developed as a non-linear function of the coordinates of fixture locators. Prior to introducing this index, a singular property of the suggested model which affects the selection of S is explained; the index is then compared with other typically used criteria.

II.1 State space variation model

There are two major fixture-related variation sources, as illustrated in Figure 4. One is the variation contributed by fixture locators on station k (Figure 4(a)) and the other is the variation induced when a sub-assembly is transferred to the next station where a different fixture layout is used to position the sub-assembly (Figure 4(b)).

The modeling procedure starts with an individual station k. Denote the

product dimensional state of part i on station k as $\mathbf{x}_{i,k} = [\delta X_{i,k} \quad \delta Z_{i,k} \quad \delta\alpha_{i,k}]^T$, which are the deviations associated with its three degrees of freedom, where δ is the perturbation operator and α is the orientation angle. Suppose that this part is located by the j^{th} fixture pair $\{P_1, P_2\}$ on station k, whose random deviations are denoted as $\mathbf{u}_{j,k} = [\delta P_1(X) \quad \delta P_1(Z) \quad \delta P_2(Z)]^T$. Therefore, $\mathbf{x}_{i,k}$ can be related to $\mathbf{u}_{j,k}$ through a linearization,

$$\mathbf{x}_{i,k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \dfrac{1}{P_1(X) - P_2(X)} & \dfrac{1}{P_2(X) - P_1(X)} \end{bmatrix} \cdot \mathbf{u}_{j,k} + \mathbf{w}_{i,k} \quad , \tag{2}$$

where $P_1(X)$ and $P_2(X)$ are the nominal coordinates of locators P_1 and P_2 and $\mathbf{w}_{i,k}$ includes the un-modeled higher order terms.

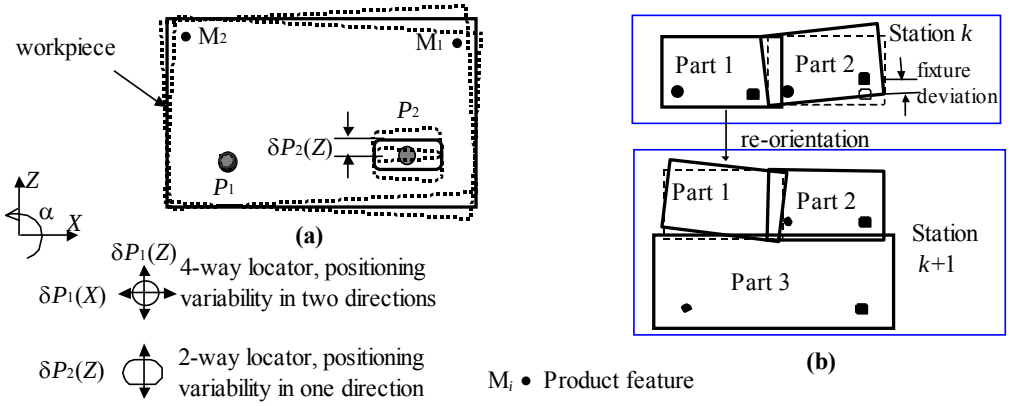


Figure 4 Fixture-related variation sources

Generally, the state of the product, which comprises n_p parts, is represented by $\mathbf{x}_k \equiv [\mathbf{x}_{1,k}^T \quad \cdots \quad \mathbf{x}_{n_p,k}^T]^T$. If part i has not yet appeared on station k, the corresponding $\mathbf{x}_{i,k}=\mathbf{0}$. The fixture deviation vector on station k is $\mathbf{u}_k = [\mathbf{u}_{1,k}^T \quad \cdots \quad \mathbf{u}_{n_k,k}^T]^T$, where n_k is the number of fixture pairs on station k. Product measurements at station k are included in $\mathbf{y}_k$. For the example in Figure 4(a), $\mathbf{y}_k = [\delta M_1(X) \quad \delta M_1(Z) \mid \delta M_2(X) \quad \delta M_2(Z)]^T$, which are the deviations associated with product features M_1 and M_2.

The basic idea of a state space variation model is to consider a multi-station process as a sequential system but replace the time index in a traditional state space model with a station index. For the process in Figure 5, the station-indexed state space model can be expressed as

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \quad \text{and} \quad \mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k, \qquad k \in \{1, 2, \ldots, N\}, \tag{3}$$

where N is the number of stations and $\mathbf{v}_k$ represents measurement noises. In this variation model, $\mathbf{B}_k$ models the effect of fixture variation ($\mathbf{u}_k$) on the product dimensional state ($\mathbf{x}_k$). It aggregates transformation matrices, each of which is similar to the one in Equation (2), for modeling all n_k fixture pairs. Matrix $\mathbf{C}_k$ includes the information of key product features (the number and locations of those features on station k). In the process described in Figure 2, $\mathbf{C}_{1,2,3}=\mathbf{0}$ and $\mathbf{C}_4\neq\mathbf{0}$ because key product features are measured only on Station 4 after assembly operations on Stations 1, 2, 3.

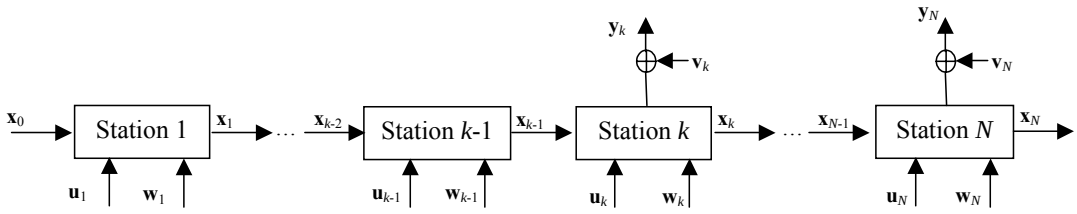


Figure 5 Diagram of a multi-station manufacturing process

Finally, we summarize the physical interpretation of **A**, **B**, and **C** in Table 2, where $\mathbf{\Phi}_{k,i} \equiv \mathbf{A}_{k-1}\mathbf{A}_{k-2}\cdots\mathbf{A}_i$, $k > i$ and $\mathbf{\Phi}_{i,i} \equiv \mathbf{I}$, and include a few more remarks regarding the state space variation model as follows.

Table 2 Interpretation of system matrices

Symbol	Name	Relationship	Interpretation	Assembly Task
A	Dynamic matrix	$\mathbf{x}_{k-1} \xrightarrow{\mathbf{A}_{k-1}} \mathbf{x}_k$	Changes of fixture layout between two adjacent stations	Assembly transfer
Φ	State transition matrix	$\mathbf{x}_i \xrightarrow{\mathbf{\Phi}_{k,i}} \mathbf{x}_k$	Changes of fixture layout across multiple stations	Assembly transfer
B	Input matrix	$\mathbf{u}_k \xrightarrow{\mathbf{B}_k} \mathbf{x}_k$	Fixture layout at station *k*	Part positioning
C	Observation matrix	$\mathbf{x}_k \xrightarrow{\mathbf{C}_k} \mathbf{y}_k$	Key product features at station *k*	Inspection

Remark 2.1. The state space variation model in this dissertation assumes a linear model structure. We acknowledge that its applications are limited to linear systems where the magnitude of fixturing errors is much smaller than the distance between

locators and when the process error is not strongly coupled with the fixturing error. Non-linearity could result from strong error-coupling and a relatively large fixturing error, both of which are cases that have been addressed in recent work (Carlson, 2001; Xiong et al., 2002).

Remark 2.2. Because we are more interested in the global variation resulting from locating pins, we assume that the NC blocks are not the major variation contributors and thus modeled only a 2D product. In situations when the NC blocks indeed significantly affect the assembly variation, a 3D locating model is more appropriate. State space models with the same structure but different matrix dimensions and parameters were used to model complicated 3D processes, e.g., the 3D machining model in Zhou et al. (2003). It should be noted that the subsequent development of the fixture design criteria and optimization methods are generally applicable to any linear system model instead of depending on particular parameters values or matrix dimensions.

Remark 2.3. In this study, the variation model for a single station is implemented to address the point geometry of the locating contacts for a fixture pair. However, products with complicated surface profiles are located using a greater number of fixture elements and product quality may also be affected by local surface properties of the locator-workpiece system. Researchers have recently spent efforts (Wang, 2000; Wang and Nagarkar, 1999; Wang, Liu and Pelinescu, 2003) to address these problems as they may be critical to meet high-precision requirements in fixturing small parts with complex geometry. The resulting models by Wang (2000), Wang and

Nagarkar (1999), and Wang et al. (2003) adopt a linear structure, which makes it less difficult to incorporate them in the state space model. For example, the fixture-quality relations for a more general product surface, modeled by Equation (8) in Wang and Nagarkar (1999) or Equation (5) in Wang (2000), are mathematically equivalent to the term $\mathbf{B}_k\mathbf{u}_k$ in Equation (3) so that $\mathbf{B}_k\mathbf{u}_k$ can be simply replaced by these relations. The local fixture contact properties modeled by Equation (28) in Wang et al. (2003) cannot directly replace $\mathbf{B}_k\mathbf{u}_k$, though, because they are expressed in velocity and not in displacement (or deviation). In that particular case, either the state vector should be augmented to include both velocity and deviation, as it is usually expressed in dynamic state space models, or a model for deviation from the integral of Equation (28) in Wang et al. (2003) should be used.

II.2 Singularity of the model

One difference between a multi-station variation model and a single-station model is the existence of matrix **A** that links product states (**x**) across different stations. Matrix **A** depends on fixture layouts on two adjacent stations. The procedure to determine **A** is conceptually similar to that of determining **B** or **C**, but is algebraically complicated (for more details, please refer to Jin and Shi, 1999; Ding et al., 2000; or Camelio et al., 2001). If there is no change in fixture layouts across stations, **A** simply becomes an identity matrix (e.g. the process described by Mantripragada and Whitney, 1999), and then the multi-station model in Equation (3) becomes a simple summation of multiple single-station models.

In the multi-station panel assembly process described in Chapter I, a change in fixture layouts occurs when the sub-assembly proceeds to a new station. Figure 4(b) illustrates the effect due to the change in fixture layouts; it results in a re-orientation of the sub-assembly. If there were fixture deviations in previous stations, the reorientation-induced error could happen to a part, even if fixtures at the current station are free of error (e.g. part 1 in Figure 4(b)).

This re-orientation is almost unavoidable for a multi-station panel-assembly process because a subset of PLPs is necessary to re-position a sub-assembly on a downstream station. Due to this re-orientation effect, **A** in the multi-station variation model takes a structure other than an identity matrix. More importantly, and maybe surprisingly, **A** is singular throughout the entire process. This singularity issue was identified for a multi-station assembly process in Carlson et al.'s (2000) work .

We present an intuitive explanation for why **A** is singular when fixture layouts change across stations. Consider the simple example in Figure 6, where several possible fixture errors on an upstream station could have caused the same resulting pattern of part deviation.

When this resulting deviation pattern is observed on Station k+1 (Figure 6(a)), the faulty fixture pair on Station k causing the deviation pattern could be either fixture-pair one (Figure 6(c)), fixture-pair two (Figure 6(b)), or both fixture pairs (Figure 6(d)). Assembly deviation at one station is related to deviation incurred at the previous station through matrix **A**, i.e., $\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k$, by neglecting other terms. With that in mind, given $\mathbf{x}_{k+1}$, there is no unique solution for $\mathbf{x}_k$ because of ambiguity and

we can conclude that $\mathbf{A}_k$ is singular. The singularity of matrix **A** is a general problem existing in panel assembly processes and will affect our choice of design criterion during later development.

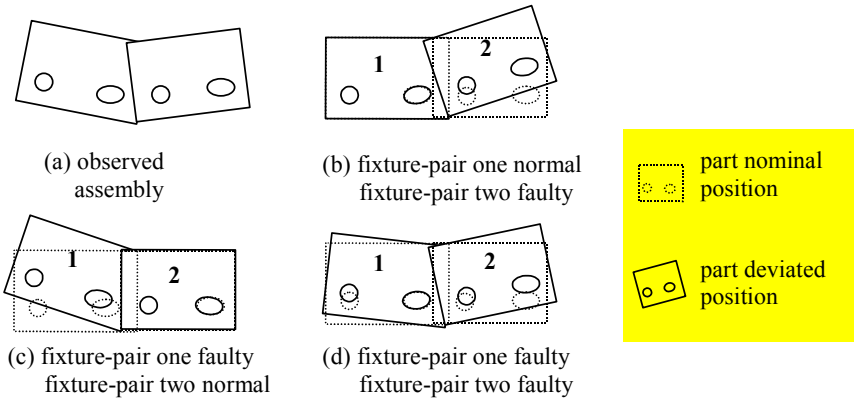


Figure 6 Singularity of **A** due to re-orientation

Following the modeling procedure in Jin and Shi (1999) and Ding et al. (2000), a state-space variation model was developed for the four-station assembly process of the SUV side frame in Figure 2. In this model, the fixture used on Station IV is considered well maintained and calibrated with much higher repeatability than those on a regular assembly station. Thus, fixture locators on the measurement station are assumed free of error, i.e. $\mathbf{u}_4=\mathbf{0}$, while deviation inputs from fixtures on three assembly stations, $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$, are included. Thus, the state space model is

$$\begin{cases} \mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k, k = 1,2,3, \quad \mathbf{x}_4 = \mathbf{A}_3\mathbf{x}_3 + \mathbf{w}_4, \\ \mathbf{y}_4 = \mathbf{C}_4\mathbf{x}_4 + \mathbf{v}_4, \end{cases} \tag{4}$$

where $\mathbf{x}_0$ represents product error resulting from the part-fabrication process (which is a stamping process for panel assembly) prior to assembly. Numerical expressions of **A**'s, **B**'s, and **C**'s are included in the Appendix. It is easy to verify that $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ are all singular.

II.3 Design criteria

We first reformulate the state space model in Equation (3) into an input-output linear model by eliminating all intermediate state variables $\mathbf{x}_k$. We have

$$\mathbf{y}_N = \Sigma_{k=1}^{N} \mathbf{C}_N \mathbf{\Phi}_{N,k} \mathbf{B}_k \mathbf{u}_k + \mathbf{C}_N \mathbf{\Phi}_{N,0} \cdot \mathbf{x}_0 + \Sigma_{k=1}^{N} \mathbf{C}_N \mathbf{\Phi}_{N,k} \mathbf{w}_k + \mathbf{v}_N \ . \tag{5}$$

In this fixture design problem, our focus is on the first term in the above equation, $\Sigma_{k=1}^{N} \mathbf{C}_N \mathbf{\Phi}_{N,k} \mathbf{B}_k \mathbf{u}_k$, because it represents fixture error inputs from all N stations. Therefore, we simplify Equation (5) as

$$\hat{\mathbf{y}} \equiv \mathbf{D}\mathbf{u} = \Sigma_{k=1}^{N} \mathbf{C}_N \mathbf{\Phi}_{N,k} \mathbf{B}_k \mathbf{u}_k \,, \tag{6}$$

where $\mathbf{D} \equiv [\mathbf{C}_N \mathbf{\Phi}_{N,1} \mathbf{B}_1 \quad \mathbf{C}_N \mathbf{\Phi}_{N,2} \mathbf{B}_2 \quad \cdots \quad \mathbf{C}_N \mathbf{B}_N]$, $\mathbf{u}^T \equiv [\mathbf{u}_1^T \quad \cdots \quad \mathbf{u}_N^T]$, and $\hat{\mathbf{y}}$ is the fixture-induced product variation. Subscript N is dropped from $\hat{\mathbf{y}}$ hereafter without causing ambiguity. For the model in Equation (4), because $\mathbf{u}_4$ is assumed zero, $\mathbf{D} = [\mathbf{C}_4 \mathbf{\Phi}_{4,1} \mathbf{B}_1 \quad \mathbf{C}_4 \mathbf{\Phi}_{4,2} \mathbf{B}_2 \quad \mathbf{C}_4 \mathbf{\Phi}_{4,3} \mathbf{B}_3]$.

The term $\hat{\mathbf{y}}^T \hat{\mathbf{y}}$, the sum of squares of product deviations, was used to benchmark the overall level of product-dimensional nonconformity; thus, product

quality is optimized if $\hat{\mathbf{y}}^T\hat{\mathbf{y}}$ is minimized. Given $\hat{\mathbf{y}}^T\hat{\mathbf{y}} = \mathbf{u}^T\mathbf{D}^T\mathbf{D}\mathbf{u}$, the problem is equivalent to minimizing $\mathbf{u}^T\mathbf{D}^T\mathbf{D}\mathbf{u}$. However, $\mathbf{u}^T\mathbf{D}^T\mathbf{D}\mathbf{u}$ is an input-dependent quantity and since our goal is to find a fixture layout in which product quality is insensitive to fixture variation input, we need a design criterion or a sensitivity index that is determined only by fixture design information (modeled by **D**) and is independent of variation input (represented by **u**).

For a single input-output pair, the sensitivity can be defined as $S_{i,j} = y_i / u_j$, where y_i is the i^{th} product feature and u_j is the j^{th} fixture error input. For the entire assembly system with multiple inputs and multiple features, an intuitive way to define the sensitivity index is as

$$S \equiv \frac{\hat{\mathbf{y}}^T\hat{\mathbf{y}}}{\mathbf{u}^T\mathbf{u}} = \frac{\mathbf{u}^T\mathbf{D}^T\mathbf{D}\mathbf{u}}{\mathbf{u}^T\mathbf{u}}. \tag{7}$$

The difficulty associated with this definition is that *S* is still input-dependent.

It is felt that $\mathbf{D}^T\mathbf{D}$ plays a determining role in the above definition, which has motivated researchers to define the sensitivity index using a measure of $\mathbf{D}^T\mathbf{D}$. Research conducted in experimental design has studied a similar problem and proposed several optimality criteria (Fedorov, 1972; Atkinson and Donev, 1992; Pukelsheim, 1993). The often used criteria include D-optimality (min det($\mathbf{D}^T\mathbf{D}$)), A-optimality (min tr($\mathbf{D}^T\mathbf{D}$)), and E-optimality (minimize the extreme eigenvalue of $\mathbf{D}^T\mathbf{D}$), where tr(·) and det(·) are the trace and the determinant of a matrix, respectively. These three measures are related to each other through the eigenvalues

of $\mathbf{D}^T\mathbf{D}$, $\{\lambda_i\}_{i=1}^{p}$, where p is the column number of $\mathbf{D}$. They can be expressed as

$$D_{opt}: \quad \det(\mathbf{D}^T\mathbf{D}) = \Pi_{i=1}^{p}\lambda_i; \; A_{opt}: \quad \mathrm{tr}(\mathbf{D}^T\mathbf{D}) = \Sigma_{i=1}^{p}\lambda_i; \text{ and } E_{opt}: \quad \lambda_{\min} \; or \; \lambda_{\max}. \quad (8)$$

The D-optimality criterion is the most widely used in experimental designs for following two reasons (Atkinson and Donev, 1992; Pukelsheim, 1993). (1) For experimental designs, this criterion has a clear interpretation. D-optimality is equivalent to minimizing the prediction variance from an estimated model or the variances of least-squares estimates of unknown parameters. (2) It possesses an invariant property under scaling, i.e. experiments can be designed using a group of standardized dimensionless variables (say, all variables are in [-1,1]) instead of the original physical variables. In fact, this D-optimality criterion was also used in solving problems of fixture design and sensor placement by Wang and his colleagues (Wang, 2000; Wang and Pelinescu, 2001; Wang and Nagarkar, 1999).

However, the singularity of matrix **A** in our variation propagation model (Equation 3) requires us to reconsider the design criterion. Because **A** is singular, the state transition matrix $\mathbf{\Phi}_{k,i}$ is also singular. It suggests that each term $\mathbf{C}_N\mathbf{\Phi}_{N,i}\mathbf{B}_i$ in **D** is less than full rank even if **C** and **B** matrices are of full rank. As a result, matrix **D** is less than full rank so that $\mathbf{D}^T\mathbf{D}$ is singular.

When $\mathbf{D}^T\mathbf{D}$ is singular, at least one of its eigenvalues is zero, i.e. $\det(\mathbf{D}^T\mathbf{D}) = 0$. Recalling the reason why **A** is singular (explained in Chapter II.2), we know that this singularity issue cannot be resolved by simply changing the positions of fixture locators on a station. It is an inherent problem caused by the fixturing mechanism in a

multi-station panel assembly process. This fact implies that even if we choose new positions for fixture locators, det($\mathbf{D}^T\mathbf{D}$) is always zero, therefore it is fair to conclude that det($\mathbf{D}^T\mathbf{D}$) is non-informative in this multi-station fixture design.

Given the singularity problem of design matrix **D**, we consider that either A-optimality or E-optimality is an informative criterion for multi-station fixture design. We recommend the use of E-optimality because it has a clearer physical interpretation. It is known (Schott, 1997) that

$$S \equiv \frac{\mathbf{u}^T\mathbf{D}^T\mathbf{D}\mathbf{u}}{\mathbf{u}^T\mathbf{u}} \le \lambda_{\max}(\mathbf{D}^T\mathbf{D}) \text{ for any } \mathbf{u}\neq\mathbf{0}. \tag{9}$$

That is, E-optimality, which minimizes $\lambda_{\max}(\mathbf{D}^T\mathbf{D})$, is equivalent to minimizing the upper sensitivity bound of the fixture system. This criterion can also be derived using the concept of matrix 2-norm. Defining the upper bound of sensitivity as $S(\boldsymbol{\theta})$, it follows the definition of matrix 2-norm (Schott, 1997) that

$$S(\boldsymbol{\theta}) \equiv \sup_{\mathbf{u}\neq\mathbf{0}} \frac{\mathbf{u}^T\mathbf{D}^T\mathbf{D}\mathbf{u}}{\mathbf{u}^T\mathbf{u}} = \|\mathbf{D}\|_2^2 = \lambda_{\max}(\mathbf{D}^T\mathbf{D}). \tag{10}$$

In other words, E-optimal is the square of the 2-norm of the design matrix **D**.

We cannot rule out the possible use of A-optimality in this multi-station fixture design problem. Since an eigenvalue of $\mathbf{D}^T\mathbf{D}$ represents the sensitivity level related to one particular input-output pair for a canonical variation model, tr($\mathbf{D}^T\mathbf{D}$) is the summation of sensitivities related to all input-output pairs, representing the overall sensitivity level of the fixture system. Using A-optimality can be considered for minimizing the summation of sensitivities.

Compared with A-optimality, E-optimality is conservative because it attempts to reduce the maximum sensitivity index. This conservativeness actually makes E-optimality more easily accepted by practitioners because the minimization of the maximum sensitivity is consistent with the Pareto Principle in quality engineering. Our experience with the automotive industry indicates the same tendency.

II.4 Discussion and summary

Based on our experience with this multi-station fixture design, we caution the use of D-optimality in general engineering system designs. Engineering system designs are different from experimental designs in many aspects. The differences could cause the advantages of using D-optimality in an experimental design to be inapplicable to an engineering design problem. The major differences include: (i) Engineering design problems are often accompanied by complex constraints, for example, the geometric constraints imposed by the shape of a part in the SUV side-frame assembly process. This type of complexity makes it almost impossible to design an engineering system based on a group of dimensionless standardized variables. In this regard, the invariant property of D-optimality becomes much less attractive to general engineering designs. (ii) The complexity of engineering systems often results in ill-conditioned systems with some eigenvalue of $\mathbf{D}^T\mathbf{D}$ close to zero or even singular systems (such as our multi-station fixture system). Since the purpose of D-optimality is to minimize the product of all eigenvalues, it is possible in the

presence of ill-conditioned systems that the near-zero eigenvalue is forced to become zero while leaving other eigenvalues uncontrolled as if a perfect D-optimal condition was achieved. Obviously, this is actually an undesirable result. This problem is less likely to occur, though, in an experimental design or to a well-posed system; see Wang and Nagarkar (1999) for a more detailed discussion. (iii) The physical interpretation of D-optimality in engineering system designs may not be as clear as in experimental designs. For instance, what det($\mathbf{D}^T\mathbf{D}$) represents in this fixture design problem is not obvious.

In the rest of this dissertation, we will use E-optimality criterion for determining a robust fixture system in a multi-station panel assembly process. Using the E-optimality, the optimization scheme in Equation 1 can then be expressed as

$$\begin{aligned} &\min_{\theta} \quad S(\boldsymbol{\theta}) \equiv \lambda_{\max}(\mathbf{D}^T\mathbf{D}) \\ &\textit{subject to} \quad G(\boldsymbol{\theta}) \geq 0. \end{aligned} \tag{11}$$

The initial or reference fixture layout as shown in Figure 2 is denoted as $\boldsymbol{\theta}_0$. It is straightforward to calculate $S(\boldsymbol{\theta}_0) = 5.397$.

CHAPTER III

INVESTIGATION OF EXCHANGE ALGORITHM

This chapter traces the development of the revised exchange algorithm. The basic exchange algorithm was developed to optimize the experimental design. First, the basic exchange algorithm and its limitations are briefly described. Then the three steps which relieve the complexity of the problem are introduced. Finally, the computational results from the basic exchange algorithm, the modified Fedrov algorithm and the revised-exchange algorithm are compared and resulting fixture layouts and properties are discussed.

III.1 Overview of exchange algorithm

The objective function $\lambda_{max}(\mathbf{D}^T\mathbf{D})$ is a non-linear function of design parameter $\boldsymbol{\theta}$, and Equation (11) thus states a constrained non-linear optimization problem. The performance of an optimization algorithm is often benchmarked by: 1) its effectiveness, measured by the closeness of its solution to the global optimum; and, 2) its efficiency, usually measured by the time it takes to find the optimal value. Unless the objective function is of a simple form such as a quadratic function (and our objective function is apparently not), the difficulty with non-linear optimization is that the global optimum is not guaranteed for almost all available algorithms without an exhaustive search.

A multi-station fixture design problem, when expressed in the format of Equation (11), might appear to be no different from a single-station fixture design. However, the challenge that a multi-station fixture design raises is that a much higher dimension design space will have to be explored. For example, even in the 2D four-panel SUV assembly process, we need to determine the positions of eight PLPs, which constitutes a sixteen-dimension design space. Consequently, this high dimension design space, embedding a lot of local optimums, makes a global optimality much more difficult and requires prohibitive computer time if an exhaustive search is used. Therefore, we soften our goal a bit in this dissertation. Instead of looking for *the* global optimum, we try to find an algorithm that yields a substantial improvement in our design criterion with a reasonable cost of computer-time.

In the research of optimal experimental design, exchange algorithms were developed to solve combinatorial optimizations based on various design criteria mentioned earlier, such as D-, E-, and A-optimality (Fedorov, 1972; Atkinson and Donev, 1992; Cook and Nachtsheim, 1980). Most of these algorithms are variants on the basic idea of an exchange, explained as follows. First, discretize the continuous design space to yield N_c candidate fixture-locator positions. Then, randomly select n_d locations from N_c candidate positions as an initial design and calculate $S(\boldsymbol{\theta})$ (in our problem, we actually already have an initial design, $\boldsymbol{\theta}_0$). In each exchange, do the following:

(EA1) for each one of the N_c candidate locations, calculate $S(\boldsymbol{\theta})$ as if the i^{th} location in

the current design was exchanged with the candidate location. Record the smallest $S_i(\boldsymbol{\theta})$ and the corresponding candidate location;

(EA2) repeat (EA1) for i=1, … ,n_d locations in the current design space ;

(EA3) find the smallest value among $\{S_i(\boldsymbol{\theta})\}_{i=1}^{n_d}$ and exchange the corresponding location in the design space and its according candidate location;

(EA4) iterate until $S(\boldsymbol{\theta})$ cannot be improved further.

The above procedure is known as the "basic exchange algorithm" (Cook and Nachtsheim, 1980). Wang and his colleagues have applied this idea in solving a single-station fixture-design problem based on the D-optimality criterion (Wang, 2000; Wang and Pelinescu, 2001).

Indeed, this basic exchange algorithm can yield a remarkably smaller value of $S(\boldsymbol{\theta})$ when it is applied to the SUV assembly process. However, the basic exchange algorithm was initially designed to determine efficient experiments for fitting simple regression models rather than for optimization problems with a high design space. It would run too slowly given a large N_c, i.e. a large number of the candidate locations. In this study, we discretized the continuous design space on each panel with candidate points 10 millimeters apart. Ten-millimeters is roughly the size of a locator's diameter. We feel that this resolution of discretization is sufficient to generate a fine enough grid on a panel. Given that the panels in the SUV assembly process have a size of several hundred millimeters, this discretizing resolution results in a total of N_c = 7,813 candidate positions on four panels. Applying the basic exchange algorithm,

we reduce $S(\boldsymbol{\theta})$ down from reference fixture layout $\boldsymbol{\theta}_0$ with $S(\boldsymbol{\theta}_0) = 5.397$ to 3.922 at the computing cost of 1,955.9 seconds.

The value of $S(\boldsymbol{\theta})$ from the basic exchange algorithm renders a 27.3 % reduction of the maximal sensitivity level of the fixture system from the initial or reference fixture layout, $\boldsymbol{\theta}_0$. Our empirical experience indicates that this $S(\boldsymbol{\theta})$ value, even if it may not be the smallest sensitivity, should be close to the global optimum. However, the basic exchange algorithm takes too much computing time for the three-station process in Figure 2, which is a simplified version of a real manufacturing process. This computation inefficiency limits its applicability in a larger scale fixture design problem. A general car body assembly that is made of over 100 panels will then correspond to a design space of hundreds of dimensions. Thus, our goal is to make the exchange algorithm faster without sacrificing too much of its effectiveness in reducing the sensitivity level of a fixture system.

III.2 Revised exchange algorithm

The fact that only one fixture location in the initial design is replaced in each iteration makes the basic exchange algorithm expensive to use. Within each iteration, the algorithm loops through all candidate sets n_d times, which makes the total computation of sensitivity function at the order of $n_d \cdot N_c$ per iteration. Meanwhile, all the PLPs in the initial design are likely to be replaced eventually. Thus, the overall computation complexity is at the order of $(n_d)^2 \cdot N_c$. It is clear that we should reduce

N_c and the number of iterations to make the exchange algorithm faster. Toward this goal, we implemented the following three improvements.

III.2.1 Increase the number of exchanges per iteration

In order to increase the number of exchanges, after Step (EA1) in the basic exchange algorithm, we can carry out the exchange that minimizes $S_i(\boldsymbol{\theta})$. Then, the number of exchanges is n_d for each iteration. This method is known as "modified Fedorov exchange" and was first suggested by Cook and Nachtsheim (1980).

Another way of increasing the exchange number is to perform the exchange whenever there is an improvement in the objective function. In this way, the exchange is performed much more frequently. However, it is easier for this algorithm to become entrapped in a local optimum since it is rushed for an exchange. This method is seldom recommended in the literature.

Alternatively, we can combine the above modifications to a basic exchange algorithm. The purpose of a combined modification is to exchange the candidate locations in the upper tail of the distribution of improvements in design criterion among all the candidate locations. A similar procedure is suggested by Lam, Welch and Young (2002) for a uniform coverage design in molecule selection.

In doing so, we should record the improvement in design criterion that a candidate location can make if the corresponding exchange is indeed carried out. The distribution of the improvement can be approximated by the recorded values. Denote the Δ as the improvement in the $S(\boldsymbol{\theta})$ criterion, i.e. $\Delta \equiv S(\boldsymbol{\theta})^{old} - S(\boldsymbol{\theta})^{new}$. Record all

Δ_j's (j=1, …, N_c) when we loop through the N_c candidate locations. Sort the value of Δ_j's in a descending order as $\Delta_{(1)} \geq \Delta_{(2)} \geq \ldots$ and so on. Select an integer number q, set $\Delta_{(q)}$ as the threshold. If there is an improvement greater than $\Delta_{(q)}$, then carry out the exchange.

It is apparent that the above combined modification is similar to the modified Fedrovo exchange algorithm if q = 1; and, if q is the value corresponding to $\Delta_{(q)} = 0$, it is the same as the one that performs an exchange whenever there is an improvement. This combined exchange algorithm is more versatile for broader applications.

In implementing this algorithm, we need to determine the value of q. Since we will likely replace all the initial design points in the final design, we decide to select $q = n_d$ so that we can replace n_d points in each iteration. However, in our fixture design, panels have a natural boundary and therefore an exchange between a design point and a candidate point can only be performed for those locations on the same panel. For this reason, we should implement the above algorithm for individual panels. Given that $n_d = 2$ for a panel (i.e. two locators per panel), we set $q = 2$. Moreover, the initial distribution of Δ is determined in the same way as in Lam et al. (2002), because it is approximated by the Δ-values of 100 randomly selected locations in the candidate set.

III.2.2 Reduce the number of locations in the candidate set

It is obvious to us that the large value of N_c is one of the key reasons that the basic exchange algorithm is computationally expensive. The N_c can be reduced if we

use a coarse grid on each panel when we discretize the continuous design space, though this could miss those low-sensitivity PLP locations and thus sacrifice the algorithm's effectiveness.

If we could rule out some areas that are unlikely to yield a "good" location, we can then discard the candidate locations in those areas entirely and thus reduce N_c. A part positioning deviation is more sensitive to locating deviations when both locators are close to each other than when they are distantly apart. This simple rule suggests that the final position of a locator is unlikely to fall into the geometrical central area on a panel. The geometrical center of a panel, which coincides with its gravity center when the panel has a homogenous density, is defined as

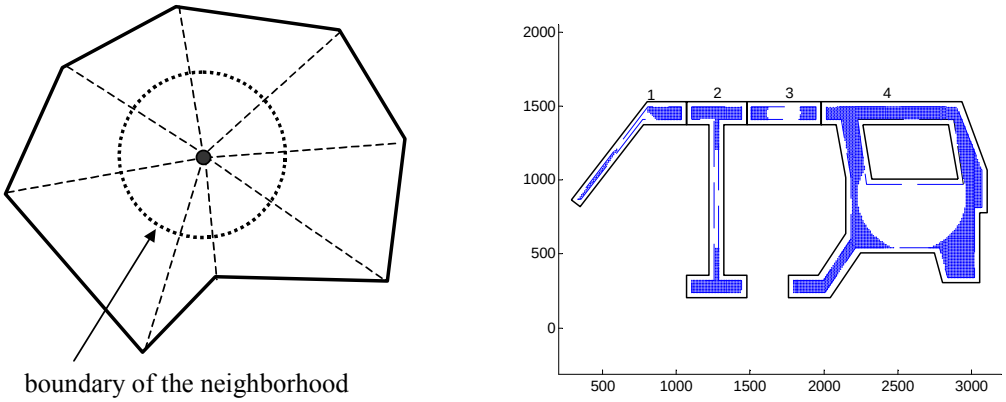
$$X_i = \frac{\iint_{Panel\ i} X dX dZ}{A_i} \quad \text{and} \quad Z_i = \frac{\iint_{Panel\ i} Z dX dZ}{A_i} \tag{12}$$

where A_i is the area of panel i.

The geometrical central area on a panel is considered to be in the neighborhood of a panel's gravity center. The determination of this neighborhood is illustrated in Figure 7(a). The distance between the gravity center and a vertex on the polygonal panel is calculated. Then, the median of these distances is chosen to represent the size of the panel, denoted as d_0. A hypothetical circle is drawn on the panel with the gravity as its center and $d_0/2$ as its radius. The area inside this hypothetical circle is considered to be the neighborhood of the gravity center. Only candidate locations outside the neighborhood will be used for exchanges with a design

point. The use of the median of all gravity-to-vertex distances in determining d_0, rather than their mean value, makes the resulting d_0 less sensitive to a very large or a very small gravity-to-vertex distance on panels with an irregular shape (recall that the median is a more robust statistic than the mean (Montgomery and Runger, 1999)).

We apply this rule to four SUV side frames. The resulting candidate areas are shown as the dark areas in Figure 7(b). One may also notice that there is a gap (35 mm) between the candidate areas and the edge of a part. This 35-mm-gap is determined by engineering safety requirement because a locating hole that is too close to the edge may not be able to endure the load exerted during fixturing. The resulting candidate area contains a total of N_c= 4,642 candidate locations, which is 59.4 % of the original N_c. The density of candidate locations is kept the same.



(a) (b)

Figure 7 Neighborhood of a gravity center (a) and candidate areas on SUV side frames (b)

III.2.3 Reduce the number of candidate locations after each iteration

After each iteration, the improvement in design criterion Δ_j is recorded for all candidate locations and sorted in a descending order. Those candidate locations with a low Δ value are less likely to be picked up by the exchange algorithm in the next iteration. Therefore, we propose removing half of the candidate locations whose Δ value is among $[\Delta_{(N_c/2+1)}, \Delta_{(N_c)}]$ after each iteration so that N_c becomes $N_c/2$ after each iteration. Our implementation of this action shows that it not only reduces the number of candidate locations but also makes convergence faster, meaning that the program will stop after fewer iterations.

By incorporating III.2.1-III.2.3, our revised exchange algorithm is summarized as follows and a flow chart is shown in Figure 8 to illustrate the algorithm.

Step 1. Generate the candidate locations in the candidate areas as shown in Figure 7(b). The resolution for discretization is 10 mm between two adjacent candidate locations.

Step 2. Initialize the Δ distribution. Randomly select 100 candidate locations on each panel. Calculate their Δ values and sort them in a descending order. Set $\Delta^* = \Delta_{(q)}$, where $q = 2$.

Step 3. For $i = 1$ to n_d (loop for each one of the current design points)

For $j = 1$ to N_i (loop through the candidate locations; N_i is the number of candidate locations on the panel that contain design

point i)

- Calculate and record $\Delta_{i,j}$;
- If $\Delta_{i,j}>\Delta^*$, then exchange design point i with candidate location j.

End of the j-indexed loop

If there is no exchange during the last j-indexed loop, then exchange design point i with the candidate point that maximizes $\Delta_{i,j}$ (for j=1 … N_i).

End of the i-indexed loop

Step 4. If there is no improvement in the $S(\boldsymbol{\theta})$ criterion during last loop (we check if $\frac{\max_{i,j} \Delta_{i,j}}{S(\boldsymbol{\theta})^{old}_{\max}} < 0.1\%$), then stop. Otherwise, sort $\Delta_{i,j}$; set $\Delta^*=\Delta_{(q)}$; remove half the candidate locations on each panel whose Δ value is less than $\Delta_{(N_i/2+1)}$; set $N_i=N_i/2$; go to Step 3 until the stopping criterion is met.

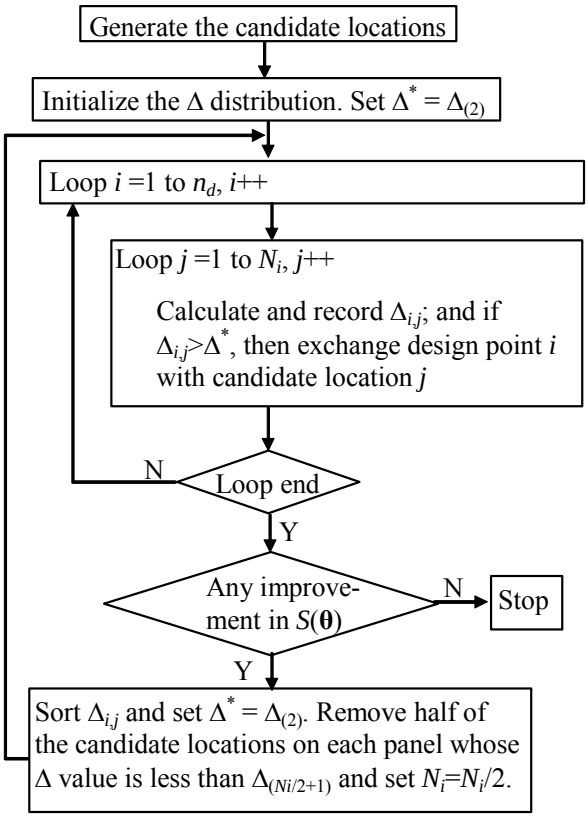


Figure 8 Flow chart of the revised exchange algorithm

III.3 Comparison and discussion

We implemented the above optimization algorithms in solving the multi-station fixture-layout design problem in the SUV assembly process. The results from the basic exchange algorithm, the modified Fedrov algorithm, and our revised exchange algorithm are respectively summarized in Table 3. Please note that our coding of exchange algorithms is implemented in MATLAB. The actual computation time of exchange algorithms should be able to be further reduced if using C or FORTUNE compiled codes.

Optimization methods are compared based on two kinds of initial designs. The first is the design currently used in industry (the $\boldsymbol{\theta}_0$ in Figure 2) and the other uses randomly generated initial designs; the performance data is the average of 10 trials. The reason to include the random initial design is to avoid any serious bias resulting from a comparison using a fixed initial design.

The results show that the revised exchange algorithm significantly reduces computing time. When we use PLP design $\boldsymbol{\theta}_0$, the computing time of the revised exchange algorithm on a computer with a 2.20GHz P4 processor is less than one-fourth (22.6%) of that needed for the basic exchange algorithm, and is 56.8% of that needed for the modified Fedorov algorithm. Surprisingly, the $S(\boldsymbol{\theta})$ value from the revised exchange algorithm is even smaller than that from the basic exchange algorithm. This indicates that more exchanges per iteration may help an algorithm escape from a local optimum and thus can improve the algorithm's effectiveness. When we use $\boldsymbol{\theta}_0$, the modified Fedorov exchange demonstrates a 60% shorter run-time than the basic exchange algorithm, yet yields a slightly larger $S(\boldsymbol{\theta})$. When we use random initial designs, the revised exchange algorithm runs about 5 times faster on average than the basic exchange algorithm, or 4 times faster than the modified Fedorov algorithm. The $S(\boldsymbol{\theta})$ it finds is slightly (1.2%) larger than the one found by the modified Fedorov, but smaller than that of the basic exchange algorithm. The number of iterations in the random initial design is roughly consistent with our previous analysis. The basic exchange algorithm used 5.1 iterations to replace all eight initial locators. The modified Fedorov exchange algorithm used less iterations

since more than one locator is replaced with a good candidate per iteration. The revised exchange algorithm further reduces the iteration to three times, about half of what the basic exchange algorithm used. Due to the nature of the stopping rule for exchange algorithms (comparing two subsequent $S(\boldsymbol{\theta})$'s), the minimum number of iterations is two. We feel that the potential for reducing the iteration number is being pushed to its limit by the revised exchange algorithm.

Using random initial designs, the modified Fedorov exchange yields a lower $S(\boldsymbol{\theta})$ value on average. The lowest value of $S(\boldsymbol{\theta})$=3.82 during those trials is also found by the modified Fedorov exchange. Since this value is only 2% lower than 3.922, it does not invalidate our prior conjecture that $S(\boldsymbol{\theta})$=3.922 should be close to the global optimum.

Table 3 Comparison of exchange algorithms

	Initial PLP design			Random initial designs		
	$S(\boldsymbol{\theta})$	Time (sec.)	# of iterations	$S(\boldsymbol{\theta})$	Time (sec.)	# of iterations
Basic Exchange	3.922	1955.9	5	4.022	1868.9	5.1
Modified Fedorov	3.952	780.312	2	3.894	1614.4	4.4
Revised Exchange (B.1-B.3)	3.903	443.528	4	3.940	373.1	3

The coordinates of the fixture layout with the lowest $S(\boldsymbol{\theta})$ value during our trials and the one determined by our revised exchange algorithm are listed in Table 4 as well as shown in Figure 9, where '+' represents a P_{4way} and "." represents a P_{2way}.

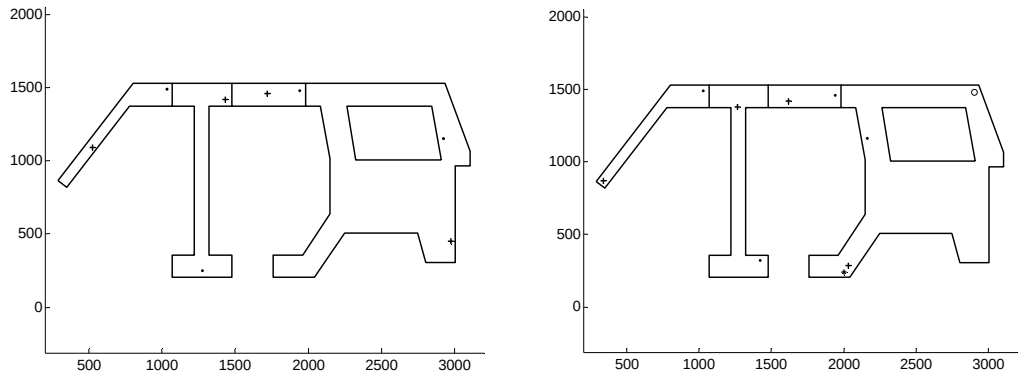
One interesting phenomenon that one may observe from Figure 9 is that the resulting fixture layout on the rear quarter panel apparently does not have the largest possible distance between the pair of locators. We performed fixture optimization for this panel alone and display the resulting positions, indicated by a "*" for P_{4way} and an "o" for P_{2way} in Figure 9(b). The pair of locators from the single-panel optimization has a much greater distance between them and is consistent with our intuition about a robust fixture layout. If we substitute this pair of PLP locations from the single-panel optimization into the multi-station assembly, we have the overall system-level $S(\boldsymbol{\theta})$=3.958, which is in fact larger than the best $S(\boldsymbol{\theta})$. This phenomenon implies that our intuitive largest-distance rule is not necessarily always right in a multi-station fixture design due to the fact that fixture locators are reused on different stations and their interaction complicates the sensitivity analysis. Thus, we should rely on an integrated variation propagation model and an effective optimization method, as developed in this dissertation.

The fact that both PLPs on the rear quarter panel in this obtained improved fixture layout are on the same side of the panel's gravity center does not cause a problem here because, in our application, the panels are positioned on a horizontal platform (refer to Figure 1). If the panels are vertically positioned, a force closure

constraint in addition to the geometrical constraint $G(\cdot)$ should be included in the optimization scheme (Equation 11) to ensure the resultant force and moment is zero. Under that circumstance, the resulting optimal fixture layout could be different. For robust fixture design considering force closure, please refer to Wang (2000).

Table 4 The fixture layout ($\boldsymbol{\theta}$) from exchange algorithms (Units: mm)

Part #	Fixture layout with the smallest $S(\boldsymbol{\theta})$		From the revised exchange algorithm	
	4-way PLP (X, Z)	2-way PLP (X, Z)	4-way PLP (X, Z)	2-way PLP (X, Z)
1	(523.8, 1091.1)	(1033.8, 1490)	(337.9, 871.9)	(1027.9, 1490)
2	(1434.5, 1418.8)	(1274.5, 248.7)	(1264.5, 1378.6)	(1424.5, 318.8)
3	(1720.9, 1460)	(1940.9, 1480)	(1620.9, 1420)	(1940.9, 1460)
4	(2973.5, 450.1)	(2923.5, 1150.1)	(2033.5, 286.6)	(2163.5, 1160.1)



(a) (b)

Figure 9 Fixture layouts; (a) fixture layout with the lowest $S(\boldsymbol{\theta})$ value (b) from revised exchange algorithm

III.4 Conclusion

For the fixture system used in a four-station SUV side-frame assembly process, the revised exchange algorithm yields the optimal fixture design whose maximal sensitivity level is only 72.3% of the currently used fixture layout design. The resulting optimal fixture layout is more robust to environmental noise – the reduction of 27.7% in sensitivity implies the same amount of reduction in product variation levels under the same variation inputs, according to the definition of sensitivity in Equation (7). The improvement in product quality will lead to a remarkable cost reduction in manufacturing systems.

For a non-linear optimization problem such as this multi-station fixture-layout design, it may be too costly, sometimes even impossible, to find the global optimum. The revised exchange algorithm is a good trade-off between optimality and algorithm efficiency. This revised exchange algorithm takes less than one-fourth of the computing time of the basic exchange algorithm and yields a fixture layout whose $S(\boldsymbol{\theta})$ is only 2.1% larger than that of the best fixture layout we obtained during all the trials.

CHAPTER IV

DATA-MINING METHOD

This chapter presents the data-mining aided optimal design method and explains how it is applied to facilitate the optimal fixture layout design in a four-station SUV side panel assembly process. Four components of the data-mining method are adopted; uniform coverage selection and carefully chosen parameters are used to make data-mining more efficient and effective.

IV.1 Overview of data-mining method

Data-mining is a discovery of unsuspected but valuable information from a large dataset (Hand, 1999). Recently, noteworthy efforts have been made in employing data-mining methods to aid the process of an optimal engineering design (Schwabacher, Ellman and Hirsh, 2001; Igusa, Liu, Schafer and Naiman, 2003). The basic idea is to use a data-mining method -- various classification methods are the major ones employed in such applications -- to extract "good" initial designs from a large volume of design candidates (design alternatives). In other words, if the design alternatives are treated as a dataset, a data-mining method may be able to discover valuable structures within the dataset and generalize design selection rules leading us to a much smaller "good" design set, which is more likely to yield a better design solution even if a local optimization method is applied.

The idea is illustrated in Figure 10. A data-mining method generalizes the design selection rules based on the training data in a design library, which is in turn created either from a collection of historical design results or from random sampling among design alternatives. The resulting selection rules are often expressed as a classification tree, or equivalently, a set of "*if-then*" rules. The large number of design alternatives pass through the selection rules and certain local optimization methods are applied to the selected "good" designs in order to find the final optimal design. Schwabacher et al. (2001), for instance, applied this idea in the prototype selection of structures of a racing yacht and a supersonic aircraft, respectively, where their design library is created from historical designs and classification method C4.5 (Quinlan, 1993) is used to generate the design selection rules.

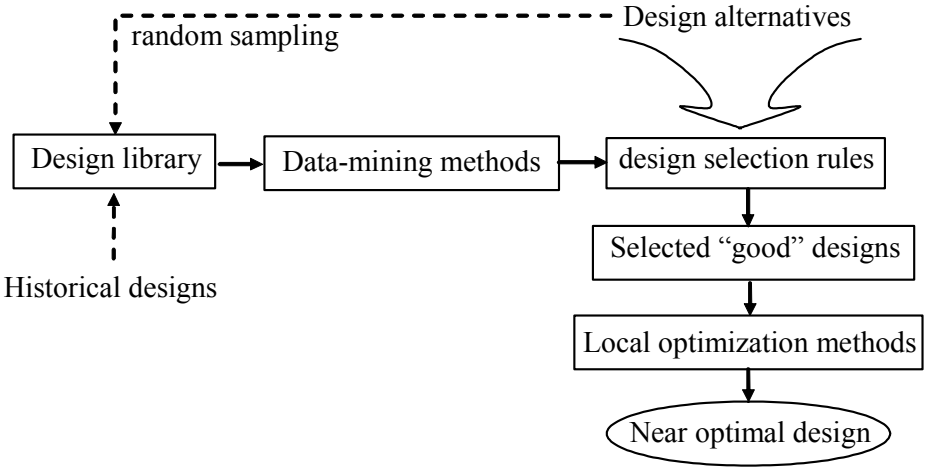


Figure 10 Design optimization utilizing data-mining methods

Although the general idea as described in Figure 10 could help in discovering valuable design selection guidelines, there is a major obstacle to applying 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, generation of design selection rules needs to evaluate the objective functions of all designs in a design library. In order for the design library to be representative of a large volume of design alternatives, one will have to include a large enough number of designs in the library – potentially too many to be computationally affordable for generating the selection rules.

IV.2 Data-mining method for engineering design problem

In the design of a civil structure, Igusa et al. (2003) proposed a more sophisticated idea, 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 needs to be evaluated for the generation of a classification tree.

Following the general idea proposed by Igusa et al. (2003), we developed in this dissertation 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 from among a large amount of original design alternatives for a non-rectangular design space; (2) feature functions of which evaluation is

computationally economical as the surrogate of the 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, eventually leading us to a competitive design.

There are eight fixture locators (P_1-P_8) in this multi-station panel assembly problem. Each part or sub-assembly is positioned by a pair of locators. We generate design alternatives by discretization using the resolution of 10 millimeters (the size of a locator's diameter) on each panel. We also rule out the central locations of the panel in order to reduce the number of design alternatives by the method introduced in Chapter III.2.2. This results in the number of candidate locations on each panel being 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.

There is virtually no efficient method which allows us to directly optimize over the huge volume of original design alternatives, such as the possible combinations of locators, as many as 8.6×10^{20}, in the multi-station 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 remarkably fewer than the original design alternatives, are still too many to be used as the design library.

In this dissertation, we use a clustering method with a set of computationally simple feature functions to facilitate the creation of a design library. This procedure

allows us to eventually have an affordable number of designs as a training dataset in a design library. The overall framework is illustrated in Figure 11, as a modification to Figure 10. We will present in the following subsections the considerations and procedures for realizing each component of the data-mining aided optimal design.

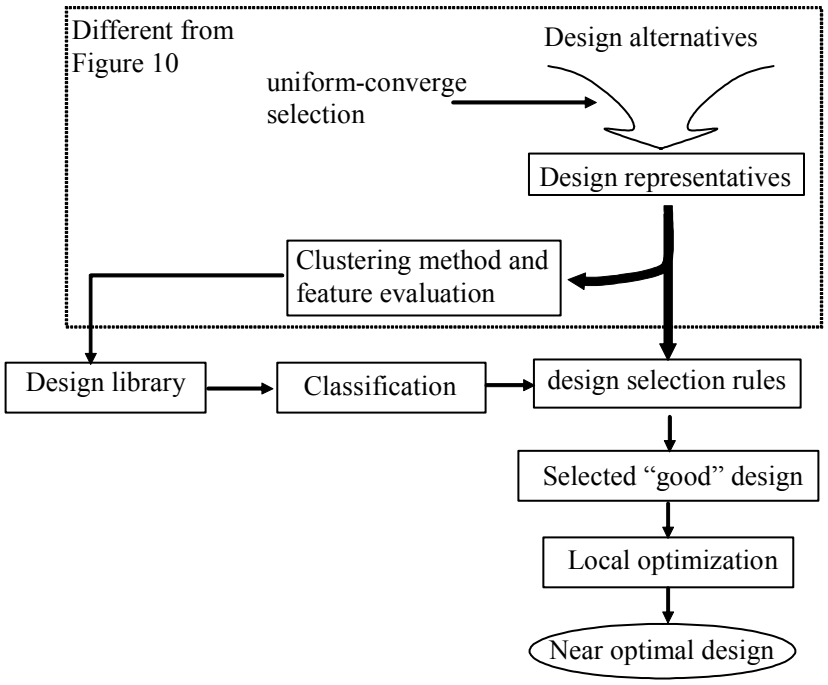


Figure 11 Modified data-mining aided design optimization procedure

IV.3 Uniform coverage selection of design representatives

The first component in the proposed optimal method is to select design representatives from original design alternatives. Unless one has profound knowledge of which part in the candidate design space (after the method in Chapter II.2.2 has been applied) is preferred in such a selection, a safer way of selecting good

representatives of the original design set is to select them from a design space as evenly as possible. Igusa et al. (2003) suggested randomly selecting design representatives, based on a uniform distribution, 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 (Fang and Wang, 1994).

A space-filling design, widely used in computer experiments (Santner, Williams, and Notz, 2003), 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) (McKay, Bechman, and Conover, 1979), a stratified sampling method, or using a uniformity criterion from the Number-Theoretic Method (NTM) (Fang, Lin, Winkle, and Zhang, 2000).

These methods can be easily implemented over a hyper-rectangular design space in experimental designs. In engineering system designs, accompanied by complicated geometric and physical constraints, the design space is often non-rectangular or even highly irregular, such as the candidate design space shown in Figure 7(b). Another constraint also comes into play in this fixture layout design, i.e. once a locator's position is chosen on a panel, the second locator on the same panel should not be located near the first one, following the same physical intuition related

to positioning variability explained in Chapter III.2.2. This is different from the factor level selection in experimental designs, where there is usually no clear prior knowledge to indicate the dependency among factors.

Given the complexity in design constraints, we have not seen a generic method to translate an LHS- or NTM-based space-filling design to an engineering system design problem. Instead, we here devised a heuristic procedure for the fixture layout design, attempting to provide a uniform-coverage selection of design representatives from original designs.

This procedure considers both physical constraints and uniformity. Considering uniformity alone, we will select the location of locators from a uniform grid generated over each panel. Considering the physical constraints alone, we will filter out all locator pairs on the same panel, whose distance is less than a minimum criterion. Thus, we select one locator from the uniform grid of candidate points and select another locator randomly from a candidate pool created by enforcing the minimum distance rule. The detailed steps are shown as follows.

Step 1. Uniformly discretize the candidate design space on each plane using the same resolution (in our implementation, the resolution is 10-mm between two adjacent locators).

Step 2. On each panel, the first locator is chosen sequentially 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 from among the locations whose distance from the first locator is greater than half of the 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 each of $\Omega_j^{(i-1)}$ 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 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. The threshold of d_0/2 is again chosen empirically, following the same spirit as in Chapter III.2.2.

After Step 2, the set $\Omega_4^{(0)}$ has the largest number of locator pairs, n_4=3496. Step 3 actually performs stratified sampling to generate locator combinations. The stratified sampling will go over $\Omega_4^{(0)}$ once but will have to go over $\Omega_j^{(0)}$ 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.

IV.4 Feature definition 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 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.

Features are often selected based on prior experience, rough engineering knowledge, or physical intuitions. The advantage of such a feature definition/selection is that vague experiences, knowledge, or understandings of a complicated engineering system can be more systemically integrated into the optimal design process. Although the selection of a feature in the proposed method is rather flexible, we do have certain generic considerations for an effective selection of features and feature functions. First, since features are used to replace the direct evaluation of an objective function, a feature function should be computationally simple, otherwise it will not serve our purpose. Second, because a feature is usually not connected to the design objective with mathematical explicitness, too few feature functions may generate a serious bias in the latter selection of design representatives. On the other hand, too many feature functions will increase the computation burden. A tradeoff will depend on specific applications, where five to fifteen feature functions may be selected. Finally, it is desirable to select scalable features, i.e. a feature

definition will remain the same when the system scale has increased. For the example of the multi-station fixture design, a scalable feature means that it can be used to characterize the system performance whether the multi-station system has three stations or ten stations.

Keeping in mind the above guidelines, we choose a set of feature functions for the fixture layout design as follows. We know that the distance between locators is an important factor related to the variation sensitivity of a fixture layout. We select the between-locator distance on a panel as one feature relevant to our design objective. We select the following five functions to characterize the feature of between-locator distance -- the five feature functions actually approximate the distribution of the set of the between-locator distances --

$F_1(\boldsymbol{\theta})$ = The largest value of the same-panel between-locator distances;

$F_2(\boldsymbol{\theta})$ = The second largest value of the same-panel between-locator distances;

$F_3(\boldsymbol{\theta})$ = The mean of the same-panel between-locator distances;

$F_4(\boldsymbol{\theta})$ = The second smallest value of the same-panel between-locator distances;

$F_5(\boldsymbol{\theta})$ = The smallest value of the same-panel between-locator distances.

For a larger scale assembly system with more parts and stations, the above feature functions can still be used; namely, they are scalable. The approximation of distribution could be improved by augmenting the number of feature functions so that they will give more refined percentile values of the set of between-locator distances.

If we are concerned only with a single part that is positioned by a pair of locators at a single station, the between-locator distance could be the only factor that

matters. However, complexity results from the fact that locating holes on a panel are re-used but usually in a different layout. For the multi-station assembly process in Figure 2, A-pillar and B-pillar are positioned on Station I by $\{P_1, P_2\}$ and $\{P_3, P_4\}$, respectively. After the assembly operation is finished, the sub-assembly becomes one piece, and it is transferred to Station II and positioned by $\{P_1, P_4\}$. This assembly transition across stations and the reuse of fixture locating holes complicates the sensitivity analysis for a multi-station system. It was shown in Chapter III.3 that a larger between-locator distance on one station may not necessarily produce a lower sensitivity for the whole process. In order to capture the across-station transition effect, we select a second feature, which is the ratio of between-locator distances on two adjacent stations. Denote by $L_1, L_2, \ldots, L_m$ the between-locator distance for m locator pairs on station k. After those parts are assembled, they are transferred to the next station and positioned by a locator pair with a between-locator distance $L^{(m)}$. The ratio of distance change r is then defined for this transition as

$$r \equiv \frac{L^{(m)}}{(\Sigma_{i=1}^{m} L_i)/m}. \tag{13}$$

Here we include three more feature functions related to the feature of distance change ratio as:

$F_6(\boldsymbol{\theta})$ = 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.

Similarly, the three feature functions approximate the distribution of the set of r. We do not include five functions as we did for the between-locator distances because the four stations in this example produce only three distance change ratios.

We have defined eight scalable feature functions for two physically intuitive features relevant to the variation sensitivity of a multi-station assembly process. Please note that the calculation of the above feature functions is very economical even for a large-scale system.

IV.5 Clustering method

Clustering is done to segment a heterogeneous population into a number of more homogeneous subgroups (Hastie, Tibshirani and Friedman, 2001). When using feature functions as the surrogate of a design objective to benchmark the dissimilarity criterion in a clustering procedure, design representatives in a resulting cluster will have a more similar distribution profile for the two selected features. Empirical evidence (Igusa et al., 2003) shows that resulting clusters are associated with a local response surface and its center will likely be around a local optimum. For this reason, a design library for classification can then be created by selecting a few designs from each cluster around the cluster center, which results in fewer designs. The generation of a design library is illustrated in Figure 12.

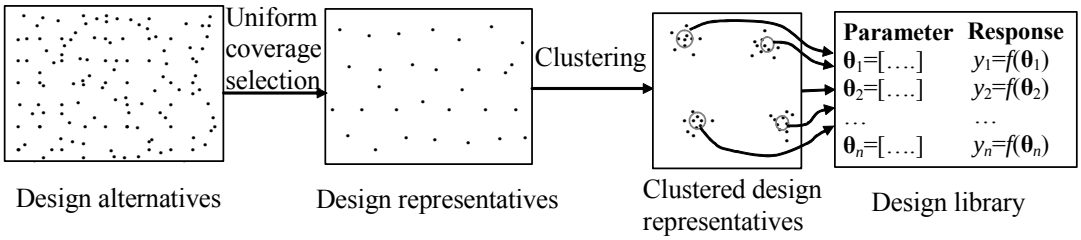


Figure 12 Generation of a design library

For the i^{th} fixture layout represented by $\boldsymbol{\theta}_i$, $\mathbf{F}_i \equiv [F_1(\boldsymbol{\theta}_i) \ \ldots \ F_8(\boldsymbol{\theta}_i)]^T$ is the vector of its feature functions and $C(i)$ denotes the cluster to which it belongs. In our solution procedure we employ a standard *K-means* clustering method (Hastie et al., 2001). Namely, for K clusters, we will find the vector mean values of cluster k, $\mathbf{m}_k$, as the cluster center, and the association of a fixture layout to cluster k (represented by $C(i)=k$) so that

$$\underset{C,\{\mathbf{m}_k\}_1^K}{Min} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \left\| \mathbf{F}_i - \mathbf{m}_k \right\|^2, \tag{14}$$

where N_k is the number of elements in cluster k and $\|\cdot\|$ is a vector 2-norm. The *K-means* method minimizes the dissimilarity measure, defined as the Euclidean distances of the elements within the same cluster. With different values of K, the minimization in Equation (14) will yield different clustering results, i.e. different cluster centers and cluster associations. We delay the discussion of how to choose K to Chapter IV.7.

Once the design representatives are clustered, i.e. each fixture layout is labeled with a cluster identification, we will choose a few designs around the cluster center to form a design library, as illustrated in Figure 12. We call the selected designs from

each cluster center *seed designs* and denote by J_k the number of seed designs chosen from cluster k. For the sake of simplicity, we use the same seed number for all clusters, i.e. $J_k=J$ for all k's. Then, the design library contains KJ data pairs $\{\mathbf{F}_i, S_i\}$ for i=1, 2, …, KJ, where S_i is the sensitivity value of the i^{th} fixture layout.

IV.6 Classification method

We perform classification on the dataset $\{\mathbf{F}_i, S_i\}$ in the design library to generate the design selection rules – this step is similar to what has been implemented before (Schwabacher et al., 2001, also refer to Figure 10). Local optimization methods can be used to evaluate a few designs chosen by the selection rules and yield the final optimal design. Many times, as we will see in Chapter V, a local optimization method may not be necessary, as a direct comparison among all the selected designs could have given us a satisfactory result.

A Classification and Regression Tree (CART) method is employed for constructing the classification tree in our problem. The one-standard-error rule (Hastie et al., 2001 pp. 57) based on a ten-fold cross-validation is used to select the final tree structure. The paths in a classification tree can be expressed as a set of "*if-then*" rules in terms of the feature functions. One resulting classification tree is shown in Figure 13. A decision condition such as $F_6 < 1.15$ is indicated at each node. One takes the left hand path if the answer to this condition is "yes" or the right hand path if "no."

An end node in the tree represents a set of designs associated with a narrow range of sensitivity values – the value indicated in Figure 13 beside an end node is the

average sensitivity value of the corresponding design set. If a certain combination of feature function values leads us to a set of designs whose expected sensitivity value is the lowest among all end nodes, then the corresponding path (one such path is highlighted in Figure 13) constitutes a design selection rule that we are looking for. The resulting selection rule is applied to the whole set of 3,496 design representatives from the uniform-coverage selection and the designs that are finally selected are considered "good" designs. Please note that because the random selection of the second locator on a panel, the resulting tree is not exactly the same each time we start the design process over. But our results show that this difference does not cause much difference in the final optimal design.

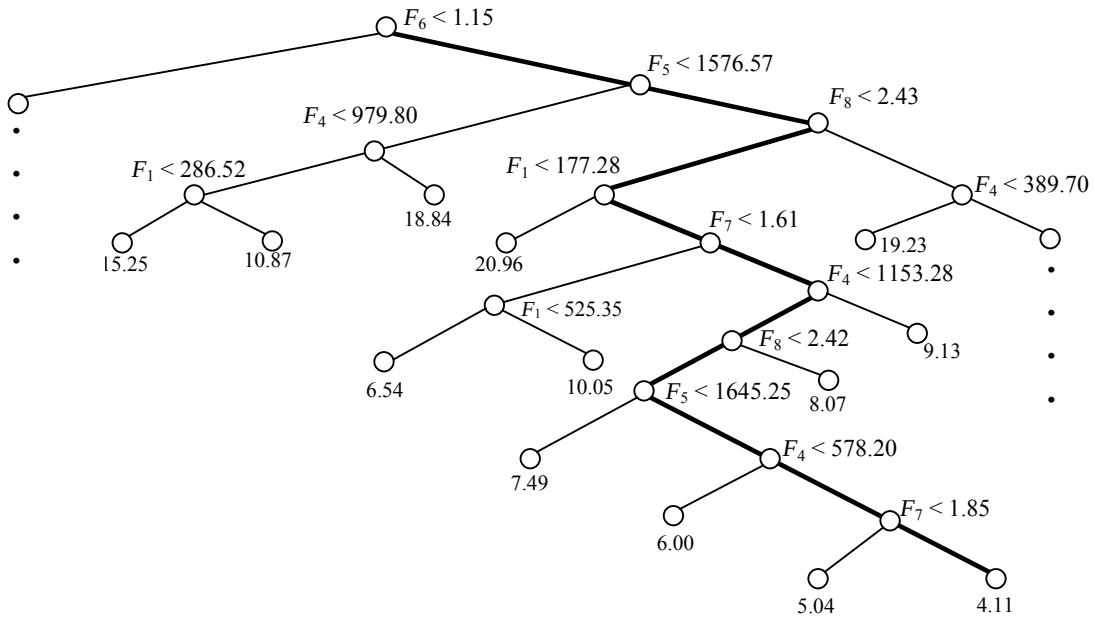


Figure 13 Part of the classification tree for the fixture layout design

IV.7 Selection of *K* and *J*

One issue we left out in IV.5 is how to select *K* (cluster number) and *J* (seed design number). The importance of these two values, *K* and *J*, is obvious since they determine the number of designs in the resulting design library. These two factors are related to both the optimal sensitivity value a design can achieve and the computation 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 issue by employing an experimental design approach. For a given combination of *K* and *J*, two response variables are chosen. These are the smallest sensitivity value (before a local optimization method is applied) and the computation time. For this data-mining aided optimal design, the overall computation time can be calculated by $T_0+KJ\cdot T+N_f\cdot T$, where T_0 is the time component in addition to that for evaluating the objective function and N_f is the number of designs in the selected "good" design set when the whole set of design representatives passes 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 given choice of *K* and *J*, the first and second time components will be largely fixed and *T* is also a constant. Hence, we use N_f as the second response variable.

We conduct a 3^2 factorial experiment, with three levels of K and J chosen at 3, 6, 9 and 5, 10, 15, respectively. We limit ourselves to the cases with $K<9$ since a large K will easily result in a large KJ, a situation less likely to be computationally advantageous. Because of the previously mentioned random selection of the second locator on a panel, for a given combination of K and J, the sensitivity and N_f are in fact random variables. Then three replications are performed at each combined level of K and J. A total of 27 computer experiments are conducted, each of which goes through the procedure as outlined in Figure 11 (before applying the local optimization). The lowest sensitivity value and the value of N_f are recorded in Table 5.

From Table 5, we find that the sensitivity value S is not significantly affected by the choice of K and J. On the other hand, the value of N_f is significantly affected, ranging from over one thousand to 52, depending on the choice of K and J. An ANOVA of S and N_f data verifies our finding and indicates that K, J, and their interaction are significant in the case of N_f data with a p-value less than 0.01 (the ANOVA tables are omitted because their significance is obvious).

Table 5 Results for different design conditions

		Sensitivity Value ($S(\boldsymbol{\theta})$)			The number of designs in the final "good" design set (N_f)		
		J			J		
		5	10	15	5	10	15
K	3	3.9746	3.9177	3.9051	1763	187	246
		3.8720	3.9450	3.9618	1536	333	116

		3.8879	3.9113	3.8722	1685	901	631
	6	3.9122	3.9023	3.9100	1261	300	274
		3.9034	3.917	3.9173	666	393	310
		3.8994	3.8722	3.9038	623	391	127
	9	3.8709	3.8989	3.9192	435	229	93
		3.8858	3.9144	3.9014	338	209	188
		3.9057	3.9268	3.9183	736	121	52

The reason that a choice of K and J will not have much effect on the sensitivity value is related to the fact that N_f changes accordingly for different K and J. When K and J are small and the designs in the library are fewer, the partition of design sets corresponding to different levels of sensitivity is rough, and thus, the resulting selection rule generated by the design library is not very discriminating. As a result, when the rule is applied to the entire set of design representatives, there will be a large number of designs that will satisfy it (e.g. the average of N_f is 1661 for K=3, J=5). Evaluation of the large number of the selected designs, however, will circumvent the limitation brought up by the non-discriminating selection rule, and the whole design process is still able to yield a low sensitivity value eventually. In the opposite case, when a relatively large number of designs is chosen to constitute the design library, the resulting selection rule will be discriminating and is able to select a small number of good designs, evaluation of which will give us a comparably low sensitivity value. The adaptive nature of N_f makes the eventual sensitivity value insensitive to the initial choice of K and J.

The choice of K and J will mainly affect the algorithm efficiency benchmarked by how many times the objective function is evaluated. The case with both small K and J is not an efficient choice because of a large N_f. However, a large K and J will not be a good choice, either, since KJ will be large even though N_f will decrease. Define the total number of function evaluations as $N_t \equiv KJ + N_f$. Utilizing the data in Table 5, we can fit a second order polynomial, expressing N_t in terms of K and J as

$$\hat{N}_t = 3918.3 - 331.1 \cdot K - 391.8 \cdot J + 16.6 \cdot KJ + 6.8 \cdot K^2 + 11.0 \cdot J^2 \ . \quad (15)$$

Based on the above expression, it is not difficult to find that the combination of K=9 and J=12 will give the lowest value of N_t. This combination of K and J is only optimal within the experimental range. Since K=9 is actually on the boundary, it seems to be the tendency that when KJ increases, N_f decreases. But the benefit of a decreasing N_f does not appear to be much beyond the point of K=9 and J=15, where KJ=135 is already more than the average value of N_f (which is 111). Further increases in KJ are likely to outnumber the decreases in N_f.

Using the following approximation, we provide a guideline for choosing K and J, which is independent of the specific relation in Equation (15). Recall that the selected "good" design set is generated by passing all design representatives through the design selection rule. In order to have a meaningful design selection rule, the corresponding end node in the classification tree must have at least one design point. Suppose that there is only one design in the end node, then the percentage of "good" designs selected from KJ designs in the library is $1/KJ$. If the same percentage applies

to all the design representatives, then $N_f=N_r/KJ$, where N_r is the number of design representatives. The total number of function evaluations can be approximated as

$$N_t \approx KJ + \frac{N_r}{KJ} \quad . \tag{16}$$

The above equation suggests that N_t is minimized when $KJ = \sqrt{N_r}$. In our problem, given N_r=3,496, KJ is roughly 60. A reasonable choice of K and J would be K=6 and J=10.

In actual cases, a classification tree pruned based on cross-validation usually keeps more than one element in its end nodes. We also observe that the percentage of "good" designs selected from the design representatives is higher than that from the design library. These factors make the actual value of KJ minimizing N_t larger than what is estimated from Equation (16). We could treat $KJ = \sqrt{N_r}$ as the lower bound for choosing K and J. As a rule of thumb, we recommend choosing a cluster number K from 6 to 9 and the number of seed designs J per cluster from 10 to 15.

Decision regarding cluster number is a major research topic in statistics. Tibshirani, Walther and Hastie (2001) proposed gap statistics for determining cluster number and also provided a comparison of several available statistical rules, including Milligan's method, Krzanowski's method, Hartigan's method, Kaufman's silhouette statistics, and their own gap statistics method. For the details of these criteria and computational procedures, please refer to Tibshirani et al. (2001). Using these criteria for our fixture design problem, the cluster number selected ranges from 2 to 5, as shown in Table 6. According to our previous discussion, these resulting cluster

numbers appears to be too small and will likely cause a large N_f. Since those criteria are originally devised for a different purpose, it is not really surprising that directly applying them here may not serve our optimal design well enough.

Table 6 The number of clusters suggested by the other methods

	Milligan's method	Krzanowski's method	Hartigan's method	Kaufman's silhouette statistics	Tibshirani's gap statistics
K	5	5	2	3	3

IV.8 Implementation and discussion

Our design algorithm is implemented with the choice of K=9 and J=12, the optimal combination found in Chapter IV.7. Two versions of the data-mining aided optimal design are realized, one is with the uniform-coverage selection of design representatives as outlined in Chapter IV.3 and another is by using simple random sampling for design representatives. Additionally, we include an alternative solution. An alternative solution for this fixture-design problem is to directly evaluate all 3,496 design representatives and select the best design among them, which provides a simple way for optimization. The direct comparison method based on a uniform selection is usually robust, as its performance is less sensitive to the properties of response functions, the properties of constraints, or the choice of initial conditions. The same philosophy of optimization was advocated by Fang and Wang (1994) using their NTM-based uniform number generation.

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 2 is not really an expensive one due to various simplifications we made in variation modeling (e.g. a 2-D assembly, rigid part assumption, only four stations). 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 can afford to perform the exploration in Chapter IV.2. When a computationally inexpensive function is used, the overhead computing cost T_0 kicks in, which may blind us to the benefit of the proposed method for a complicated system with a more expensive objective function. In order to show the potential benefit for expensive objective functions, we also include for comparison the number of times the objective function is evaluated -- when T is large, the time of function evaluation essentially dominates the entire computation cost.

We implemented the above-mentioned optimization algorithms to solve the multi-station fixture-layout design in the MATLAB environment; for example, MATLAB function "*kmeans*" is used for the *K*-means clustering method, "*treefit*" and "*treeprune*" for the CART. All implementations are executed on the same computer and the average performance data of 10 trials are included in Table 7. Based on the Table 7, we observe the following:

Table 7 Comparison of optimization methods

Optimization Methods	$S(\boldsymbol{\theta})$	Time (sec.)	The time for evaluating the objective function
Direct evaluation of design representatives	3.891	79.3	3,496 T
Data-mining aided (simple random selection)	4.060	51.9	276 T
Data-mining aided (uniform converge selection)	3.894	54.3	283 T

(1) If we consider both the value of $S(\boldsymbol{\theta})$ and the total computation time, the best design was found by the data-mining method with uniform coverage selection. Comparison between the data-mining aided designs with a uniform selection and a random selection shows an improvement of 5% in the sensitivity value by using the uniform design.

(2) We also find that the direct evaluation method is indeed quite competitive. In the above example, for instance, this direct evaluation method finds the best design among the chosen optimization methods. The limitation of this solution is that it may need to evaluate a rather large number of design representatives and thus becomes computationally unaffordable when the objective function is expensive (3,496T versus 283T in the case of fixture layout design). How to reduce the number of function evaluations is exactly where the data-mining method can help.

Regarding the design selection rule found by the CART in Figure 13, we find that for the feature of between-locator distance, only the extreme values (the largest

one, F_1, the smallest one, F_5, and the second smallest one, F_4) matter. In fact, the restriction on F_1 is $F_1 > 177.28$ mm, which will be satisfied in most designs. Hence, more insights come from the rules associated with F_4 and F_5, which provide non-trivial conditions, leading us to a design with low sensitivity values. For the feature of distance change ratio, all three related feature functions play a role, $F_6 > 1.15$, $F_7 > 1.85$, and $F_8 < 2.43$. Basically, this set of rules suggests that a good design will probably have a distance change ratio between 1.15 and 2.43. But the average ratio should be more than 1.85. The set of design selection rules makes our original intuitions on the across-station transition effect more quantitatively understood. This understanding can be extended to facilitate the design of a larger system with many more stations.

The best fixture layout found by our data-mining aided method is shown in Figure 14, where a fixture location is denoted by a "+" for P_1 and "." for P_2. The optimal fixture layout founded by the data-mining method is quite consistent with that of the revised exchange algorithm in Figure 9. For example, the distance between the pair of locators on each panel is large, but the distance between the pair of locators on the rear quarter panel does not have the largest possible one. This is also the case on Figure 9. Both PLPs on the rear quarter panel are on the same side of the panel's gravity center because the panels are positioned on a horizontal platform in our application.

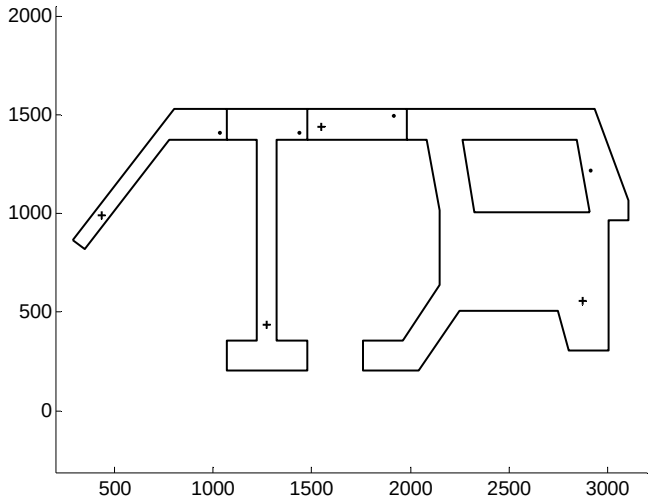


Figure 14 Fixture layouts with the lowest $S(\boldsymbol{\theta})$ value from data-mining method

CHAPTER V

COMPARISON WITH EXISTING ALGORITHMS

The non-linear optimization method and random search based heuristic methods such as simulated annealing algorithm (SA) and a genetic algorithm (GA) are applied to our example. The results are compared with the results from the revised exchange algorithm and data-mining aided method. The advantages of the data-mining method and results are discussed.

V.1 Non-linear programming method

A non-linear programming method such as the simplex search method (Nelder and Mead, 1965) is available in MATLAB as "*fminsearch*". It is a direct search method that does not require gradients or other derivative information. It stops at a local optimum and does not take a long time to compute. Our calculation reveals that $S(\boldsymbol{\theta})= 4.420$ and it takes 85.6 seconds to converge in the MATLAB environment on the machine, 2.20GHz P4 processor (other algorithms below are also executed under the same software and hardware computing conditions). The simplex search procedure operates on a continuous design space.

V.2 Simulated annealing method (SA)

Simulated annealing is used for finding near globally optimization procedures for solving combinatorial optimization problems based on stochastic computational techniques though it also considers many aspects related to iterative improvement algorithms. The application of iterative improvement algorithms supposes a specific definition of the problem; this includes the definition of configurations, neighborhood, cost of the configuration, and an initial configuration. The generation mechanism defines a neighborhood for each configuration, consisting of all configurations that can be reached with one transition.

Kirkpatrick, Gelatt and Vecchi (1983) introduced the first annealing technique that corresponds to an increase in the cost function in a limited way. It is generally known as simulated annealing due to the analogy with the simulation of the annealing of solids it is based upon. It is also known as Monte-Carlo annealing, statistical cooling, probabilistic hill climbing, or the probabilistic exchange algorithm. Solutions obtained by simulated annealing do not depend on the initial configuration and have a solution near global optimum. Further, it is possible to give a polynomial upper bound for the computation time for some implementations of the algorithm.

The generic parameters to be determined before experiments are: initial temperature (T), cooling ratio (k_B), known as the Boltzmann's constant, and the stopping criterion. An initial configuration is also needed, and is generally selected at random from all design configuration alternatives. The specific procedure is controlled as a function based on initial temperature and Boltzmann's constant. Δ is

defined as the difference in cost between the current solution and the neighboring solution; if the difference means reduction in an objective function, then the process is continued with the new solution. If the difference means an increase in the objective function, the new solution is accepted according to the specific probability, which is expressed as $e^{(-\Delta/k_B T)}$ where T is the control parameter. A random number r is selected from the interval [0,1]. If $r < e^{(-\Delta/k_B T)}$ then the step is accepted or otherwise denied. This probability depends highly on the Boltamann's constant k_B, and this condition means that the simulated annealing algorithm can violate local optimality in its quest for a global optimum.

The stopping criterion which will determine whether the system is cool enough affects the efficiency of the solution as it depends on the number of iterations per each temperature, the total number of temperature changes, and the configuration changes at each temperature stage. The algorithm proceeds until the temperature reaches the final temperature, which corresponds in the analogy to the frozen or solid temperature. To find a good solution quality within a comparably short period of time using this simulated annealing algorithm, the parameters must be chosen carefully.

The simulated annealing algorithm is implemented in solving our multi-station fixture layout design problem in the SUV assembly process. The number of configuration changes to be attempted at each temperature and the number of iterations are set proportional to the number of fixtures, e.g. n=8 for the process in Figure 2, at (10n) and (100n), respectively. The total number of temperature steps, which affects the run time of the procedure, is set at 100 iterations. Generally, the

most important parameters in using simulated annealing is the initial temperature T and Boltamann's constant k_B.

The initial temperature is determined by identifying the lowest value at which at least 80 percent of a certain number of random configuration changes are accepted. Extremely high initial temperatures without relatively long iteration times can not guarantee favorable solutions since they provide too many chances to accept an inferior objective function value and hence the procedure could stop before it reaches a solid state that is the best solution. Alternatively, assigning too small of a value for initial temperatures will make the simulated annealing algorithm behave as a steepest descent algorithm, which does not allow uphill moves, and it may easily become trapped in a local optimum. For our problem, T is tested from 10 to 20. There was no appreciable difference in the objective function value and T = 12.5 was selected as a best initial temperature.

The Boltamann's constant k_B is determined by the experiments. A general guideline given by Viswanadham, Sharma and Taneja (1996) is to choose k_B between 0.85 and 0.95. Table 8 shows the results when the Boltamann's constant k_B is changed from 0.85 to 0.99 with initial temperature T = 12.5. It is seen that $S(\boldsymbol{\theta})$ value is decreased with lower Boltamann's constant. The value of $S(\boldsymbol{\theta})$ is similar but the running time is decreased when the value of k_B is increased from 0.85 to 0.9. When the value of k_B is increased from 0.90 to 0.95, the value of $S(\boldsymbol{\theta})$ is increased but the running time is decreased too. The value of $S(\boldsymbol{\theta})$ is quite large when k_B = 0.99. Thus,

if we consider the efficiency, as well as optimality, we find that k_B = 0.90 or 0.95 is a good trade off.

Table 8 Result comparison from different Boltamann's constant (k_B)

		k_B = 0.85		k_B = 0.90		k_B = 0.95		k_B = 0.99	
	initial $S(\boldsymbol{\theta})$	$S(\boldsymbol{\theta})$	Time (sec.)	$S(\boldsymbol{\theta})$	Time (sec.)	$S(\boldsymbol{\theta})$	Time (sec.)	$S(\boldsymbol{\theta})$	Time (sec.)
1	43.892	3.864	1051.3	3.885	640.09	3.960	309.2	7.150	206.9
2	16.812	3.883	1080.2	3.818	722.02	3.957	313.8	9.376	213.3
3	27.443	3.820	1092.1	3.821	655.77	3.909	310.1	4.402	213.2

V.3 Genetic algorithm (GA)

Genetic algorithm is an evolutionary algorithm which employs stochastic optimization techniques that simulate the natural evolution process for solving complicated real-world problems (Gen and Cheng, 2000). The genetic algorithm methodology finds solution by mimicking processes of evolution, such as selection, recombination, and mutation. Selection determines which individuals are chosen for mating in recombination and which offspring are selected for the next generation. Recombination produces new individuals by combining the information contained in parents. After recombination, every offspring undergoes mutation. In the mutation process, offspring variables are mutated by small perturbations with low probability.

Several parameters should be carefully selected for the efficiency and the effectiveness of the algorithms. Recombination rate (P_c) is the ratio of the number of

offspring produced for each generation relative to the population size. Permutation rate (P_m) is the percentage of the total number of genes in the population. Population size (M) is the size of population that evolves. Each parameter affects the computation time, CPU memory and optimality. For example, if we increase the value of M, then the likelihood of reaching a global optimum also increases, with a corresponding increase in the total computation time and memory usage. Large values for recombination rate (P_c) and permutation rate (P_m) help to make the solution space broader, but reduce the probability to obtain the best solution.

In our example, the acceptable parameter was selected based on multiple experimental trials. Table 9 shows the results comparison from different parameters of the GA. The acceptable solution was found when $P_c = 0.2$, $P_m = 0.01$ and $M = 10$. From this result, we know that the GA also finds a reasonable solution ($S(\boldsymbol{\theta}) = 4.1958$) with the above parameters. The experiment results show the various outcomes when the value of the parameters are changed. When the recombination rate (P_c) and permutation rate (P_m) becomes larger, the value of the objective function and the running time also become larger. In addition, having a large population size (M) increased the total computation time and decreased the chance of obtaining a good solution.

Table 9 Result comparison from different parameters of GA

	P_c	0.2					
	P_m	0.005		0.01		0.015	
M		$S(\boldsymbol{\theta})$	Time(sec.)	$S(\boldsymbol{\theta})$	Time(sec.)	$S(\boldsymbol{\theta})$	Time(sec.)
10		4.7197	720.16	4.1958	672.03	5.0784	758.31
20		5.0565	1188	4.5896	1297.2	4.7397	1151.5
	P_c	0.4					
	P_m	0.005		0.01		0.015	
M		$S(\boldsymbol{\theta})$	Time(sec.)	$S(\boldsymbol{\theta})$	Time(sec.)	$S(\boldsymbol{\theta})$	Time(sec.)
10		4.6556	1137.6	4.7488	1147.9	4.2726	1094.4
20		4.6199	1984.2	4.1892	1969.1	4.6183	2101.3
	P_c	0.6					
	P_m	0.005		0.01		0.015	
M		$S(\boldsymbol{\theta})$	Time(sec.)	$S(\boldsymbol{\theta})$	Time(sec.)	$S(\boldsymbol{\theta})$	Time(sec.)
10		4.3161	1432.7	4.7496	1446.7	4.7468	1444.5
20		4.3941	2844.1	4.2786	2509.1	4.6884	2614.5

V.4 Comparison

Table 10 summarizes all the results of the methods which were applied in this dissertation. The non-linear programming method finds results quickly but at the local solution level. SA and GA find acceptable solutions, but the limitation of these algorithms is the large number of evaluations of the objective function involved. For example, in simulated annealing there were 28,503 number of objective function evaluations for k_B = 0.9, and 13,606 evaluations for k_B = 0.95. For the genetic algorithm, there were 18,094 number of objective function evaluations on the best solution. In this fixture layout example, the evaluation time for objective functions is

not really expensive, so simulated annealing could obtain the solution in a reasonable amount of time. If our assembly system contains more stations so that objective function evaluations become more expensive, the total computing time of these methods becomes more expensive as well.

The basic exchange algorithm, modified Fedorov exchange algorithm, and revised exchange algorithm also find good solutions. The revised exchange algorithm is superior among them for finding an acceptable solution in a reasonable time-frame, but it also involves a large number of objective function evaluations, and might be inefficient when evaluation time is very expensive.

The data-mining method shows the best results on both objective function value criteria and total running time. If we consider the amount of evaluation time for the objective function, data-mining attains the result in a very small number of evaluations. Based on the comparison, we have a few remarks and elaboration.

Table 10 Comparison of result from all methods

Optimization methods	$S(\boldsymbol{\theta})$	Time (sec.)	The time for evaluating the objective function
Simplex search	6.825	73.8	3,200 T
Genetic Algorithm	4.196	672.1	18,094 T
Simulated Annealing (k_B =0.9)	3.831	542.8	28,503 T
Simulated Annealing (k_B =0.95)	3.979	259.5	13,606 T
Basic Exchange	4.022	1868.9	124,954 T
Modified Fedorov	3.894	1614.4	62,472 T
Revised Exchange	3.940	373.1	17,579 T
Direct evaluation of design representatives	3.891	79.3	3,496 T
Data-mining aided (simple random selection)	4.060	51.9	276 T
Data-mining aided (uniform converge selection)	3.894	54.3	283 T

(1) The best design is found by the simulated annealing with $k_B = 0.9$ at the cost of 542.8 seconds of computation time or over 28,000 times of function evaluation. The detailed description of simulated annealing is in Chapter V.2. By comparison, the data-mining aided design reaches a very close sensitivity value (only 1.6% higher than what the SA found) but used one-tenth of the computation time. We also notice that the data-mining aided design evaluates the objection function only one-hundredth of the number of times that SA did. The SA with a larger k_B is not advantageous – the computing time is still long (five times the data-mining aided method for $k_B = 0.95$) but the resulting sensitivity value increases considerably.

(2) Because of our current choice of objective function, the time that a data-mining method takes is dominated by its overhead time, roughly 45 seconds for a simple random selection or 50 seconds for a uniform-coverage selection. Since we used scalable feature functions, these overhead time components will not change much even for a system with an expensive objective function. The computation for other algorithms, however, is mainly the result of evaluating the objective function. Therefore, the benefit of our data-mining aided design method will be more obvious -- 28,503T for SA versus 283T for our design -- for a larger, more complicated system where the evaluation of the objective function will dominate the overall computation cost.

(3) The time for evaluating the objective function is significantly decreased when we use a revised exchange algorithm than basic exchange algorithm (124,954T as apposed to 17,579T). The $S(\boldsymbol{\theta})$ value from the revised exchange algorithm is also lower than that from the basic exchange algorithm. From this result, we can conclude that the three improvements addressed in Chapter III.2 helped to reduce the evaluation time of the objective function and that they eventually reduced the total computation time without sacrificing the objective function value.

V.5 Summary and conclusion

In summary, the advantage of the data-mining aided design is noteworthy. For the multi-station fixture layout design, it yields a solution with a sensitivity value as low as a random search method while taking a shorter amount of time than a local

optimization method (the simplex search takes 73.8 seconds). Local optimization can be applied to the best design found by the data-mining method and it will reduce the sensitivity value to 3.864, an improvement which is not significant. It is our observation that the data-mining aided design can oftentimes produce a satisfactory design result without the need of applying a local optimization method.

The reason that data-mining methods can facilitate optimal engineering design lies in its capability in knowledge discovery, knowledge transferring, and knowledge encapsulation. The clustering method actually connects, without performing direct evaluation of an objective function, vague human knowledge about an engineering system to design parameters and objectives that are mathematically defined. The reduction in evaluating an objective function eventually generates a remarkable benefit in terms of algorithm efficiency. Meanwhile, the knowledge about the performance of an engineering system will become more explicit and numerical once the set of design selection rules is formed from a classification method. The accumulated knowledge, expressed in design rules and the better design conditions, can be translated into the optimization of a similar yet larger system.

A final note is on the use of feature function, which transfers engineering knowledge for statistical treatments. Such an integration of engineering knowledge and statistical methods is considered an important way of improving the performance of statistical solutions for solving messy engineering problems. Traditional ways of transferring engineering knowledge include expert systems (Jackson, 1999) or physical modeling. The former is usually too qualitative and the latter is highly

quantitative but less flexible – in many sophisticated physical systems, an accurate physical modeling of the system is almost impossible. We feel that the inclusion of feature function strikes a balance of being more quantitative, as well as being flexible enough, in incorporating engineering knowledge and understanding into the process of design optimization.

CHAPTER VI

CONCLUSION AND FUTURE WORK

The conclusions of this dissertation are summarized, and the overall findings derived from the research are clarified in this chapter. Furthermore, possibilities for future research are discussed.

VI.1 Conclusion

This dissertation investigates various aspects of near optimal fixture layout design in a multi-station panel assembly processes: variation modeling, design criteria, and optimization methods. Due to the singularity of the design matrix of a multi-station fixture system, the widely used D-optimal criterion is not an appropriate measure. Instead, the E-optimality criterion is recommended, which minimizes the maximum sensitivity level of a fixture system to the input variation. Different optimization methods are explored and compared. The revised exchange algorithm and data-mining aided method are developed to solve a high-dimension optimization for the multi-station fixture layout designs.

The revised exchange algorithm incorporates three improvements to make it more efficient. These are 1) increase the number of exchanges per iteration, 2) reduce the number of locations in the candidate sets, and 3) reduce the number of candidate locations after each iteration. The revised exchange algorithm then generates fairly applicable solutions to engineering system designs.

Data-mining aided methods have been further enhanced to be more effective and reliable in design. Four steps included in the refinement are 1) a uniform coverage method, 2) feature and feature function, 3) clustering method, and 4) classification method. These components lead us to find an near optimal design and compared with other available optimization methods, this method demonstrates clear advantages in terms of both the sensitivity value it can find (only 1.6% higher than what SA found) and the computation time it consumes (shorter than a simplex search and one-tenth of what SA takes).

VI.2 Suggestions for future work

This study developed a general framework to optimize the fixture layout design in the multi-station panel assembly process. However, much more research needs to be done in order to expand the applicability and increase the generality of the resulting methodology. Some issues for future study include the following:

- Since the state space model is a linear model structure, its applications are limited to the situation where the magnitude of fixturing error is smaller than the locaters distance. For more general variation modeling, non-linear effects should be addressed and eventually, a non-linear model should be applied.
- Because our interest is more focused on the global variation resulting from locating pins, a 2D model is used and the result is for the location design of locating pins. Efforts need to be spent to gradually relax the 2D assumption to

accommodate more complex factors such as NC blocks, 3D fabrication, and part compliance.

- It is worthwhile to study some theoretical aspects of the developed algorithms. For example, in the data-mining method the value of *K* and *J* highly affect the optimality and total computation time. Since the mathematical relationship between *K* and *J* value and the optimality or total computation time is difficult to define, this dissertation selects both values through the experimental design. A theoretic study revealing the relationship between *K* and *J* and algorithm efficiency can help use the algorithm wisely.
- It is worthwhile to explore how to apply the developed methods to new applications. Although the algorithms are discussed in the specific context of fixture layout design, we feel that the variation propagation model, the selection of design criterion, and the resulting algorithms such as the revised exchange algorithm and data-mining method are fairly general and should be applicable to other engineering system designs. For example, when a sensor system is designed for a discrete parts manufacturing process, the methods used in this dissertation may be employed to maximize the detection sensitivity.

REFERENCES

Asada, H. and By, A.B. (1985), "Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixture," *IEEE Journal of Robotics and Automation*, RA-1, 86-94.

Atkinson, A.C. and Donev, A.N. (1992), *Optimum Experimental Designs*, New York, Oxford University Press.

Cai, W.J. and Hu, S.J. (1996), "Optimal fixture configuration design for sheet metal assembly with spring back," *Transactions of NAMRI/SME*, XXIV, 229-234.

Cai, W.J., Hu, S.J. and Yuan, J.X. (1997), "A variational method of robust fixture configuration design for 3-d workpieces," ASME *Journal of Manufacturing Science and Engineering*, 119, 593-602.

Camelio, A.J., Hu, S.J. and Ceglarek, D.J. (2001), "Modeling variation propagation of multi-station assembly systems with compliant parts," *Proceedings of the 2001 ASME Design Engineering Technical Conferences*, Pittsburgh, PA, September 9-12.

Carlson, J.S. (2001), "Quadratic sensitivity analysis of fixtures and locating schemes for rigid parts," ASME *Journal of Manufacturing Science and Engineering*, 123, 462-472.

Carlson, J.S., Lindkvist, L. and Soderberg, R. (2000), "Multi-fixture assembly system diagnosis based on part and sub-assembly measurement data," *Proceedings of the*

2000 ASME Design Engineering Technical Conference, Baltimore, MD, September 10-13.

Ceglarek, D. and Shi, J. (1995), "Dimensional variation reduction for automotive body assembly," *Manufacturing Review*, 8, 139-154.

Ceglarek, D. and Shi, J. (1996), "Fixture failure diagnosis for auto-body assembly using pattern recognition," ASME *Journal of Engineering for Industry,* 118, 55-66.

Chang, M. and Gossard, D.C. (1998),"Computational method for diagnosis of variation-related assembly problem," *International Journal of Production Research*, 36, 2985-2995.

Chou, Y-C., Chandru, V. and Barash, M.M. (1989), "A mathematical approach to automatic configuration of machining fixtures: analysis and synthesis," ASME *Journal of Engineering for Industry*, 111, 299-306.

Choudhuri, S.A. and DeMeter, E.C. (1999), "Tolerance analysis of machining fixture locators," ASME *Journal of Manufacturing Science and Engineering,* 121, 273-281.

Cook, R.D. and Nachtsheim, C.J. (1980), "A comparison for constructing exact D-optimal designs," *Technometrics*, 22, 315-324.

Ding, Y., Ceglarek, D. and Shi, J. (2000), "Modeling and diagnosis of multi-station manufacturing processes: state space model," *Proceedings of the 2000 Japan/USA Symposium on Flexible Automation*, Ann Arbor, MI, 2000JUSFA-13146, July 23-26.

Ding, Y., Ceglarek, D. and Shi, J. (2002a), "Design evaluation of multi-station

manufacturing processes by using state space approach," ASME *Journal of Mechanical Design*, 124, 408-418.

Ding, Y., Ceglarek, D. and Shi, J. (2002b), "Fault diagnosis of multi-station manufacturing processes by using state space approach," ASME *Journal of Manufacturing Science and Engineering*, 124, 313-322.

DeMeter, E.C. (1995), "Min-max load model for optimizing machining fixture performance," ASME *Journal of Engineering for Industry,* 117, 186-193.

Fang, K.T. and Wang, Y. (1994), *Number-theoretic Methods in Statistics*, New York , Chapman and Hall.

Fang, K.T., Lin, D.K.J., Winkle, P. and Zhang, Y. (2000), "Uniform design: theory and application," *Technometrics*, 42, 237-248.

Fedorov, V.V. (1972), *Theory of Optimal Experiments*, Academic Press, New York.

Ferreira, P.M., Kochar, B., Liu, C.R. and Chandru, V. (1985), "AIFIX: an expert system approach to fixture design," ASME *Winter Annual Meeting on Computer Aided/Intelligent Process Planning*, Miami Beach, FL.

Gen, M. and Cheng, R (2000), *Genetic Algorithms and Engineering Optimization*, New York John Wiley and Sons.

Hand, A.J. (1999), "Statistics and data mining: intersection disciplines," SIGKDD *Explorations,*1-1,16-19.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Element of Statistical Learning*, New York, Springer-Verlag.

Hillier, F.S. and Lieberman, G.J. (2001), *Introduction to Operations Research*, 7th edition, New York, McGraw Hill.

Hockenberger, M.J. and DeMeter, E.C. (1996), "The application of meta functions to the quasi-static analysis of workpiece displacement within a machining fixture," ASME *Journal of Manufacturing Science and Engineering*, 118, 325-331.

Huang, Y. and Hoshi, T. (1999), "Study for optimum fixture design considering flatness error due to moving cutting heat source," *Technical Paper – Society of Manufacturing Engineers, MS99-177*, MS99-177-1 – MS99-177-6.

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 Design, Service and Manufacturing Grantees and Research Conference*, Birmingham AL, January 4-9.

Jackson, P. (1999), *Introduction to Expert System*, 3rd edition, Reading, MA, Addison-Wesley,.

Jin, J. and Shi, J. (1999), "State space modeling of sheet metal assembly for dimensional control," ASME *Journal of Manufacturing Science and Engineering*, 212, 756-762.

Kim P. and Ding, Y. (2004), "Optimal design of fixture layout in multi-station assembly process," *IEEE Transactions on Automation Science and Engineering*, to appear.

Kirkpatrick, S., Gelatt, C.D. Jr. and Vecchi, M.P. (1983), "Optimization by simulated

annealing", *Science*, 220, 671-680.

Lam, R.L.H., Welch, W.J. and Young, S.S. (2002), "Uniform coverage designs for molecule selection," *Technometrics*, 44, 99-109.

Mantripragada, R. and Whitney, D.E. (1999), "Modeling and controlling variation propagation in mechanical assemblies using state transition models," *IEEE Transaction on Robotics and Automation*, 15, 124-140.

McKay, M.D., Bechman, R.J. and 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*, 21, 239-245.

Melkote, S.N. (1995), "Prediction of the reaction force system for machining fixtures based on machining process simulation," *Transactions of NAMRI/SME.* XXIII, 207-214.

Menassa, R.J. and DeVries, W.R. (1991), "Optimization methods applied to selecting support positions in fixture design," ASME *Journal of Engineering for Industry,* 113, 412-418.

Montgomery, D.C. and Runger, G.C. (1999), *Applied Statistics and Probability for Engineers*, 2nd edition, New York, John Wiley and Sons.

Nelder, J.A. and Mead, R. (1965), "A simplex method for function minimization," *The Computer Journal*, 7, 308-313.

Paul, R.P. (1981), *Robot Manipulators: Mathematics, Programming, and Control*, Cambridge, MA, MIT Press.

Pukelsheim, F. (1993), *Optimal Design of Experiments*, New York, John Wiley and Sons.

Quinlan, J.R. (1993), *C4.5: Programs for Machine Learning*, San Mateo, CA, Morgan Kaufmann.

Rearick, M.R., Hu, S.J. and Wu, S.M. (1993), "Optimal fixture design for deformable sheet metal workpieces," *Transactions of NAMRI/SME,* XXI, 407-412.

Rong, Y. and Bai, Y. (1996), "Machining accuracy analysis for computer-aided fixture design verification," ASME *Journal of Manufacturing Science and Engineering*, 118, 289-300.

Rong, Y., Li, W. and Bai, Y. (1995), "Locating error analysis for computer-aided fixture design and verification," *Proceedings of the 1995 ASME Database Symposium, Computer in Engineering*, 825-832, Boston, MA, September 17-21.

Santner, T.J., Williams, B.J. and Notz, W.I. (2003), *Design & Analysis of Computer Experiments*, New York, Springer Verlag.

Schott, J.R. (1997), *Matrix Analysis for Statistics*, New York, John Wiley & Sons.

Schwabacher, M., Ellman, T. and Hirsh, H. (2001), "Learning to set up numerical optimizations of engineering designs," *Data Mining for Design and Manufacturing*, Norwell, MA, Kluwer Academic Publishers, 87-125.

Shalon, D., Gossard, D., Ulrich, K. and Fitzpatrick, D. (1992), "Representing geometric variations in complex structural assemblies on CAD systems,"

Proceedings of the 19th Annual ASME Advances in Design Automation Conference, DE-vol.44-2, 121-132, Scottsdale, AZ, September 13-16.

Soderberg, R. and Carlson, J.S. (1999), "Locating Scheme Analysis for Robust Assembly and Fixture Design," *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, Las Vegas, NE, September 12-15.

Tibshirani, R., Walther, G. and Hastie, T. (2001), "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society - series B*, 63, 411-423.

Trappey, J.C. and Liu, C.R (1990), "A literature survey of fixture design automation," *International Journal of Advanced Manufacturing Technology*, 5, 240-255.

Viswanadham, N., Sharma, S. and Taneja, M. (1996), "Inspection allocation in manufacturing systems using stochastic search techniques," *IEEE Transactions on Systems, Man and Cybernetics – Part A*, 26, 222-230.

Wang, M.Y. (2000), "An optimum design for 3-D fixture synthesis in a point set domain," *IEEE Transactions on Robotics & Automation*, 16, 839-846.

Wang, M.Y. and Pelinescu, D.M. (2001), "Optimizing fixture layout in a point-set domain," *IEEE Transactions on Robotics and Automation*, 17, 312-323.

Wang, M.Y., Liu, T. and Pelinescu, D.M. (2003), "Fixture kinematic analysis based on the full contact model of rigid bodies," ASME *Journal of Manufacturing Science and Engineering*, 125, 316-324.

Wang, Y. and Nagarkar, S.R. (1999), "Locator and sensor placement for automated coordinate checking fixtures," ASME *Journal of Manufacturing Science and Engineering*, 121, 709-719.

Xiong, C., Rong, Y., Koganti, R., Zaluzec, M. and Wang, N. (2002), "Geometric variation prediction in automotive assembly", *Assembly Automation Journal*, 22, 260-269.

Zhou, S., Huang, Q. and Shi, J. (2003), "State space modeling for dimensional monitoring of multistage machining process using differential motion vector," *IEEE Transactions on Robotics and Automation*, 19, 296-308.

APPENDIX

SYSTEM MATRICES A, B, C FOR THE INITIAL DESIGN

$\mathbf{A}_0$ is an identity matrix with appropriate dimensions.

$$\mathbf{A}_1 = \left[\begin{array}{cccccc:c} 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & \\ 0.0004 & 0.0009 & 1.0000 & -0.0004 & -0.0009 & -0.2855 & \mathbf{0}^{6\times6} \\ -1.1425 & -0.3497 & 0 & 1.1425 & 0.3497 & 105.3621 & \\ 0.3492 & -0.1430 & 0 & -0.3492 & 0.1430 & -258.1800 & \\ 0.0004 & 0.0009 & 0 & -0.0004 & -0.0009 & 0.7145 & \\ \hdashline \multicolumn{6}{c:}{\mathbf{0}^{6\times6}} & \mathbf{I}^{6\times6} \end{array}\right]_{12\times12}$$

$$\mathbf{A}_2 = \left[\begin{array}{ccc:ccc:ccc:c} 0 & 0 & 0 & & 0 & 0 & 0 & \\ 0 & 0 & 0 & \mathbf{0}^{3\times3} & 0 & 0 & 0 & \mathbf{0}^{3\times3} \\ -0.0003 & 0.0007 & 1 & & 0.0003 & -0.0007 & 0.2056 & \\ -0.8859 & -0.2529 & 0 & & -0.1141 & 0.2529 & -75.8764 & \\ -0.2797 & -0.3802 & 0 & \mathbf{I}^{3\times3} & 0.2797 & -0.6198 & 185.9280 & \mathbf{0}^{3\times3} \\ -0.0003 & 0.0007 & 0 & & 0.0003 & -0.0007 & 0.2056 & \\ -0.8308 & -0.3749 & 0 & & 0.8308 & 0.3749 & -112.4780 & \\ -0.4677 & 0.0364 & 0 & \mathbf{0}^{3\times3} & 0.4677 & -0.0364 & 310.9287 & \mathbf{0}^{3\times3} \\ -0.0003 & 0.0007 & 0 & & 0.0003 & -0.0007 & 1.2056 & \\ \hdashline \multicolumn{3}{c:}{\mathbf{0}^{3\times3}} & \mathbf{0}^{3\times3} & \multicolumn{3}{c:}{\mathbf{0}^{3\times3}} & \mathbf{I}^{3\times3} \end{array}\right]_{12\times12}$$

$$\mathbf{A}_3 = \left[\begin{array}{ccc:ccc:c} 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{0}^{3\times6} & 0 & 0 & 0 \\ -0.0000 & 0.0000 & 1 & & 0.0000 & -0.0000 & 0.0004 \\ -0.0010 & -0.0002 & 0 & & -0.0000 & 0.0002 & -0.1468 \\ -0.0001 & -0.0005 & 0 & & 0.0001 & -0.0005 & 0.3597 \\ -0.0000 & 0.0000 & 0 & & 0.0000 & -0.0000 & 0.0004 \\ -0.0009 & -0.0003 & 0 & \mathbf{I}^{6\times6} & -0.0001 & 0.0003 & -0.2176 \\ -0.0002 & -0.0002 & 0 & & 0.0002 & -0.0008 & 0.6015 \\ -0.0000 & 0.0000 & 0 & & 0.0000 & -0.0000 & 0.0004 \\ -0.0009 & -0.0003 & 0 & & 0.0009 & 0.0003 & -0.2049 \\ -0.0003 & 0.0003 & 0 & \mathbf{0}^{3\times6} & 0.0003 & -0.0003 & 1.0133 \\ -0.0000 & 0.0000 & 0 & & 0.0000 & -0.0000 & 0.0014 \end{array}\right]_{12\times12}$$

$$\mathbf{B}_1 = \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & & & & \\ 0.0016 & -0.0012 & -0.0016 & 0.0012 & & \mathbf{0}^{3\times4} & & \\ \hdashline & & & & 1 & 0 & 0 & 0 \\ & \mathbf{0}^{3\times4} & & & 0 & 1 & 0 & 0 \\ & & & & -0.0014 & 0 & 0.0014 & 0 \\ \hdashline & & \mathbf{0}^{6\times8} & & & & & \end{array}\right]_{12\times8}$$

$$\mathbf{B}_3 = \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & & & & \\ 0.0003 & -0.0007 & -0.0003 & 0.0007 & & & & \\ 0.8859 & 0.2529 & 0.1141 & -0.2529 & & & & \\ 0.2797 & 0.3802 & -0.2797 & 0.6198 & & \mathbf{0}^{9\times4} & & \\ 0.0003 & -0.0007 & -0.0003 & 0.0007 & & & & \\ 0.8308 & 0.3749 & 0.1692 & -0.3749 & & & & \\ 0.4677 & -0.0364 & -0.4677 & 1.0364 & & & & \\ 0.0003 & -0.0007 & -0.0003 & 0.0007 & & & & \\ \hdashline & & & & 1 & 0 & 0 & 0 \\ & \mathbf{0}^{3\times4} & & & 0 & 1 & 0 & 0 \\ & & & & -0.0002 & 0.0013 & 0.0002 & -0.0013 \end{array}\right]_{12\times8}$$

$$\mathbf{B}_2 = \left[\begin{array}{cccc:cccc} 1 & 0 & 0 & 0 & & & & \\ 0 & 1 & 0 & 0 & & & & \\ -0.0004 & -0.0009 & 0.0004 & 0.0009 & & & & \\ 1.1425 & 0.3497 & -0.1425 & -0.3497 & & \mathbf{0}^{6\times4} & & \\ -0.3492 & 0.1430 & 0.3492 & 0.8570 & & & & \\ -0.0004 & -0.0009 & 0.0004 & 0.0009 & & & & \\ \hdashline & & & & 1 & 0 & 0 & 0 \\ & & & & 0 & 1 & 0 & 0 \\ & \mathbf{0}^{6\times4} & & & 0.0000 & 0.0033 & -0.0000 & -0.0033 \\ & & & & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \end{array}\right]_{12\times8}$$

$$\mathbf{C}_4 = \left[\begin{array}{ccc:ccc:ccc:ccc} 1 & 0 & 46 & & & & & & & & & \\ 0 & 1 & -89.8 & & \mathbf{0}^{4\times3} & & & \mathbf{0}^{4\times3} & & & \mathbf{0}^{4\times3} & \\ 1 & 0 & -62.9 & & & & & & & & & \\ 0 & 1 & 43.2 & & & & & & & & & \\ \hdashline & & & 1 & 0 & -25 & & & & & & \\ & & & 0 & 1 & 3 & & & & & & \\ & \mathbf{0}^{6\times3} & & 1 & 0 & 641.5 & & \mathbf{0}^{6\times3} & & & \mathbf{0}^{6\times3} & \\ & & & 0 & 1 & 34.5 & & & & & & \\ & & & 1 & 0 & 1074 & & & & & & \\ & & & 0 & 1 & -27.5 & & & & & & \\ \hdashline & \mathbf{0}^{2\times3} & & & \mathbf{0}^{2\times3} & & 1 & 0 & -82 & & \mathbf{0}^{2\times3} & \\ & & & & & & 0 & 1 & -149.9 & & & \\ \hdashline & & & & & & & & & 1 & 0 & -114 \\ & & & & & & & & & 0 & 1 & 30 \\ & & & & & & & & & 1 & 0 & 1115.9 \\ & \mathbf{0}^{8\times3} & & & \mathbf{0}^{8\times3} & & & \mathbf{0}^{8\times3} & & 0 & 1 & -108.7 \\ & & & & & & & & & 1 & 0 & 1221 \\ & & & & & & & & & 0 & 1 & -877.1 \\ & & & & & & & & & 1 & 0 & 71 \\ & & & & & & & & & 0 & 1 & -830 \end{array}\right]_{20\times12}$$

VITA

Name: Pansoo Kim

Permanent Address: Hwamok Apt #101-406, Anrak 1 Dong, Dongrae-Gu

Pusan, Republic of Korea, 607-770

Educational Background: B.S. 1991. Department of Industrial Engineering

Pusan National University

M.S. 1994. Department of Industrial Engineering

Pusan National University