# INDUCTIVE AND ANALOGICAL LEARNING: DATA-DRIVEN IMPROVEMENT OF PROCESS OPERATIONS[1]

**Pedro M. Saraiva**

**Department of Chemical Engineering**
**University of Coimbra**
**3000 Coimbra, Portugal**



[1] The work reported in this chapter was performed while the author was on leave as a Ph.D. student in the Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139.

Informed and systematic observation of data, naturally generated, can lead to the formulation of interesting and effective generalizations. Whereas some statisticians believe that experimentation is the only safe and reliable way to "learn" and achieve operational improvements in a manufacturing system, other statisticians and all the empirical machine learning researchers contend that by looking at past records and sets of examples, it is possible to extract and generate important new knowledge. This chapter draws from *inductive and analogical learning* ideas in an effort to develop systematic methodologies for the extraction of structured new knowledge from operational data of manufacturing systems. These methodologies do not require any a priori decisions and assumptions either on the character of the operating data (e.g., probability density distributions) or the behavior of the manufacturing operations (e.g., linear or nonlinear structured quantitative models), and make use of *instance-based learning* and *inductive symbolic learning* techniques, developed by artificial intelligence (AI). They are aimed to be complementary to the usual set of statistical tools that have been employed to solve analogous problems. Thus, one can see the material of this chapter as an attempt to fuse statistics and machine learning in solving specific engineering problems. The framework developed in this chapter is quite generic and, as the subsequent sections illustrate, it can be used to generate operational improvement opportunities for manufacturing systems (1) that are simple or complex (with internal structure), (2) whose performance is characterized by one or multiple objectives, and (3) whose performance metrics are categorical (qualitative) or continuous (real numbers). A series of industrial case studies illustrates the learning ideas and methodologies.

## I. Introduction

With rapid advances in hardware, database management and information processing systems, efficient and competitive manufacturing has become an information-intensive activity. The amounts of data presently collected in the field on a routine basis are staggering, and it is not unusual to find plants where as many as 20,000 variables are continuously monitored and stored (Taylor, 1989).

It has also been recognized that the ability of manufacturing companies to become *learning* organizations (Senge, 1990; Shiba *et al.*, 1993; Hayes *et al.*, 1988), achieving high rates of continuous and never-ending improve-

ments (with a special focus on *total manufacturing quality*), is a critical factor determining the survival and growth of the company in an increasingly competitive global economy (Deming, 1986; Juran, 1964).

As a result, there has been a significant and increasing awareness and interest on how the accumulation of data can lead to a better management of operational quality, which involves two complementary steps (Saraiva and Stephanopoulos, 1992a): (a) *control within prespecified limits* and (b) *continuous improvement of process performance*. The aim of the first is to detect "abnormal" situations, identify and eliminate the *special causes* that produced them. But, bringing the process under bounded, statistical control is not enough. The final level of variability thus achieved is the result of *common and sustained causes*, present within the process itself, and must not be considered as unavoidable. Therefore, the second step dictates that one should move from process control to process improvement, i.e., continuously search for common causes, ways of reducing their impact and challenge the current levels of performance (Juran, 1964).

The bulk chemical commodity producing companies (e.g., refineries, petrochemicals) have been practicing this philosophy for some time, using dynamic models to contain operational variability through feedback controllers, and employing static models to determine the optimal levels of operating conditions (Lasdon and Baker, 1986; Garcia and Prett, 1986).

On the other hand, plants involving solids processing (e.g., pulp and paper) with poorly understood physicochemical phenomena, not lending themselves to description through first-principles models, have not been using effectively the mountains of accumulated operational data. This lack of exploration and analysis of data is particularly severe for records of data collected while the processes were kept under "normal" operating contexts. Thus, given that less than 5% of the points contained in a conventional $6\sigma$ control chart are likely to attract any attention, the bulk of the acquired operating records is simply "stored," i.e., wasted. For such manufacturing plants there is a strong need and incentives to develop theoretical frameworks and practical tools that are able to extract useful and operational knowledge from existing records of data, leading to continuous improvement of process operational performance. This chapter represents a specific effort that attempts to fulfill that need.

Several statistical, quality management, and optimization data analysis tools, aimed at exploring records of measurements and uncover useful information from them, have been available for some time. However, all of them require from the user a significant number of assumptions and a priori decisions, which determine in a very strict manner the validity of the final results obtained. Furthermore, these classical tools are guided

essentially by an academic attitude that emphasizes numerical accuracy over the extraction of substantive knowledge, whereas the formats chosen to express the solutions are often difficult to be interpreted and understood by human operators or to be implemented by them in manufacturing operations.

Rather than trying to replace any of the above traditional techniques, this chapter presents the development of complementary frameworks and methodologies, supported by symbolic empirical machine learning algorithms (Kodratoff and Michalski, 1990; Shavlik and Dietterich, 1990; Shapiro and Frawley, 1991). These ideas from machine learning try to overcome some of the weaknesses of the traditional techniques in terms of both (1) the number and type of a priori decisions and assumptions that they require and (2) the knowledge representation formats they choose to express final solutions.

One can thus state the primary goal of the approaches summarized in this chapter as follows:

> *To develop and apply assumption-free learning frameworks and methodologies, aimed at uncovering and expressing in adequate solution formats performance improvement opportunities, extracted from existing data which were acquired from plants that cannot be described effectively through first-principles quantitative models.* (1)

The remaining material of this chapter has the following structure: Section II provides a more specific definition of the problems that are addressed, and it identifies the scope of applications and the type of manufacturing systems that the presented methodologies aim to cover. Then, Section III introduces a generic framework for the development and description of the learning algorithms that will allow us to present the several methodologies on a common basis. In the same section we will also enumerate the most critical characteristics and features that differentiate the approaches of this chapter from previous ones. Sections IV and V illustrate the problem statements and solution methodologies employed when the manufacturing system's performance is measured by a *categorical* (qualitative) or *continuous* (real-valued) variable. Section VI extends the ideas presented in Sections IV and V to manufacturing systems whose performance is characterized by multiple, possibly noncommensurable objectives, and Section VII makes extensions to complex manufacturing systems with internal structure. Concrete applications are presented at the end of these sections. Some final conclusions together with a critical summary of the main results are presented in Section VIII.

## II. General Problem Statement and Scope of the Learning Task

In its most general form, the problem that we address in this chapter can be stated as follows:

***Given*** a set of existing $(\mathbf{x}, y)$ data records,
where $\mathbf{x}$ is a vector of operating or decision variables, which are believed to influence the values taken on by $y$;
$y$ is a performance metric, usually assumed to be a quality characteristic of the product or process under analysis;
***Learn*** how to improve the system performance above its current levels (2)

The work described in this chapter revisits this old problem by adopting a new perspective, exploring alternative formats for the presentation of solutions and problem-solving strategies. Such alternative problem-solving strategies are particularly useful in addressing systems where the weaknesses of traditional approaches become particularly severe. It is thus important to clarify the scope of application and the type of situations within the broad area covered by the problem statement (2) that the methodologies of this chapter aim to cover. That's what we will try to do in the following paragraphs.

1. *Supervisory control layer of decisionmaking*. Although learning capabilities and data analysis activities should be present across all levels of decisionmaking (Fig. 1), countless studies and authors (National Research Council, 1991; Latour, 1976, 1979; Launks *et al.*, 1992; Klein, 1990; Sargent, 1984; Ellingsen, 1976; Moore, 1990) have shown that the greatest

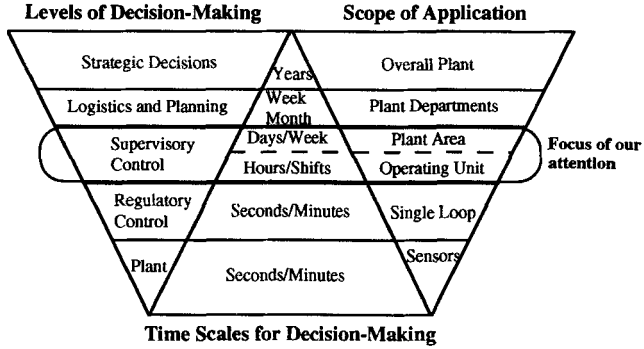


FIG. 1. Levels, time scales, and application scopes of decisionmaking activities.

opportunities and benefits are associated with improvements at the steady-state optimization or supervisory control layer. Shinnar (1986) makes clear this point by stating that in the chemical industry 80% of all practical control problems and 90% of the financial gains must be found and addressed at this level. Accordingly, we will be performing a steady-state analysis of data that has been collected and defined according to the supervisory control layer of decisionmaking.

2. *Plants lacking credible first-principles models.* The learning methodologies of this chapter require very few assumptions and decisions to be made a priori by the user, and rely on an exploratory analysis of data (Tukey, 1977) to uncover operational knowledge. Thus, they are particularly useful to analyze systems poorly understood and for which no first-principles models with acceptable accuracy are available. In such plants supervisory control decisions are quite often taken by human operators, rather than by computers. Reflecting on this fact, across the scale of automation in the spectrum of human–machine interaction (Sheridan, 1985) the approaches of this chapter fall within the scope of decision support systems (Turban, 1988); i.e., they do not intend to replace human intervention, but instead to interact with the users, and provide them as much useful information as possible. Yet, it is still the human operator who is responsible for the choice of a particular final solution, among a number of promising alternatives presented to him or her, and for selecting the particular course of action to follow. That being the case, it is critical for the knowledge extracted from the data to be represented in operational, explicit, and easy-to-understand formats, so that better use of that knowledge can be made by the human operators, who are not necessarily experts in pattern recognition, optimization, or statistics. Thus, the learning methodologies must reflect these concerns. Indeed, a strong emphasis is placed on the language used by the techniques of this chapter to express final solutions.

3. *Considerable amounts of data available on routine basis.* In learning activities, there is a fundamental tradeoff between the a priori existing knowledge and data intensity (Gaines, 1991). The more one already knows, the less data are needed to reach a given conclusion. As was said earlier, we are especially interested in studying complex systems for which no accurate quantitative models of behavior are available. Therefore, the learning methodologies must not rely or depend on the validity of any strong assumptions made about the system under study. A price has to be paid, as a result of this strategic choice; to reach the same level of accuracy and resolution in the final solutions found, more data are needed than if stronger assumptions could be made a priori and they happened to

be valid ones. This data-intensive nature of the approaches is not however likely to be problematic, since today in most manufacturing organizations there is no real shortage of data, but rather large amounts of unexplored measurement records. Furthermore, the case studies that were conducted show that even with moderate amounts of data (always less than 1000 records), it is possible to find solutions that result in quite significant improvements over the current levels of performance.

4. *Flexibility in problem definition.* Within the scope of application problems, outlined in the previous paragraphs, the learning methodologies are quite flexible and can handle a number of different situations and problem formulations. Both types of systems, where the performance metric used to evaluate performance, $y$, is continuous and categorical (i.e., can assume only one among a discrete number of possible values, such as "good," "bad," and "excellent"), can be analyzed. In systems of nontrivial size several performance objectives must often be taken into account. Extensions of the basic learning methodologies to handle such multiobjective problem formulations have been developed and tested. Finally, besides simple systems without internal structure, assumed to be isolated from the remaining world and self-sufficient for decisionmaking purposes, one has frequently to consider more complex systems, such as a complete plant, composed of a number of interconnected subsystems. A learning architecture for these complex systems was conceived and applied to a specific case study. It is based on the same procedures applied for simple systems, on top of which we have added articulation, coordination and propagation procedures. Thus, the learning methodologies of this chapter cover a broad range of possible industrial applications, and are able to handle simple or complex systems, with single or multiple objectives, and categorical or continuous performance evaluation variables.

5. *Prototypical application examples.* To provide a more concrete notion of the type of systems where our approaches are expected to be particularly helpful and useful, we conclude this section with a sample of prototypical examples of what the performance metric, $y$, in the problem statement (2) may represent, together with a definition of the corresponding systems:

*Example 1*: kappa ($\kappa$) index of the unbleached pulp produced by a Kraft digester

*Example 2*: activity, yield, and total production cost of proteins from fermentation units

*Example 3*: composition, molecular-weight distribution, and structure of complex copolymers

*Example 4*: selectivity and uniformity of etching rates at a chemical vapor deposition unit that produces silicon wafers

*Example 5*: analysis of the operation of an activated sludge wastewater treatment unit or of an entire pulp plant

As can be inferred from this sample of examples, industrial sectors where the learning methodologies of this chapter look particularly promising are pulp and paper, biotechnology, cement and ceramics, microelectronics, and discrete-parts manufacturing, i.e., complex plants or pieces of equipment that include solids processing, whose behavior is poorly understood and for which no accurate first-principles models exist. It should not be construed, though, that the methodologies of this chapter cannot also be applied to the analysis of other well-understood systems. However, as we move to such well-known systems, for which reliable quantitative models are already available, the suggested problem-solving approaches gradually loose their competitive advantages over traditional ones.

## III. A Generic Framework to Describe Learning Procedures

In this section we present a formal framework to describe learning procedures that is sufficiently rich and general to allow us to concisely express (1) all the previous approaches that have been proposed for the formulation and solution of the problems stated in statement (2), and (2) all the different alternative definitions and methodologies that will be discussed in later sections of the chapter. This description language provides a common basis to introduce our different application contexts (categorical or continuous $y$, single or multiple objectives, simple or complex systems) and corresponding algorithms. It also allows us to make a clear statement of their most important departures from previous conventional approaches, thus facilitating a comparative analysis.

Subsequently, we use the common description framework in order to identify and analyze the following three most important distinguishing features that differentiate the learning frameworks of this chapter from traditional optimization and statistical techniques:

(a) Formats chosen to express the solutions
(b) Criteria used to evaluate the solutions
(c) Procedures used to estimate the solutions' evaluation criteria

## A. A Generic Formalism

Any learning procedure, aimed to address and solve problems given by the problem statement (2) at the supervisory control level of decisionmaking, can be expressed by the following quartuple:

$$L = (\xi, \psi, f, S), \tag{3}$$

where

(a) $\xi \in \Xi$ represents a generic solution, $\xi$, defined in the solution space, $\Xi$.

(b) $\psi \in \Psi$ is the performance criterion, $\psi$, defined in the performance space, $\Psi$, that one chooses to evaluate the merit of a generic solution $\xi$.

(c) $f$ is the model or procedure that maps the solution into the performance space, i.e., it allows one to compute an estimate of the performance criterion, $\psi^{\text{est}}$, for any potential solution $\xi$, $\psi^{\text{est}} = f(\xi)$.

(d) $S$ is a search procedure that explores $f$ in order to identify specific solutions, $\xi^* \in \Xi$, that look particularly promising according to their estimated performances, $f(\xi^*)$. This final fourth element is optional and it is absent from conventional approaches whose goal is strict estimation (prediction of $\psi$ for a given particular choice of $\xi$).

## B. Major Departures from Previous Approaches

An examination of previous classical learning procedures reveals that they differ from each other only with respect to the choices of $\psi$, $f$, and $S$. All of them share the same basic format for $\xi$ and the corresponding solution space, $\Xi$. Let's assume that each $(\mathbf{x}, y)$ pair in the problem statement (2) contains a total of $M$ decision variables:

$$\mathbf{x} \equiv [x_1, \ldots, x_m, \ldots, x_M]^T \in \Xi_{\text{decision}}, \tag{4}$$

where $\Xi_{\text{decision}}$ stands for the decision space, composed of all feasible $\mathbf{x}$ vectors, a subspace of $\mathscr{R}^M$.

Traditional approaches adopt a solution space $\Xi$ that coincides with $\Xi_{\text{decision}}$, and thus any final solution ($\xi^*$ or $\mathbf{x}^*$) has the same format as $\mathbf{x}$, consisting of a real vector that defines a single point in the decision space.

By considering such a language to express solutions, one ignores the fact that in the type of problems we want to study decision variables behave as random variables, and there is always some variability associated with them. No matter how good control systems happen to be, in

reality we will always have to live with ranges of values for the decision variables (concentrations, pressures, flows, etc.), eventually bounded within a narrow, but not null, operation window. As a consequence of not taking into account that one has to operate within a given zone of the decision space, rather than at a single point, the final solutions found by conventional approaches may be suboptimal even when perfect $f(\mathbf{x})$ models are available, since their evaluation criterion, $\psi$, reflects only the performance achieved at a particular point in the decision space, and completely ignores the system behavior around that point. The zone of the decision space that surrounds the $\mathbf{x}^*$ solution with the best $\psi(\mathbf{x}^*)$ does not correspond in general to the zone of the decision space, $\mathbf{Z}^*$, where best average performance, $\psi(\mathbf{Z}^*)$, can be achieved.

### *1. Hyperrectangles as a Convenient Solution Format*

As a result of the above observations, once one chooses to adopt solution formats where each $\xi$ represents a region rather than a point in the decision space, it has yet to be decided what type of zones and shapes should be considered. Since our goal is not to achieve full automation, but to provide support to human operators, it is critical for the new language, used to express solutions, to be understood by them and lead to results that are easy to implement. In that regard, after observing how people tend to articulate their reasoning activities in the control rooms we concluded that they essentially follow an "orthogonal thinking" paradigm: *human operators express themselves by means of conjunctions of statements about individual decision variables* (e.g., the concentration is high and the pressure low), $x_m$, *and not through some more or less intricate linear or nonlinear combination of them* (as is the case with multivariate statistical techniques such as principal-components analysis, partial least squares, factor analysis, or neural networks). Combining the need to identify zones in the decision space, rather than points, with the need to preserve the individuality of each decision variable, rather than losing it in linear or nonlinear combinations, hyperrectangles (conjunctions of ranges of $x_m$ values) in the decision space appear as a very convenient and the natural format to express solutions.

Thus, a critical departure from previous approaches, common to all our learning methodologies, is the adoption of a solution format that consists of hyperrectangles (not points) defined in the decision space.

### *2. Interval Analysis Nomenclature*

Thus, interval analysis (Moore, 1979; Alefeld and Herzberger, 1983) provides the adequate support and notation formalism to express solu-

tions. To distinguish real numbers from intervals, we will use capital letters for intervals. Also, bold typing is employed to represent both real variable and real interval vectors. A real interval $X$ is a subset of $\mathscr{R}$ of the form

$$X \in \mathbf{I} \equiv \{x \in \mathscr{R} \mid i(X) \leq x \leq s(X)\}, \tag{5}$$

where

(a) $\mathbf{I}$ is the space of all closed real intervals.
(b) $i(X)$ is the lower bound of $X$.
(c) $s(X)$ is the upper bound of $X$.
(d) $w(X) = s(X) - i(X)$ is the width of $X$.
(e) $m(X) = [i(X) + s(X)]/2$ is the midpoint of $X$.

Extending the notation to hyperrectangles in $\mathscr{R}^M$, an $M$-dimensional interval vector, $\mathbf{X}$, has as its components real intervals, $X_m$, defined by ranges of $x_m$:

$$\mathbf{X} \equiv [X_1, \ldots, X_m, \ldots, X_M]^T \in \mathbf{I}^M;$$

$$\mathbf{m}(\mathbf{X}) = [m(X_1), \ldots, m(X_m), \ldots, m(X_M)]^T \in \mathscr{R}^M. \tag{6}$$

Since a real vector is a degenerate interval vector whose components are null width intervals, previous conventional pointwise solution formats can be considered a particular case of the suggested alternative and more general solution space, obtained when the minimum allowed region size is reduced to zero, thus converting hyperrectangles into single points.

### 3. *Major Differences*

The notation introduced above allows us to make now a more explicit and condensed enumeration of the major characteristics and differences, with respect to the $(\xi, \psi, f, S)$ key components, that separate our learning methodologies from other approaches.

*Solution format*, $\xi$. The solution space consists of hyperrectangles ($\xi = \mathbf{X} \in \mathbf{I}^M$), instead of points ($\xi = \mathbf{x} \in \mathscr{R}^M$), defined in the decision space.

*Performance criterion*, $\psi$. The quality of any potential solution, $\mathbf{X}$, is determined by the average system performance achieved within the zone of the decision space identified by $\mathbf{X}$, $\psi(\mathbf{X})$, not by the individual performance obtained at any particular point $\mathbf{x}$, $\psi(\mathbf{x})$.

*Mapping procedure*, $f$. The models that perform the mapping from the solution to the performance space have as argument a given hyperrectangle, $\mathbf{X}$, rather than a point, $\mathbf{x}$. They compute an estimate of the

average performance that is expected within $\mathbf{X}$, $\psi^{est}(\mathbf{X}) = f(\mathbf{X})$, not a single pointwise prediction, $\psi^{est}(\mathbf{x}) = f(\mathbf{x})$.

*Search procedure*, $S$. The search procedures explore the modified mapping models, $\psi^{est}(\mathbf{X}) = f(\mathbf{X})$, in order to generate and identify a set of final solutions, $\mathbf{X}^*$, that look particularly promising according to the corresponding estimated performance scores, $\psi^{est}(\mathbf{X}^*)$.

As we will see in subsequent sections, the mapping procedures, $f$, adopted in our learning methodologies, are based on direct sampling approaches:

*For any given* $\mathbf{X}$ *and* $\psi$, *we find those* $(\mathbf{x}, y)$ *pairs for which* $\mathbf{x} \in \mathbf{X}$, *and use this random sample to compute* $\psi^{est}(\mathbf{X})$ *and build confidence intervals for* $\psi(\mathbf{X})$.

These direct sampling strategies provide $\psi^{est}(\mathbf{X})$ estimators that are consistent, unbiased, and do not require a priori assumptions about the system behavior, thus remaining consistent and unbiased regardless of the validity of any assumptions. In addition to a point estimate of performance, they also provide a probabilistic bound on the uncertainty associated with it, through the construction of confidence intervals for $\psi(\mathbf{X})$. Furthermore, the accuracy of the estimates obtained is limited only by the amounts of data that are available, not by any of the structural or functional form choices that have to be made with other mapping models. Finally, they are also computationally efficient, since all the effort involved consists of a search for those pairs of data falling inside $\mathbf{X}$, followed by a computation of the average and standard deviation among these records.

The preceding set of characteristics and properties of the $\psi^{est}(\mathbf{X})$ estimators makes our type of mapping procedures, $f$, particularly appealing for the kinds of systems that we are especially interested to study, i.e., manufacturing systems where considerable amounts of data records are available, with poorly understood behavior, and for which neither accurate first-principles quantitative models exist nor adequate functional form choices for empirical models can be made a priori. In other situations and application contexts that are substantially different from the above, while much can still be gained by adopting the same problem statements, solution formats and performance criteria, other mapping and search procedures (statistical, optimization theory) may be more efficient.

A solution space, $\Xi$, consisting of hyperrectangles defined in the decision space, $\mathbf{X}$, is a basic characteristic common to all the learning methodologies that will be described in subsequent sections. The same does not happen with the specific performance criteria $\psi$, mapping models $f$, and search procedures $S$, which obviously depend on the particular nature of the systems under analysis, and the type of the corresponding performance metric, $y$.

## IV. Learning with Categorical Performance Metrics

The nature of the performance metric, $y$, is determined by the characteristics of the specific process under analysis. Since we are particularly interested in analyzing situations where $y$ is related to product or process quality, it is quite common to find systems where a categorical variable $y$ is chosen to classify and evaluate their performance. This may happen due to the intrinsic nature of $y$ (e.g., it can only be measured and assume qualitative values, such as "good," "high," and "low"), or because $y$ is derived from a quantization of the values of a surrogate continuous measure of performance (e.g., $y =$ "good" if some characteristic $z$ of the product has value within the range of its specifications, and $y =$ "bad," otherwise).

In this section we will introduce the problem statements adopted for this type of performance metric, briefly describe the learning methodology employed to address it [for a more complete presentation, see Saraiva and Stephanopoulos (1992a)], and show a specific application case study.

### A. Problem Statement

When $y$ is a categorical variable, which can assume only one among a total of $K$ discrete possible values, pattern recognition (Duda and Hart, 1973; James, 1985) provides an adequate context for the introduction of the learning methodologies. The most important features that separate these learning methodologies from existing classification techniques (such as linear discriminant functions, nearest-neighbor and other nonparametric classifiers, neural networks, etc.), are summarized in Table I, and briefly discussed below.

TABLE I

Conventional Pattern Recognition and Suggested Alternative

| | Conventional classification | Suggested alternative |
|---|---|---|
| $\xi$ | $\mathbf{x} \in \mathcal{R}^M$ | $\mathbf{X} \in \mathbf{I}^M$ |
| $\psi$ | $y(\mathbf{x})$ or $p(y = j\|\mathbf{x})$ | $y(\mathbf{X})$ or $p(y = j\|\mathbf{X})$ |
| $f$ | Technique-dependent | $\dfrac{n_j(\mathbf{X})}{n(\mathbf{X})}$ |
| $S$ | Nonexistent | Induction of decision trees |

*1. Conventional Classification Techniques*

All conventional classification procedures are aimed at answering one of the following questions:

*Given a generic vector of values* $\mathbf{x} \equiv [x_1, \ldots, x_m, \ldots, x_M]^T \in \mathscr{R}^M$, *what is the corresponding y value?* (7a)

and/or

*Given a generic vector of values* $\mathbf{x} \equiv [x_1, \ldots, x_m, \ldots, x_M]^T \in \mathscr{R}^M$, *what are reasonable estimates for the conditional probabilities* $p(y = j|\mathbf{x})$, $j = 1, \ldots, K$? (7b)

Thus, they share exactly the same solution ($\Xi$) and performance criteria ($\Psi$) spaces. Furthermore, since their role is simply to estimate $y$ for a given $\mathbf{x}$, no search procedures $S$ are attached to classical pattern recognition techniques. Consequently, the only element that differs from one classification procedure to another is the particular mapping procedure $f$ that is used to estimate $y(\mathbf{x})$ and/or $p(y = j|\mathbf{x})$. The available set of $(\mathbf{x}, y)$ data records is used to build $f$, either through the construction of approximations to the decision boundaries that separate zones in the decision space leading to different $y$ values (Fig. 2a), or through the construction of approximations to the conditional probability functions, $p(y = j|\mathbf{x})$.

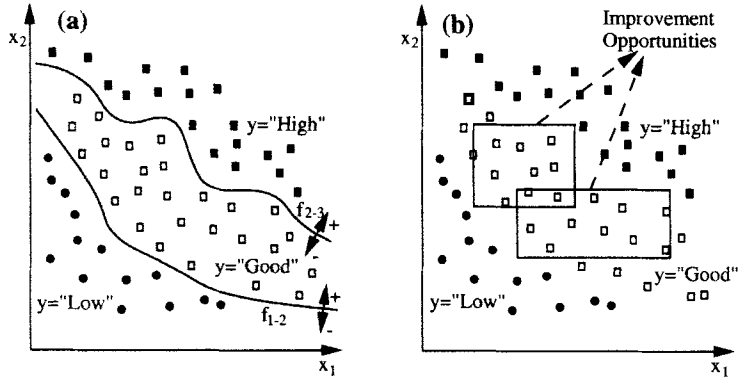


FIG. 2. (a) Conventional pattern recognition; (b) alternative problem statement and solution format.

*2. The Learning Methodology*

On the other hand, the question that we want to see answered is the following one (Fig. 2b):

*What are hyperrectangles in the decision space, $\mathbf{X} \in \mathbf{I}^M$, inside which one gets only a desired $y$ value, or at least a large fraction of that $y$ value?* (8)

The solution space thus consists of hyperrectangles in the decision space, $\mathbf{X} \in \mathbf{I}^M$, and the corresponding performance criteria are the conditional probabilities of getting any given $y$ value inside $\mathbf{X}$, $p(y=j|\mathbf{X})$, $j = 1, \ldots, K$, or the single most likely $y$ value within $\mathbf{X}$.

The mapping procedure, $f$, that allows us to compute $p(y=j|\mathbf{X})$ estimates, starts with a search performed over all the available $(\mathbf{x}, y)$ pairs that leads to the identification of the $n(\mathbf{X})$ cases for which $\mathbf{x} \in \mathbf{X}$. If we designate as $n_j(\mathbf{X})$ the number of such records with $y=j$, the desired estimates, $p^{\text{est}}(y=j|\mathbf{X})$, are given by

$$p^{\text{est}}(y=j|\mathbf{X}) = \frac{n_j(\mathbf{X})}{n(\mathbf{X})}, \qquad j = 1, \ldots, K. \tag{9}$$

Using the normal approximation to a binomial distribution, confidence intervals (CIs) for $p(y=j|\mathbf{X})$ can be established for a specific significance level, $\alpha$:

$$\text{CI} = \left[ p^{\text{est}}(y=j|\mathbf{X}) \pm t_{(\alpha/2,\, n(\mathbf{X})-1)} \cdot \sqrt{\frac{p^{\text{est}}(y=j|\mathbf{X}) \cdot (1 - p^{\text{est}}(y=j|\mathbf{X}))}{n(\mathbf{X})}} \right], \tag{10}$$

where $t$ stands for the critical value of the Student's distribution.

The search procedure, $S$, used to uncover promising hyperrectangles in the decision space, $\mathbf{X}^*$, associated with a desired $y$ value (e.g., $y =$ "good"), is based on *symbolic inductive learning algorithms*, and leads to the identification of a final number of promising solutions, $\mathbf{X}^*$, such as the ones in Fig. 2b. It is described in the following subsection.

## B. Search Procedure, $S$

In order to introduce the search procedure, $S$, we start by showing how classification decision trees lead to the definition of a set of hyperrectangles, and how they can be constructed from a set of $(\mathbf{x}, y)$ data records.

Then we describe the conversion of the knowledge captured by the induced decision trees into a set of final solutions, **X***.

### *1. Classification Decision Trees and Their Inductive Construction*

A classification decision tree allows one to predict in a sequential way the $y$ value (or corresponding conditional probabilities) that is associated with a particular **x** vector of values. At the top node of the tree ($A$ in Fig. 3), a first test is performed, based on the value assumed by a particular decision variable ($x_3$). Depending on the outcome of this test, vector **x** is sent to one of the branches emanating from node $A$. A second test follows, being carried out at another node ($B$), and over the values of the same or a different decision variable (e.g., $x_6$). This procedure is

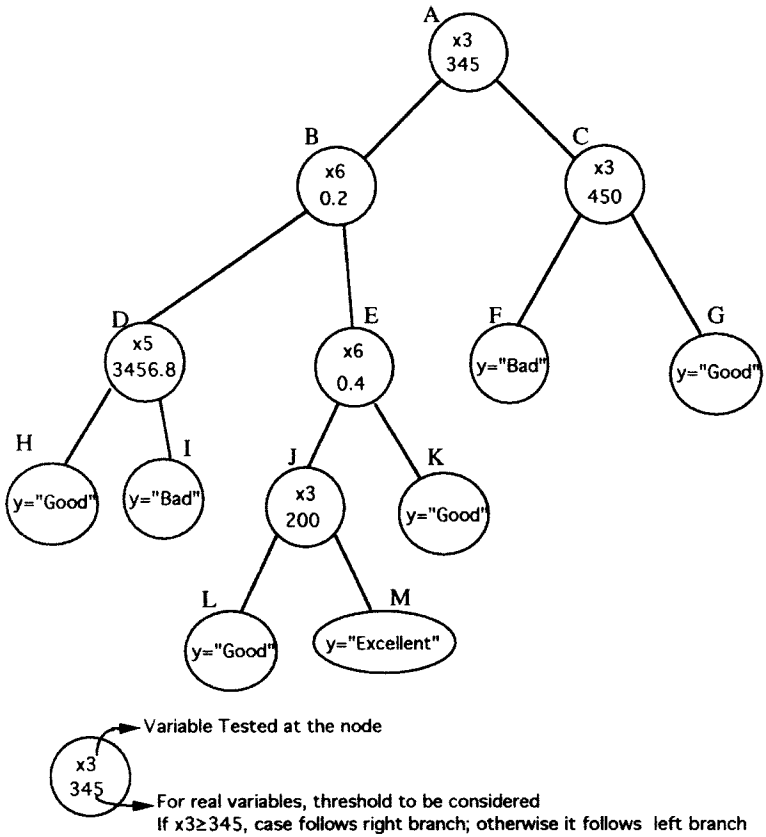


FIG. 3. A classification decision tree.

repeated until, after a last test, vector **x** follows a branch that leads to a terminal node (or leaf), labeled with a particular $y$ value and/or set of conditional probabilities, which provides the $y(\mathbf{x})$ and/or $p(y=j|\mathbf{x})$ estimates.

Although decision trees contain a number of attractive features, including competitive accuracy, when considered strictly as classification devices (Saraiva and Stephanopoulos, 1992a), the most important point for our purposes is that each of the tree's terminal nodes identifies a particular hyperrectangle in the decision space, **X**, associated with a given $y$ value. For example, node $M$ defines a $y =$ "excellent" rectangle that corresponds to the following rule:

$$\textit{If } 200 \leq x_3 < 345 \textit{ and } 0.2 < x_6 \leq 0.4, \textit{ then } y = \text{"Excellent,"} \quad (11)$$

Once a decision tree such as this has been constructed from the available set of $(\mathbf{x}, y)$ pairs, by picking up those leaves labeled with the desired $y$ value, one gets an initial collection of promising hyperrectangles. Some additional transformations, summarized in the next paragraph, convert this initial collection into a final set of solutions, $\mathbf{X}^*$.

The algorithm that we employ to build a classification decision tree from $(\mathbf{x}, y)$ data records belongs to a group of techniques known as *top–down induction of decision trees* (TDIDT) (Sonquist *et al.*, 1971; Fu, 1968; Hunt, 1962; Quinlan, 1986, 1987, 1993; Breiman *et al.*, 1984).

The construction starts at the root node of the tree, where all the available $(\mathbf{x}, y)$ pairs are initially placed. One identifies the particular split or test, $s$, that maximizes a given measure of information gain (Shannon and Weaver, 1964), $\Phi(s)$. The definition of a split, $s$, involves both the choice of the decision variable and the threshold to be used. Then, the $(\mathbf{x}, y)$ root node pairs are divided according to the best split found, and assigned to one of the children nodes emanating from it. The information gain measure, $\Phi(s)$, for a particular parent node $t$, is

$$\Phi(s) = -\sum_{k=1}^{K} P_k(t)\log_2 P_k(t) + \sum_{c=1}^{R} \frac{N(t_c)}{N(t)} \sum_{k=1}^{K} P_k(t_c)\log_2 P_k(t_c), \quad (12)$$

where

(a) $N(t)$ is the number of $(\mathbf{x}, y)$ pairs included in $t$.
(b) $R$ is the total number of children nodes, $t_c$, $c = 1, \ldots, R$, emanating from $t$.
(c) $P_k(t) = N_k(t)/N(t)$ is an estimate of the $p(y=k|t)$ conditional probability.
(d) $N_k(t)$ is the number of $(\mathbf{x}, y)$ pairs assigned to node $t$ for which $y = k$.

This splitting procedure is now applied recursively to each of the children nodes just created. The successive expansion process continues until terminal nodes or leaves, over which no further partitions are performed, can be identified.

The preceding strategy for the construction of decision trees provides an efficient way for inducing compact classification decision trees from a set of $(\mathbf{x}, y)$ pairs (Moret, 1982; Utgoff, 1988; Goodman and Smyth, 1990). Furthermore, tests based on the values of irrelevant $x_m$ variables are not likely to be present in the final decision tree. Thus, the problem dimensionality is automatically reduced to a subset of decision variables that convey critical information and influence decisively the system performance.

### 2. *From Decision Trees to Final Solutions*, $\mathbf{X}^*$

Once an induced decision tree has been found, it is subjected to an additional pruning and simplification treatment, whose details can be found in Saraiva and Stephanopoulos (1992a). Pruning and simplification lead to the definition of a revised and reasonably sized final decision tree, with greater statistical reliability. Each of the terminal nodes of this final decision tree corresponds to a particular hyperrectangle in the decision space, $\mathbf{X}$, which is labeled either with the $y$ value that is most likely to be obtained within that hyperrectangle, or with the corresponding conditional probabilities estimates, $p^{\text{est}}(y = j|\mathbf{X})$, $j = 1, \ldots, K$. These terminal nodes are further refined through an expansion process aimed at enlarging the zone of the decision space that they cover, followed by statistical significance tests that may introduce additional simplifications (Saraiva and Stephanopoulos, 1992a). By the end of this refinement stage, we get a group of improved, statistically significant, simplified, and partially overlapped hyperrectangles. Those that lead predominantly to the desired $y$ value constitute the final group of solutions, $\mathbf{X}^*$, and are presented to the user together with a number of auxiliary evaluation scores (Saraiva and Stephanopoulos, 1992a). It is the user's responsibility to analyze this set of hyperrectangles, $\mathbf{X}^*$, make a selection among them, and define the course of action to follow.

## C. Case Study: Operating Strategies for Desired Octane Number

To illustrate the potential practical capabilities of the learning methodology, we will now present the results obtained through its application to

records of operating data collected from a refinery unit (Daniel and Wood, 1980). Additional industrial case studies can be found in Saraiva and Stephanopoulos (1992a).

The $y$ variable that we will consider derives from a quantization of the octane numbers of the gasoline product, $z$, assuming one out of three possible values:

(a) $y = 1$ ("very low") for $z \leq 91$
(b) $y = 2$ ("low") when $91 < z < 92$
(c) $y = 3$ ("good") for $z \geq 92$

In this particular problem, one wishes to achieve values of $z$ as high as possible, and thus to identify zones in the decision space where one gets mostly $y = 3$.

There are four decision variables: three different measures of the feed composition $(x_1, x_2, x_3)$ and the value of an unspecified operating condition $(x_4)$.

In Fig. 4 we present the final induced decision tree, as well as the partition of the $(x_1, x_4)$ plane defined by its leaves, together with a projection of all the available $(\mathbf{x}, y)$ pairs on the same plane. These two decision variables are clearly influencing the current performance of the refinery unit, and the decision tree leaves perform a reasonable partition of the plane. To achieve better performance, we must look for operating zones that will result in obtaining mostly $y = 3$ values. Terminal nodes 2

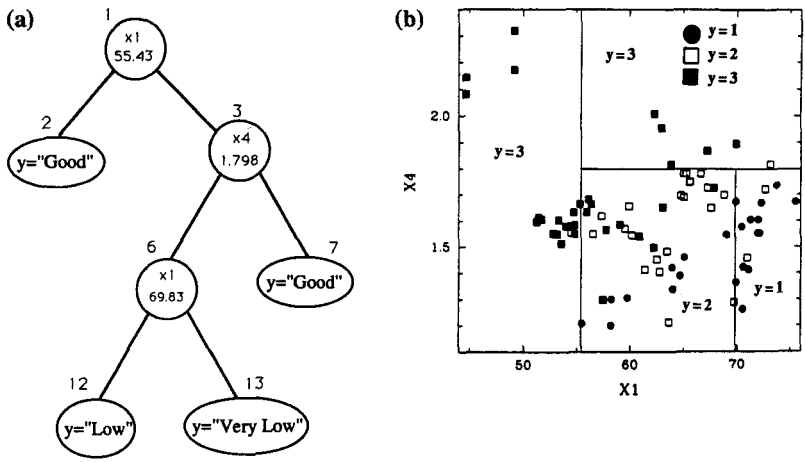


FIG. 4. (a) Induced decision tree; (b) partition of the plane defined by its leaves.

and 7 identify two such zones. The corresponding final solutions, $\mathbf{X}^*$, found after going through all the additional steps of refinement and validation, are

$$\mathbf{X}_1^* = \{x_1 \in [44.6, 55.4]\}, \text{ with } p^{\text{est}}(y = 3|\mathbf{X}_1^*) = 0.9,$$

$$\mathbf{X}_2^* = \{x_4 \in [1.8, 2.3]\}, \text{ with } p^{\text{est}}(y = 3|\mathbf{X}_2^*) = 1.0$$

It can be noticed from the conditional probability estimates that one should expect to get almost only "good" $y$ values while operating inside these zones of the decision space, as opposed to the current operating conditions, which lead to just 40% of "good" $y$ values.

## V. Continuous Performance Metrics

In Section IV we considered a categorical performance metric $y$. Although that represents a common practice, especially when $y$ defines the quality of a product or process operation, there are many instances where system performance is measured by a continuous variable. Even when $y$ is quality-related, it is becoming increasingly clear that explicit continuous quality cost models should be adopted and replace evaluations of performance based on categorical variables.

This Section addresses cases with a continuous performance metric, $y$. We identify the corresponding problem statements and results, which are compared with conventional formulations and solutions. Then Taguchi loss functions are introduced as quality cost models that allow one to express a quality-related $y$ on a continuous basis. Next we present the learning methodology used to solve the alternative problem statements and uncover a set of final solutions. The section ends with an application case study.

### A. Problem Statement

A number of different techniques have been suggested and applied to address situations where $y$ is a continuous variable. Table II summarizes the most important characteristics of our approach and major features that differentiate it from conventional procedures.

TABLE II

CONVENTIONAL APPROACHES AND SUGGESTED ALTERNATIVE

| | Conventional approaches | Suggested alternative |
|---|---|---|
| $\xi$ | $\mathbf{x} \in \mathscr{R}^M$ | $\mathbf{X} \in \mathbf{I}^M$ |
| $\psi$ | $y(\mathbf{x})$ or $E(y\|\mathbf{x})$ | $E(y\|\mathbf{X})$ |
| $f$ | Technique-dependent | $\frac{1}{n(\mathbf{X})} \sum_{j=1}^{n(\mathbf{X})} y_j$ |
| $S$ | Optimization | Exploratory search |

### 1. *Conventional Procedures*

All conventional approaches (mathematical and stochastic programming, parametric and nonparametric regression analysis) adopt as a common solution format real vectors, $\mathbf{x} \in \mathscr{R}^M$, and as performance criterion, $\psi$, the expected value of $y$, $E(y|\mathbf{x})$, given $\mathbf{x}$, or the single $y$ value that corresponds to a specific $\mathbf{x}$, $y(\mathbf{x})$, if one assumes a fully deterministic relationship between $y$ and $\mathbf{x}$. Just as in Section IV, the element that essentially distinguishes the several techniques is the mapping procedure, $f$, used to compute $\psi^{\text{est}} = f(\mathbf{x})$. In order to find a final solution $\mathbf{x}^*$, optimization algorithms form the search procedure, $S$, that leads to the identification of the particular point in the decision space that maximizes or minimizes $\psi^{\text{est}} = f(\mathbf{x})$.

These conventional approaches usually follow a two-step sequential process:

Step 1. The $(\mathbf{x}, y)$ records of data are employed to build the $f$ mapping.

Step 2. $S$ does not make direct use of the original data, but rather employs $f$ to find the final solution, $\mathbf{x}^*$, that optimizes $\psi^{\text{est}} = f(\mathbf{x})$.

### 2. *The Learning Methodology*

For the reasons already discussed in Section III, our solution space consists of hyperrectangles in the decision space, $\mathbf{X} \in \mathbf{I}^M$, not single points, $\mathbf{x}$. The corresponding performance criterion used to evaluate solutions, $\psi$, is the expected $y$ value within $\mathbf{X}$:

$$\psi(\mathbf{X}) = E(y|\mathbf{X}). \tag{13}$$

These conceptual changes in both solution formats and performance

criteria are independent of the particular procedures chosen to estimate $\psi(\mathbf{X})$ and search for a set of final solutions, $\mathbf{X}^*$. However, as was also discussed in Section III, for the types of systems that we are specially interested in analyzing, direct sampling strategies to estimate $\psi(\mathbf{X})$ offer a number of advantages. The mapping model that we employ, $f$, is similar to the one adopted for categorical $y$ variables. A search is performed over all the available $(\mathbf{x}, y)$ data records, leading to the identification of a total of $n(\mathbf{X})$ pairs for which $\mathbf{x} \in \mathbf{X}$. The performance criterion estimate, $\psi^{\text{est}}(\mathbf{X})$, is the sample $y$ average among these $n(\mathbf{X})$ pairs:

$$\psi^{\text{est}}(\mathbf{X}) = f(\mathbf{X}) = \frac{1}{n(\mathbf{X})} \sum_{j=1}^{n(\mathbf{X})} y_j \tag{14}$$

The corresponding confidence interval, CI, for $E(y|\mathbf{X})$, at a given significance level, $\alpha$, is

$$\text{CI} = \left[ f(\mathbf{X}) \pm t_{(\alpha/2,\, n(\mathbf{X})-1)} \cdot \frac{s_y(\mathbf{X})}{\sqrt{n(\mathbf{X})}} \right], \tag{15}$$

where $s_y(\mathbf{X})$ stands for the sample standard deviation:

$$s_y(\mathbf{X}) = \left\{ \frac{1}{n(\mathbf{X}) - 1} \sum_{j=1}^{n(\mathbf{X})} \left[ y_j - f(\mathbf{X}) \right]^2 \right\}^{0.5}. \tag{16}$$

If one is interested only in finding the single feasible hyperrectangle (i.e., respecting minimum width constraints imposed due to control limitations) that minimizes $\psi^{\text{est}}(\mathbf{X})$, to find that hyperrectangle one may choose as search procedure, $S$, any optimization routine. However, our primary goal is to conduct an exploratory analysis of the decision space, leading to the identification of a set of particularly promising solutions, $\mathbf{X}^*$, that are presented to the decisionmaker, who is responsible for a final selection and a choice of the course of action to follow. The search procedure adopted in our learning methodology, $S$, reflects this goal, and will be described in a subsequent paragraph.

## B. Alternative Problem Statements and Solutions

Recognizing that, due to unavoidable variability in the decision variables, one has to operate within a zone of the decision space, and not at a single point, we might still believe that finding the optimal pointwise solution, $\mathbf{x}^*$, as usual, would be enough. The assumption behind such a

belief is that centering the operation around $\mathbf{x}^*$ will correspond in practice to the adoption of a zone in the decision space to conduct the operation, $\mathbf{X}$ [with $\mathbf{m}(\mathbf{X}) = \mathbf{x}^*$], that is equivalent or close to the best possible zone, $\mathbf{X}^*$. However, because the evaluation of performance at a single point in the decision space, $\mathbf{x}$, completely ignores the system behavior around that point, the preceding assumption in general does not hold: *searching for an optimal hyperrectangle leads to a final solution*, $\mathbf{X}^*$, *that is likely to lie in a region of the decision space quite distant from* $\mathbf{x}^*$, *and* $\mathbf{m}(\mathbf{X}^*) \neq \mathbf{x}^*$. This observation emphasizes how critical it is to adopt the modified problem statements described at the beginning of Section V, where a direct and explicit search for the best zone to operate replaces the classical optimization paradigm, which ignores variability in the decision variables and seeks to identify as precisely as possible the optimal $\mathbf{x}^*$. No matter how accurately $\mathbf{x}^*$ is determined, targeting the operation around it can represent a quite suboptimal answer when the random nature of the decision variables is taken into account.

To illustrate how different $\mathbf{m}(\mathbf{X}^*)$ and $\mathbf{x}^*$ may happen to be, let's consider as a specific example (others can be found in Saraiva and Stephanopoulos, 1992c) a Kraft pulp digester. The performance metric $y$, that one wishes to minimize, is determined by the kappa index of the pulp produced and the cooking yield. Two decision variables are considered: *H-factor* ($x_1$), and *alkali charge* ($x_2$). Furthermore, we will assume as perfect an available deterministic empirical model (Saraiva and Stephanopoulos, 1992c), $f$, which expresses $y$ as function of $\mathbf{x}$, i.e., that $y = f(x_1, x_2)$ is perfectly known.

If one follows the conventional optimization paradigm, adopting pointwise solution formats, the best feasible answer, $\mathbf{x}^*$, which minimizes $f(\mathbf{x})$, is $\mathbf{x}^* = (200; 17.9)$, as can be confirmed by examining the contour plots of $f$ shown in Fig. 5a.

On the other hand, when the unavoidable variability in the decision space is considered explicitly, the goal of the search becomes the identification of the optimal hyperrectangle, $\mathbf{X}^*$, which solves the following problem:

$$\min_{\mathbf{X} \in \mathbf{I}^2} f(\mathbf{X}), \tag{17}$$

subject to

$$w(X_1) \geq \Delta x_1 = 300,$$

$$w(X_2) \geq \Delta x_2 = 1.0,$$

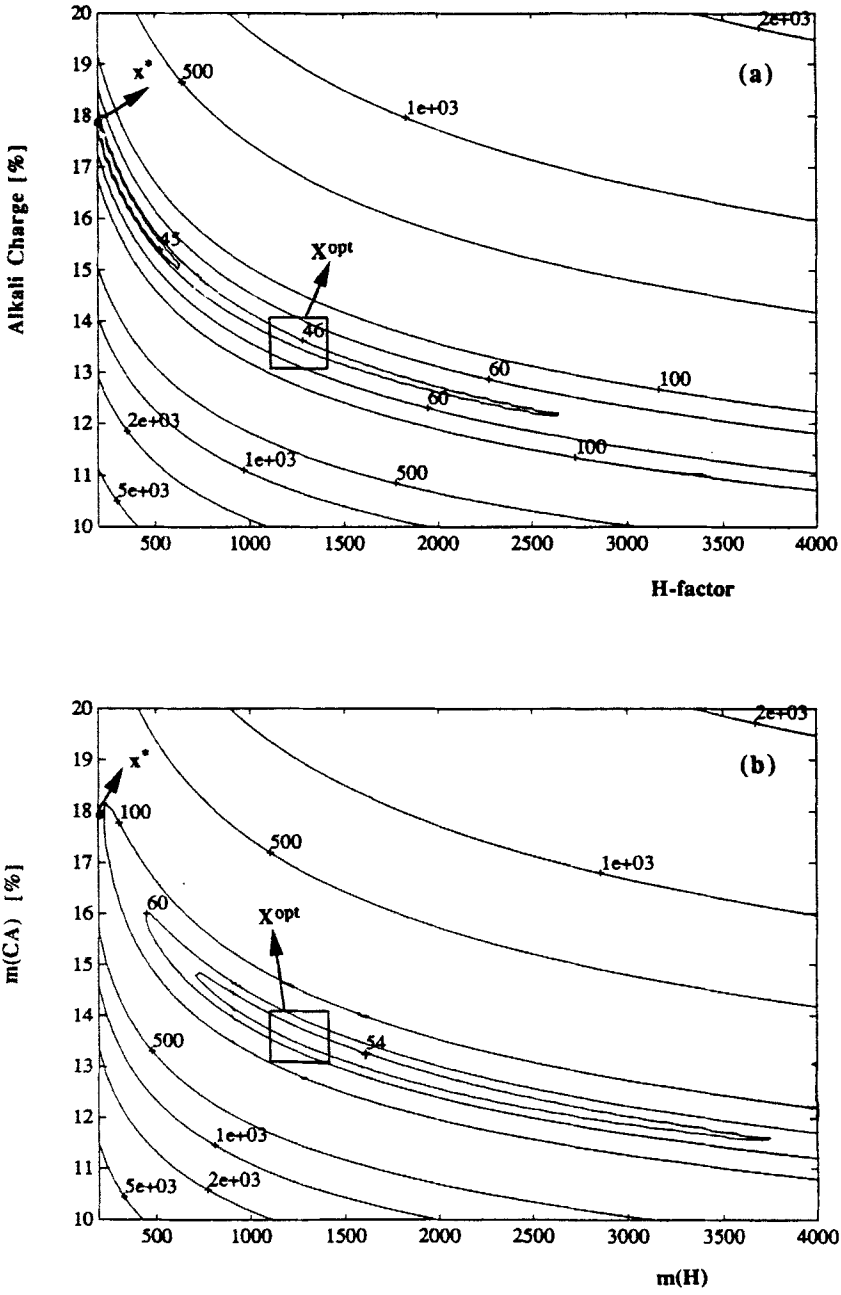


FIG. 5. Contour plots and optimal solutions for (a) $f(\mathbf{x})$ versus $\mathbf{x}$ and (b) $E(f(\mathbf{x})|\mathbf{X})$ versus $\mathbf{m}(\mathbf{X})$.

with

$$f(\mathbf{X}) = E(y|\mathbf{X}) = \frac{\int_{i(X_1)}^{s(X_1)} \int_{i(X_2)}^{s(X_2)} p(\mathbf{x}) f(\mathbf{x})\, dx_1\, dx_2}{\int_{i(X_1)}^{s(X_1)} \int_{i(X_2)}^{s(X_2)} p(\mathbf{x})\, dx_1\, dx_2},$$

and $p(\mathbf{x})$ representing the $(x_1, x_2)$ joint probability density function (for this particular study we assumed that both decision variables are independent and uniformly distributed).

In Fig. 5b we present the contour plots for $E(y|\mathbf{X})$ as a function of $\mathbf{m}(\mathbf{X})$. The corresponding optimal rectangle, $\mathbf{X}^*$, is

$$\mathbf{X}^* = \{x_1 \in [1114; 1414] \wedge x_2 \in [13.1; 14.1]\}. \tag{18}$$

This solution lies in a zone of the decision space that is quite distant from $\mathbf{x}^* = (200; 17.9)$ and targeting the operation around $\mathbf{x}^*$ results in clearly suboptimal performance.

This discrepancy illustrates some of the dangers associated with looking exclusively for pointwise solutions and performances, while neglecting the decision variables' variability. Although the specific problem-solving strategies developed in this chapter are aimed at the analysis of systems for which no good quantitative $f(\mathbf{x})$ and $p(\mathbf{x})$ exist a priori, the example presented shows the more general nature of the benefits that may derive from adopting the suggested alternative problem statements even when such models are available and used to find the final optimal hyperrectangle, $\mathbf{X}^*$.

Finally, it should be added that the conventional problem statement and pointwise solution format can be interpreted as a particular degenerate case of our more general formulations. As the minimum acceptable size for zones in the decision space decreases, the different performance criteria converge to each other and $\mathbf{X}^*$ gets closer and closer to $\mathbf{x}^*$. Both approaches become exactly identical in the extreme limiting case where $\Delta x_m = 0$, $m = 1, \ldots, M$, which is the particular degenerate case adopted in traditional formulations.

## C. Taguchi Loss Functions as Continuous Quality Cost Models

The development of most of the optimization and operations research techniques was motivated and focused on the minimization of operating costs, which are usually expressed on a quantitative basis. However, when

$y$ represents a quality related measure, performance has been traditionally evaluated through a categorical variable, whose values depend on whether the product is inside or outside the range of desired specifications. But it is recognized today that just being within any type of specification limits is not good enough, and the idea that any product is equally good inside a given range of values and equally bad outside it must be revised (Deming, 1986; Roy, 1990). This points to the need for assuming continuous performance metrics even when $y$ is quality related. One of the most powerful contributions of Taguchi (1986) to quality management was the proposition of loss functions as ways of quantifying and penalizing on a continuous basis any deviations from a desired nominal target (Phadke, 1989; Clausing, 1993). Given a quality functional characteristic $z$, with a nominal target $z^*$, any deviation from $z^*$ has some quality cost associated with it, and this cost increases gradually as we move away from the target (Fig. 6). To operationalize and quantify this quality degradation process, Taguchi loss functions express quality costs on a monetary basis, commensurate with operating costs, and define the particular quality cost associated with a generic $z$ value, $L(z)$, as

$$L(z) = k(z - z^*)^2, \tag{19}$$

where $k$ is a constant known as quality loss coefficient. The value of $k$ is

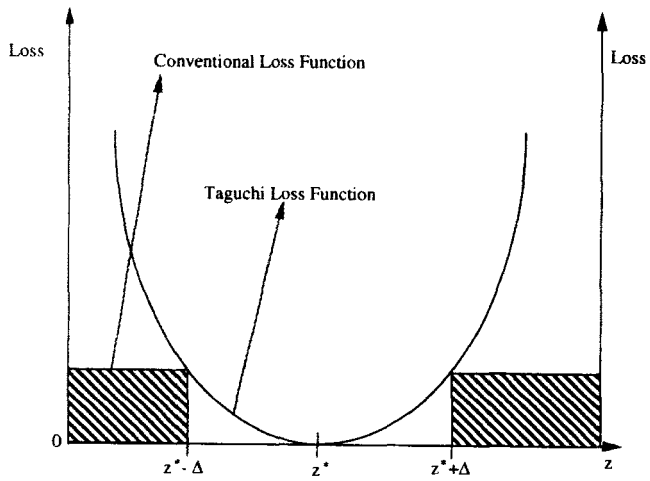


FIG. 6. Conventional and Taguchi quality cost models.

usually found by assigning a loss value to the specification limits, established at $z^* \pm \Delta$:

$$k = L(z^* \pm \Delta)/\Delta^2, \tag{20}$$

$$L(z) = k(z - z^*)^2 = L(z^* \pm \Delta) \cdot (z - z^*)^2/\Delta^2. \tag{21}$$

It is also important to realize that Taguchi loss functions not only bring into consideration both issues of location and dispersion of $z$ but also provide a consistent format for combining them. By taking expectations on both sides of Eq. (19), and after a few algebraic rearrangements, we can show that the expected loss, $E[L(z)]$ is

$$E[L(z)] = k\left[\sigma_z^2 + (\mu_z - z^*)^2\right], \tag{22}$$

where $\sigma_z^2$ is the $z$ variance and $\mu_z$ is the $z$ expectation.

For a zone $\mathbf{X}$ in the decision space to lead to a small conditional expected loss, $E[L(z)|\mathbf{X}]$, it must achieve both precise (reduced $\sigma_z$) and accurate ($\mu_z \cong z^*$) performance with respect to $z$. Finding such robust zones and operating on them results in inoculating the process against the transmission of variation from disturbances and the decision space to the performance space (Taylor, 1991).

If, besides the quality-related measure, $z$, one also wishes to include operating costs, $\zeta$, in the analysis, because quality loss functions express quality costs on a monetary basis, commensurate with operating costs, the final global performance metric, $y$, which reflects total manufacturing cost, is simply the sum of both quality and operating costs (Clausing, 1993),

$$y = L(z) + \zeta. \tag{23}$$

Consequently, the goal of our learning methodology is the identification of hyperrectangles in the decision space, $\mathbf{X}$, that minimize expected total manufacturing cost, $E(y|\mathbf{X})$, a performance measure that combines in a consistent form and a quantitative basis both operating and quality costs.

### D. Learning Methodology and Search Procedure, $S$

In the previous paragraphs we defined the solution format $\xi$, performance criterion $\psi$, mapping procedure $f$, and performance metric $y$ that characterize our learning methodology for systems with a quantitative metric $y$. Here we will assemble all these pieces together and briefly discuss the search procedure, $S$ (further details can be found in Saraiva

and Stephanopoulos, 1992c), that is employed to identify a set of final solutions, $\mathbf{X}^*$, which achieve low $\psi^{\text{est}}(\mathbf{X})$ scores. The two main stages of the learning procedure are examined in the following paragraphs.

### 1. *Problem Formulation*

The total loss function, $y$, given by Eq. (23), is not directly measured and has to be computed from information that is available and collected from the process, consisting of $(\mathbf{x}, z)$ pairs. After defining an adequate loss function, $L(z)$, and considering operating costs, $\zeta$, one can identify the $(\mathbf{x}, y)$ pairs that correspond to each of the initial $(\mathbf{x}, z)$ data records.

Before starting the search for solutions, it is necessary to select among the $M$ decision variables a subset of $H$ variables, $x_h$, $h = 1, \ldots, H$, which influence significantly the system performance, and thus will be used by $S$ and included in the definition of the final set of hyperrectangles, $\mathbf{X}^*$. For this preliminary choice of critical decision variables, other than his or her own specific process knowledge, the decisionmaker can count on a number of auxiliary techniques enumerated in Saraiva and Stephanopoulos (1992c).

This first stage of the learning procedure is concluded with the definition of a set of constraints related to

(a) Minimum acceptable width of operating windows $[w(X_h) \geq \Delta x_h, h = 1, \ldots, H]$

(b) Minimum number of data records, $N_{\min}$, that must be covered by any final solution, $\mathbf{X}^*[n(\mathbf{X}) \geq N_{\min}$ for $\mathbf{X}$ to be considered a feasible solution]. This constraint is identical to specifying a given maximum acceptable width for the $E(y|\mathbf{X})$ confidence interval that is obtained at a significance level, $\alpha$, as defined by Eq. (15).

### 2. *Search Procedure*

Rather than finding the exact location of the single feasible hyperrectangle that optimizes $\psi^{\text{est}}(\mathbf{X})$, our primary goal is to conduct an exploratory analysis of the decision space, leading to the definition of a set of particularly promising solutions, $\mathbf{X}^*$, to be presented to the decisionmaker.

To identify this set of final feasible solutions, $\mathbf{X}^* \in \mathbf{I}^H$, with low $\psi^{\text{est}}(\mathbf{X})$ scores, we developed a greedy search procedure, $S$ (Saraiva and Stephanopoulos, 1992c), that has resulted, within an acceptable computation time, in almost-optimal solutions for all the cases studied so far, while avoiding the combinatorial explosion with the number of $(\mathbf{x}, y)$ pairs of an exhaustive enumeration/ evaluation of all feasible alternatives. The algorithm starts by partitioning the decision space into a number of isovolu-

metric and contiguous hyperrectangular seed cells, where for each cell the width associated with variable $x_h$ is smaller than the corresponding $\Delta x_h$. Then, we gradually enlarge these seed cells, until they satisfy all imposed constraints, and further growth is found to degrade their estimated performance. Each initial cell is thus converted into a feasible solution candidate, $\mathbf{X}$, and the corresponding $\psi^{\text{est}}(\mathbf{X})$ score is evaluated. Those feasible solution candidates receiving the lowest $\psi^{\text{est}}(\mathbf{X})$ are included in the set of final solutions, $\mathbf{X}^*$, that is presented to the decisionmaker. It is the user's responsibility to analyze this set, make a selection among its elements, and thus choose the ultimate target zone to conduct the operation of the process.

## E. Case Study: Pulp Digester

In order to verify how close to a known true optimum the final solutions found by our learning methodology happen to be, we will describe here its application to a pulp digester, for which a perfect empirical model $f(\mathbf{x})$ is assumed to be available. Other applications are discussed in Saraiva and Stephanopoulos (1992c).

The original data format consists of $(x_1, x_2, z, \omega)$ records, where

$x_1$ stands for the *H-factor*.

$x_2$ is the *alkali charge*.

$z$ is the pulp *kappa index*, with nominal target set at 30.0.

$\omega$ is the cooking yield, an indirect measure of operating cost that one wishes to maximize.

After defining the $z$ loss function as

$$L(z) = 10(z-30)^2 \quad \$/\text{ton of pulp} \tag{24}$$

(where \$ = U.S. dollars), and combining it with a commensurate measure of operating cost, expressed as a very simple function of $\omega$

$$\zeta = 100 - \omega \quad \$/\text{ton of pulp}, \tag{25}$$

one finally arrives at the identification of our total manufacturing cost performance metric,

$$y = 10(z-30)^2 + 100 - \omega \quad \$/\text{ton of pulp}, \tag{26}$$

which leads to the conversion of the original data records into the usual $(\mathbf{x}, y)$ format.

Let's consider that under the current operating conditions the values of $\mathbf{x}$ fall within a rectangle $\mathbf{X}_{\text{current}} = \{x_1 \in [200; 4000] \wedge x_2 \in [10; 20]\}$. Furthermore, we will assume that the two decision variables ($x_1$ and $x_2$) are independent and have uniform probability distributions. Using the available model, $f(\mathbf{x})$, we computed the current average total manufacturing cost, $E(y|\mathbf{X}_{\text{current}}) = 743.5$, a reference value that can be used to estimate the savings achieved with the implementation of any uncovered final solutions, $\mathbf{X}^*$.

To support the application of the learning methodology, $f(\mathbf{x})$ was used to generate 500 $(\mathbf{x}, z, \omega)$ records of simulated operational data, transformed by Eq. (26) into an equivalent number of $(\mathbf{x}, y)$ pairs. Finally, the following constraints were imposed to the search procedure, $S$:

(a) $w(X_1) \geq \Delta x_1 = 300$
$w(X_2) \geq \Delta x_2 = 1.0$
(b) $N_{\min} = 15$

Given the preceding problem definition, and after going through $S$, the final solution, $\mathbf{X}^*$, chosen for implementation is (Fig. 7):

$$\mathbf{X}^* = \{x_1 \in [910.2; 1566.3] \wedge x_2 \in [12.8; 14.6]\} \qquad E(y|\mathbf{X}^*) = 69.9. \tag{27}$$

Thus, $\mathbf{X}^*$ indeed leads to a quite significant average total cost reduction, because $E(y|\mathbf{X}^*) \ll E(y|\mathbf{X}_{\text{current}})$. Both $\mathbf{X}^*$ and $E(y|\mathbf{X}^*)$ are also close approximations (Fig. 7) to the true optimal solution given by Eq. (18), i.e., $\mathbf{X}_{\text{opt}}$ and $E(y|\mathbf{X}_{\text{opt}})$ are

$$\mathbf{X}_{\text{opt}} = \{x_1 \in [1114; 1414] \wedge x_2 \in [13.1; 14.1]\} \qquad E(y|\mathbf{X}_{\text{opt}}) = 52.2. \tag{28}$$

To benchmark our learning methodology with alternative conventional approaches, we used the same 500 $(\mathbf{x}, y)$ data records and followed the usual regression analysis steps (including stepwise variable selection, examination of residuals, and variable transformations) to find an approximate empirical model, $f^{\text{est}}(\mathbf{x})$, with a coefficient of determination $R^2 = 0.79$. This model is given by

$$y \approx f^{\text{est}}(\mathbf{x}) = a \ln(x_1) + b \ln(x_2) + cx_2 + dx_1^2 + ex_2^2 + gx_1 \cdot x_2, \tag{29}$$

whose parameters were fitted by ordinary least squares.

By employing $f^{\text{est}}(\mathbf{x})$ in Eq. (17), we used this approximate model to find a final solution, $\mathbf{X}_{\text{est}}$ (Fig. 7), that satisfies the (a) constraints and

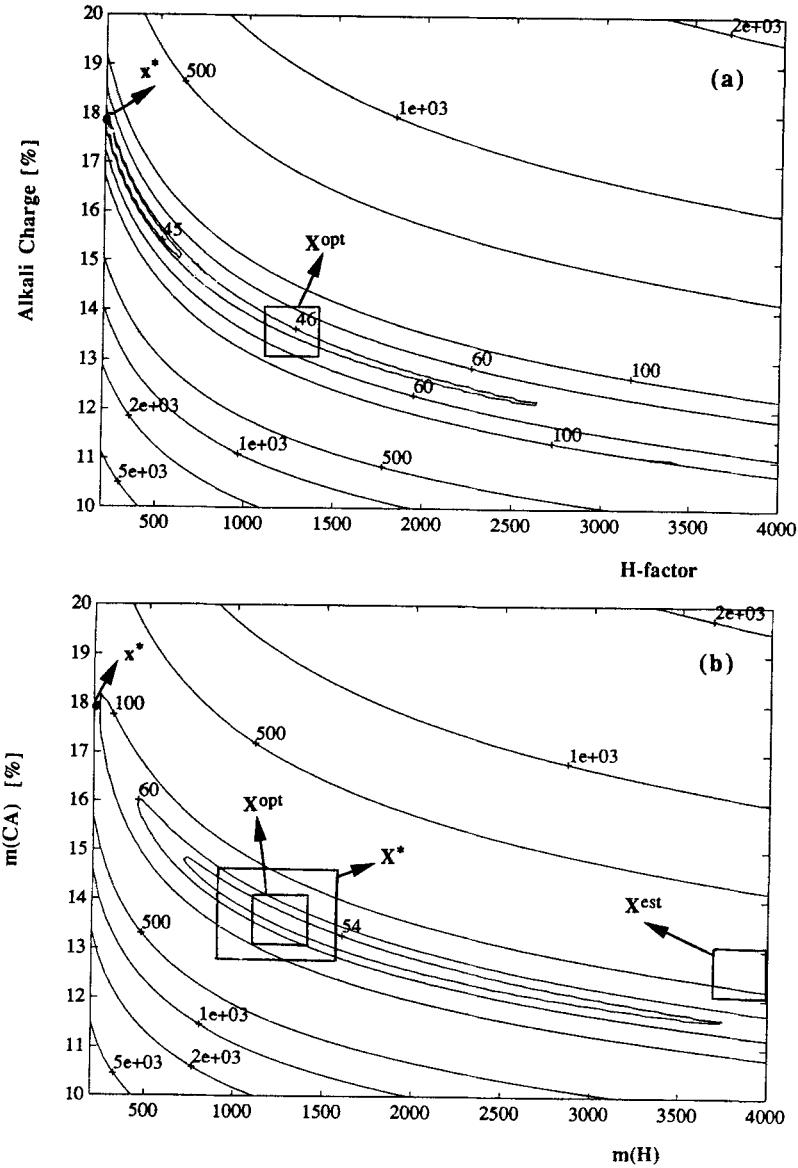


FIG. 7. Locations of $\mathbf{x}^*$, $\mathbf{X}^*$, $\mathbf{X}^{opt}$, and $\mathbf{X}^{est}$ in the decision space, and contour plots of (a) $y = f(\mathbf{x})$ versus $\mathbf{X}$, (b) $E(y|\mathbf{X})$ versus $\mathbf{m}(\mathbf{X})$.

minimizes $E[f^{est}(\mathbf{x})|\mathbf{X}]$:

$$\mathbf{X}_{est} = \{x_1 \in [3698; 3998] \wedge x_2 \in [12.1; 13.1]\} \qquad E(y|\mathbf{X}_{est}) = 145.6. \tag{30}$$

This solution leads to a considerably worse performance, $E(y|\mathbf{X})$, than $\mathbf{X}^*$, and it is also much more distant from the zone of the decision space where the true optimum, $\mathbf{X}_{opt}$, is located.

Results obtained with other $f^{est}(\mathbf{x})$ functional forms, for this and other similar problems, seem to indicate that even when only moderate amounts of data are available our direct sampling estimation procedure, $f$, and search algorithm, $S$, provide better final solutions than classical regression analysis followed by the use of Eq. (17) unless one is able to build from the data almost perfect $f^{est}(\mathbf{x})$ empirical models with the appropriate functional forms.

## VI. Systems with Multiple Operational Objectives

Except for the combination of quality loss and operating costs given by Eq. (23), in the previous sections we assumed the system performance to be determined by a single objective. However, in the analysis of pieces of equipment or plant segments of nontrivial size/ complexity, a multitude of objectives has usually to be taken into account in order to evaluate the system's global performance, and find ways to improve it.

In this section we describe extensions of the basic learning methodologies introduced in Sections IV and V that, while preserving the same premises and paradigms, enlarge considerably their scope by adding the capability to consider simultaneously multiple objectives. As before, and without loss of generality, we will focus our attention on the coexistence of several quality-related objectives.

Both situations with categorical and continuous, real-valued performance metrics will be considered and analyzed. Since Taguchi loss functions provide quality cost models that allow the different objectives to be expressed on a commensurate basis, for continuous performance variables only minor modifications in the problem definition of the approach presented in Section V are needed. On the other hand, if categorical variables are chosen to characterize the system's multiple performance metrics, important modifications and additional components have to be incorporated into the basic learning methodology described in Section IV.

A case study on the operational improvement of a plasma etching unit in microelectronics fabrication ends the section. This case study illustrates that if similar preference structures are used in both types of formulation, identical final solutions are found when either categorical or continuous performance evaluation modes are employed.

### A. Continuous Performance Variables

Instead of a single quality-related performance variable, $z$, as in Section V, let's suppose that one has to consider a total of $P$ distinct objectives and the corresponding continuous performance variables, $z_i$, $i = 1, \ldots, P$, which are components of a performance vector $\mathbf{z} = [z_1, \ldots, z_P]^T$. In such case, one has to identify the corresponding Taguchi loss functions, $L(z_i)$, $i = 1, \ldots, P$, for each of the performance variables:

$$L(z_i) = k_i(z_i - z_i^*)^2. \tag{31}$$

Since these loss functions express quality costs on a common and commensurate basis, extending the learning methodology of Section V to a situation with $P$ objectives is straightforward. All one has to do is replace the original definition of the $y$ performance metric [Eq. (23)] by the following more general version:

$$y = \sum_{i=1}^{P} k_i(z_i - z_i^*)^2 + \zeta. \tag{32}$$

Except for this modification, all the procedures and steps discussed in Section V carry over to the solution of multiobjective problems.

### B. Categorical Performance Variables

Rather than a single objective, $y$, as in Section IV, we now have a total of $P$ distinct categorical performance variables, $y_i$, $i = 1, \ldots, P$, associated with an equivalent number of objectives. Consequently, each data record is now composed of a $(\mathbf{x}, \mathbf{y})$ pair, where $\mathbf{y}$ is a performance vector defined by

$$\mathbf{y} = [y_1, \ldots, y_i, \ldots, y_P]^T. \tag{33}$$

The most important changes and adaptations that were introduced in order to handle such multiobjective problems are summarized in Table III. The solution space remains the same as for the single objective case.

TABLE III

SINGLE AND MULTIPLE CATEGORICAL PERFORMANCE VARIABLES

| | Single objective | Multiple objectives |
|---|---|---|
| $\xi$ | $\mathbf{X} \in \mathbf{I}^M$ | $\mathbf{X} \in \mathbf{I}^M$ |
| $\psi$ | $y(\mathbf{X})$ or $p(y=j\|\mathbf{X})$ | $\mathbf{y}(\mathbf{X})$ or $p(y_i=j\|\mathbf{X})$ |
| $f$ | $\frac{n_j(\mathbf{X})}{n(\mathbf{X})}$ | $\frac{n_{i,j}(\mathbf{X})}{n(\mathbf{X})}$ |
| $S$ | Induction of decision trees | Multiple agents and lexicographic search |

However, since now we have $P$ different objectives, the performance criteria, $\psi$, must include all of them. They may assume one of the following formats:

1. $\mathbf{y}(\mathbf{X}) = [y_1(\mathbf{X}), \ldots, y_i(\mathbf{X}), \ldots, y_P(\mathbf{X})]^T$,
   where $y_i(\mathbf{X})$ stands for the most likely $y_i$ value within the zone of the decision space defined by $\mathbf{X}$.
2. $p(y_i = j|\mathbf{X})$, $i = 1, \ldots, P$ and $j = 1, \ldots, K_i$,
   the set of conditional probabilities for getting any given value for each of the $y_i$ variables, with $K_i$ representing the total number of different possible values that $y_i$ can assume.

The mapping procedure, $f$, is identical to the one adopted for a single objective:

$$p^{\text{est}}(y_i = j|\mathbf{X}) = \frac{n_{i,j}(\mathbf{X})}{n(\mathbf{X})}, \qquad j = 1, \ldots, K_i, \tag{34}$$

where $n_{i,j}(\mathbf{X})$ is the total number of $(\mathbf{x}, \mathbf{y})$ pairs for which $\mathbf{x} \in \mathbf{X}$, and $y_i = j$.

The search procedure, $S$, requires major modifications in order to account for multiple objectives. As we will see, $S$ becomes now a highly interactive process, with progressive articulation/ elicitation of the user's preference structure, successive relaxation of aspiration levels and lexicographic construction of final solutions, $\mathbf{X}^*$, that lead to satisfactory joint performances according to all the $P$ objectives. It includes $P$ replicates of the basic learning methodology introduced in Section IV for a single objective, designated as *agents*, and a *coordination mechanism* that combines the results found by individual agents, in an attempt to identify conjunctions of zones uncovered by the agents that lead to the formation of the desired final solutions, $\mathbf{X}^*$. In the following paragraphs we will summarize the main steps of this search procedure (for details, see Saraiva and Stephanopoulos, 1992b).

*1. Problem Definition*

Besides the identification of the decision variables, $x_m$, $m = 1, \ldots, M$, and the performance variables, $y_i$, the user is asked to

(a) Rank the $P$ objectives in order of decreasing relative importance.
(b) Provide constraints on minimum acceptable widths [$w(X_m) \geq \Delta x_m$, $m = 1, \ldots, M$], and coverage, $N_{\min}$.
(c) Identify minimum acceptability criteria or constraints in the performance space (e.g., no more than 10% of "very low" $y_3$ values will be tolerated) that must be satisfied irrespectively of how good or bad the corresponding performances for the other objectives may be.

*2. Identification and Refinement of Initial Aspiration Levels*

From past experience and an examination of the results provided by each of the $P$ agents, a first tentative group of aspiration levels, $\mathbf{y}^*$, that one should aim to reach, is defined by the user ($y_1^* =$ "excellent$_1$," $y_2^* =$ "good-or-excellent$_2$," etc.):

$$\mathbf{y}^* = [y_1^*, \ldots, y_i^*, \ldots, y_P^*]^T. \tag{35}$$

These aspiration levels may be expressed either on an absolute ($y_1^* =$ "excellent$_1$") or on a probabilistic basis [$y_1^* \equiv p(y_1 =$ "excellent$_1$") $\geq$ 0.90].

Before beginning the search for feasible zones of the decision space where the preceding tentative aspiration levels can be achieved, a preliminary check for the possibility of existence of such a zone is conducted. If the perceived ideal, $\mathbf{y}^*$, does not pass this preliminary check, i.e., there is no commensurable solution to the multiobjective problem, the decisionmaker is asked to relax $\mathbf{y}^*$, in order to transform the problem into one with commensurable solutions. For instance, if the initial tentative perceived ideal were $\mathbf{y}^* =$ ("excellent$_1$," "excellent$_2$," "excellent$_3$"), but there aren't any available $(\mathbf{x}, \mathbf{y})$ pairs with $\mathbf{y} = \mathbf{y}^*$, the user might adopt as a revised combination of aspiration levels $\mathbf{y}^* =$ ("excellent$_1$," "excellent$_2$," "good-or-excellent$_3$"). This relaxation process, guided by the decisionmaker, continues until a final perceived ideal, $\mathbf{y}^*$, for which there are at least $N_{\min}(\mathbf{x}, \mathbf{y})$ pairs with $\mathbf{y} = \mathbf{y}^*$, can be identified. This is the set of aspiration levels that will be used to initiate the interactive search procedure for final solutions, $\mathbf{X}^*$.

*3. Search for Solutions*

The aspiration levels inherited from the previous step, $\mathbf{y}^*$, are used to guide the search process. Each agent $i$ employs the corresponding aspiration level, $y_i^*$, and through the application of the learning methodology presented in Section IV tries to identify feasible hyperrectangles, $\mathbf{X}_i^*$, that lead to performance consistent with $y_i^*$.

Then, we try to combine the partial solutions uncovered by the several agents, $\mathbf{X}_i^*$, in order to find feasible final solutions, $\mathbf{X}^*$, that lead to joint satisfactory behavior in terms of all the objectives, and thus consistent with the current $\mathbf{y}^*$. This is achieved by building conjunctions of multiple $\mathbf{X}_i^*$, uncovered by different agents, according to a breadth-first type of search (Winston, 1984), that takes into account the relative importance assigned to the objectives. Following this lexicographic approach, final solutions $\mathbf{X}^*$ are gradually constructed, and partial paths expanded to accomplish less important goals only when the aspiration levels for the most important ones have already been satisfied. The construction process (whose details are given in Saraiva, 1993; Saraiva and Stephanopoulos, 1992b) relies heavily on the interaction with the decisionmaker to overcome dead-ends, examine arising conflicts, establish tradeoffs, and guide the procedure.

*4. Validation of Results*

After the search has been concluded, all the uncovered feasible final solutions, $\mathbf{X}^*$, leading to satisfactory joint performances, consistent with $\mathbf{y}^*$, are presented to the decisionmaker for close examination and for the selection of a particular hyperrectangle within this group for eventual implementation.

However, conflicts between the fulfillment of different objectives and aspiration levels may prevent any feasible zone of the decision space from leading to satisfactory joint performances. If the search procedure fails to uncover at least one feasible final solution, $\mathbf{X}^*$, consistent with $\mathbf{y}^*$, a number of options are available to the decisionmaker to try to overcome this impasse. Namely, the decisionmaker can revise the initial problem definition, by either

(a) Redefining any of the constraints originally imposed.
(b) Introducing further relaxations of aspiration levels, which result in new and less demanding perceived ideals, $\mathbf{y}^*$.
(c) Excluding from the search space of agents the particular decision variable, $x_m$, which creates the conflict among objectives.

Such revisions to the problem statement in order to overcome unsuccessful applications of the search procedure may have to be repeated a

number of times before the problem possesses commensurable solutions, and one can find at least one final feasible solution, $\mathbf{X}^*$, that satisfies all the imposed constraints and achieves performances consistent with the current set of aspiration levels, $\mathbf{y}^*$.

## C. Case Study: Operational Analysis of a Plasma Etching Unit

To conclude this section on systems with multiple objectives, we will consider a specific plasma etching unit case study. This unit will be analyzed considering both categorical and continuous performance measurement variables. Provided that similar preference structures are expressed in both instances, we will see that the two approaches lead to similar final answers. Additional applications of the learning methodologies to multiobjective systems can be found in Saraiva and Stephanopoulos (1992b, c).

### *1. System Characterization*

This case study is based on real industrial data collected from a plasma etching plant, as presented and discussed in Reece *et al.* (1989). The task of the unit is to remove the top layer from wafers, while preserving the bottom one. Four different objectives and performance variables are considered:

(a) Maximize a measure of etching selectivity, $z_1$, expressed as the ratio of etching rates for the top and bottom layers.
(b) Minimize dispersion, $z_2$, of etching rate values across the wafer bottom-layer surface.
(c) Minimize dispersion, $z_3$, of etching rate values across the wafer, but now for the top layer.
(d) Maximize the average etching rate for the top layer, $z_4$.

A quantization of the $z_i$ variables resulted in the definition of the following categorical performance variables, $y_i$:

*Etching selectivity*:

$$y_1 = \begin{cases} \text{“bad}_1\text{”} & \text{if} \quad z_1 \leq 3.4, \\ \text{“good}_1\text{”} & \text{if} \quad 3.4 < z_1 \leq 4.0, \\ \text{“excellent}_1\text{”} & \text{if} \quad 4.0 < z_1. \end{cases}$$

*Bottom-layer etching dispersion*:

$$y_2 = \begin{cases} \text{“bad}_2\text{”} & \text{if} \quad 7.7 < z_2, \\ \text{“good}_2\text{”} & \text{if} \quad 7.0 < z_2 \leq 7.7, \\ \text{“excellent}_2\text{”} & \text{if} \quad z_2 \leq 7.0. \end{cases}$$

*Top-layer etching dispersion*:

$$y_3 = \begin{cases} \text{“bad}_3\text{”} & \text{if} \quad 4.6 < z_3, \\ \text{“good}_3\text{”} & \text{if} \quad 3.0 < z_3 \leq 4.6, \\ \text{“excellent}_3\text{”} & \text{if} \quad z_3 \leq 3.0. \end{cases}$$

*Average top-layer etching rate*:

$$y_4 = \begin{cases} \text{“bad}_4\text{”} & \text{if} \quad z_4 \leq 1870, \\ \text{“good}_4\text{”} & \text{if} \quad 1870 < z_4 \leq 2000, \\ \text{“excellent}_4\text{”} & \text{if} \quad 2000 < z_4. \end{cases}$$

Similarly, for the case where a continuous performance metric, $y$, is employed, the following loss functions were defined (Saraiva and Stephanopoulos, 1992c):

(a) $L(z_1) = 1.924(4.321 - z_1)^2$;
(b) $L(z_2) = 0.033(z_2 - 4.50)^2$;
(c) $L(z_3) = 0.022(z_3 - 0.31)^2$;
(d) $L(z_4) = 5.1387 \cdot 10^{-7}(2595.0 - z_4)^2$;

and $y = \sum_{i=1}^{4} L(z_i)$.

The three decision variables, and the corresponding ranges of values in $\mathbf{X}_{\text{current}}$, are

$x_1$: power at which the unit is operated, with values ranging between 75 and 150 W (watts)

$x_2$: pressure in the apparatus, ranging from 200 to 255 mtorr (millitorr)

$x_3$: flow of etchant gas, varying between 20 and 40 sccm (cubic centimeters per minute of gas flow at standard temperature and pressure conditions)

Since only 20 data records were collected from the system during the execution of the designed experiments conducted by Reece *et al.* (1989), we used their response surface models, deliberately contaminated with small Gaussian noise terms, to generate a total of 500 $(\mathbf{x}, \mathbf{z})$ pairs (assuming that the three variables, $x_1, x_2, x_3$, have independent and uniform

probability distributions). Finally, the following constraints were considered:

(a) As minimum acceptable window sizes, values close to 10% of the $\mathbf{X}_{\text{current}}$ ranges are used, leading to $\Delta x_1 = 7.5$, $\Delta x_2 = 5.5$, and $\Delta x_3 = 2.05$.
(b) As minimum coverage, we set $N_{\min} = 5$.

*2. Categorical Performance Variables*

The initial tentative perceived ideal, $\mathbf{y}^*$, was set at

$$\mathbf{y}^* = [\text{“excellent}_i\text{”}]^T, \qquad i = 1,\ldots,4.$$

After successive interactive relaxations, all of them leading to an insufficient number of $(\mathbf{x},\mathbf{y})$ pairs that jointly satisfy the aspiration levels, we finally came down to the following revision of the perceived ideal:

$$\mathbf{y}^* = [\text{“good-or-excellent}_i\text{”}]^T, \qquad i = 1,\ldots,4.$$

After going through the complete search procedure, given the perceived ideal shown above, the following final solution, $\mathbf{X}_1^*$, was selected:

$$\mathbf{X}_1^* = \{x_1 \in [134.6, 149.1] \wedge x_2 \in [235.1, 243.2] \wedge x_3 \in [20.9, 25.4]\}. \tag{36}$$

A projection of this solution into the $(x_1$–$x_2)$ plane is shown in Fig. 8, together with the available $(\mathbf{x},\mathbf{y})$ pairs that verify the condition imposed over $x_3$ values. One can qualitatively confirm the validity of condition (36): $\mathbf{X}_1^*$ is indeed a good approximation of the zones in the $x_1$–$x_2$ plane that by visual inspection one would associate with leading simultaneously to “good-or-excellent$_i$” performances for all the four objectives.

*3. Continuous Performance Variables*

The solution found when the plasma etching was analyzed in terms of continuous performance metrics is also presented in Fig. 8, and is given by

$$\mathbf{X}_2^* = \{x_1 \in [129.79; 140.39] \wedge x_2 \in [230.17; 243.22] \wedge x_3 \in [20.22; 23.18]\}. \tag{37}$$

This solution is similar to the one found [see hyperrectangle defined by Eq. (36)] previously, when categorical performance evaluation variables were employed. Since the preference structures expressed under both

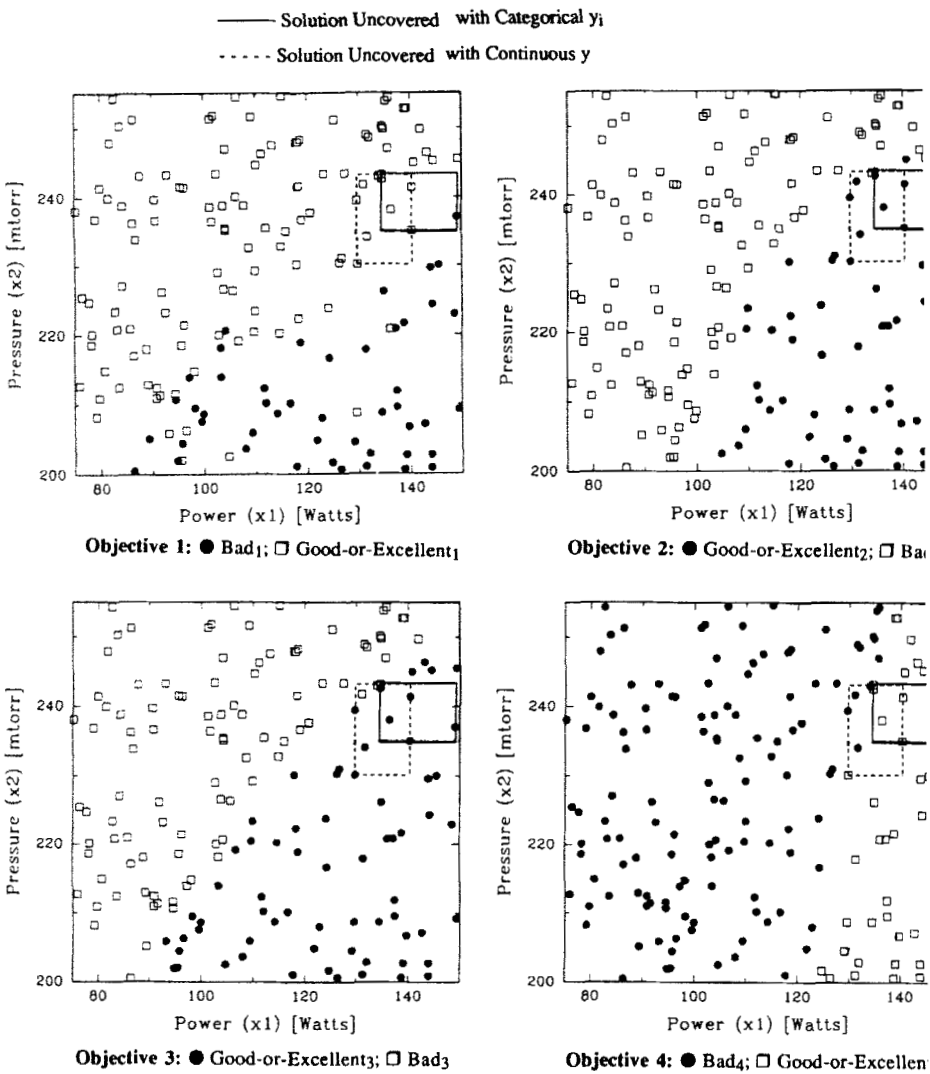


FIG. 8. Data records and final solutions.

formats were chosen to be consistent with each other, this is what one should expect and desire to happen.

Additional studies documented in Saraiva and Stephanopoulos (1992b, c) also illustrate how the introduction of changes in preference structures is translated into displacements of the final uncovered solutions in the decision space.

## VII. Complex Systems with Internal Structure

In previous sections we covered a variety of possible applications, including single or multiple objectives, continuous or categorical performance variables. However, so far it has always been assumed that the systems studied are simple systems without any type of internal structure, isolated from the remaining world, and self-sufficient for decisionmaking purposes. In this section we discuss additional extensions of the basic learning methodologies, in order to address complex systems composed of a number of interconnected subsystems. Although situations with categorical performance variables can be treated in similar ways, requiring only minor changes and adaptations, we will consider here only continuous performance metrics.

First, we discuss the problem statements and key features of the learning architecture that are specific to complex systems. This is followed by a brief presentation of the search procedures that are used to build a final solution. The section ends with a summary of the application of the learning architecture to the analysis of a Kraft pulp mill.

A more detailed description of this section's contents can be found in Saraiva (1993) or Saraiva and Stephanopoulos (1992d).

### A. Problem Statement and Key Features

Complex manufacturing systems, such as an unbleached Kraft pulp plant (Fig. 9), are almost always characterized by some type of internal structure, composed of a number of interconnected subsystems with their own data collection and decisionmaking responsibilities. This raises a number of additional issues, not addressed in previous sections. For instance, if the learning methodology described in Section VI is applied to the digester module of a pulp plant (Fig. 9), it is possible for the final selected solution, $\mathbf{X}_{\text{digester}}$, to include ranges of desired values of sulfidity or other composition properties of the white liquor that enters the digester. This being the case, and since an adjustment of the liquor composition has to be achieved elsewhere in the plant, the request over white liquor sulfidity values has to be propagated backward, to the causticizing area, and eventually from this module to the one preceding it, before a final solution can be found.

In the development of a learning architecture able to extend the methodologies introduced in other sections to complex systems, we looked

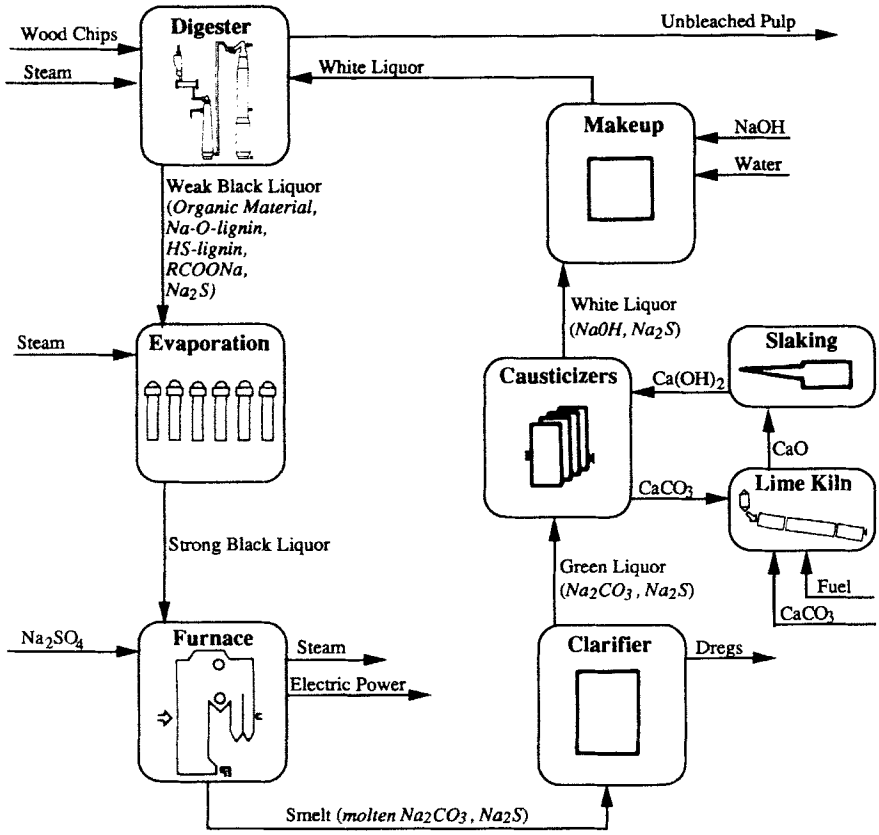


FIG. 9. General overview of unbleached Kraft pulp plant.

explicitly (for reasons presented and justified in Saraiva and Stephanopoulos, 1992c) for approaches that

(a) Are modular and decentralized, allowing each subsystem to take the initiative or make its own decisions, and assigning coordination roles to the upper hierarchical level.
(b) Support and reflect the existing organizational decisionmaking structures, responsibilities, data collection, and analysis activities.

Most of the existing tools to improve process operations fail to provide a systematic and formal process of handling complex systems, or do so in ways that do not fulfill the preceding set of requests. In this paragraph we provide a more formal characterization of a complex system and its several

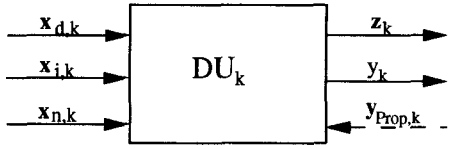


FIG. 10. Schematic representation of a decision unit.

subsystems, which will then be used to introduce our problem statement, and compare it with analogous conventional formulations.

### 1. *Complex Systems as Networks of Interconnected Subsystems*

The basic building block in the definition of a complex system, as well as the key element in our learning architecture, is what we will designate as an *infimal decision unit* or subsystem (Mesarović *et al.*, 1970; Findeisen *et al.*, 1980), $DU_k$ (Fig. 10). These decision units will in general correspond to a particular piece of equipment or section of the plant. The overall system is represented by a single *supremal decision unit* (Mesarović *et al.*, 1970; Findeisen *et al.*, 1980), $DU_0$, and contains a total of $K$ interconnected infimal decision units (Fig. 11), $DU_k$, $k = 1, \ldots, K$.

For each of the infimal decision units, $DU_k$, one has to consider several groups of input and output variables (Fig. 10). Among the inputs are

(a) A vector of decision variables, $\mathbf{x}_{d,k}$, which are variables that fall under the scope of authority and can be directly manipulated by $DU_k$.

(b) A vector of connection variables, $\mathbf{x}_{i,k}$, containing those variables that link consecutive infimal decision units, because they are simultaneously inputs to $DU_k$ (although not under its control) and outputs from the preceding decision unit, $DU_{k+1}$.

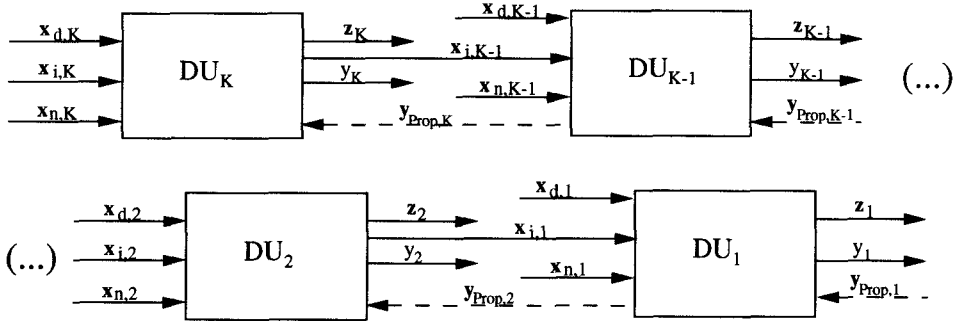


FIG. 11. Complex system as a sequence of infimal decision units.

(c) A vector of disturbance factors, $\mathbf{x}_{n,k}$, including variables that reside within the boundaries of $DU_k$, but nonetheless over which neither $DU_k$ nor any of the other decision units have any control.

As for possible outputs, one has

(a) A vector of performance variables, $\mathbf{z}_k = [z_{1,k}, \ldots, z_{p(k),k}]$ [*note:* $p(k)$ stands for the total number of subsystem $k$ objectives], that reflect operating costs, quality related measures and/or the violation of existing constraints on either the input or output spaces.
(b) A scalar global measure of performance for $DU_k$, $y_k$, which depends on $\mathbf{z}_k$ and results from the combination on a common basis of operating, quality, and constraint violation costs. This measure is global in the sense that it aggregates together all the operational objectives of the decision unit, but local in the sense that it is limited in scope to $DU_k$ goals.
(c) A vector of temporary propagation loss functions, $\mathbf{y}_{\text{Prop},k}$, used to transmit the requests over the values of connection variables from one decision unit to the ones that precede it, during the backpropagation process that takes place in the construction of a final solution (for details about the procedure adopted to define propagation loss functions, see Saraiva, 1993; Saraiva and Stephanopoulos, 1992d). Once that solution has been found, these propagation loss functions cease to exist.

## 2. *Conventional and Alternative Problem Definitions*

In order to define and compare in a concise way the different problem formulations, let's designate as

$n$ the total number of decision variables, distributed among the $K$ subsystems, $x_{d,j}$, $j = 1, \ldots, n$.

$\mathbf{x}_{\text{DP}}$ a generic pointwise decision policy, i.e., a vector whose components are values of decision variables,

$$\mathbf{x}_{\text{DP}} = [x_{d,1}, \ldots, x_{d,j}, \ldots, x_{d,n}].$$

$\mathbf{X}_{\text{DP}}$ a generic interval vector decision policy, whose components are ranges of decision variables

$$\mathbf{X}_{\text{DP}} = [X_{d,1}, \ldots, X_{d,n}].$$

TABLE IV

CONVENTIONAL APPROACHES AND SUGGESTED ALTERNATIVE

| | Conventional approaches | Suggested alternative |
|---|---|---|
| $\xi$ | $\mathbf{x}_{DP} \in \mathcal{R}^M$ | $\mathbf{X}_{DP} \in \mathbf{I}^M$ |
| $\psi$ | $y_0(\mathbf{x}_{DP})$ or $E(y_0\|\mathbf{x}_{DP})$ | $E(\mathbf{y}_k\|\mathbf{X}_{DP})$ and $\sigma(\mathbf{y}_k\|\mathbf{X}_{DP})$ |
| $f$ | Technique-dependent | Sample averages and standard deviations |
| $S$ | Optimization | Top-down and bottom-up |

$\mathbf{y}_k$ a performance vector whose components are all the unit $k$ performance variables,

$$\mathbf{y}_k = [\mathbf{z}_k, y_k], \qquad k = 0, 1, \ldots, K.$$

The main differences between conventional approaches and our learning architecture are summarized in Table IV, and are discussed below:

1. As final solution formats, interval vector decision policies, $\mathbf{X}_{DP}$, replace their pointwise counterparts, $\mathbf{x}_{DP}$. Thus, a decision policy, $\mathbf{X}_{DP}$, in the context of this section is an interval vector whose components are intervals of decision variables associated with one or more of the infimal decision units. No connection variables or disturbance factors are involved in their definition;

2. As performance criteria, $\psi$, the traditional centralized optimization approach considers an overall objective function, $y_0$, as its single evaluation measure, and the role assigned to the search procedure is the identification of a pointwise decision policy that minimizes $y_0(\mathbf{x}_{DP})$ or $E(y_0|\mathbf{x}_{DP})$. However, any single aggregate measure alone, such as $y_0$, can not provide a complete and meaningful evaluation of performance for a complex plant that includes several subsystems, a large variety of objectives and local performance measurement variables. Although such aggregate measures may be used to facilitate the search for a promising solution, they should not be interpreted as leading to the very best possible answer, neither should the solutions found be implemented without a detailed analysis of how they will affect all subsystems and the achievement of their own local objectives. Thus, in our approach each subsystem is assumed to have its own goals, and these are taken into account explicitly in the definition of the performance criteria, $\psi$, which include the conditional expectations and standard deviations of the several goal achievement measures associated with the multiple subsystems, as

well as the system as a whole:

$$E(\mathbf{y}_k|\mathbf{X}_{DP}) = \left\{E(z_{1,k}|\mathbf{X}_{DP}),\ldots,E\left(z_{p(k),k}|\mathbf{X}_{DP}\right),E(y_k|\mathbf{X}_{DP})\right\}^T,$$

$$k = 0,1,\ldots,K, \quad (38)$$

$$\sigma(\mathbf{y}_k|\mathbf{X}_{DP}) = \left\{\sigma(z_{1,k}|\mathbf{X}_{DP}),\ldots,\sigma\left(z_{p(k),k}|\mathbf{X}_{DP}\right),\sigma(y_k|\mathbf{X}_{DP})\right\}^T,$$

$$k = 0,1,\ldots,K. \quad (39)$$

3. Traditional techniques use $f(\mathbf{x}_{DP})$ quantitative models to perform the mapping from the solution to the performance space. These models essentially reduce the problem to a simple system, having the decision variables $x_{d,j}$ as inputs and $y_0$ as output, because they allow one to express the $y_0(\mathbf{x}_{DP})$ or $E(y_0|\mathbf{x}_{DP})$ estimates as a function of the decision variables, $\psi^{est} = f(\mathbf{x}_{DP})$. On the other hand, our mapping procedures, $f$, are based on the construction of direct sampling estimates. Given a decision policy, $\mathbf{X}_{DP}$, both $E(\mathbf{y}_k|\mathbf{X}_{DP})$ and $\sigma(\mathbf{y}_k|\mathbf{X}_{DP})$ are estimated by identifying the available data records for which $\mathbf{x} \in \mathbf{X}_{DP}$, and computing the corresponding sample averages and standard deviations for all the $z_{i,k}$ and $y_k$ variables.

4. In conventional centralized approaches the goal of the search procedures is to find a final feasible decision policy, $\mathbf{x}^*_{DP}$, that optimizes $f(\mathbf{x}_{DP})$. This solution is then imposed to the several subsystems for implementation, although these were not directly involved in its construction. Other similar methodologies include problem decomposition strategies (Lasdon, 1970; Biegler, 1992) that explore particular properties of the complex system structure and dynamic programming (Bellman and Dreyfus, 1962; Roberts, 1964; Nemhauser, 1966). They share the same basic assumptions, problem statement and solution formats as centralized approaches, although, for the sake of computational efficiency gains, a multistage and sequential identification of the final result, $\mathbf{x}^*_{DP}$, is adopted as the problem solving strategy.

Our search procedures represent a departure from the above type of paradigm. Rather than simply accepting and implementing a decision policy found by $DU_0$, that optimizes an overall measure of performance, the infimal subsystems and corresponding plant personnel play an active role in the construction and validation of solutions. One tries to build a consensus decision policy, $\mathbf{X}_{DP}$, validated by all subsystems, $DU_k$, $k = 1,\ldots,K$, as well as by the whole plant, $DU_0$, and only when that consensus has been reached does one move toward implementation. Within this context, the upper-level decision unit, $DU_0$, assumes a coordination role,

and does not have the power to impose solutions to the several subsystems. Two different search procedures, used to find such consensus decision policies, $\mathbf{X}_{DP}$, and designated respectively as *bottom–up* and *top–down*, are described in the following paragraphs.

### 3. *Final Problem Statement*

The ultimate goal of our learning architecture is to uncover at least one decision policy, $\mathbf{X}_{DP}$, that

(a) Is feasible, i.e., satisfies constraints imposed over the minimum acceptable size of operating windows $[w(X_{d,j}) \geq \Delta x_{d,j},\ j = 1, \ldots, n]$ and coverage $[n(\mathbf{X}_{DP}) \geq N_{min}]$.
(b) Leads to a significant improvement over the current levels of performance for one or more of the decision units.
(c) Is accepted and validated by all decision units involved with or affected by the use of $\mathbf{X}_{DP}$ as the zone to conduct the operation.

A decision policy that satisfies all these requirements is designated as an active decision policy, $\mathbf{X}^*_{DP}$.

To declare a decision policy, $\mathbf{X}_{DP}$, as either unacceptable, acceptable or leading to a significant improvement, each decision unit compares its current levels of performance with the ones that are expected within $\mathbf{X}_{DP}$. The current levels of performance for unit $k$ are provided by the $E(\mathbf{y}_k|\mathbf{X}_{current})$ and $\sigma(\mathbf{y}_k|\mathbf{X}_{current})$ estimates obtained from a sample of data records, just as in the case of $E(\mathbf{y}_k|\mathbf{X}_{DP})$ and $\sigma(\mathbf{y}_k|\mathbf{X}_{DP})$. A comparison of these reference performance values for $\mathbf{X}_{current}$ with the ones achieved by $\mathbf{X}_{DP}$ is made by each decision unit. As a result of this comparison, a final evaluation of $\mathbf{X}_{DP}$ by decision unit $k$ leads to one of the three possible outcomes:

1. $f_k(\mathbf{X}_{DP}) \gg f_k(\mathbf{X}_{current})$, meaning that significant improvement is expected.
2. $f_k(\mathbf{X}_{DP}) \approx f_k(\mathbf{X}_{current})$, in case $\mathbf{X}_{DP}$ is accepted and validated by decision unit $k$, although no significant improvements are expected.
3. $f_k(\mathbf{X}_{DP}) \ll f_k(\mathbf{X}_{current})$, meaning that $\mathbf{X}_{DP}$ can not be accepted by unit $k$, because it would result in performance deterioration down to levels that fall below what the unit can tolerate for at least one of its performance variables.

Thus, an active decision policy, $\mathbf{X}^*_{DP}$, is a feasible decision policy such that

$$\exists_{k \in \{0,1,\ldots,K\}} \mid f_k(\mathbf{X}_{DP}) \gg f_k(\mathbf{X}_{current});$$

$$\forall_{k \in \{0,1,\ldots,K\}} \{f_k(\mathbf{X}_{DP}) \approx f_k(\mathbf{X}_{current}) \vee f_k(\mathbf{X}_{DP}) \gg f_k(\mathbf{X}_{current})\}.$$

## B. Search Procedures

Two different search procedures (bottom–up and top–down) can be followed to build active decision policies, $\mathbf{X}^*_{DP}$.

In the bottom–up approach the initiative to start the learning process is taken by one of the infimal decision units. Since solutions found at this unit may include connection variables, the request for given values of these variables is propagated backward, to unit $k+1$, through temporary loss functions. After successive backpropagation steps, the participation of several other $DU_k$ and the operators associated with them, a final decision policy, accepted and validated by all infimal decision units, is eventually found. Then, this policy is brought to the attention of the supremal decision unit, $DU_0$, who is responsible for detecting whether it leads to an improved performance of the system as a whole. If so, the uncovered policy is an active decision policy, and one can proceed with its implementation.

On the other hand, the top–down approach starts the learning process at the supremal decision unit, $DU_0$, and only on a second stage does it move down to the infimal decision units for approval and validation.

The next paragraphs provide a brief description of both the bottom–up and top–down search procedures (for further details, see Saraiva, 1993; Saraiva and Stephanopoulos, 1992d).

### *1. Bottom–Up Approach*

The bottom–up approach contains two distinct stages. First, by successive backpropagation steps one builds a decision policy. Then, this uncovered policy is evaluated and refined, and its expected benefits confirmed before any implementation actually takes place. This two-stage process is conceptually similar to dynamic programming solution strategies, where first a decision policy is constructed by backward induction, and then one finds a realization of the process for the given policy, in order to check its expected performance (Bradley *et al.*, 1977).

*a. Decision Policy Construction.* Learning activities may be initiated by any of the infimal decision units, $DU_k$, to which one applies the ***basic*** learning methodology introduced in Sections V and VI, leading to the identification of a particular final solution, $\mathbf{X}_k$.

If $\mathbf{X}_k$ involves only ranges of decision variables attached to unit $k$, $x_{d,k}$, it defines a decision policy, and thus one can move directly to the validation and refinement phase.

However, $\mathbf{X}_k$, besides decision variables, may also include ranges of connection variables, $x_{i,k}$, that link units $k+1$ and $k$. That being the

case, the learning process has to be propagated backward, toward decision unit $DU_{k+1}$. To induce this propagation, one has first to identify temporary loss functions (Saraiva, 1993; Saraiva and Stephanopoulos, 1992d) for all connection variables present in the definition of $\mathbf{X}_k$. These temporary propagation loss functions are combined with other $DU_{k+1}$ goals, and ***basic*** is now applied to decision unit $DU_{k+1}$. Further propagations, from unit $k+1$ to unit $k+2, k+2$ to $k+3$, etc., are identical to the one that occurred from $DU_k$ to $DU_{k+1}$. This backpropagation through different decision units continues until one eventually reaches an infimal decision unit, $DU_{k+j}$, where a solution, $\mathbf{X}_{k+j}$, involving only ranges of decision variables, is found. When that happens, a final decision policy, $\mathbf{X}_{DP}$, can be immediately constructed by simply assembling all of its pieces together: it is the conjunction of the decision variable intervals, distributed among several decision units, that were uncovered during the upstream propagation process.

*b. Validation and Refinement.* Because the construction of $\mathbf{X}_{DP}$ resulted from the contributions of multiple infimal decision units, taking into consideration their specific goals and imposed constraints, it may be already an active decision policy. However, and even if that is the case, before proceeding to any implementation, it is necessary to evaluate the benefits that would derive from operating within $\mathbf{X}_{DP}$.

The process of $\mathbf{X}_{DP}$ validation and refinement starts with a final detailed analysis of its realization, through the computation of $E(\mathbf{y}_k|\mathbf{X}_{DP})$ and $\sigma(\mathbf{y}_k|\mathbf{X}_{DP})$, $k = 0, 1, \ldots, K$, estimates. If for some reason one or more infimal decision units judge the implementation of $\mathbf{X}_{DP}$ to be unacceptable, an additional attempt is made to refine the decision policy and find a revised version of it, $\mathbf{X}_{DP}^{final}$, that is accepted and validated by all infimal decision units, through additional applications of ***basic*** to the decision units in conflict with the initial $\mathbf{X}_{DP}$ version (see Saraiva, 1993; Saraiva and Stephanopoulos, 1992d, for details).

Once agreement and consensus have been reached by all infimal decision units, for $\mathbf{X}_{DP}^{final}$ to be declared an active decision policy, it is necessary to bring it to the attention of $DU_0$. The effects of a possible implementation of $\mathbf{X}_{DP}^{final}$ on the system as a whole are examined, to check that it also translates into improved performance from a global perspective. If this final validation test is passed, $\mathbf{X}_{DP}^{final}$ represents an active decision policy, and one can proceed to its implementation.

### 2. *Top–Down Approach*

In the top–down approach the supremal decision unit, $DU_0$, starts the learning process by itself, and identifies a decision policy, $\mathbf{X}_{DP}$. Then, in a

second stage, one moves down to the infimal subsystems, seeking their support and validation for $\mathbf{X}_{DP}$.

Let's assume that the inputs to the supremal decision unit are a subset of all the decision variables attached to infimal decision units, consisting of those $x_{d,k}$ variables that are believed to be particularly influential with respect to the operation of the overall system. Then, an application of ***basic*** to $DU_0$ results directly in the identification of a decision policy, $\mathbf{X}_{DP}$. This decision policy is then passed down to the lower level in the hierarchy, where it is submitted to a process of validation and refinement by all infimal decision units that is identical to the one that takes place in the bottom–up approach.

## C. Case Study: Operational Analysis of a Pulp Plant

The overall system that we will analyze comprises the unbleached Kraft pulp line, chemicals and energy recovery zones of a specific paper mill (Melville and Williams, 1977). We will employ a somewhat simplified but still realistic representation of the plant, originally developed in a series of research projects at Purdue University (Adler and Goodson, 1972; Foster *et al.*, 1973; Melville and Williams, 1977). The records of simulated operation data, used to support the application of our learning architecture, were generated by a reimplementation, with only minor changes, of steady-state models (for each individual module and the system as a

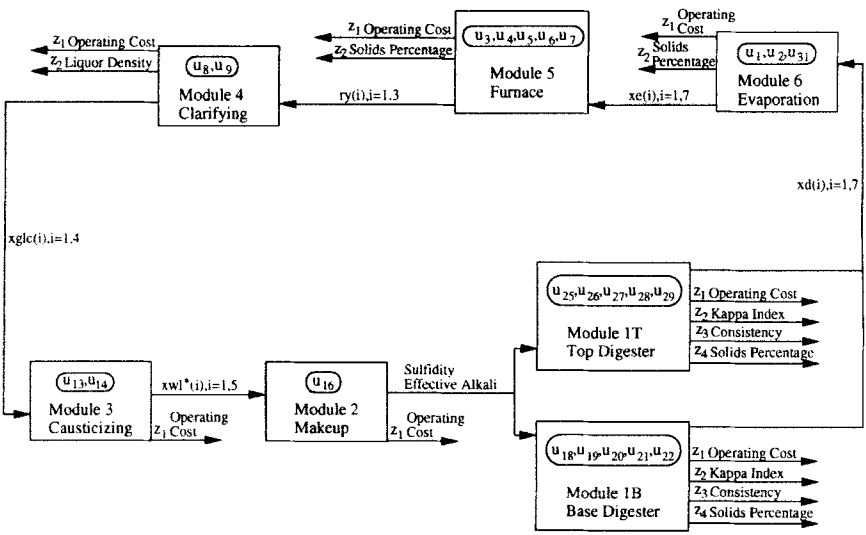


FIG. 12. Plant internal structure and modular representation.

whole) presented in the references cited above. A more detailed description of this case study can be found in Saraiva (1993), or Saraiva and Stephanopoulos (1992d).

The global structure of a Kraft pulp plant was illustrated in Fig. 9. The corresponding modular representation adopted in this study is shown schematically in Fig. 12. It includes 7 infimal decision units, with a total of 23 different decision variables distributed among them [to keep the notation consistent with Melville and Williams (1977), decision variables are designated as $u_d$]. A complete list of these variables, including their physical meaning, measurement units and ranges of values in $\mathbf{X}_{\text{current}}$, is provided in Table V. There are also 28 connection variables, linking successive infimal subsystems, and 16 local performance variables, $z_{i,k}$. All the performance measures (operating costs, quality losses, penalty functions) are expressed on a common basis of U.S. dollars per ton of air-dried pulp produced ($/ TADP).

TABLE V

LIST OF ALL 23 DECISION VARIABLES AND CORRESPONDING WINDOWS UNDER CURRENT OPERATING CONDITIONS

| $u_d$ | $u_d$ description | Units | Range |
|---|---|---|---|
| $u_1$ | Oxidation tower efficiency | Adimensional | [0.0; 0.9] |
| $u_2$ | Steam flow to evaporators | lb/h | $[0.9 \cdot 10^5; 1.25 \cdot 10^5]$ |
| $u_3$ | Sodium sulfate addition | lb/h | [2000; 4000] |
| $u_4$ | Black liquor temperature at nozzles | °F | [245; 250] |
| $u_5$ | Primary airflow to furnace | lb/h | $[3 \cdot 10^5; 4 \cdot 10^5]$ |
| $u_6$ | Secondary airflow to furnace | lb/h | $[1.75 \cdot 10^5; 2.75 \cdot 10^5]$ |
| $u_7$ | Furnace flue gas temperature | °F | [550; 750] |
| $u_8$ | Steam/condensate added to smelt tank | lb/TADP | [4000; 5000] |
| $u_9$ | Washing water flow to dregs filter | lb/TADP | [0.0; 75.0] |
| $u_{13}$ | Water fraction in lime mud slurry | Adimensional | [0.3; 0.4] |
| $u_{14}$ | Water spray flow to white liquor filter | lb/TADP | [1000; 2000] |
| $u_{16}$ | White liquor effective alkali | lb/gal | [0.8; 0.9] |
| $u_{18}$ | Lower heater temperature | °F | [295; 310] |
| $u_{19}$ | Digester blow flow | gpm[a] | [1100; 1250] |
| $u_{20}$ | Washing circulation temperature | °F | [225; 280] |
| $u_{21}$ | Black liquor extraction flow | gpm | [1150; 1250] |
| $u_{22}$ | White liquor added to digester | gpm | [400; 500] |
| $u_{25}$ | Lower heater temperature | °F | [295; 308] |
| $u_{26}$ | Digester blow flow | gpm | [800; 950] |
| $u_{27}$ | Washing circulation temperature | °F | [225; 280] |
| $u_{28}$ | Black liquor extraction flow | gpm | [950; 1150] |
| $u_{29}$ | White liquor added to digester | gpm | [400; 500] |
| $u_{31}$ | Steam flow to flash system | lb/h | [0; 10000] |

[a]Gallons per minute.

*1. Top–Down Approach*

When examined at the supremal level, the system's primary goal is the production of pulp with the desired kappa indices. Consequently, as $DU_0$ performance variables we will consider the following:

(a) $z_{1,0}$, total operating cost, which is the sum of all the 7 subsystems operating costs, $z_{1,k}$, $k = 1,\ldots,7$
(b) $z_{2,0}$, kappa index of the pulp produced at the base digester
(c) $z_{3,0}$, kappa index of the pulp produced at the top digester

Furthermore, we will consider the 23 different decision variables, $u_d$, as the $DU_0$ inputs.

Both operating costs and quality losses were combined together, leading to the following overall performance measure, $y_0$, for the supremal decision unit:

$$y_0 = z_{1,0} + (z_{2,0} - 100)^2 + 5(z_{3,0} - 60)^2. \tag{40}$$

A preliminary analysis of the available $DU_0$ data records showed that $y_0$ is currently dominated by the behavior of $z_{3,0}$, and its deviations from the target. After going through the search procedure described in Section VII, the following final active decision policy, $\mathbf{X}_{DP}^{final}$, was identified:

$$\mathbf{X}_{DP}^{final} = \{u_2 \in [112{,}000; 122{,}000] \wedge u_3 \in [2060; 2700] \wedge u_4 \in [245.3; 247.3] \wedge u_{25} \in [304.2; 307.8]\}. \tag{41}$$

In Table VI we compare the performance achieved through the implementation of the above strategy with that defined by the current values. An implementation of $\mathbf{X}_{DP}^{final}$ results in a significant decrease of the $y_0$ average, primarily as a consequence of a reduction in the average cost associated with the operation of the top digester, $y_{1T}$. On its own hand, the $y_{1T}$ average decrease derives from the fact that $\mathbf{X}_{DP}^{final}$ centers the average kappa index of the top-digester pulp much closer to its target of 60, while also reducing its standard deviation. All these results are consistent with the observation made earlier that pointed to $z_{3,0}$ as the key performance variable that conditions the current levels of overall system performance, $E(y_0|\mathbf{X}_{current})$.

*2. Bottom–Up Approach*

The learning process was initiated at the top-digester infimal decision unit, leading to a solution, $\mathbf{X}_{1T}$, that involves local decision variables and a range of white liquor sulfidity (fraction of active reactants in the white

TABLE VI

COMPARISON OF PERFORMANCE MEASURES BEFORE AND AFTER IMPLEMENTATION OF $\mathbf{X}_{DP}^{final}$

| | Average ($\mathbf{X}_{current}$) | Average ($\mathbf{X}_{DP}^{final}$) |
|---|---|---|
| $y_0$ | 1786.5 | 895.6 |
| $y_{1T}$ | 1500.6 | 695.3 |
| $y_{1B}$ | 281.5 | 187.8 |
| $y_2$ | 8.8 | 9.2 |
| $y_3$ | 7.4 | 7.8 |
| $y_4$ | 185.6 | 168.8 |
| $y_5$ | 120.0 | 50.1 |
| $y_6$ | 5.8 | 4.8 |
| Kappa index (top) | 67.4 | 59.9 |
| Kappa index (base) | 102.9 | 102.0 |

liquor that are present as $HS^-$ rather than as $OH^-$) values. Successive upstream propagations of this request had to be performed before a solution involving only decision variables, $\mathbf{X}_5$, was found at the furnace module, thus leading to the identification of a decision policy, $\mathbf{X}_{DP}$, which consists of the conjunction of all the decision variable ranges identified up to that point.

It is worth noticing that the path that was followed in the construction of $\mathbf{X}_{DP}$ is coherent and logical with respect to a physical understanding of the plant. It was found at the top digester that one of the most important variables, conditioning the location and dispersion of the pulp kappa index, is the amount of sulfur present in the white liquor added to the digester. But sodium sulfate enters the plant to compensate for sulfur losses at the recovery furnace, and thus it is basically within this infimal decision unit that the levels of sulfidity can be adjusted. Accordingly, the backpropagation process involved requests over the values of sulfur flows in several intermediate streams, and it stopped only at the furnace module, where the decision policy construction was concluded, leading to a $\mathbf{X}_{DP}$ that involves the amount of sodium sulfate added to the furnace.

After submitting $\mathbf{X}_{DP}$ through the validation and refinement stage, the final uncovered active decision policy, $\mathbf{X}_{DP}^{final}$, was given by

$$\mathbf{X}_{DP}^{final} = \{u_3 \in [3200;3635] \wedge u_4 \in [247.5;248.7] \wedge u_8 \in [4800;5000] \wedge u_{18} \in [306,309] \wedge u_{25} \in [304.5;308.0] \wedge u_{29} \in [460.0;490.0]\} \quad (42)$$

Table VII summarizes the levels of performance that are achieved within $\mathbf{X}_{DP}^{final}$, and compares them with the performance corresponding to the current values.

TABLE VII

COMPARISON OF PERFORMANCE MEASURES BEFORE AND AFTER IMPLEMENTATION OF $\mathbf{X}_{DP}^{final}$

| | Average ($\mathbf{X}_{current}$) | Average ($\mathbf{X}_{DP}^{final}$) |
|---|---|---|
| $y_0$ | 1786.5 | 489.1 |
| $y_{1T}$ | 1500.6 | 387.4 |
| $y_{1B}$ | 281.5 | 89.2 |
| $y_2$ | 8.8 | 11.8 |
| $y_3$ | 7.4 | 8.6 |
| $y_4$ | 185.6 | 14.9 |
| $y_5$ | 120.0 | 91.7 |
| $y_6$ | 5.8 | 4.9 |
| Kappa index (top) | 67.4 | 60.7 |
| Kappa index (base) | 102.9 | 99.5 |

### 3. *Brief Comparison of Final Decision Policies*

In the previous paragraphs it was shown how by following the top–down and bottom–up approaches one arrived at the construction of two distinct and promising decision policies, Eqs. (41) and (42). Both of these decision policies include intervals of values for certain critical decision variables (e.g., $u_3, u_4, u_{25}$). Since there is some consistency between the infimal and supremal decision unit goals, this communality of variables should be expected. However, different infimal decision unit local goals were taken into account in the construction of these policies, and choices among several possible solutions were made by the user during that construction. Thus, it does not come as a surprise that the two final decision policies also involve ranges of different decision variables. Similarly, the performances achieved are not entirely identical.

A final comparison of the results obtained with each policy, as well as under the current operating conditions, is given in Fig. 13. Both the top–down and the bottom–up approaches uncovered promising decision policies, which lead to considerable improvements over the current plant performance levels. When one compares in a more detailed way the differences in infimal decision unit performances for the two policies, one can see that they reflect the participation of the corresponding subsystems in their construction. Finally, it should be added that, besides performance related issues, the upstream propagation associated with the bottom–up approach provided some important insight into understanding the main causes of dispersion and location for the top-digester pulp kappa index. Specifically, it was uncovered that the kappa index is highly dependent on

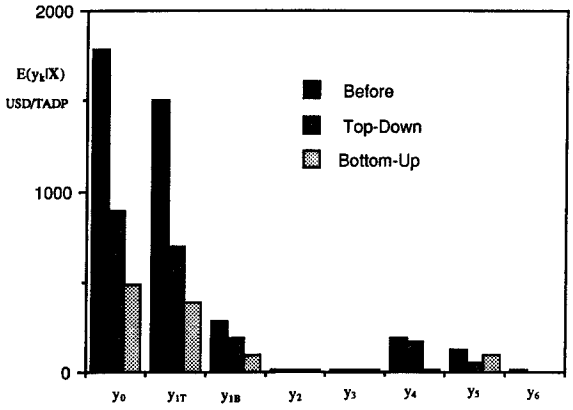


FIG. 13. Final comparison of performances for both decision policies.

the white liquor sulfidity, a relationship translated later on into the definition of intervals for decision variables associated with the recovery boiler, which is the module where one has the ability to manipulate the amounts of sulfur present in all the plant liquor streams. This type of knowledge would have been hard or impossible to acquire by merely examining the final decision policies or through an analysis of the top–down approach alone.

## VIII. Summary and Conclusions

In this chapter we revisited an old problem, namely, exploring the information provided by a set of $(\mathbf{x}, y)$ operation data records and learn from it how to improve the behavior of the performance variable, $y$. Although some of the ideas and methodologies presented can be applied to other types of situations, we defined as our primary target an analysis at the supervisory control level of $(\mathbf{x}, y)$ data, generated by systems that cannot be described effectively through first-principles models, and whose performance depends to a large extent on quality-related issues and measurements.

We have introduced modified statements and solution formats for the preceding problem, with hyperrectangles in the decision space replacing the conventional pointwise results. The advantages and implications of adopting this alternative language to express final solutions were discussed, and it was also shown that traditional formulations can be interpreted as a particular degenerate case of the suggested more general

problem definitions, where one searches for ranges of decision variables rather than single values.

To address the modified problem statements and uncover final solutions with the desired alternative formats, data-driven nonparametric learning methodologies, based on direct sampling approaches, were described. They require far fewer assumptions and a priori decisions on the part of the user than most conventional techniques. These practical frameworks for extracting knowledge from operating data present the final uncovered solutions to the decisionmaker in formats that are both easy to understand and implement.

We presented extensions and variations of the basic learning methodologies aimed at enlarging their flexibility and cover a number of different situations, including systems where performance is evaluated by categorical or continuous variables, with single or multiple objectives, simple or complex plants containing some type of internal structure and composed of a number of interconnected subsystems.

The potential practical capabilities of the described learning methodologies, and their attractive implementational features from an industrial point of view, were illustrated through the presentation of a series of case studies with both real-world industrial and simulated operating data.

## Acknowledgments

The author would like to acknowledge financial an other types of support received from the Leaders for Manufacturing program at MIT, Fulbright Program, Rotary Foundation, Comissão Permanente da INVOTAN, Fundação Luso-Americana para o Desenvolvimento, and Comissão Cultural Luso-Americana. Special thanks also to Professor George Stephanopoulos, who always provided the right amount of support and guidance, while at the same time allowing me to have all the freedom that I needed to pursue my own research dreams and try to convert them into reality.

## References

Adler, L., and Goodson, R., "An Economic Optimization of a Kraft Pulping Process," Laboratory for Applied Industrial Control, Report 48. Purdue University, West Lafayette, IN, 1972.

Alefeld, G., and Herzberger, J., "Introduction to Interval Computations." Academic Press, New York, 1983.

Bellman, R., and Dreyfus, S., "Applied Dynamic Programming." Princeton University Press, Princeton, NJ, 1962.

Biegler, L., Optimization strategies for complex process models. *Adv. Chem. Eng.* **18**, 197 (1992).

Bradley, S., *et al.*, "Applied Mathematical Programming." Addison-Wesley, Reading, MA, 1977.

Breiman, L., *et al.*, "Classification and Regression Trees." Wadsworth, Belmont, CA, 1984.

Clausing, D., "Total Quality Development." ASME Press, New York, 1993.

Daniel, C., and Wood, F., "Fitting Equations to Data." 2nd ed. Wiley, New York, 1980.

Deming, W., "Out of the Crisis." Massachusetts Institute of Technology, Center for Advanced Engineering Study, Cambridge, MA, 1986.

Duda, R., and Hart, P., "Pattern Classification and Scene Analysis." Wiley, New York, 1973.

Ellingsen, W., Implementation of advanced control systems. *AIChE Symp. Ser.* **159**, 150 (1976).

Findeisen, W., *et al.*, "Control and Coordination in Hierarchical Systems." Wiley, New York, 1980.

Foster, R., *et al.*, "Optimization of the Chemical Recovery Cycle of the Kraft Pulping Process," Laboratory for Applied Industrial Control, Report 54. Purdue University, West Lafayette, IN, 1973.

Fu, K., "Sequential Methods in Pattern Recognition and Machine Learning." Academic Press, New York, 1968.

Gaines, B., The trade-off between knowledge and data in knowledge acquisition. *In* "Knowledge Discovery in Databases" (G. Shapiro and W. Frawley, eds.), p. 491. MIT Press, Cambridge, MA, 1991.

Garcia, C., and Prett, D., Advances in industrial model-predictive control. *In* "Chemical Process Control, CPC-III." (Morari, M. and McAvoy, T. J., eds.). CACHE-Elsevier, New York, 1986.

Goodman, R., and Smyth, P., Decision tree design using information theory. *Knowl. Acquis.* **2**, 1 (1990).

Hayes, R., *et al.*, "Dynamic Manufacturing: Creating the Learning Organization." Free Press, New York, 1988.

Hunt, E., "Concept Learning: An Information Processing Problem." Wiley, New York, 1962.

James, M., "Classification Algorithms." Wiley, New York, 1985.

Juran, J., "Managerial Breakthrough." McGraw-Hill, New York, 1964.

Klein, J., "Revitalizing Manufacturing." R.D. Irwin, Homewood, IL, 1990.

Kodratoff, Y., and Michalski, R., eds., "Machine Learning: An Artificial Intelligence Approach." Vol. 3. Morgan Kaufmann, San Mateo, CA, 1990.

Lasdon, L., "Optimization Theory for Large Systems." Macmillan, New York, 1970.

Lasdon, L., and Baker, T., The integration of planning, scheduling and process control. *In* "Chemical Process Control, CPC-III." (Morari, M. and McAvoy, T. J., eds.). CACHE-Elsevier, New York, 1986.

Latour, P., Comments on assessment and needs. *AIChE Symp. Ser.* **159**, 161 (1976).

Latour, P., Use of steady-state optimization for computer control in the process industries. *In* "On-line Optimization Techniques in Industrial Control" (Kompass, E. J. and Williams, T. J., eds.). Technical Publishing Company, 1979.

Launks, U., *et al.*, On-line optimization of an ethylene plant. *Comput. Chem. Eng.* **16**, S213 (1992).

Melville, S., and Williams, T., "Application of Economic Optimization to the Chemical Recovery System of a Kraft Pulping Process," Laboratory for Applied Industrial Control, Report 107. Purdue University, West Lafayette, IN, 1977.

Mesarović, M., *et al.*, "Theory of Hierarchical, Multilevel Systems." Academic Press, New York, 1970.

Moore, R., "Methods and Applications of Interval Analysis." SIAM, Philadelphia, 1979.
Moore, R., What and who is in control. *In* "The Second Shell Process Control Workshop" (D.M. Prett, C.E. Garcia, and B.L. Ramaker, eds.). Butterworth, Stoneham, MA, 1990.
Moret, B., Decision trees and diagrams. *ACM Comput. Surv.* **14**(4), 593 (1982).
National Research Council, "The Competitive Edge. "National Academy Press, Washington, DC, 1991.
Nemhauser, G., "Introduction to Dynamic Programming." Wiley, New York, 1966.
Phadke, M., "Quality Engineering Using Robust Design." Prentice Hall, Englewood Cliffs, NJ, 1989.
Quinlan, J., Induction of decision trees. *Mach. Learn.* **1**, 81 (1986).
Quinlan, J., Simplifying decision trees. *Int. J. Man-Mach. Stud.* **27**, 221 (1987).
Quinlan, J., "C4.5: Programs for Machine Learning." Morgan Kaufmann, San Mateo, CA, 1993.
Reece, J., Daniel, D. and Bloom R., Identifying a plasma etch process window. *In* "Understanding Industrial Designed Experiments" (Schmidt S. and Launsby R., eds.), 2nd ed. AIR Academy Press, Colorado Springs, CO, 1989.
Roberts, S., "Dynamic Programming in Chemical Engineering." Academic Press, New York, 1964.
Roy, R., "A Primer on the Taguchi Method." Van Nostrand-Reinhold, Princeton, NJ, 1990.
Saraiva, P., Data-driven learning frameworks for continuous process analysis and improvement. Ph.D. Thesis, Massachusetts Institute of Technology, Dept. Chem. Eng., Cambridge, MA, 1993.
Saraiva, P., and Stephanopoulos, G., Continuous process improvement through inductive and analogical learning. *AIChE J.* **38**(2), 161 (1992a).
Saraiva, P., and Stephanopoulos, G., "Learning to Improve Processes with Multiple Pattern Recognition Objectives," Working paper. Massachusetts Institute of Technology, Dept. Chem. Eng., Cambridge, MA, 1992b.
Saraiva, P., and Stephanopoulos, G., "An Exploratory Data Analysis Robust Optimization Approach to Continuous Process Improvement," Working paper. Massachusetts Institute of Technology, Dept. Chem. Eng., Cambridge, MA, 1992c.
Saraiva, P., and Stephanopoulos, G., "Data-Driven Learning Architectures for Process Improvement in Complex Systems with Internal Structure," Working paper. Massachusetts Institute of Technology, Dept. Chem. Eng., Cambridge, MA, 1992d.
Sargent, R., The future of digital computer based industrial control systems. *In* "Industrial Computing Control After 25 Years" (Kompass, E. J. and Williams, T. J., eds.), p. 63 Technical Publishing Company, 1984.
Senge, P., "The Fifth Discipline." Currency, 1990.
Shannon, C., and Weaver, W., "The Mathematical Theory of Communication," 11th printing. University of Illinois Press, Urbana, 1964.
Shapiro, G., and Frawley, W., eds., "Knowledge Discovery in Databases." MIT Press Cambridge, MA, 1991.
Shavlik, J., and Dietterich, T., eds., "Readings in Machine Learning." Morgan Kaufmann San Mateo, CA, 1990.
Sheridan, T., "45 Years of Man-Machine Systems." Massachusetts Institute of Technology Dept. Mech. Eng., Cambridge, MA, 1985.
Shiba, S., *et al.*, "The Four Revolutions of Management Thinking: Planning and Implementation of TQM for Executives." Productivity Press, 1993.
Sinnar, R., Impact of model uncertainties and nonlinearities on modern controller design *In* "Chemical Process Control, CPC-III." (Morari, M. and McAvoy, T. J., eds.), p. 53 CACHE-Elsevier, 1986.

Sonquist, J., *et al.*, "Searching for Structure." University of Michigan, Ann Arbor, Michigan, 1971.

Taguchi, G., "Introduction to Quality Engineering." Asian Productivity Association, 1986.

Taylor, W., "What Every Engineer Should Know About Artificial Intelligence." MIT Press, Cambridge, MA, 1989.

Taylor, W., "Optimization and Variation Reduction in Quality." McGraw-Hill, New York, 1991.

Tukey, J., "Exploratory Data Analysis." Addison-Wesley, Reading, MA, 1977.

Turban, E., "Decision Support and Expert Systems." Macmillan, New York, 1988.

Utgoff, P., Perception trees: A case study in hybrid concept representations. *In* "Proceedings of AAAI88," Vol. 2, p. 601. Morgan Kaufmann, San Mateo, CA, 1988.

Winston, P., "Artificial Intelligence," 2nd ed. Addison-Wesley, Reading, MA, 1984.