IIE Transactions
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713772245

Optimal sensor distribution in multi-station assembly processes for maximal variance detection capability

Yuan Ren [a]; Yu Ding [a]
[a] Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA

Online Publication Date: 01 September 2009

To cite this Article Ren, Yuan and Ding, Yu(2009)'Optimal sensor distribution in multi-station assembly processes for maximal variance detection capability',IIE Transactions,41:9,804 — 818

To link to this Article: DOI: 10.1080/07408170902789050

URL: http://dx.doi.org/10.1080/07408170902789050

IIE Transactions (2009) **41**, 804–818

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170902789050

Optimal sensor distribution in multi-station assembly processes for maximal variance detection capability

YUAN REN and YU DING*

Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843-3131, USA
E-mail: yuding@iemail.tamu.edu

Received January 2007 and accepted September 2007

Recent advances in sensor technology now allow manufacturers to distribute multiple sensors in multi-station assembly processes. A distributed sensor system enables the continual monitoring of manufactured products and greatly facilitates the determination of the underlying process variation sources that cause product quality defects. This paper addresses the problem of optimally distributing sensors in a multi-station assembly process to achieve a maximal variance detection capability. A sensitivity index is proposed for characterizing the detection ability of process variance components and the optimization problem for sensor distribution is formulated for a multi-station assembly process. A data-mining-guided evolutionary method is devised to solve this non-linear optimization problem. The data-mining-guided method demonstrates a considerable improvement compared to the existing alternatives. Guidance on practical issues such as the interpretation of the rules generated by the data mining method and how many sensors are required are also provided.

Keywords: Multi-station assembly process, sensitivity, data mining, evolutionary algorithms

1. Introduction

Variation propagation is a common phenomenon in a production system and profoundly affects the quality control practice in a multi-station manufacturing process (Ceglarek and Shi, 1995). Establishing sound quality assurance strategies hinges upon how well one can observe changes of variation in the multi-station manufacturing processes. Recent innovations in sensor technology provide manufacturers the ability to deploy automated sensing devices at multiple stations along the production line for tracking the variation level of a product. A major challenge is where in the production process to distribute these sensing devices so as to achieve the maximum detection capability of variation contributors at an affordable sensor cost. The complex interaction between various constraint factors and operational alternatives makes the solution far from trivial.

Research studies on sensor distribution in discrete manufacturing processes were surveyed in Mandroli *et al.* (2006). Relevant research efforts can be classified into two categories: (i) inspection-oriented quality assurance strategies; and (ii) diagnosis-oriented sensor-distribution strategies. The goal of an inspection-oriented strategy is to optimally allocate inspection resources by considering trade-offs of various cost components associated with inspection, repair and scraping due to product defects, and warranty penalties for the case where defective products have been shipped to customers. Inspection-oriented strategies do improve the quality of products customers eventually receive but do nothing to the underlying manufacturing process to stop it producing defective products. Diagnosis-oriented sensor distribution strategies, on the other hand, focus on identifying as well as eliminating the root causes of product defects. To establish such strategies, people usually assume that a set of underlying yet unknown variables (e.g., fixture errors in assembly processes) is responsible for the product quality defects. Since the underlying variables are not directly observable, statistical inferences have to be made about them from sensory information in order to determine which underlying variables cause the product defects. Therefore, the goal in a diagnosis-oriented strategy is to find a deployment of sensors such that the root causes of product defects are identifiable in a certain optimal sense. This paper intends to address a diagnosis-oriented sensor distribution strategy.

From the survey made by Mandroli *et al.* (2006), especially Tables 5 and 6 therein, it is clear that research on sensor placement distribution in multi-station systems is relatively limited; by comparison the majority of work on sensor placement has focused on a single-station setting (Mandroli *et al.*, 2006). The following is a quick recap

*Corresponding author

of the relevant literature related to sensor distribution in a multi-station process. We will review previous work according to different performance measures of a sensor system. Primarily three performance measures have been reported in the literature: (i) minimum distance between variation patterns; (ii) diagnosability; and (iii) sensitivity.

Ceglarek and Shi (1996) have developed a pattern matching procedure for variation diagnosis. Based on the variation patterns, researchers tried to maximize the minimum distance among different patterns. Using this performance measure, Khan *et al.* (1998) studied "end-of-line sensing," where the sensing station is located at the end of a manufacturing system but variation sources include those from upstream stations. Khan and Ceglarek (2000) studied "distributed sensing," which allows sensors to be placed at any station. Their methodology is based on a single-fixture model and thus does not consider interaction effects of variation sources between stations. However, such interaction effects between stations are actually the reason causing the additional complexity in multi-station manufacturing processes and thus should be considered. Recently, researchers developed a recursive station-indexed state space model to capture the complicated phenomenon of variation propagation in a multi-station process and this technique has been successfully applied to assembly processes (Jin and Shi, 1999; Ding *et al.*, 2000; Camelio *et al.*, 2003) as well as machining processes (Djurdjanovic and Ni, 2003; Huang *et al.*, 2003; Zhou *et al.*, 2003).

Based on a state space variation model, Ding *et al.* (2003) presented a sensor distribution strategy in a multi-station assembly process to guarantee a full diagnosability, where all the variation sources can be uniquely identified by the resulting sensor system. The diagnosability condition is essentially to make the corresponding diagnosability matrix non-singular but it does not regulate how well-conditioned a diagnosability matrix should be, or equivalently, how large should be the eigenvalues of a diagnosability matrix. When a sensor distribution strategy achieves full diagnosability, it means that any change in the underlying process variation will produce a difference in the sensor outputs so that such a change can theoretically be detected. However, in reality, the change in sensor readings could be very small and therefore easily overwhelmed by background disturbances, thereby becoming not useful for diagnosis purposes. For example, for a fully diagnosable system with no other requirements, one unit change in the variance of variation sources might only result in a 0.01 unit change in the variance of sensor measurements. Moreover, using the diagnosability criterion alone, as in Ding *et al.* (2003), does not differentiate between diagnosable systems in terms of their capability to detect small changes in variation sources. Using the diagnosability criterion, the diagnosable system in the above example is no different from another diagnosable system that can produce a 10-unit change in sensor readings for every unit change in the variation sources.

Another often used criterion for sensor placement is the sensitivity of a sensor system (Wang and Nagarkar, 1999; Camelio *et al.*, 2005; Liu *et al.*, 2005) – we will present the formal definition of the sensitivity in Section 2. Intuitively, the sensitivity is related to the detection capability of a sensor system. Mathematically, it regulates the eigenvalues of a diagnosability matrix to make sure that a change in the underlying process will produce a large enough change in sensor readings. Liu *et al.* (2005) discussed the mathematical relationship between diagnosability and sensitivity, and stated that "a sensor system that has zero sensitivity to any one of the variation sources provides no diagnosability, whereas a sensor system with non-zero sensitivity to all variation sources possesses a certain level of diagnosability." In summary, optimizing the sensitivity criterion will lead to a maximized detection capability for a sensor system and ensure a maximum separation of variation sources. The objective of this paper is to study the sensitivity measure of a distributed sensor system in a multi-station assembly process and devise an algorithm that efficiently solves the corresponding optimization problem.

The aforementioned work using the sensivity criterion (Wang and Nagarkar, 1999; Camelio *et al.*, 2005; Liu *et al.*, 2005) was conducted primarily under a single-station setting. This is not surprising because solving the sensor distribution problem for a multi-station system is generally more difficult, a point argued previously in Ding *et al.* (2006) and Mandroli *et al.* (2006). Unlike the sensor distribution problem on single-station systems, each sensor in our problem could be installed at any location of any part at any station. As a result, the pool of candidate sensor locations is much larger. The sensitivity measure is a non-linear function of the sensor locations and is accompanied by complicated geometric constraints (which typically come from the parts or subassemblies that the sensors are meant to measure). According to our experience, the sensitivity measure may not be continuous over the design space. This is because when a sensor location is changed from one station to another station or from a part to another part, the value of the sensitivity measure could have a point of discontinuity.

Given the relationship between the diagnosability and sensitivity measures, one may wonder if we could integrate the strategy in Ding *et al.* (2003) and a single-station sensor placement algorithm to solve our sensor distribution problem. The procedure might go as follows. First use the approach in Ding *et al.* (2003) to determine on which stations to distribute sensors in order to achieve full diagnosability. Then, optimize the sensor locations within individual stations to maximize the sensitivity measure. The problem with this two-step procedure is that solving an optimization problem in two suboptimal steps typically does not produce an optimal or nearly optimal result. As will be shown in Section 4 for a three-station example, doing so could incorrectly filter out some promising solution regions and eventually lead to a solution that is much worse than solving the optimization problem in an integrated manner.

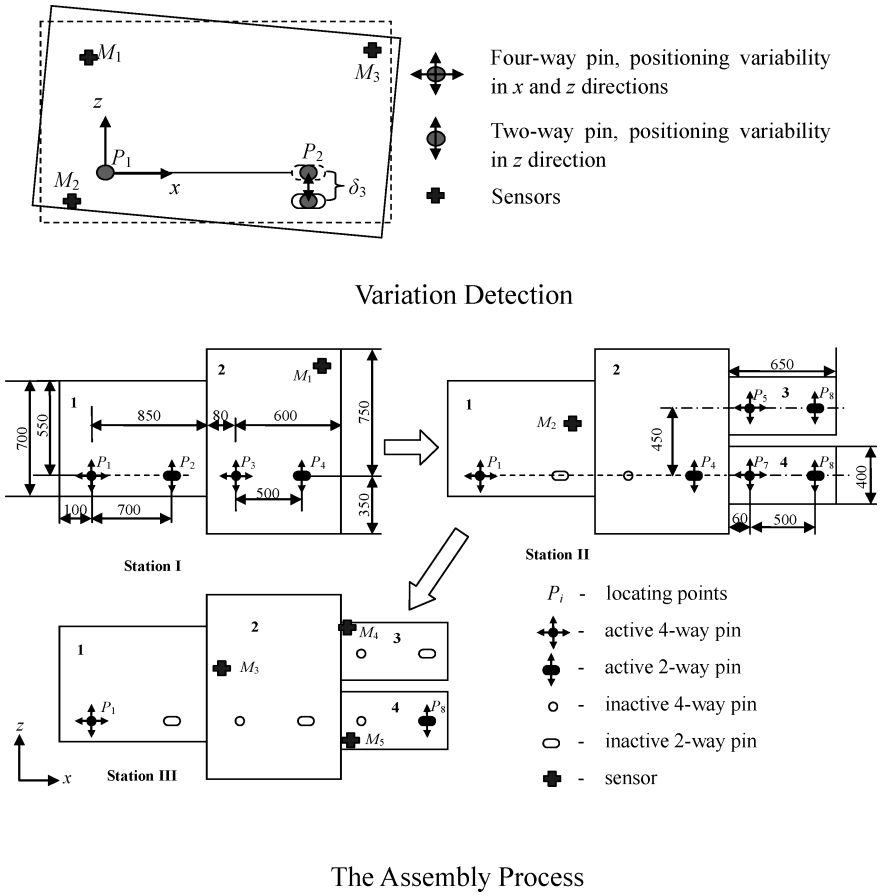

Fig. 1. Illustrative example: a multi-station assembly process.

Thus, this paper solves the sensor distribution problem using a single optimization formulation. The solution approach we undertake is a data-mining-guided evolutionary approach, following a general idea initially proposed in Michalski (2000). This data-mining-guided method has two ingredients: a data-mining procedure that predicts the subregions where a "good" sensor location could be and an evolutionary procedure that diversifies, and thus enhances the quality of, the samples to be used for the data mining and prediction purposes. Our results demonstrate that this approach can handle the non-linear optimization problem for sensor distribution very well and that it can find a better solution in shorter time as compared with the procedures using a data-mining approach or an evolutionary approach alone.

Following this introduction, we unfold our investigation as follows. Section 2 presents the problem formulation and discusses the design criterion. Section 3 explains the general idea of the proposed approach to solve the optimal sensor distribution problem, and also presents the details for realizing the data-mining guided approach. Section 4 implements the proposed method and investigates its performance. Section 5 concludes the paper.

2. Formulation of the sensor distribution problem

Consider as an example the three-station two-dimensional (2-D) assembly process in Fig. 1. The three-station assembly process proceeds as follows: (i) at the first station, part 1 and part 2 are assembled; (ii) at the second station, the subassembly consisting of parts 1 and 2 receives part 3 and part 4; and (iii) at the third station, no assembly operation is performed but the key dimensional features of the final assembly are measured. The quality assurance objective here is about the dimensional integrity of the assembly product. Optical coordinate sensors M_1 to M_5 are distributed throughout the assembly process to monitor the dimensional quality of the final assembly and/or of the intermediate subassemblies. The major variation sources in such a process are associated with the fixture locators on different stations – as shown in Fig. 1, each part or subassembly is held by a set of fixtures, which consists of a four-way pin P_1 that constrains the part motion in both the x and the z directions, and a two-way pin P_2 that constrains the part motion in the z-direction. An optimal sensor distribution should be able to identify these fixturing variation sources uniquely and accurately.

In order to calculate the sensitivity index, we first need to have the relationship between the fixturing variation and the sensor measurements in a multi-station assembly process. This relation is in fact represented using a station-indexed state space model. The detailed model development was reported in Ding *et al.* (2002). We just summarize the results that will be used in this paper. For a multi-station assembly process, the state space model can be expressed as

$$\begin{aligned} \mathbf{x}_{k-1} &= \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \boldsymbol{\xi}_k, \\ \mathbf{y}_{k-1} &= \mathbf{C}_k \mathbf{x}_k + \boldsymbol{\eta}_k, \end{aligned} \tag{1}$$

where $k \in 1, 2, \ldots, N$ is the station index, and N is the number of stations. The state vector $\mathbf{x}_k$ and input vector $\mathbf{u}_k$ are the product accumulated deviations and the fixture deviation on station k. The process disturbances and sensor noises are denoted by $\boldsymbol{\xi}_k$ and $\boldsymbol{\eta}_k$, respectively. Product measurements at station k are included in $\mathbf{y}_k$. In Figure 1, for instance, $\mathbf{y}_3$ comprises the deviations measured by sensors M_3 to M_5. The $\mathbf{A}_k$ is the state transition matrix which links the part deviation states on adjacent stations, $\mathbf{B}_k$ is the input matrix which represents the effect from the fixture deviations and $\mathbf{C}_k$ is the observation matrix corresponding to the number and locations of sensors.

By eliminating the intermediate state variables and aggregating the information associated with individual stations, Equation (1) could be further formulated into an input–output relation as

$$\mathbf{y} = \boldsymbol{\Gamma} \times \mathbf{u} + \boldsymbol{\Gamma}_0 \times \mathbf{x}_0 + \boldsymbol{\varepsilon}. \tag{2}$$

where $\mathbf{y}^{\mathrm{T}} \equiv [\mathbf{y}_1^{\mathrm{T}}\ \mathbf{y}_2^{\mathrm{T}} \ldots \mathbf{y}_N^{\mathrm{T}}]$, $\mathbf{u}^{\mathrm{T}} \equiv [\mathbf{u}_1^{\mathrm{T}}\ \mathbf{u}_2^{\mathrm{T}} \ldots \mathbf{u}_N^{\mathrm{T}}]$, $\boldsymbol{\varepsilon}^{\mathrm{T}} \equiv [\boldsymbol{\varepsilon}_1^{\mathrm{T}}\ \boldsymbol{\varepsilon}_2^{\mathrm{T}} \ldots \boldsymbol{\varepsilon}_N^{\mathrm{T}}]$, $\boldsymbol{\varepsilon}_k \equiv \Sigma_{i=1}^{k} \mathbf{C}_k \boldsymbol{\Phi}_{k,i} \boldsymbol{\xi}_i + \boldsymbol{\eta}_k$, $\boldsymbol{\Phi}_{k,i} \equiv \mathbf{A}_{k-1}\mathbf{A}_{k-2} \cdots \mathbf{A}_i$, and

$$\boldsymbol{\Gamma} \equiv \begin{bmatrix} \mathbf{C}_1\mathbf{B}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}_2\boldsymbol{\Phi}_{2,1}\mathbf{B}_1 & \mathbf{C}_2\mathbf{B}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_N\boldsymbol{\Phi}_{N,1}\mathbf{B}_1 & \mathbf{C}_N\boldsymbol{\Phi}_{N,2}\mathbf{B}_2 & \cdots & \mathbf{C}_N\mathbf{B}_N \end{bmatrix},$$

$$\boldsymbol{\Gamma}_0 \equiv \begin{bmatrix} \mathbf{C}_1\boldsymbol{\Phi}_{1,0} \\ \mathbf{C}_2\boldsymbol{\Phi}_{2,0} \\ \vdots \\ \mathbf{C}_N\boldsymbol{\Phi}_{N,0} \end{bmatrix}. \tag{3}$$

Our goal is to detect the variance change in the variation sources $\mathbf{u}$. For that purpose, we transform model (2) into a variation model. We assume that the product deviation $\mathbf{x}_0$, the fixture deviation $\mathbf{u}$ and the noise term $\boldsymbol{\varepsilon}$ are independent, and we have

$$\boldsymbol{\Sigma}_{\mathrm{y}} = \boldsymbol{\Gamma} \times \boldsymbol{\Sigma}_{\mathbf{u}} \times \boldsymbol{\Gamma}^{\mathrm{T}} + \boldsymbol{\Gamma}_0 \times \boldsymbol{\Sigma}_{\mathbf{x}_0} \times \boldsymbol{\Gamma}_0^{\mathrm{T}} + \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}. \tag{4}$$

Note that $\boldsymbol{\Sigma}_{\mathbf{u}}$ is a diagonal matrix since the fixture deviations are physically uncorrelated. As such, $\mathrm{diag}(\boldsymbol{\Sigma}_{\mathbf{u}})$ is the vector of the variances of variation sources. For the sake of notational simplicity, we use $\boldsymbol{\sigma}$ to represent $\mathrm{diag}(\boldsymbol{\Sigma}_{\mathbf{u}})$ in the following discussions. The following simplification follows what was done in Ding *et al.* (2002) and Ding *et al.* (2003). Consider that $\boldsymbol{\Sigma}_{\mathbf{x}_0}$ is known from measurements at the end of the precedent fabrication process and that $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$ can be estimated using data from a normal process condition when no outstanding fixture deviation occurs, then a new covariance matrix is introduced as $\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{y}} - \boldsymbol{\Gamma}_0 \times \boldsymbol{\Sigma}_{\mathbf{x}_0} \times \boldsymbol{\Gamma}_0^{\mathrm{T}} - \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$ to summarize the known quantities. Then Equation (4) can be rewritten as

$$\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}} = \boldsymbol{\Gamma} \times \boldsymbol{\Sigma}_{\mathbf{u}} \times \boldsymbol{\Gamma}^{\mathrm{T}}. \tag{5}$$

Using the $\boldsymbol{\pi}(\cdot)$-transform introduced in Ding *et al.* (2002), Equation (5) can be expressed as

$$\mathrm{vec}(\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}}) = \boldsymbol{\pi}(\boldsymbol{\Gamma}) \times \mathrm{diag}(\boldsymbol{\Sigma}_{\mathbf{u}}) = \boldsymbol{\pi}(\boldsymbol{\Gamma}) \times \boldsymbol{\sigma}, \tag{6}$$

where $\mathrm{vec}(\cdot)$ is the vector operator (Schott, 1997), $\boldsymbol{\pi}(\cdot)$ is a matrix transform defined as

$$\begin{aligned} \boldsymbol{\pi}(\boldsymbol{\Gamma}) = [&(\boldsymbol{\tau}^1 \otimes \boldsymbol{\tau}^1)^{\mathrm{T}} \cdots (\boldsymbol{\tau}^1 \otimes \boldsymbol{\tau}^n)^{\mathrm{T}} \cdots (\boldsymbol{\tau}^n \otimes \boldsymbol{\tau}^1)^{\mathrm{T}} \cdots \\ &(\boldsymbol{\tau}^n \otimes \boldsymbol{\tau}^n)^{\mathrm{T}}]^{\mathrm{T}}, \end{aligned} \tag{7}$$

$\boldsymbol{\tau}^j$ is the jth row of $\boldsymbol{\Gamma}$, $j = 1, \ldots, n$, and $\otimes$ represents the Hadamard product (Schott, 1997).

The diagnosability criterion used in Ding *et al.* (2003) ensures that $\boldsymbol{\pi}(\boldsymbol{\Gamma})$ is full rank so that the variance components in $\boldsymbol{\sigma}$ are uniquely identifiable. Under a single-station setting, Liu *et al.* (2005) defined the variance-detecting sensitivity as the ratio of the change in the variance of measurements over a perturbation of the variance components in $\boldsymbol{\sigma}$, that is to say

$$S \equiv \min_{\delta\boldsymbol{\sigma} \neq \mathbf{0}} \frac{\mathrm{tr}(\delta\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}}^{\mathrm{T}} \delta\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}})}{(\delta\boldsymbol{\sigma})^{\mathrm{T}}(\delta\boldsymbol{\sigma})} = \min_{\delta\boldsymbol{\sigma} \neq \mathbf{0}} \frac{\mathrm{vec}(\delta\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}})^{\mathrm{T}} \times \mathrm{vec}(\delta\tilde{\boldsymbol{\Sigma}}_{\mathbf{y}})}{(\delta\boldsymbol{\sigma})^{\mathrm{T}}(\delta\boldsymbol{\sigma})}, \tag{8}$$

where δ denotes the perturbation operator. Using Equation (6) and the eigenvalue property of a symmetric matrix (Schott, 1997, Theorem 3.16, pp. 105), it is easy to show that:

$$S = \lambda_{\min}(\boldsymbol{\pi}(\boldsymbol{\Gamma})^{\mathrm{T}} \boldsymbol{\pi}(\boldsymbol{\Gamma})), \tag{9}$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue of a matrix. It is now clear that the variance-detecting sensitivity actually belongs to the family of the so-called alphabetic optimality criteria in the optimal experiment design, including A-optimality, D-optimality, and E-optimality, which were defined using an algebraic form of the eigenvalues of the Fisher information matrix (Pukelsheim, 1993). Here, $\boldsymbol{\pi}(\boldsymbol{\Gamma})^{\mathrm{T}}\boldsymbol{\pi}(\boldsymbol{\Gamma})$ is actually the Fisher information matrix for detecting the variance components and this index S is in fact the E-optimality criterion. Liu *et al.* (2005) further stated that maximizing S is also equivalent to minimizing the maximum variance of a linear parametric function of the variation components to be estimated.

Apparently, the sensitivity index defined in Equation (8) is quite general and is not limited to a single station process. We therefore adopt this E-optimality sensitivity index

as the design criterion for our multi-station assembly problem. Then, our optimal sensor distribution problem is to maximize the sensitivity index in Equation (9), i.e., maximize the smallest eigenvalue associated with the Fisher information matrix $\boldsymbol{\pi}(\boldsymbol{\Gamma})^{\mathrm{T}}\boldsymbol{\pi}(\boldsymbol{\Gamma})$.

To formulate this optimization problem, denote by s the number of sensors, by X_i, Y_i and Z_i the coordinates of where the ith sensor is located and by T_i the station on which the ith sensor is placed. Then, the design of the sensor distribution can be represented by $\mathbf{w} = [X_1\ Y_1\ Z_1\ T_1 \ldots X_s\ Y_s\ Z_s\ T_s]$. The geometric constraint (since a sensor can only measure some valid area on a product) is represented by $G(\mathbf{w}) \geq 0$. As such, the sensor distribution problem for a specified number of sensors is to find the optimal sensor locations that maximize S, namely:

$$\max_{\mathbf{w}} S(\mathbf{w}) \equiv \lambda_{\min}(\boldsymbol{\pi}(\boldsymbol{\Gamma})^{\mathrm{T}}\boldsymbol{\pi}(\boldsymbol{\Gamma})),$$
$$\text{subject to } G(\mathbf{w}) \geq 0. \tag{10}$$

Several optimization methods have been applied to solve the sensor distribution problem; please refer to Table 6 in Mandroli *et al.* (2006) for the applications of specific algorithms. These optimization methods include nonlinear programming methods such as sequential quadratic programming (Khan *et al.*, 1999), exchange algorithms (Camelio *et al.*, 2005; Liu *et al.*, 2005) and Genetic Algorithms (GAs) (Djurdjanovic and Ni, 2004). Recently, data-mining methods have also been used to solve optimization problems in engineering designs (Schwabacher *et al.* 2001; Igusa *et al.*, 2003; Kim and Ding, 2005). In this paper, we use a data-mining-guided evolutionary approach, which borrows strengths from both methods and compensates the weakness of the individual ones in solving a non-linear optimization problem. Our research finds that integrating the two methods produces a desirable outcome in solving the sensor distribution problem at hand – it can find a good solution in a timely fashion as compared to other alternatives.

In order for the subsequent data-mining method to be easily applied, which needs to perform sampling operations and other operations on datasets, we first discretize the continuous design space to form a dataset of all potential sensor locations. For the 2-D assembly process demonstrated in Fig. 1, we discretize the valid geometric area of each part using a resolution of 10 mm (which is the size of a locator's diameter). Our engineering experience and prior studies (e.g., Liu *et al.* (2005)) indicate that this resolution is fine enough for a part size of several hundred millimeters. For an s-sensor design problem (i.e., the sensor number is fixed at s), denote by N_{c} the resulting number of candidate locations for any single sensor and by θ_k the index of the location of the kth sensor, where $1 \leq \theta_k \leq N_{\mathrm{c}}$, $k = 1, 2, \ldots, s$. An instance of sensor distribution can be denoted by $\boldsymbol{\theta} = [\theta_1, \ldots \theta_s]$.

After discetization, we label each candidate sensor location uniquely so that the original representation $\mathbf{w}$ of a location in the continuous design space, comprising the coordinates, (X, Y, Z), and the station index T, is replaced by a single index θ. Consequently, the original continuous design space, which is $4s$-dimensional, is reduced to an s-dimensional discrete space, denoted by $\boldsymbol{\Omega} = [1, N_{\mathrm{c}}]^s \cap \mathbf{Z}^s$, where $\mathbf{Z}^s$ is the set of s-dimensional vectors with integer elements. Since all s sensors to be installed are assumed to be identical, the cardinality of the set $\boldsymbol{\Omega}$ is $|\boldsymbol{\Omega}| = C_s^{N_{\mathrm{c}}}$, where C_a^b denotes the number of ways to choose a objects from a set of size b. The continuous formulation has a station index T because two sensor locations having the same (X, Y, Z) but on different stations are considered to be different. Having this station index would have created additional complexity for the subsequent data-mining methods to be applied and for the mined knowledge to be utilized. The discretization streamlines the representation, where the station index is no longer needed.

Given the new discrete design space for sensor distribution, the optimal sensor distribution problem in Equation (10) can be rewritten as follows: for a given number of sensors, find the optimal sensor locations that maximize S, namely:

$$\max_{\boldsymbol{\theta} \in \boldsymbol{\Omega}} S(\boldsymbol{\theta}) = \lambda_{\min}(\boldsymbol{\pi}(\boldsymbol{\Gamma})^{\mathrm{T}}\boldsymbol{\pi}(\boldsymbol{\Gamma})). \tag{11}$$

For the assembly process shown in Fig. 1, the 10 mm resolution level will result in the number of candidate sensor locations on each part as $n_1 = 6650$, $n_2 = 7480$, $n_3 = 2600$ and $n_4 = 2600$. Because parts 1 and 2 appear on all three stations, parts 3 and 4 appear on the second and the third stations, there is a total of $3 \times (n_1 + n_2) + 2 \times (n_3 + n_4) = 52\ 790$ candidate locations for each sensor. Thus, $N_{\mathrm{c}} = 52\ 790$ in this paper. Suppose that $s = 5$, then the total number of design alternatives, namely $|\boldsymbol{\Omega}|$, is $C_5^{52\ 790} \approx 3.4 \times 10^{21}$. Apparently this number is overwhelmingly large, which makes the exchange algorithms difficult to apply or a GA-based random search less likely to be efficient.

3. Optimal sensor distribution by a data-mining-guided evolutionary approach

3.1. *Basic idea*

Applying data-mining methods to optimization problems is a recent development (Schwabacher *et al.*, 2001; Igusa *et al.*, 2003; Kim and Ding, 2005). By data-mining methods, people typically refer to sampling, classification and clustering methods. The basic idea is to consider an optimization problem as selecting the best solution (or a better one) from a large set of solution alternatives. If treating the solution alternatives as a dataset, a data-mining method may be able to discover valuable structures within it and then generalize insightful selection rules that could eventually lead to a better solution, or when lucky enough, to the optimal solution. There are a few recent successes in applying this idea to various optimal design problems. Igusa

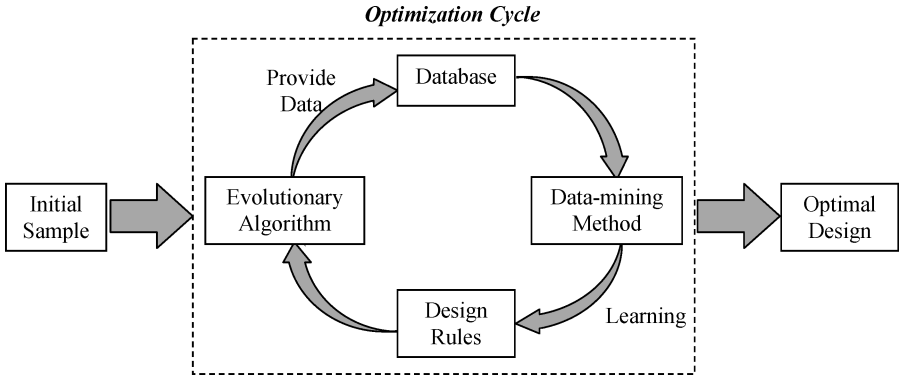

Fig. 2. General framework of combining an evolutionary algorithm and the data-mining approach.

et al. (2003) and Kim and Ding (2005) demonstrated that the data-mining ingredients could greatly speed up computation; the former reported a 15-fold time reduction (as compared to GAs) in their civil structure optimization and the latter reported a 10-fold time reduction (as compared to a simulated annealing algorithm) in their optimal fixture layout design.

However, there are also some limitations in the current version of a data-mining-guided method. For example, one of the reasons behind the computation benefit shown in Igusa *et al.* (2003) and Kim and Ding (2005) is because a set of feature functions is used to replace the computationally more expensive objective function at some stage. However, introducing the appropriate set of feature functions is *ad hoc* and may not be universally applicable. A more common shortcoming of data-mining-guided methods is that their effectiveness strongly depends on how representative the initially and subsequently sampled solution alternatives are of the solution space (also called the design space). Without representative data of the solution alternatives, data mining can give us inaccurate information, leading to the choice of a wrong direction. In particular, when the design space is too large and the good solution candidates are too sparse, data obtained from a uniform sampling may not be sufficiently representative. Taking the sensor distribution in the three-station assembly process as an example, when the number of sensors is five, we found that only 5% of the solution alternatives can be considered as promising candidates, having relatively large sensitivity values. Consequently, purely based on a data-mining method, we might not be able to obtain enough representative solution candidates to produce informative guidelines. If we proceed with non-informative guidelines, our efforts to find good designs could become tainted or even wasted. For this reason, we feel that an effective procedure is needed to compensate the ability of the current data-mining methods to find good solution alternatives with which to work.

It turns out that evolutionary algorithms, though slow as a stand-alone optimization tool, are able to improve the quality of the solution alternatives for their use in the data-mining methods. This is because when conducting random searches over a design space, an evolutionary algorithm will iteratively and stochastically direct the current solution sample toward the local optima so that the candidate solution alternatives are more representative of the solution space. With the representative datasets produced by an evolutionary algorithm, a data-mining method could generate more informative and accurate solution-selecting rules to characterize the promising subregions of the original design space. Also, this knowledge obtained in the data-mining process will be fed back to the evolutionary algorithm so that the subsequent evolution process can be guided to search only in promising regions to gather more representative data.

The above arguments actually point to integrating a data-mining method and an evolutionary algorithm and execute them alternately in a fashion as shown in Fig. 2. In fact, some efforts exploring this integration have been reported recently. Michalski (2000) presented a new evolutionary process called the Learnable Evolution Model (LEM), which employs both evolutionary algorithms and machine learning to generate new solutions. The LEM switches between a Darwinian evolution mode and a machine learning mode and enables possible quantum leaps. Huyet (2006) applied Michalski's idea to the analysis of simulated production systems.

We follow the general framework of the LEM to solve the sensor distribution problem. However, we make a number of modifications so that it better fits our problem. First, a Classification And Regression Tree (CART), instead of the AQ-learner (Michalski, 2000), is used as our data-mining method because a CART is computationally efficient (Hastie *et al.*, 2001). Since the data-mining method will be repeatedly used, being computationally efficient is essential otherwise it will significantly slow down the whole optimization process. Second, because of the difficulty in obtaining a representative dataset in our sensor distribution problem (due to the sparsity of the good solutions), we run the evolutionary algorithm at the beginning of

the combined procedure to collect enough representative data for the subsequent data-mining operation. Michalski (2000) did the opposite, which does not work well in our problem. Third, we use the information learned from data mining to regulate the evolutionary algorithm, i.e., limiting the evolutionary algorithm to search in promising regions in order to increase the convergence rate to the optimal solution. Michalski (2000) did not have this regulation component. A similar idea of limiting the search space was proposed in Mandal *et al.* (2006), where they proposed a modified genetic algorithm that utilizes the idea of a forbidden array and weighted mutation to reach the best solution in fewer runs. However, using a forbidden array may not be well suited for large-scale problems such as the one introduced in Section 2, where the decision variables could have 52 790 levels (in Mandal *et al.* (2006), the number of levels for each decision variable is 11). Having a large number of levels would make the required orthogonal array unattainable. Details of our proposed methods will become clear in the next subsection. We label this modified LEM as a data-mining-guided evolutionary approach.

Including the data-mining ingredient is beneficial to the optimal sensor distribution problem in the sense that in addition to obtaining the optimal solution for the current process configuration, one also garners guidelines about how to construct good sensor distribution alternatives, which can facilitate future optimization. For example, in the three-station assembly process, if the shape of a part is changed, a previously determined sensor location could become unfeasible. With the guidelines mined from previous design optimization processes, manufacturers could adjust their previous sensor distribution without having to run the optimization procedure again from the beginning.

3.2. *Data-mining guided evolutionary approach*

Borrowing the language used in a standard GA, each solution (or design) alternative of sensor distribution $\boldsymbol{\theta}$ is called an *individual*; a sample or collection of individuals is called a *population*, denoted by $\mathbf{P}$; a collection of populations is called a *database*, denoted by $\boldsymbol{\Theta}$, and obviously, $\boldsymbol{\theta} \in \mathbf{P} \subseteq \boldsymbol{\Theta}$. Moreover, if $\theta_k \in [1, N_c] \cap \mathbf{Z}$ for any k, we say $\boldsymbol{\theta} \in \boldsymbol{\Omega}$, and if for any $\boldsymbol{\theta} \in \mathbf{P}$, $\boldsymbol{\theta} \in \boldsymbol{\Omega}$, then we say $\mathbf{P} \subseteq \boldsymbol{\Omega}$.

The first step of the proposed method is to sample an initial population $\mathbf{P}_0$ of sensor distribution alternatives from the candidate design space $\boldsymbol{\Omega}$, where $\mathbf{P}_0 = \{\boldsymbol{\theta}_0^1, \boldsymbol{\theta}_0^2, \ldots, \boldsymbol{\theta}_0^M\}$ and M is the population size. At the stage of initial sampling, people typically do not have detailed knowledge about which regions of the space might contain good individuals. Under this circumstance, the most sensible way to perform the initial sampling has been suggested to be to sample individuals from $\boldsymbol{\Omega}$ as evenly as possible, a concept also known as "space-filling" design (Fang and Wang, 1994; Santner *et al.*, 2003). In our proposed method, a uniform sampling will be repeatedly used to obtain samples from some sub-regions of $\boldsymbol{\Omega}$. Thus, the sampling method should be easy to implement and computationally economic, which makes the sampling method based on the *minimax* or *maximin* criteria inappropriate for our application. We instead choose the Latin Hypercube Sampling method. Empirical evidence indicates that its use for uniform sampling purposes is satisfactory (Santner *et al.*, 2003).

After getting the initial population $\mathbf{P}_0 \subset \boldsymbol{\Omega}$, there will be two modes of operations executed alternately, the evolution mode and the data-mining mode. The evolution mode goes first, where an evolutionary algorithm (we use the standard GA as described in Holland (1992)) is used to randomly search over $\boldsymbol{\Omega}$ and gather representative individuals for the subsequent data-mining mode. The fitness (or representativeness) of an individual $\boldsymbol{\theta}$ is measured by its sensitivity value $S(\boldsymbol{\theta})$. Each iteration is called a generation, indexed by t. We denote the population in the tth generation (t starts from zero) by $\mathbf{P}_t = \{\boldsymbol{\theta}_t^1, \boldsymbol{\theta}_t^2, \ldots, \boldsymbol{\theta}_t^M\}$, where the population size is always kept at M. At this stage, the evolutionary algorithm selects the parental individuals from the current population according to their fitness, such that Prob($\boldsymbol{\theta}$ is selected) $\geq$ Prob($\boldsymbol{\theta}'$ is selected) if $S(\boldsymbol{\theta}) \geq S(\boldsymbol{\theta}')$. Then, the crossover operators are applied with probability p_c on two parental individuals to produce an offspring; and the mutation operators are employed with probability p_m to inject variation into a population.

After running evolutionary algorithms for a number of generations, we are ready to switch to the data-mining mode that is supposed to produce a new population for the subsequent evolution search. Each iteration in the data-mining mode also counts as a generation. Denote the database we have retained up to generation t by $\boldsymbol{\Theta}_t \subset \boldsymbol{\Omega}$. We classify $\boldsymbol{\Theta}_t$ into three categories: (i) those with relatively high sensitivity values (high-performance individuals, or HPI); (ii) those with relatively low sensitivity values (low-performance individuals, or LPI); and (iii) the others between the first two classes. The HPI and LPI are the representatives of the most and the least promising search regions, respectively. These two classes of individuals inform us about where to search and where to avoid. The individuals with a fitness in between, i.e., the category (iii) as mentioned above, are considered non-informative and will not be directly utilized. We denote the h and l percentiles of the S values in $\boldsymbol{\Theta}_t$ by $S_t^{(h)}$ and $S_t^{(l)}$, respectively. Then the class of HPI, $\mathbf{H}_t$, and the class of LPI, $\mathbf{L}_t$, at generation t, are defined as follows:

$$\mathbf{H}_t = \{\boldsymbol{\theta} : \boldsymbol{\theta} \in \boldsymbol{\Theta}_t \text{ and } S(\boldsymbol{\theta}) \geq S_t^{(h)}\}, \tag{12}$$

$$\mathbf{L}_t = \{\boldsymbol{\theta} : \boldsymbol{\theta} \in \boldsymbol{\Theta}_t \text{ and } S(\boldsymbol{\theta}) \leq S_t^{(h)}\}. \tag{13}$$

Naturally, $h > l$ so that the two classes of data do not overlap. As such, the three classes are represented by $\mathbf{H}_t$, $\mathbf{L}_t$ and $\boldsymbol{\Theta}_t \backslash (\mathbf{H}_t \cup \mathbf{L}_t)$, respectively, where $\mathbf{A} \backslash \mathbf{B}$ represents the set of elements that are in $\mathbf{A}$ but not in $\mathbf{B}$.

The method we recommend for the data-mining mode is CART (Breiman, 1984). The reason is as follows. Our goal of solving an optimization problem places several requirements on the data-mining method. As we mentioned earlier, first and foremost, it must be fast and computationally scalable to accommodate the large dataset of individuals. Since the data-mining method is repeatedly used to select new samples of individuals, a complicated, computationally expensive method will unavoidably slow down the optimization process. Second, the data-mining method should generate explicit descriptions that could differentiate several classes of individuals. These requirements and preferences make CART a favorable candidate to be used in the data-mining mode for optimization purpose.

CART is applied to both $\mathbf{H}_t$ and $\mathbf{L}_t$ and produces a set of "if–then" rules. In fact, the set of rules is the mined knowledge characterizing the design space and will be used to determine the promising search subareas. When the set of rules is applied to $\mathbf{\Omega}$, the candidate design space will be partitioned into a number of rectangular regions, generally expressed as $a_{tk} \le \theta_{tk} \le b_{tk}$, $k = 1, \ldots, s$, where a_{tk} and b_{tk} are between one and N_c, defining each region. Applying the set of rules learned from $\mathbf{H}_t$ and $\mathbf{L}_t$, we can identify the most promising search regions, denoted by $\mathbf{D}_t$ and the least promising search regions, denoted by $\mathbf{V}_t$. CART may produce multiple, non-connected regions in both $\mathbf{D}_t$ and $\mathbf{V}_t$; denote by r_t the number of rectangular regions in $\mathbf{D}_t$ and by q_t as the ones in $\mathbf{V}_t$. We have:

$$\mathbf{D}_t = \cup_{i=1}^{r_t} \mathbf{D}_{ti} \quad \text{and} \quad \mathbf{V}_t = \cup_{j=1}^{q_t} \mathbf{V}_{tj}, \tag{14}$$

where $\mathbf{D}_{ti} = [a_{ti1}, b_{ti1}] \times \cdots \times [a_{tis}, b_{tis}]$, $\mathbf{V}_{tj} = [c_{tj1}, d_{tj1}] \times \cdots \times [c_{tjs}, d_{tjs}]$, a_{tik} and b_{tik} are for region i of $\mathbf{D}_t$, $i = 1, \ldots, r_t$, and c_{tjk} and d_{tjk} are for region j of $\mathbf{V}_t$, $j = 1, \ldots, q_t$.

Subsequently, $\mathbf{D}_t$ and $\mathbf{V}_t$ are used to guide the next evolutionary search. Specifically, $\mathbf{D}_t$ is used for pointing out where to search. We uniformly sample individuals in $\mathbf{D}_t$ (instead of in $\mathbf{\Omega}$) and then deliver the sampled individuals to the evolutionary algorithm for further runs. The samples of individuals are generated as follows. For region i of $\mathbf{D}_t$, we first obtain a uniform design $\{\phi_{ik}^m\}$ of $\lceil M/r_t \rceil$ points, $m = 1, \ldots, \lceil M/r_t \rceil$, over a unit cube $\mathbf{C}^s = [0,1]^s$, where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x. Then, we map the design onto $\mathbf{D}_{ti}$ by calculating:

$$\theta_{tik}^m = a_{tik} + (b_{tik} - a_{tik}) \times \varphi_{ik}^m, \quad i = 1, \ldots, r_t, \\ m = 1, \ldots, \lceil M/r_t \rceil, k = 1, \ldots, s. \tag{15}$$

Then, $\boldsymbol{\theta}_{ti}^m = [\theta_{ti1}^m \cdots \theta_{tis}^m]$ is an individual in $\mathbf{D}_{ti}$. The set $\cup_{i=1}^{r_t} \{\boldsymbol{\theta}_{ti}^1, \boldsymbol{\theta}_{ti}^2, \ldots, \boldsymbol{\theta}_{ti}^{\lceil M/r_t \rceil}\}$ is the new sample of individuals generated by the data-mining mode. Combining this set with the current population $\mathbf{P}_t$ produces a new population $\mathbf{P}_{t+1}$. The individuals sampled over $\mathbf{D}_t$ are expected to have, on average, larger sensitivities than a uniform sample of individuals over $\mathbf{\Omega}$. Hence, the optimization procedure could converge to a better solution faster than the traditional, "blind" Darwinian-type evolution.

In the meanwhile, $\mathbf{V}_t$ is used to regulate the evolutionary algorithm to avoid certain "unwanted" individuals produced during the evolution process. Due to its nature of random search, the evolutionary algorithm may produce a new individual that is actually not a good choice for sensor distribution. Thus, we need to check if $\boldsymbol{\theta} \in \mathbf{V}_t$ once a new individual $\boldsymbol{\theta}$ is generated by the Darwinian-type operations (crossover or mutation). If $\boldsymbol{\theta}$ is indeed in the least promising region, we discard it and generate another individual until we have M individuals in the new population. In this way, we can prevent the evolutionary algorithm from going into some search regions that are not promising.

Because there is always misclassification in the data-mining mode and randomness in the evolution mode, the above procedure needs to iterate a number of times until a stopping rule is met.

One more thing we want to articulate is the construction of database $\mathbf{\Theta}_t$, on which the most and least promising search regions $\mathbf{D}_t$ and $\mathbf{V}_t$ are identified in the data-mining mode. Huyet (2006) constructed $\mathbf{\Theta}_t$ by using only the populations from the most recent generations, i.e., $\boldsymbol{\theta}_t = \cup_{t=B-b+1}^{B} \mathbf{P}_t$, where B is the generation count of the current population, and b is the number of generations used to construct $\mathbf{\Theta}_t$. Doing so may make the optimization process converge relatively quickly to an optimum. However, it also becomes more difficult for the algorithm to find a way of escaping a local optimum when only the recently sampled individuals are considered and the information from earlier generations is discarded. In this paper we instead choose to keep all the populations we obtained so far in $\mathbf{\Theta}_t$, i.e., $\mathbf{\Theta}_t = \cup_{t=1}^{B} \mathbf{P}_t$. The benefit of retaining all the available information up to the current generation is that it can help prevent all the individuals in $\mathbf{\Theta}_t$ being from a narrow region. That certainly boosts the algorithm's ability to avoid local optima. Of course, retaining all the populations could make the learning process in the data-mining mode slower. For our application of sensor distribution, it does not show any significant adverse effects on the computation.

3.3. *The algorithm and discussions*

We present a summary of the data-mining-guided evolutionary approach in the following.

Step 1. Set the generation count $t = 0$. Start with a random population $\mathbf{P}_0 = \{\boldsymbol{\theta}_0^1, \boldsymbol{\theta}_0^2, \ldots, \boldsymbol{\theta}_0^M\}$ by uniformly sampling M individuals over $\mathbf{\Omega}$, and evaluate the sensitivity S of each individual. Let $\mathbf{D}_0 = \mathbf{\Omega}$, $\mathbf{V}_0 = \emptyset$ and $\mathbf{\Theta}_0 = \mathbf{P}_0$.

Step 2. Evolution mode: run the GA until a switching condition is met.

2.1. Set $t = t + 1$, $\mathbf{D}_t = \mathbf{D}_{t-1}$ and $\mathbf{V}_t = \mathbf{V}_{t-1}$. Set $m = 1$. Let $\mathbf{P}_t = \emptyset$. Repeat the following actions until $|\mathbf{P}_t| = M$.

Select two individuals from $\mathbf{P}_{t-1}$ according to their fitness, and generate a new offspring $\boldsymbol{\theta}_t^m$ with a crossover rate p_c. If no crossover was performed, the offspring is an exact copy of its parents. Mutate new offspring with a mutation rate p_m. If $\boldsymbol{\theta}_t^m \notin \mathbf{V}_t$, $\mathbf{P}_t = \mathbf{P}_t \cup \{\boldsymbol{\theta}_t^m\}$, set $m = m + 1$.

2.2. Let $\boldsymbol{\Theta}_t = \boldsymbol{\Theta}_{t-1} \cup \mathbf{P}_t$. Evaluate the sensitivity index S for $\mathbf{P}_t$.

Step 3. Run the data-mining mode until a switching condition is met.

3.1. Update $\mathbf{H}_t$ and $\mathbf{L}_t$ from $\boldsymbol{\Theta}_t$. Apply CART to update $\mathbf{D}_t$ and $\mathbf{V}_t$. Uniformly sample $\lceil M/r_t \rceil$ individuals from $\mathbf{D}_{ti}$, $i = 1, \ldots, r_t$, and denote by $\cup_{i=1}^{r_t}\{\boldsymbol{\theta}_{ti}^1, \boldsymbol{\theta}_{ti}^2, \ldots, \boldsymbol{\theta}_{ti}^{\lceil M/r_t \rceil}\}$ the set of new individuals from $\mathbf{D}_t$.

3.2. Set $t = t + 1$, and $\mathbf{P}_t = (\cup_{i=1}^{r_t}\{\boldsymbol{\theta}_{t-1i}^1, \boldsymbol{\theta}_{t-1i}^2, \ldots, \boldsymbol{\theta}_{t-1i}^{\lceil M/r_t \rceil}\}) \cup \mathbf{P}_{t-1}$. Sort the elements in $\mathbf{P}_t$ in descending order according to their sensitivity values and delete the last $r_t \times \lceil M/r_t \rceil$ elements so that the population size is always M (i.e., $|\mathbf{P}_t| = M$). Let $\boldsymbol{\Theta}_t = \boldsymbol{\Theta}_{t-1} \cup \mathbf{P}_t$.

Step 4. Alternate between the two modes until the stopping rule is met. The algorithm could terminate when the computational budget (the number of generations) is consumed or when the change in the best sensitivity value does not exceed a given threshold for several generations.

In order to implement this algorithm, there are a few more details that need one's attention. We discuss them in the following remarks.

Remark 1. We omit most of the details associated with the evolutionary algorithm because we use a standard GA in the proposed optimization procedure. When implementing a GA, we use binary encoding, rank selection, one-point crossover and uniform mutation. Commonly, p_c should be relatively large, (e.g., between 0.6 and 0.9), and p_m should be small, (e.g., between 0.001 and 0.1). The optimal choice of the control parameters (M, p_c, and p_m) for the GA strongly depends on the nature of the objective function $S(\boldsymbol{\theta})$ (Lobo and Goldberg, 2004). In this paper, we fine-tune the control parameters and find that $p_c = 0.9$ and $p_m = 0.01$ provide the best performance in our application.

Remark 2. In order to truly adopt the strengths of the two modes (evolutionary algorithm and data mining) and compensate their weakness, proper switching conditions are needed to guide the proposed optimization procedure to select an appropriate mode of operations. One typically runs the evolution mode for f_o generations, then switches to the data-mining mode and runs it for f_d generations, and then switches back to the evolution mode, as illustrated in Fig. 3. Because of the inability of a stand-alone data-mining mode to find representative individuals (will be shown in Section 4), it is not beneficial to run the data-mining mode for multiple generations. Thus, a natural choice is $f_d = 1$. On the other hand, running the evolutionary algorithm for a large number of generations will make the combined procedure virtually a pure evolutionary algorithm by overshadowing the learning effect from the data-mining mode, and it will certainly slow down the convergence of the whole procedure. Thus, we recommend running the proposed algorithm for f_o values between one and five and selecting the one producing the most desirable outcome.

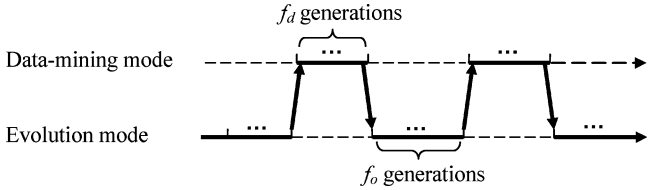

Fig. 3. Switching condition for the data-mining-guided evolutionary algorithm.

Remark 3. The correct choices of h and l could vary for different problems. For a maximization problem, too small an h value could bring many uninteresting individuals into the set of supposedly high-performance solutions which will slow down the optimization process. On the other hand, too large an h value would increase the chance of the algorithm falling into local optima. However, the danger of falling into local optima is not grave because the data-mining mode is followed by a GA that randomizes the population again and makes it possible to escape from local optima. In light of this, we recommend an aggressive choice for the h value, i.e., $h \geq 90\%$.

Selection of the l value requires greater care because the supposedly unpromising regions will be completely excluded in the subsequent optimization process. A large l value might filter out the regions containing desirable solutions. Thus, we recommend a conservative choice for l, namely, $l \leq 20\%$.

Apparently, the current selection of h and l values is still *ad hoc*. It is not difficult to see that the values of h and l should not stay constant but may need to be adjusted adaptively when the whole algorithm is progressing. That is indeed what our on-going research tries to achieve, an adaptive control that sets the correct h and l values in the optimization process.

4. Examples and performance comparisons

To illustrate the potential of the proposed method, we apply it in two different applications: the sensor distribution problem and neural network training. We show the performance of the data-mining-guided method in three different versions: the stand-alone data-mining method (i.e., no

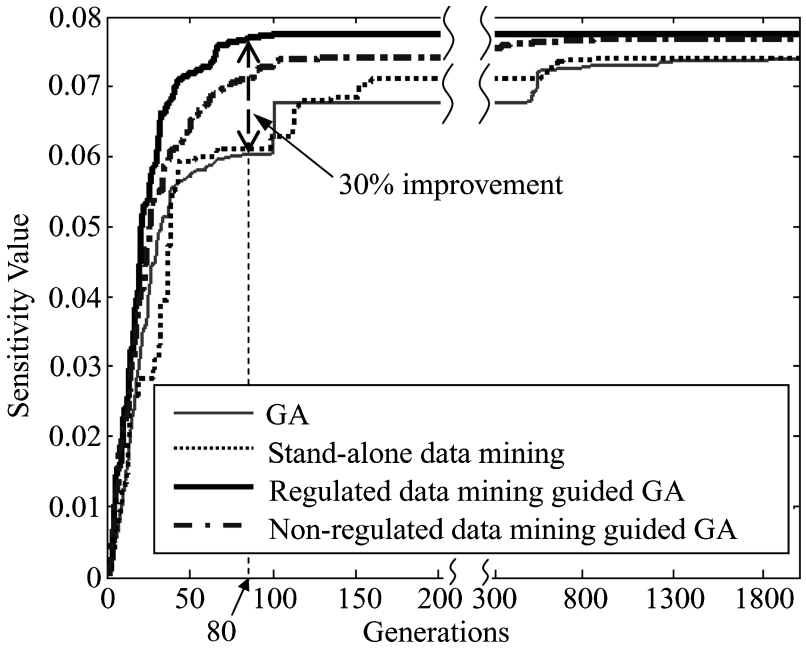

Fig. 4. Algorithm performance when $s = 5$.

evolution mode), the data-mining-guided GA without using regulation from $\mathbf{V}_t$, and the data-mining-guided GA using the regulation from $\mathbf{V}_t$ (i.e., the full version as described in Section 3.3). For the tuning parameters in the proposed algorithm we choose values of $f_o = 2$, $f_d = 1$, $h = 95\%$ and $l = 20\%$. We also compare the results of the proposed algorithm with a standard GA. All optimization algorithms are implemented in the MATLAB environment, and all reported algorithm performances are the averaged results of five trials.

4.1. *Sensor distribution examples*

We use the data-mining guided GA to solve the sensor distribution problem in the three-station 2-D assembly process under two settings: when $s = 5$ and $s = 9$; we will explain why we choose the two sensor numbers at the end of this section. For the scenario where $s = 5$, we set the population size $M = 80$. The result is shown in Fig. 4. The x-axis uses 50 as its increment before 200 and 500 as the increment after 300. This is done in order to clearly present both the early generation transient rate and the steady-state performance. Please also note that the value associated with the x-axis is the number of generations. To translate it to the number of objective function evaluations, one would need to multiply it by the population size $M = 80$. For example, the GA reaches $S = 0.057$ at the 50th generation, which is equivalent to 4000 objective function evaluations.

Apparently, the data-mining-guided evolutionary approach outperforms the GA both in terms of finding a better sensitivity value and the computation time. This is more obvious for the regulated data-mining-guided evolutionary approach, whose performance curve covers almost all the other curves. It demonstrates a key advantage of using the regulation from $\mathbf{V}_t$ in the proposed procedure, that is, it will expedite the algorithm's convergence to the best solution compared to the one that did not use the regulation. As can be seen in Fig. 5, at 80 generations, the regulated data-mining-guided GA reaches its steady-state performance, where it achieves about a 30% improvement over the GA (0.077 versus 0.060). The stand-alone data-mining method performs similarly to a stand-alone GA (both are inferior to

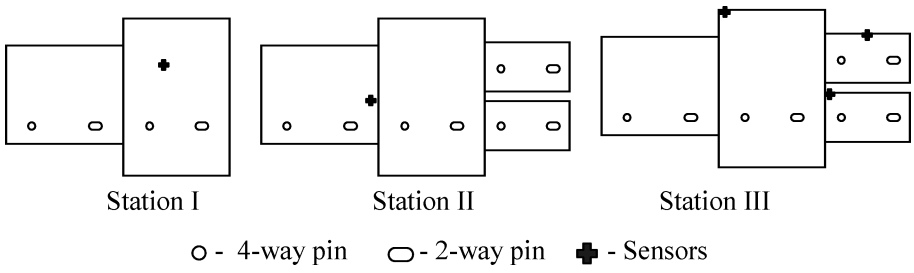

Fig. 5. Sensor distribution strategy when $s = 5$.

Table 1. Generated rules about high-performance solutions

Rules	*Meaning*
After 10 generations	
$1 \le \theta_1 \le 15\,193$	Sensor 1 is put on part 1 or 2 on station I
$13\,868 \le \theta_2 \le 15\,193$	Sensor 2 is put on part 1 on station II
$13\,868 \le \theta_3 \le 40\,617$	Sensor 3 is put on any part on station II or part 1 on station III
$44\,000 \le \theta_4 \le 52\,790$	Sensor 4 is put on part 2, 3 or 4 on station III
$49\,649 \le \theta_5 \le 52\,790$	Sensor 5 is put on part 3 or 4 on station III
$8605 \le \theta_1 \le 8811$	Sensor 1 is put on part 2 on station I
$8605 \le \theta_2 \le 20\,335$	Sensor 2 is put on part 2 on station I or part 1 on station II
After 100 generations	
$39\,893 \le \theta_3 \le 48\,831$	Sensor 3 is put on part 1, 2 or 3 on station III
$48\,783 \le \theta_4 \le 48\,831$	Sensor 4 is put on part 3 on station III
$50\,346 \le \theta_5 \le 50\,351$	Sensor 5 is put on part 4 on station III

the combined procedure), which to some extend verifies our previous arguments on the shortcomings of each individual algorithm. In order for a GA to considerably narrow its difference from the regulated data-mining-guided method, it needs to run for more than 1800 generations, which demands a great deal of computational effort. The sensor distribution strategy with the largest S value is presented in Fig. 5.

Our proposed algorithm, thanks to its data-mining ingredients, also generates several rules describing the high-performance solutions. Table 1 presents the rules that are generated after a certain number of generations. With more generations evolved and more data mined in the proposed procedure, the rules become more and more specific. Figure 6 visualizes the rules after 100 generations by highlighting the areas on a part where a sensor should be placed in order to get a good sensitivity. From Fig. 6, we observe that: (i) more sensors should be put on latter stations; (ii) if multiple sensors are placed on the same station, they are more likely to be distributed on all parts within that station instead of being clustered on individual parts (station III is a good example); and (iii) at least one sensor should be put on each part. These observations are largely consistent with, but not completely the same as, a set of guidelines regarding the strategy of sensor distribution obtained in Section 3.3 of Ding *et al.* (2003), when the same assembly process was studied.

As mentioned before, one may wonder if we could employ the results in Ding *et al.* (2003) to help solve our sensor distribution problem, i.e., use the approach in Ding *et al.* (2003) to obtain a diagnosable sensor system first and then maximize the variance-detecting sensitivity measure among those diagnosable systems. The result from Ding *et al.* (2003) says that four sensors should be installed on station III (and install one sensor on every part) and one sensor should be installed on station I (either on part 1 or part 2). Then, use a single-station sensor placement approach to optimize the sensor location on each part, the best sensitivity of the whole assembly process we could find is 0.049, which is only about 60% of the best sensitivity value shown in Fig. 4. This is in part due to a difference between the guidelines found from our data-mining-guided approach and the results in Ding *et al.* (2003): as can been seen from Figs. 5 and 6, the best sensor distribution strategy installs sensors on every station but the strategy in Ding *et al.* (2003) chose to skip the second station because installing a sensor on it does not make a difference in their diagnosability criterion.

In Fig. 4, the best sensitivity value is 0.078. It indicates that, for some sensor distribution strategies, although the full diagnosability is guaranteed ($S > 0$ implies full diagnosability), only 7.8% of variation changes in the variation sources is reflected in the sensor measurements. Thus, for this kind of sensor distribution strategy, practically it is

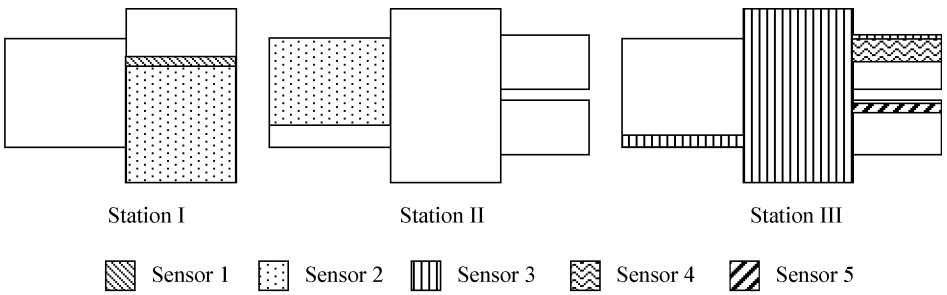

Fig. 6. Generated rules about high-performance solutions.

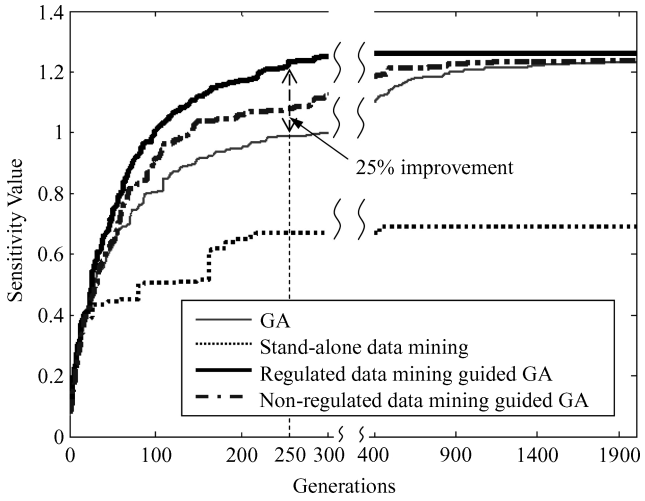

Fig. 7. Algorithm performance when $s = 9$.

not very useful to detect the small variation changes in the variation sources. Consequently, more sensors are needed to increase the variance-detecting capability of the sensor system. Thus, we study the scenario with $s = 9$ sensors.

For the scenario of $s = 9$, we set the population size $M = 100$. The performance results of various methods are presented in Fig. 7, where the x-axis uses 100 as its increment before 300 and 500 as the increment after 400. Figure 7 also demonstrates a clear advantage of the data-mining-guided methods over the stand-alone GA or the stand-alone data-mining and places the regulated version as the clear winner. The regulated data-mining-guided method comes close to its steady-state value at 250 generations, where it does 25% better than the GA (1.219 versus 0.988). For a GA to catch up with the performance of a regulated data-mining method, it needs to run more than 2000 generations, almost ten times longer. It is also worth noting that the stand-alone data-mining method performs much worse than other algorithms. We believe this happens mainly because it is more difficult to gather representative data in a higher-dimensional solution space. Figure 8 presents the best (i.e., the largest S) sensor distribution strategy found in this example.

From both Fig. 4 and Fig. 7, it is not difficult to notice that the performance curves of all four algorithms are very similar in early generations. However, the data-mining-guided methods take the lead after a small number of generations (it is about 40 generations in Fig. 7). What this tells us is the following. During the early generations when not enough individuals are generated by the GA, a data-mining method may not get "sufficient" knowledge to guide the evolution search. When enough representative individuals are collected after a number of generations, the knowledge learned by a data-mining method starts to lead the evolution process in a better direction and thus boosts the performance of the proposed algorithm.

So far, we have used the data-mining-guided evolutionary algorithm to solve the optimal sensor distribution problem when the sensor number is specified. One may wonder how the sensor number s is determined. According to Liu *et al.* (2005), the sensitivity S is defined as the ratio of the change in the variance of sensor measurements over a perturbation of the variance of the input variation sources. Thus, $S = 1$ implies that the measurement sensitivity of the input variation sources is as good as if they were measured directly by a set of coordinate sensors. It is preferable for S to be larger than or equal to one so that the detection capability of the input variation sources will not be compromised when the sources cannot be directly measured. Thus, we set a lower bound c for S and $c \geq 1$. One way to determine s, as explained in Liu *et al.* (2005), is to start from a small s and obtain the corresponding optimal sensor distribution. Then, add one more sensor sequentially, which maximizes the resulting sensitivity value, until $S \geq c$. However, this routine may miss the optimal sensor distribution, because the sensor distribution with s sensors may not be a subset of the one with $(s + 1)$ sensors. Nevertheless, we can find a sensor number which yields $S \geq c$ by this sequential routine and then use the data-mining-guided evolutionary approach to find the optimal sensor distribution for a given sensor number. In practice, it works quite effectively and often offers as good a solution as exhausting the optimization for every sensor number. When applying to the three-station assembly process by setting $c = 1$, it turns out that the smallest number of sensors yielding $S \geq 1$ is nine; that is one of the scenarios we tested above. The other choice of sensor number $s = 5$ is the minimum number of sensors that can make the three-station process fully diagnosable, as shown in Ding *et al.* (2003). In other words, if $s < 5$, the

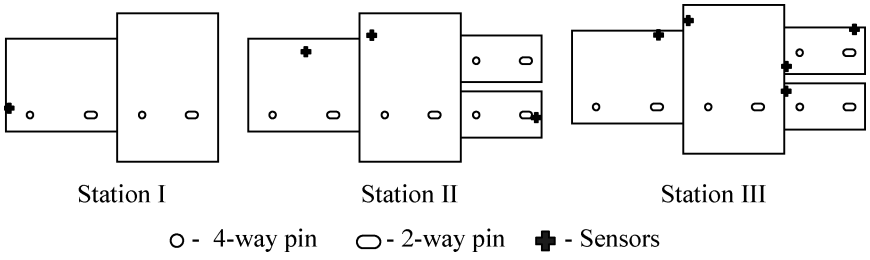

Fig. 8. Sensor distribution strategy when $s = 9$.

S value will always remain zero no matter how the sensors are distributed.

Finally, please note that the integrated approach will see its limit when the scale of a manufacturing process gets very large. Under that circumstance, using the two-step procedure is warranted. For a large-scale system, one may first partition the whole system into smaller segments. Subsequently, use the method proposed in Ding *et al.* (2003) to find out how many sensors are needed for each segment and then use our proposed algorithm to find out the optimal layout within individual segments.

4.2. *Neural network training example*

To show the potential of our proposed method in other applications, we use it to train a neural network for classification problems. Multiple-layer Perceptrons (MLPs) (Minksy and Papert, 1969) have become popular as an effective data-mining method. Given a group of connection weights $\mathbf{z} = (\alpha, \beta, \gamma)$, the MLP predictor is written as

$$\hat{f}(\mathbf{x}_k|\mathbf{z}) = \varphi_0\left(\alpha_0 + \sum_{j=1}^{p}\gamma_j x_{jk} + \sum_{i=1}^{O}\alpha_i\varphi_h\left(\beta_{i0} + \sum_{j=1}^{p}\beta_{ij}x_{kj}\right)\right), \tag{16}$$

where p is the number of inputs, $\mathbf{x}_k = (x_{k1}, \ldots, x_{kp})$ is the kth input pattern, O is the number of hidden units in the MLP, α_i, β_{ij}, γ_j are the weights on the connections from the ith hidden unit to the output, from the jth input to the ith hidden unit, and from the jth input to the output, respectively. The functions $\varphi_0(\cdot)$ and $\varphi_h(\cdot)$ are called activation functions. A common choice for $\varphi_h(\cdot)$ is the sigmoid function. For regression problems, $\varphi_0(\cdot)$ is usually set to be $\varphi_0(x) = x$; and for classification problems, $\varphi_0(\cdot)$ is usually set to the sigmoid function. Training a MLP is to minimize the error function

$$U(\mathbf{z}) = \sum_{k=1}^{R}\left(\hat{f}(\mathbf{x}_k|\mathbf{z}) - y_k\right)^2 + \lambda\left(\sum_{i=0}^{O}\alpha_i^2 + \sum_{i=1}^{O}\sum_{j=0}^{p}\beta_{ij}^2 + \sum_{j=1}^{p}\gamma_j^2\right), \tag{17}$$

with respect to the connection weights α_i, β_{ij}, γ_j. Here y_k is the observed output associated with $\mathbf{x}_k$, and R is the number of input patterns. The second term is the regulation term which avoids overfitting problems. Minimizing the function (17) is generally difficult (Hastie *et al.*, 2001), because one has to escape many local optima to reach the global minimum.

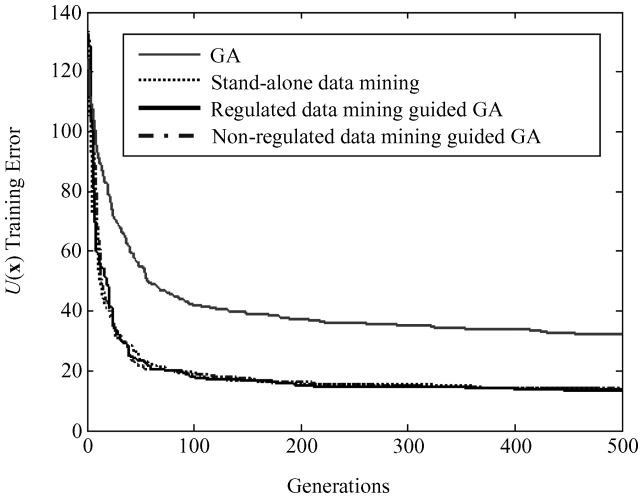

Fig. 9. Algorithm performance for neural network training.

In this paper we train a MLP for the breast cancer classification problem (Mangasarian, 1993). The breast cancer dataset is collected at the University of Wisconsin Hospitals and is available at *http://www.ics.uci.edu/~mlearn/MLRepository.html*. There are 683 samples of nuclear features of fine needle aspirates from patients' breasts. Each sample consists of nine components, with each component having values from one to ten. The samples are classified into two classes, benign and malignant. The benchmark classification error in the literature is 2% (Mangasarian, 1993).

The breast cancer data is modeled by a MLP with three hidden units, and there are totally 43 connections weights. Each weight is restricted to the interval [−10, 10]. The first 342 samples are used as training data and the remaining 341 samples are test data. We applied the proposed methods to minimize Equation (17) on the training data. The population size is set to $M = 100$. We run each algorithm for 500 iterations. The algorithm performances are shown in Fig. 9. One can notice that the performances of the regulated data-mining-guided GA, the non-regulated data-mining-guided GA and the stand-alone data-mining method are comparable, and they all outperform the GA. Numerical results are presented in Table 2. The regulated version of our proposed algorithm

Table 2. Algorithm performance comparison

	Training error		*Test error*	
Algorithms	*Average U(x)*	*Classification error (%)*	*Average U(x)*	*Classification error (%)*
Regulated data-mining-guided GA	13.36	3.4	6.73	1.7
Non-regulated data-mining-guided GA	14.16	3.3	8.28	1.9
Stand-alone data mining	13.68	3.6	8.34	2.1
GA	32.00	5.2	33.82	6.3

outperforms others both in terms of training error and test error. Both data-mining-guided methods perform better than the 2% test error reported in Mangasarian (1993).

5. Concluding remarks

This paper investigates the problem of maximizing the variance-detecting capability of a distributed sensor system in a multi-station assembly process. The sensitivity index to be optimized is the smallest eigenvalue of the associated Fisher information matrix and optimizing it can provide a guarantee of a sensor system's capability to detect the underlying process variance changes. We devised a data-mining-guided approach to solve the optimization problem, which outperforms some popular alternatives including the GA approach.

The data-mining-guided evolutionary approach presented in this paper is essentially a heuristic method for optimization. Fundamentally, a data-mining method is also an optimization routine. Thus, a data-mining-guided approach is equivalent to breaking an originally complex optimization into a set of simpler problems that may be solved by the optimization routines embedded in the data-mining methods. The method is realized in this paper to solve a sensor distribution problem but should be extendable to other types of applications as well.

In the proposed method, because CART is used as the data-mining method, the design space is sliced into rectangles. There could be situations where a rectangular partition may not be optimal, for example, when the original design space is of more complex, non-regular shape. However, finding a good replacement for CART may not be straightforward because the repeated use of the data-mining method demands that any viable candidate must be time-efficient and simple. Some other data-mining methods, such as artificial neural networks, are computationally expensive themselves, and are therefore less likely to be a good candidate for the proposed combination.

References

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and Regression Trees*, Chapman & Hall, New York, NY.

Camelio, J., Hu, S.J. and Ceglarek, D. (2003) Modeling variation propagation of multi-station assembly systems with compliant parts. *Journal of Mechanical Design*, **125**, 673–681.

Camelio, J., Hu, S.J. and Yim, H. (2005) Sensor placement for effective diagnosis of multiple faults in fixturing of compliant parts. *ASME Journal of Manufacturing Science and Engineering*, **127**, 68–74.

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 autobody assembly using pattern recognition. *Transactions of the ASME, Journal of Engineering for Industry*, **188**, 55–65.

Ding, Y., Ceglarek, D. and Shi, J. (2000) Modeling and diagnosis of multistage manufacturing processes: part I – state space model. Presented at the 2000 Japan/USA Symposium on Flexible Automation, July 23–26, Ann Arbor, MI.

Ding, Y., Shi, J. and Ceglarek, D. (2002) Diagnosability analysis of multi-station manufacturing processes. *ASME Transactions, Journal of Dynamic Systems, Measurement, and Control*, **124**, 1–13.

Ding Y., Kim, P., Ceglarek, D. and Jin, J. (2003) Optimal sensor distribution for variation diagnosis in multistation assembly processes. *IEEE Transactions on Robotics and Automation*, **19**, 543–556.

Ding, Y., Elsayed, E.A., Kumara, S., Lu, J.C., Niu, F. and Shi, J. (2006) Distributed sensing for quality and productivity improvements. *IEEE Transactions on Automation Science and Engineering*, **3**, 344–359.

Djurdjanovic, D. and Ni, J. (2003) Dimensional errors of fixtures, locating and measurement datum features in the stream of variation modeling in machining. *Journal of Manufacturing Science and Engineering*, **125**, 716–730.

Djurdjanovic, D. and Ni, J. (2004) Measurement scheme synthesis in multi-station machining systems. *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, **126**, 178–188.

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

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

Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, Cambridge, MA.

Huang, Q., Shi, J. and Yuan, J. (2003) Part dimensional error and its propagation modeling in multi-operational machining processes. *Journal of Manufacturing Science and Engineering*, **125**, 255–262.

Huyet, A.L. (2006) Optimization and analysis aid via data-mining for simulated production systems. *European Journal of Operational Research*, **173**, 827–838.

Igusa, T., Liu, H., Schafer, B. and Naiman, D.Q. (2003) Bayesian classification trees and clustering for rapid generation and selection of design alternatives. Presented at the 2003 NSF Design, Service and Manufacturing Grantees and Research Conference, Birmingham, AL, January 4–9.

Jin, J. and Shi, J. (1999) State space modeling of sheet metal assembly for dimensional control. *Journal of Manufacturing Science and Engineering*, **121**, 756–762.

Khan, A. and Ceglarek, D. (2000) Sensor optimization for fault diagnosis in multi-fixture assembly systems with distributed sensing. *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, **122**, 215–226.

Khan, A., Ceglarek, D. and Ni, J. (1998) Sensor location optimization for fault diagnosis in multi-fixture assembly systems. *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, **120**, 781–792.

Khan, A., Ceglarek, D., Shi, J., Ni, J. and Woo, T.C. (1999) Sensor optimization for fault diagnosis in single fixture systems: a methodology. *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, **121**, 109–121.

Kim, P. and Ding, Y. (2005) Optimal engineering system design guided by data-mining methods. *Technometrics*, **47**, 336–348.

Liu, C., Ding, Y. and Chen, Y. (2005) Optimal coordinate sensor placements for estimating mean and variance components of variation sources. *IIE Transcations*, **37**, 877–889.

Lobo, F.G. and Goldberg, D.E. (2004) The parameter-less genetic algorithm in practice. *Information Sciences*, **167**, 217–232.

Mandal, A., Wu, C.F.J. and Johnson, K. (2006) SELC: Sequential elimination of level combinations by means of modified genetic algorithms. *Technometrics*, **48**, 273–283.

Mandroli, S.S., Shrivastava, A.K. and Ding, Y. (2006) A survey of inspection strategy and sensor distribution studies in discrete-part manufacturing processes. *IIE Transactions*, **38**, 309–328.

Mangasarian, O.L. (1993) Mathematical programming in neural networks. *ORSA Journal on Computing*, **5**, 349–360.

Michalski, R.S. (2000) Learnable evolution model: evolutionary processes guided by machine learning. *Machine Learning*, **38**, 9–40.

Minsky, M. and Papert, S. (1969) *Perceptrons: An Introduction to Computational Geometry*, MIT Press, Cambridge, MA.
Pukelsheim, F. (1993) *Optimal Design of Experiments*. Wiley, New York, NY.
Santner, T.J., Williams, B.J. and Notz, W.I. (2003) *Design & Analysis of Computer Experiments*. Springer-Verlag, New York, NY.
Schott, J.R. (1997) *Matrix Analysis for Statistics*. Wiley, New York, NY.
Schwabacher, M., Ellman, T. and Hirsh, H. (2001) Learning to set up numerical optimizations of engineering designs, in *Data-mining for Design and Manufacturing*, Braha, D (ed). Kluwer, Boston, MA, pp. 87–125.
Wang, Y. and Nagarkar, S.R. (1999) Locator and sensor placement for automated coordinate checking fixtures, *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, **121**, 709–719.
Zhou, S., Huang, Q. and Shi, J. (2003) State space modeling of dimensional variation propagation in multistage machining process using differential motion vectors, *IEEE Transactions on Robotics and Automation*, **19**, 296–309.

Biographies

Yuan Ren received his B.E. degree in Automation from Tsinghua University, China in 2003. He is currently a Ph.D. candidate in the Department of Industrial and Systems Engineering at Texas A&M University, College Station, TX. His research interests are in the area of quality engineering, applied statistics and applied optimization, including analysis and design of distributed sensor systems for quality improvement, data mining methods and Markov chain Monte Carlo. He is a student member of IIE and INFORMS.

Yu Ding received a B.S. degree in Precision Engineering from the University of Science and Technology of China in 1993, M.S. in Precision Instruments from Tsinghua University, China in 1996, M.S. in Mechanical Engineering from the Pennsylvania State University in 1998 and a Ph.D. in Mechanical Engineering from the University of Michigan in 2001. He is currently an Associate Professor in the Department of Industrial and Systems Engineering at Texas A&M University. His research interests are in the area of quality engineering and applied statistics. His current research is sponsored by the National Science Foundation, the Department of Homeland Security, the State of Texas and industry. He has received a number of awards for his work, including an *IIE Transactions* Best Paper Award in 2006, a CAREER Award from the National Science Foundation in 2004 and a Best Paper Award from the ASME Manufacturing Engineering Division in 2000. He currently serves as a Department Editor of *IIE Transactions* and an Associate Editor of *IEEE Transactions on Automation Science and Engineering*. He is a member of IIE, INFORMS, IEEE and ASME.