Crystallization Process Optimization via a Revised Machine Learning Methodology

Ahmad Y. Sheikh and Alan G. Jones
Dept. of Chemical Engineering, University College London, London WC1E 7JE, U.K.

A revised machine learning methodology is applied to a simulated crystallization process flowsheet for continual improvement of its performance by generating and analyzing process data. The aim is to identify bands of crucial decision variables leading to zones of best average process performance. The methodology comprises two components: symbolic induction and case-based reasoning. It uses an incremental algorithm to update performance classification rules, which not only improves the efficiency of the symbolic induction of the classification rules by eliminating the need for their periodic reinduction, but also simplifies the case-based reasoning step. The new concepts and procedures are illustrated by application to a potassium nitrate crystallization process comprising a mixed-suspension, mixed-product-removal crystallizer, a hydrocyclone, and a fines dissolver. By identifying and establishing ranges for the most crucial decision variables, namely, feed concentration, flow rate, and cooling stream temperature, three zones leading to an improvement of nearly 12% on nominal average performance are detected within two generations of the classification rules.

Introduction

Process operational quality, which has emerged as an essential precondition to increase profitability by fundamentally improving the design and operation of the process, involves two complementary steps: (1) control within prespecified limits, and (2) continuous improvement of operational performance (Saraiva and Stephanopoulos, 1992a). The first step deals with the rectification of abnormal process behavior (as a result of special causes) through efficient control. The final level of performance thereby achieved, however, is a result of common and sustained causes within the process itself that are avoidable. It has been estimated that only 6–30% of production problems are due to special causes, while the remaining 70–94% are due to common and sustained causes (Hart and Hart, 1989). The magnitude of common-cause contributions can only be reduced through the introduction of appropriate changes in operating conditions and strategies by searching for better levels and ranges of decision variables.

Crystallization processes have seldom been subjected to process improvement techniques at the operational level; rather process improvement studies have been restricted to the design level through simple "design-and-cost" relationships (Rossiter and Douglas, 1986; Jones, 1991), for they are characterized by mathematical complexities associated with the adequate representation of crystal size distribution (CSD) and functional discontinuities in kinetic processes, beyond current optimization algorithms (Cuthrell and Biegler, 1987; Biegler et al., 1995). These calculations require a thorough understanding of the process so that it can be effectively represented in simplified mathematical form to find the best solution to a process within the imposed constraints.

Another sound basis for the prediction of better levels of decision variables for performance improvement is from extrapolation of past known cases. Multivariate statistical techniques, such as principal-components analysis, partial least squares, factor analysis, or neural networks, provide a wide range of tools for this purpose, usually based on the concept of fitting a particular class of models to the data and then hypothesizing that the solution for future levels of decision variables will conform to the fitted model. All of these methods formulate a final solution consisting of a vector comprising the decision variables $(x_1, \ldots, x_n)$ that defines a single best performance point (y) in the performance space. Most decision variables, however, have some variability associated with them, even under the best control schemes. As a conse-

Correspondence concerning this article should be addressed to A. G. Jones.

quence of ignoring that processes operate in narrow bands of decision variables, the final solution obtained by these approaches may be suboptimal, even when accurate models are available, since their evaluation criterion ignores the system behavior around the optimal. The area surrounding such a point does not in general correspond to the zone where best average performance can be achieved (Saraiva, 1995).

Machine learning, the study and computer modeling of learning processes in their multiple manifestations, has been used for the similar task of developing and analyzing systems to improve performance from existing data, often from a less model-driven standpoint (Saraiva and Stephanopoulos, 1992a; Saraiva, 1995). The essence of problem formulation with this approach is one in which a procedure, shown a set of process data (x, y) comprising quantitative and or qualitative features of the process, employs inductive inference to extract classification rules for the division of decision space into hyperrectangles (not points) representing different levels of performance without losing the individuality of each decision variable. The procedures that discover these rules in the form of decision trees are the most mature and widely used of all the interval analysis-based rules representations. These trees are developed through top-down, divide-and-conquer strategy that successively partitions the given set of data into smaller and smaller subsets with the growth of the tree (Quinlan, 1990). The symbolic induction is based on a direct sampling approach where random data sets (objects) are used to build confidence intervals for performance levels, and therefore does not suffer from simplifying assumptions and numerical inaccuracies inherent to mathematical formulation in traditional optimization procedures. The method relies on the quality of data, as for quantity Saraiva (1995) has shown that even with moderate amounts of data it is possible to construct trees resulting in significant performance improvements. Furthermore, since the trees exhibit explicit ranges of decision variables and associated levels of performance, they can lead either directly to changes in current operation practices or to the design of a set of confirmatory experiments for validating the findings. The implementation of these suggestions would generate more novel data capable of providing operating schemes for the extension of process performance beyond currently achieved levels. These schemes are identified and passed on to the symbolic induction stage for updating classification rules by the case-based reasoning component. Crystallization processes will therefore be best served by machine learning methodology, which offers a flexible and mathematically deconvoluted procedure identifying performance improvement zones by establishing ranges on the crucial decision variables. A thorough comparison of machine learning methodology and other multivariate statistical methods can be found in Saraiva and Stephanopoulos (1992b).

In this article, we present a procedure for performance improvement of a simulated potassium nitrate crystallization process using machine learning methodology after first modifying it. Initially, however, we develop sufficiently complex population-balance-based models for crystallization process unit operations within the conventional framework of process modeling. In the absence of the required process plant data and lack of rigorous flow-sheeting models in the literature, the process will be simulated using these new models under ranges of decision variables to generate the necessary data. In the next section, we explain the modifications made to the existing machine learning methodology (Saraiva and Stephanopoulos, 1992a), leading to its simplification and increase in the efficiency of both of its components, that is, symbolic induction and case-based reasoning. This section is followed by the stepwise illustration of the revised methodology as applied to the simulated crystallization process problem, and finally conclusions are drawn and possibilities for further refinements identified.

Simulation of the Crystallization Process

Along with the representation of CSD, another distinguishing attribute of crystallization processes is that they are not static but characterized by a trajectory, that is, progressive change in the size of individual crystals due to different size-varying phenomena. This feature of the system signifies the importance of the transient behavior of the process along with steady-state and perturbed responses and the necessity of dynamic models, an inference well supported by the heuristic that up to ten residence times are not uncommon for most substances to reach a steady-state (Randolph and Larson, 1988). Discretizing the size axis of the integro-partial differential equation representing the population balance to transform it into a set of ordinary differential equations has emerged as the most suitable solution technique for flowsheet applications of population balance models (Marchal et al., 1988; Hounslow et al., 1988; Hill and Ng, 1996). In the following subsections, the general framework for mathematical representation of unit operations pertinent to crystallization processes and the main features of the models used to simulate the process as depicted in Figure 1 are summarized.

MSMPR crystallizer

The distinctive nature of the crystallization process necessitates additional equations for sufficient mathematical representation of a mixed-suspension, mixed-product-removal (MSMPR) crystallizer in each of the three partitions: balances, rate, and constitutive equations of the conventional scheme to process modeling. Population balances and rate equations for crystal nucleation, growth, and attrition are the add-ons to the first two divisions, whereas functions describing growth mechanisms, agglomeration kernel, and the de-

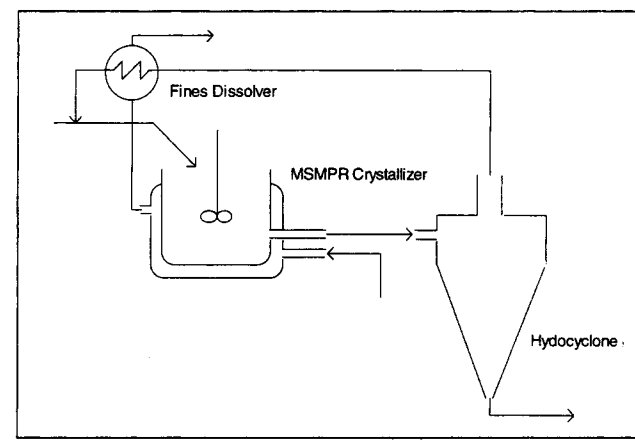

Figure 1. Crystallization process flow sheet.

pendence of physical properties on changing density are among the many extra equations falling into the third category. The discretized population, mass and energy balances, and the rate equations for nucleation and growth of potassium nitrate crystallization are as follows:

$$\frac{dN_i}{dt} = \left(\frac{dN_i}{dt}\right)_{\text{growth}} + \left(\frac{dN_i}{dt}\right)_{\text{agg}} + \left(\frac{dN_i}{dt}\right)_{\text{att}} - \frac{q}{\text{vol}}(N_i - N_i^{\text{in}}) \qquad i = 1, 2, \ldots, \infty, \quad (1)$$

where

$$\left(\frac{dN_1}{dt}\right)_{\text{growth}} = B + \frac{G}{L_1}[(b+cr)N_1 + cN_2]$$

$$\left(\frac{dN_i}{dt}\right)_{\text{growth}} = \frac{G}{L_i}[aN_{i-1} + bN_i + cN_{i+1}]$$

$$\left(\frac{dN_i}{dt}\right)_{\text{agg}} = N_{i-1}\sum_{j=1}^{i-2} 2^{j-1+1}\beta_{i-1,j}N_j + \frac{1}{2}\beta_{i-1,i-1}N_{i-1}^2 - N_i\sum_{j=1}^{i-1} 2^{j-1}\beta_{i,j}N_j - N_i\sum_{j=1}^{\infty}\beta_{i,j}N_j$$

$$\left(\frac{dN_i}{dt}\right)_{\text{att}} = \frac{3}{4}\sum_{j=i+1}^{\infty}\left(2\frac{v_i^{1/3} - v_{i-1}^{1/3}}{v_{j-1}^{1/3}}\right)X_jN_j - \frac{1}{2}X_iN_i$$

$$\frac{dC}{dt} = \frac{q}{\text{vol}}(C_{\text{in}} - C) - (k_sG\bar{m}_2 - k_vBL_0^3)\rho \qquad (2)$$

$$\text{vol}\, c_P\rho\frac{dT}{dt} = P + Q + \Delta H_C Cq + q(\rho_{\text{in}}c_{P\text{in}}T_{\text{in}} - \rho c_P T) \qquad (3)$$

$$B = A\exp\left(\frac{16\pi\sigma^3\nu^2}{3k^3T^3(\ln S)^2}\right) + k_N\frac{N_P}{N_Q}\bar{m}_3 s^{1.7} \qquad (4)$$

$$G = \left(\frac{k_dk_r}{k_d + k_r}\right)s. \qquad (5)$$

Crystal growth and agglomeration contributions to the population balance are adapted from Hounslow et al. (1988), whereas attrition is included using the equation developed by Hill and Ng (1995). Details on the formulation of rate equations can be found in Marchal et al. (1988), and the kinetic coefficients have been taken from Garside and Davey (1980).

The complete system of differential algebraic equations is categorized into ordinary differential (ODE), implicit algebraic (IAE), and explicit algebraic (EAE) equations. The Sargent–Weterberg algorithm is used on the matrix of EAEs to identify an effective partitioning structure for this subset of equations and to embed them into procedures that are solved internally. This structured partitioning reduces the apparent size of the problem and directs the order of precedence analysis (done by SPEEDUP) on the procedures rather than the individual equations, thereby modularizing the model itself to allow testing of different physical models for kinetics without reformulating the entire model. The model is found capable of predicting most of the phenomena of interest with sufficient detail and numerical precision and is versatile enough to be used in any configuration within a flow sheet.

Hydrocyclone and fines dissolver

Effective mathematical representation of the behavior of a hydrocyclone requires adequate analysis of three distinct physical phenomena taking place in these devices: the understanding of fluid flow, its interactions with the dispersed solid phase, and the quantification of shear induced attrition of crystals. The fluid flow is described by analytical solution to conservation of mass and momentum equations developed by Bloor and Ingham (1987). The resulting equations for dimensionless fluid velocities in tangential, axial, and radial directions, respectively, in cylindrical coordinates are

$$\frac{v_\lambda}{V} = \frac{\left[1 - Q^2\sigma\psi/(\pi R_0 V)^2\right]^{1/2}}{r\sin\theta} \qquad (6)$$

$$v_r = \left[2A\cos\theta - 2\sigma\left[\cos\theta\ln\left(\frac{1}{2}\tan\theta\right)\right]\right]\sin\theta + \left[\frac{2\psi}{\sin 2\theta}\cos\theta\right] \qquad (7)$$

$$v_\theta = \left[2A\cos\theta - 2\sigma\left[\cos\theta\ln\left(\frac{1}{2}\tan\theta\right)\right]\right]\cos\theta + \left[\frac{2\psi}{\sin 2\theta}\sin\theta\right]. \qquad (8)$$

The population balance is solved for particle attrition induced by high swirl velocities in the vessel; the breakage rate is this case depends on specific power input. Particles are assumed to follow fluid flow in a tangential direction, whereas the velocities in axial and radial directions are calculated using the following equations (Brayshaw, 1990):

$$\frac{\pi d_i^3}{6}(\rho_s - \rho_f)\frac{v_i^2}{r_L} = 3\pi\mu_f d_i U \qquad Re_p \le 0.1 \qquad (9)$$

$$\frac{\pi d_i^3}{6}(\rho_s - \rho_f)\frac{v_i^2}{r_L} = 3\pi\mu_f d_i U\left(1 + \frac{3}{16}Re_p\right) \qquad 0.1 < Re_p \le 1.0 \qquad (10)$$

$$\frac{\pi d_i^3}{6}(\rho_s - \rho_f)\frac{v_i^2}{r_L} = 3\pi\mu_f d_i U\left(1 + \frac{3}{16}Re_p\right)^{0.5} \qquad 1.0 < Re_p \le 100 \qquad (11)$$

The efficiency of the hydrocyclone (Dietz, 1981) is given by

$$\eta = 1 - C\exp\left[\frac{-2\pi R_C U_{pw}(s - a/2)}{Q}\right] \qquad (12)$$

where

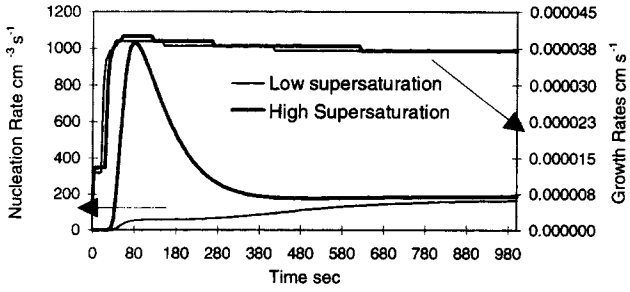

Figure 2. Nucleation and growth-rate profiles at two different levels of supersaturation.

$$C = \frac{R_C U_{pw} + R_V U_{ro} + R_V U_{pv}}{2R_V U_{pv}} - \left[\left\{\frac{R_V U_{pv} - R_V U_{ro} - R_C U_{pw}}{2R_V U_{pv}}\right\}^2 + \frac{R_C U_{pw}}{R_V U_{pv}}\right]^{1/2}.$$

Since the separation efficiency is represented by flow characteristics and dimensions of the vessel, the model provides a simulation tool with physically realizable control for desired performance. Fines dissolution is employed in high nucleating systems to dissolve smaller crystals so that the driving force for crystal growth, that is, the extent of supersaturation could be directed toward the growth of larger crystals. The overflow stream from the hydrocyclone is subjected to desupersaturation through increase in temperature to dissolve crystals below a certain size to extinction. Dissolution rate and the subsequent crystal number contributions to the population are modeled by equations similar to those used for the crystal growth process. The diffusion coefficient in this case, however, is calculated as a function of the individual size of the particle and not the average size as in the growth rate representation.

Figure 2 shows the nucleation and growth-rate profiles under different operating conditions, while the development of CSD with time within the crystallizer and the product stream is depicted in Figure 3.

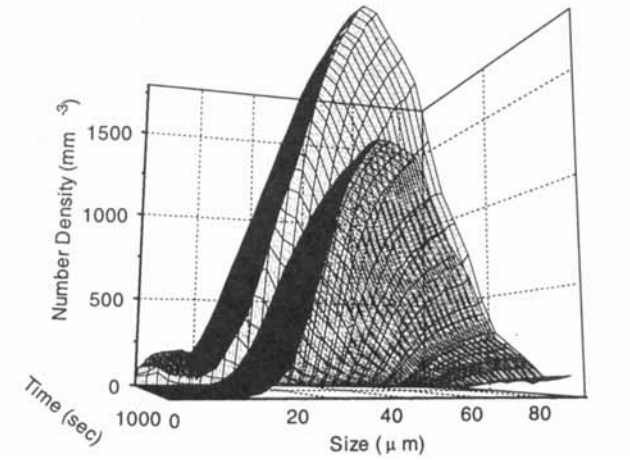

Figure 3. Development of KNO_3 CSD in the crystallizer and product stream.

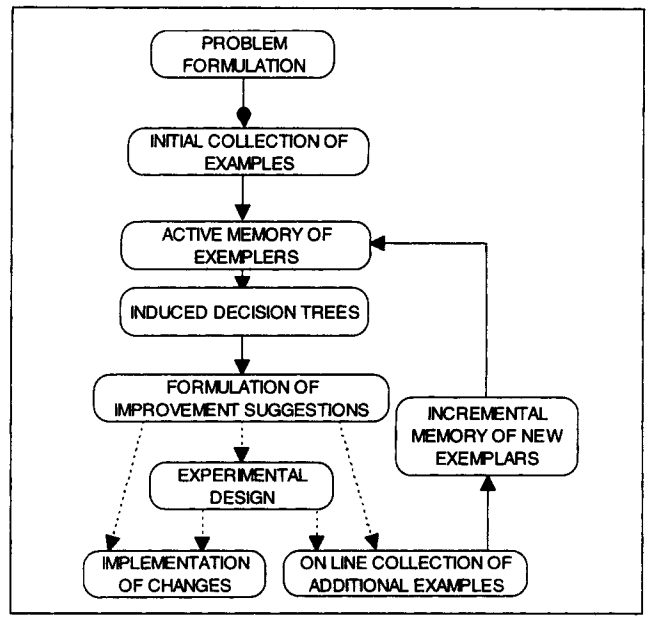

Figure 4. Global machine learning methodology (Saraiva and Stephanopoulos, 1992a).

Machine Learning Approach to Process Improvement

The sequence of steps undertaken by the global machine learning approach is depicted in Figure 4 (see Saraiva and Stephanopoulos, 1992a, for details). It comprises two major components: one for screening the objects to maintain a reduced subset of data conveying novelty (these constitute the feedback from process's attempt to perform at the desired level), and the other for detecting interesting conceptual patterns or revealing structure in collections of observations. The former employs case-based reasoning, whereas the latter is based on symbolic inductive learning that involves operations of generalizing, specializing, transforming, correcting, and refining knowledge representations. In the following, we summarize the features of the procedures used by these two components as employed in the existing methodology before highlighting the merits of the modifications suggested in this work.

ID3-CART induction of decision trees

The nonincremental construction of a decision tree based on the *ID3* algorithm starts at the root node of the tree where all the objects are located. For continuous valued attributes, the threshold value for discretizing the range is calculated by minimizing information entropy function (E-score) over all the possible splits (for N objects there are $N-1$ possible splits for each of the continuous attributes). The E-score, which is a measure of the ambiguity associated with a particular split, can be calculated by

$$E = -\sum_{K=1}^{K} P_k \log P_k + \sum_{c=1}^{R} \frac{N_c}{N} \sum_{k=1}^{K} P_k^c \log P_k^c,$$

where P_k is the relative frequency of objects that belong to

class k of the performance variable among all the objects N, and N_c is the number of objects allocated to the cth node. The attribute with lowest E-score is selected as test attribute for splitting the node and sending the subsets of objects through branches emanating from it to children nodes. E-score-based selection of the test attribute ensures that only the most significant attributes contribute to the classification rules. New threshold values are calculated at each of the new nodes for all the attributes, and test attributes are selected for further splitting of the objects. The procedure continues until a terminal node is reached where all the objects belong to the same class.

The lower levels of the tree are developed from a progressively smaller number of objects, and therefore many of branches could be reflecting chance occurrences in the particular data rather than representing underlying principles. Pruning methods identify the least reliable branches and remove them. The error-complexity (CART) pruning method of Breiman et al. (1984) is used in the existing methodology. It is a two-stage method, first generating a series of trees pruned by different amounts, and then selecting one of these by examining the number of classification errors each of them make on an independent set of objects. During pruning, the error-complexity method takes account of both the number of errors and the size (complexity) of the tree. The scheme, however, requires an independent set of objects for selecting the tree.

Case-based reasoning

This step involves a screening procedure whereby existing and continually generated data from on-line applications is analyzed to obtain a reduced data set (active memory of exemplars, AME) situated near the decision boundaries. The procedure is also capable of periodically updating AME's contents during serial learning tasks (process generating a stream of training objects where one would like to be able to detect novelty instantaneously and revise the contents accordingly). The distance nearest-neighbor (DNN) classifier is used to predict the classes of incoming objects. It uses a set of objects and a distance metric on them to find the nearest neighbors of the incoming object to find its class. The misclassified objects are stored in an incremental training set and whenever the total number in this training set reaches a predetermined fraction of the present cardinality of AME, a revision point is reached. An object is excluded from AME if the misclassifications made by it exceed a predetermined threshold. A global measure of significance, considering reliability, frequency of use and age is used as a second criterion to remove objects from AME. The revised AME is used to induce a new generation of decision trees.

Modifications

Revised Scheme for the Induction of Decision Trees. For nonincremental learning tasks, *ID3* is a useful concept learning algorithm because it can efficiently construct a decision tree that generalizes well. Serial learning tasks would, however, be better served by an algorithm that could accept objects continually to revise the tree, without needing to build a new tree each time. Most of the incremental algorithms perform on decision trees induced through nonincremental methods by restructuring them to account for the incoming object that cannot be classified by the existing tree. Failure of the tree to classify the new object is attributed to the fact that at one of the nodes traversed by this object, the criterion for selection of the test attribute does not hold. The *ID5R* algorithm (Utgoff, 1989) is one such algorithm and involves the use of E-scores to change the test attribute at a node to one with the lowest value, followed by a tree restructuring process that preserves consistency with the existing objects without re-examining them. The algorithm is depicted below:

1. If the object is from the same class then update the number of objects at the leaf.
2. otherwise,
   a. For the test attribute and all the nontest attributes at all the nodes traversed by the incoming objects, update the attribute counts.
   b. If the current node contains an attribute test that does not have the lowest E-score then,
      i) Restructure the tree so that an attribute with the lowest E-score is at the root.
      ii) Recursively reestablish a best test attribute in each subtree except the one updated in step 2c.
   c. Recursively update the decision tree below the current decision node along the branch for the value of the test attribute that occurs in the object description. Grow the branch if necessary.

When a new object is introduced to the existing tree, attribute value counts (number of attribute values in each class) are updated for the test and nontest attributes. If the test attribute no longer has the lowest score, the tree is restructed by the procedure outlined below:

1. If the attribute a_{new} to be pulled up is already at the root then stop.
2. otherwise,
   a. Recursively pull the attribute a_{new} to the root of each immediate subtree.
   b. Transpose the tree, resulting in a new tree with a_{new} at the root, and the old root attribute a_{new} at the root of each immediate subtree.

All the algorithms for inducing decision trees require the attributes to be categorical and therefore continuous valued attributes (the term continuous covers both real and integer values) must be discretized prior to test attribute selection. We employ a corollary due to Fayyad and Irani (1992) on the discretization of continuous attributes based on the information entropy minimization heuristic to gain computational efficiency by reducing the number of possible candidates for the cut point. The corollary states that a binary partition based on information entropy minimization principle will always partition the data on a boundary point in the sequence of objects ordered by the attribute value. The boundary points refer to those attribute values in their ascending sequence where the class of performance variables changes. Speedups of up to seven times have been reported for some types of objects as a direct consequence of the corollary (Fayyad and Irani, 1992).

Pessimistic pruning (Quinlan, 1993) has been adapted in preference to CART for two reasons: 1) it is much faster than any of the other pruning methods, because it only makes one pass and only looks at each node once, and (2) it does not require an independent set of objects for choosing the

pruned tree. The method, however, results in larger trees than those pruned by the error-complexity method (CART). Pruning proceeds from the root downward by comparing the corrected number of misclassifications at a node with the leaves in the corresponding subtree. A tree is pruned if the following condition is satisfied:

$$SE[n'(T_t)] + n'(T_t) > n'(t), \tag{13}$$

where $n'(t)$ and $n'(T_t)$ are correct misclassification rates at a node and for the subsequent subtree, respectively. $SE[n'(T_t)]$ denotes standard error for the subtree. The following equations are used to calculate these quantities:

$$n'(t) = e(t) + 1/2 \tag{14}$$

$$n'(T_t) = \sum e(i) + \frac{N_i}{2} \tag{15}$$

$$SE(n'(T_t)) = \sqrt{\frac{n'(T_t) \times [N(t) - n'(T_t)]}{N(t)}}, \tag{16}$$

where $e(t)$ is the actual number of objects misclassified at node t; $e(i)$ is the number at the ith leaf of the subtree; N_i is the total number of leaves under the node; and $N(t)$ is the number of objects at the node. A thorough comparison of different methods available for pruning decision trees can be found in Mingers (1989). Though, the incremental algorithm leads to instantaneous incorporation of novel objects into the decision trees, it is the evaluation process that determines the maturity of newly acquired information for application. The branches representing obsolete objects are pruned automatically by the chosen pruning method.

Present State of Case-Based Reasoning Process. The existing procedure for screening objects is applied to the initial training set to obtain a reduced subset of objects. The application of DNN also provides an indication of the adequacy of the data where similar cardinality of the complete and reduced set indicates the need for further data collection. With the modifications to the symbolic learning procedure, the need for dynamic memory to adapt AME for updating the decision tree vanishes as the objects reflecting drifts or other temporal changes in the process behavior once detected could lead to an instantaneous update of the classifer. We propose to replace the DNN method with the current version of the tree, to serve as the classifier at the case-based reasoning step for tightening the passage of objects to the incremental induction step. The DNN classifier is not the most effective classifier, especially when used in isolation. Furthermore, its effectiveness is very sensitive on the metric definition and it is computationally expensive as well, because a representative set of objects must be stored and the interpoint distances and classification rule calculated for each incoming object (Breiman et al., 1984).

The simplified case-based reasoning procedure, however leaves only the pruning mechanism to exclude obsolete objects, which can be an inefficient and slow procedure. To overcome the potential problem of exceedingly large decision trees with a significant proportion of obsolete rules, we suggest eliminating all the branches leading to these leaves after a proposed scheme of the tree for performance improvement has been agreed upon for implementation. The implementation would ensure that no objects are subsequently generated for the lower levels of performance represented by the eliminated branches.

Formulation of improvement suggestions

Some of the procedures and tests developed by Saraiva and Stephanopoulos (1992a) for the formulation of improvement suggestions from the decision trees are summarized in the following. The process can be divided into three distinct activities: refinement, evaluation, and validation.

Refinement entails the enlargement of the partitions of feature space defined by the leaves of the decision tree without significantly lowering their purity (fraction of objects with the correct class label). The surface boundaries for each of the leaves are examined to identify borders with partitions of the same class. The leaf is then expanded by extending each of these borders in turn (while leaving the others fixed), until no further extensions are possible for any boundary, without including space of different class. This procedure is repeated for all the leaves. A statistical test of significance is performed on all the expanded leaves, followed by a check to ensure that hard process constraints on any of the variables are not violated due to the expansion.

Evaluation tests include (1) *certainty factors:* the fraction of objects from an independent test set that are correctly classified by the extended leaves; (2) *Pareto index:* this index allows the identification of those regions of the feature space where most of the objects of each class from the test set are being placed, and (3) *demographic density:* the ratio of total number of objects in the set that belong to a particular leaf.

The validation process involves an extensive analysis of the evaluation scores for the identified promising and feasible pathways to process improvement. This distilled information is reevaluated to ensure that all the process and safety constraints are satisfied prior to plant personnel deciding on whether to perform pilot tests or implement the improvement schemes on-line (see Saraiva and Stephanopoulos (1992a) for details).

Performance Improvement of the Crystallization Process

In this section we present the stagewise development of two generations of decision trees for the performance improvement of KNO_3 crystallization through the revised machine learning methodology. The first-generation construction of decision trees is closely based on the existing methodology (Figure 4); however, for continual improvement of process performance the revised scheme presented in Figure 5 is employed. Three zones of high-level performance are identified, with each representing an improvement of up to 12% on the nominal average levels.

Problem formulation and process simulations

The performance of the crystallization system is measured as a combination of the total number of crystals and their average size in the underflow from the hydrocyclone. These two features of crystal size distribution manifest the effects of nucleation and growth rates, and there is always an optimum

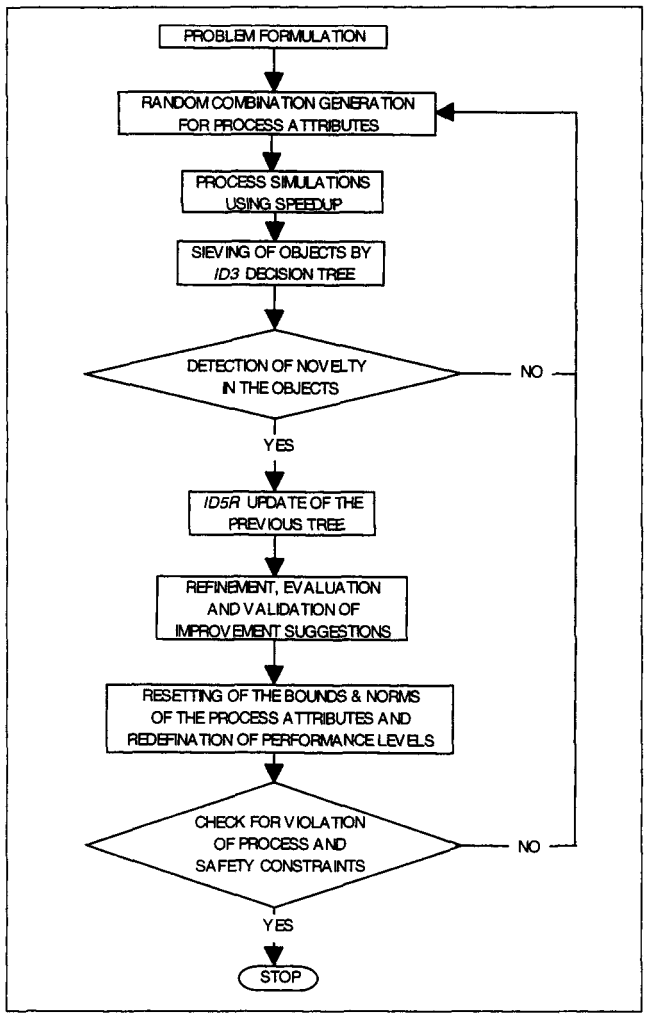

Figure 5. Revised machine learning scheme as applied to crystallization problem.

driving force for the two kinetic processes beyond which they do not increase simultaneously; rather, the average size falls while the number of crystals increases. In the process under consideration, the operation for all three units—crystallizer, hydrocyclone, and fines dissolver—can be affected to achieve the desired performance. The behavior of the crystallizer is modified through the feed temperature, concentration and flow rate, and coolant temperature. These variables allow us to manipulate both the supersaturation driving force and the residence time within the crystallizer. Axial velocity of the stream into the hydrocyclone is used to adjust the particle efficiency curve, whereas fines dissolution temperature allows us to control the number of fine crystals recycled. The operating data are constructed by generating 150 random combinations of these six variables with the following means and standard deviations: TF (295.00 K, 2.0); CF (3.52 kmol/m^3, 0.03), FF (0.10 L/s, 0.01); TC (281.25 K, 1.5); TD (296.5 K, 0.35); W (0.01257, 0.00075). Dynamic simulations are carried out for each combination of operating data by interfacing it to SPEEDUP through an external data interface. The results for total crystal number and average size are stored with the corresponding vector of the decision variables to develop a complete training set.

Case-based reasoning to obtain a reduced subset of objects

The application of the DNN classification procedure to the complete training set results in a reduced subset of 125 objects conveying all the information relevant for the identification of the decision boundaries. Reduction in size indicates the adequacy of the data for the effective induction.

ID5R induction of decision trees

The induction algorithm can handle fuzzy, functional, or crisp discretization for the performance criterion (Saraiva and Stephanopoulos, 1992a). In this study, however, with the objective being the identification of zones encompassing best average performance, we restrict ourselves to crisp discretization of the performance criterion based on the following thresholds: $y < 2.5\times10^6$ s^{-1} ($L_{av.} < 55.0$ μm) as low (A); $2.5\times10^6 < y < 4.2\times10^6$ ($55.0 < L_{av.} < 63.5$ μm) as normal (B) and $y > 4.2\times10^6$ ($L_{av.} > 63.5$ μm) as high (C). Each object in the reduced subset is labeled a performance class according to these thresholds. Of the 125 objects, 100 are used to induce the decision tree while 25 are reserved to estimate the true misclassification rate of the pruned tree. The average performance of these 100 objects is 3.58×10^6 s^{-1} crystals with an average size of 56.3 μm. In Figure 6 we present a projection of these objects in FF-TC coordinates along with the corresponding classes. Although these two variables are not enough to produce a complete discrimination, there are clearly zones of the FF-TC plane where it is very likely to get points from only one particular class. It is the boundaries of these zones that will be discretely identified by the decision tree.

The *ID5R* algorithm, which constructs an identical tree to *ID3* for the same training set, was used to develop a fully expanded decision tree. All the attributes in the training set were discretized using an information entropy minimization heuristic, and the attribute value counts associated with each split were recorded at each node to enable the update of E-scores for incorporating new process information into the tree as it becomes available. Pessimistic pruning was used to prune the tree by removing statistically insignificant branches to improve its understanding. The pruned tree, depicted in Figure 7, exhibits a misclassification rate of 20% on an independent test set.

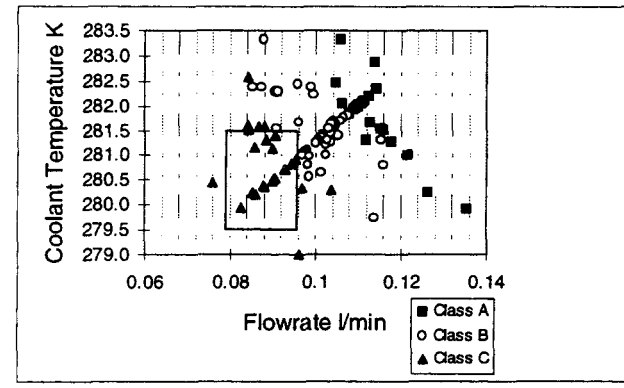

Figure 6. FF-TC scatter of the reduced training set.

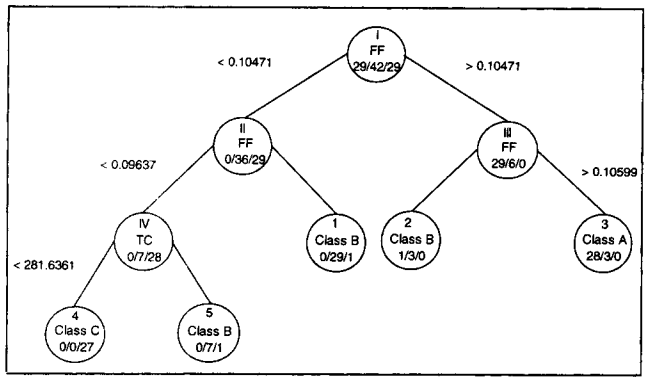

Figure 7. *ID5R* decision tree from the reduced training set.

Evaluation, refinement, and validation

It can be seen from the pruned tree that only the most significant among all the attributes appear in the trees. For example, feed temperature, feed concentration, fines dissolver temperature, and axial velocity of fluid into the hydrocyclone are found to have undetectable effects on the performance variable within their allowed variations. Only feed flow rate which is a measure of residence time, is sufficient to distinguish normal from low levels of performances, whereas a high or class "C" leaf requires three preconditions in two variables to be satisfied. Leaf 4 representing class C performance has a certainty factor of 1.00, which means that all the objects within this constrained feature space represent class C, whereas the *Pareto* index evaluation for it indicates that 93% of all the high-performance objects are concentrated here. The average performance within this zone is 5.34 $\times 10^6$ s^{-1} crystals with an average size of 65.5 μm. The best class "B" leaf scores 0.966 and a mere 0.69 on the two indices, respectively. The subsequent refinement procedures leave most of the rules essentially unchanged especially those regarding class B. The findings are validated by simulating the process under the identified conditions.

Implementation and case-based reasoning using the decision tree

To illustrate the incremental induction algorithm we explore the boundaries between classes B and C and try to explain multiple regions of class C in the feature space for increased flexibility of operation. It can be seen from the induced decision tree that by setting the flow rate at less than 0.1047 L/min, only class B or C performance will be achieved. The bounds and norms are adjusted accordingly, and a set of 50 objects is compiled through simulations. Of the 50 objects, only 30 pass through revised case base reasoning employing the current decision tree. The FF-TC and CF-TC scatters of these and other objects from the original reduced training set meeting the flow-rate criterion (Figures 8a and 8b), show that only CF-TC scatter is capable of vaguely identifying the decision boundaries.

Second-generation decision tree

The current tree is updated using E-scores through the procedures sketched in the algorithms presented earlier. The *ID5R* algorithm uses original discretization of continuous-

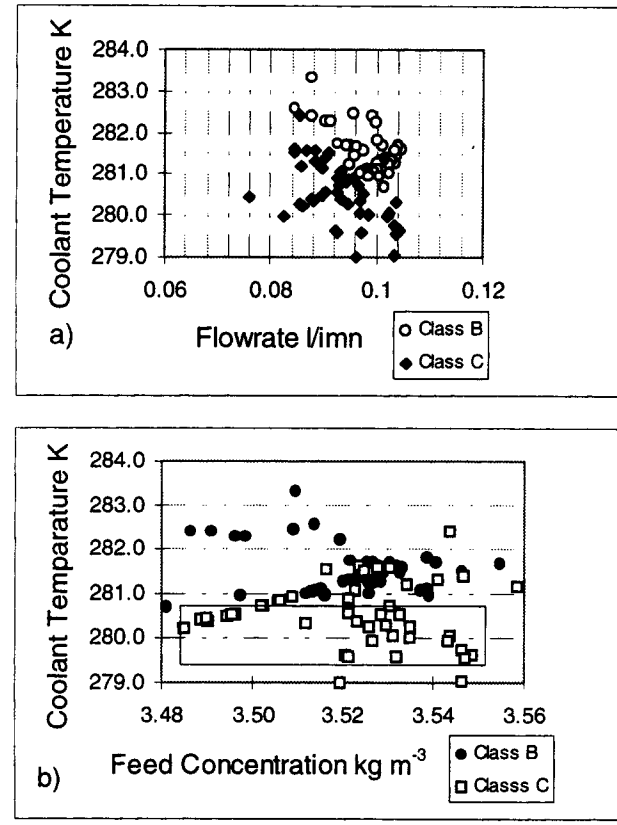

Figure 8. (a) FF-TC and (b) CF-TC scatter of all objects with feed flow rate less than 0.1047 L/min.

valued attributes during the incremental induction process. As a consequence, even though the algorithm updates the tree to a form similar to one that would be induced by the nonincremental *ID3* algorithm for the enlarged training set, the attribute splits associated with the latter are more accurate with lower entropy information. This added ambiguity resulting from the use of old discretization in the revised tree is, however, offset by the exclusion of their calculation at each node in the subtree during restructuring that would otherwise involve substantial computational effort and reexamination of the original training set. The updated version (Figure 9) presents a more complicated representation of the interactions between the decision variables. The original test attribute at node II is pushed down two levels (TC and CF) appearing at levels 2 and 3, respectively), while restructuring to ensure that all the test attributes in the subtree below the new node II have minimum E-scores. The updated tree provides us with three alternative schemes for refinement, evaluation, and validation to improve process performance beyond class C levels. If it were to be decided to reduce the flow rate to a maximum of 0.1047 L/min, then based on the argument presented earlier the tree (Figure 9) it could be simplified by eliminating nodes I and III and leaves 1 and 2.

Evaluation, refinement, and validation

Of the three leaves, leaf 4, with an average crystal production rate of 5.83 $\times 10^6$ s^{-1} and a size of 67.6 μm, seems most promising, for it represents 76.59% of class C objects and has a certainty factor of 0.923. The operating conditions corre-

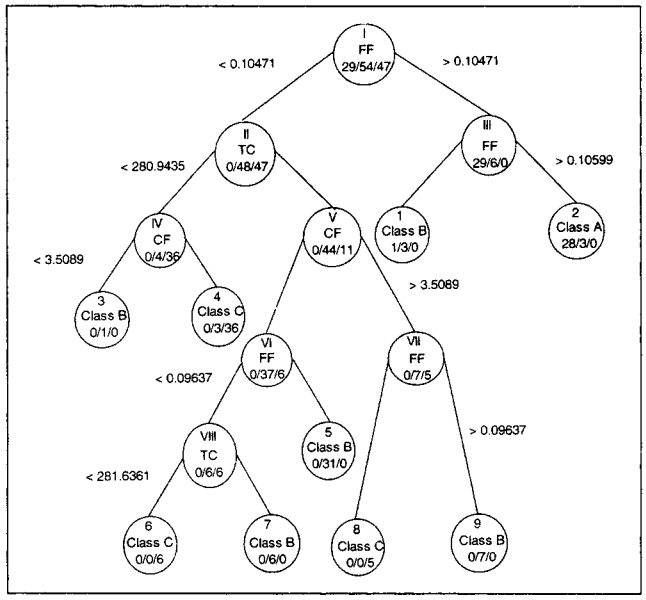

Figure 9. Revised decision tree through *ID5R* algorithm.

sponding to this leaf are (1) FF < 0.1047 L/min, (2) TC < 280.94 K, and (3) CF > 3.508 kmol/m^3. Even though the certainty factors associated with leaves 6 and 8 are a perfect 1.0, they only represent 12.76 and 10.65% of the total current class C objects, respectively. The average performances of objects in these two zones are 4.68×10^6 s^{-1} and 65.4 μm and 5.0×10^6 s^{-1} and 66.3 μm, respectively. Process and safety constraints and ease of implementation would determine which of the three schemes would be implemented.

The update procedure can be reinitiated whenever a novel object is detected by the current-generation tree (at case-based reasoning step), after implementing one of the performance improvement schemes suggested by it. Since the level of supersaturation and residence time characterize the crystallization process, it seems reasonable that the three significant features capturing their effect are cooling stream temperature, feed concentration, and feed flow rate.

Conclusions

While there have been previous efforts to simulate crystallization process flow sheets through simplified or "short-cut" models (Evans, 1989; Hounslow and Wynn, 1993) and optimize them at design level through semiempirical "design-and-cost" relationships (Rossiter and Douglas, 1986; Jones, 1991), this is the first attempt to optimize a process flowsheet at operational level through a consistent and sufficiently detailed population-balance-based models capable of predicting complete crystal size distribution by considering all the size and number varying processes. The existing short-cut models are limited in their scope to the prediction of the total number of crystals and their average size, which does not allow us to consider agglomeration and disruption processes and to exercise a physically realizable control on units, including fines dissolver and hydrocyclones; for example, effective operation of a fines dissolver requires an estimation of the number of fine crystals. Simulations of the process under consideration not only help in understanding some of the very complicated and highly coupled interactions within and across the units, but could also lead to interesting observations on the conceptual design aspects of the process.

The machine learning approach for continuous process improvement has been applied to the crystallization system, because of its superior representation of process improvement suggestions in the form of explicit operational and symbolic descriptive language identifying zones of the best average performance and comparable accuracy with alternative techniques for enhancing process performance (Saraiva and Stephanopoulos, 1992a). The symbolic inductive component of the approach has been modified to include an incremental algorithm for the induction of decision trees to update them in the wake of incoming process data. The knowledge embraced by the tree is easily convertible into improvement suggestions for refinement, evaluation, and validation, before implementing them on the process itself. The observed response of the system to these changes creates novel data, which are passed on to the case-based reasoning component. The modification to the induction process, however, results in a knock-on effect on this component of the overall methodology. Its scope reduces from keeping an active memory of exemplars, updating it as new data become available, identifying novelty, removing unreliable data, and determining when to reinduce the tree, to simply recognizing an incoming object as novel and passing it onto the induction stage for continual update of classification rules through the incremental algorithm.

The revised methodology presented here has been successfully used to quantify performance of the crystallization process in terms of operating conditions. In two generations of symbolic induction three different operational schemes, representing an increase of nearly 12% on nominal average performance, are identified by the methodology without altering the variation levels specified at the start. The performance level is found to be a strong function of feed concentration, and flow-rate and cooling stream temperature. These variables reflect the underlying characterization of the process by supersaturation levels and residence time in the crystallizer. After the implementation of one of the schemes, the performance levels could be redefined and the procedure for searching further improvements reinitiated. Application of revised machine learning methodology thus provides a new tool for optimization of crystallization processes that can utilize either simulated or real plant data from crystallization processes.

Future work will include the development of multifaceted dynamic models for settling and filtration devices, and the incorporation of model and parameter uncertainty in the simulation process to observe the effects on the findings of the methodology. After validating the procedure and its findings on plant data, an incremental algorithm for the induction of a multivariate decision tree will be formulated. Unlike the univariate decision tree, a multivariate decision tree is not restricted to orthogonal partition of the feature space (Brodley and Utgoff, 1995), and therefore is able to find more concise, effective, and accurate decision trees. Multiple objectives for the process would also be considered with the growth of the problem through the developments due to Saraiva (1995).

Notation

Crystallizer

A = frequency factor for primary nucleation, $cm^{-3} \cdot s^{-1}$
a, b, c = constants in Eq. 3
B = overall nucleation rate, $cm^{-3} \cdot s^{-1}$
β = agglomeration kernel, $cm^3 \cdot s^{-1}$
C = solute concentration, $kg \cdot m^{-3}$
c_P = specific heat capacity, $kJ \cdot kg^{-1}$
ΔH_C = heat of crystallization, $kJ \cdot kg^{-1}$
G = linear growth rate, $cm \cdot s^{-1}$
k = Boltzmann constant, $J \cdot K^{-1}$
k_d, k_r = diffusion and surface integration coefficients
k_N = secondary nucleation rate coefficient
k_s = surface shape factor
L_0 = size of the nucleus, μm
$\overline{m}_i$ = ith moment, $\mu m^i \cdot cm^{-3}$
N_P, N_Q = power and pumping number for the stirrer
P = specific power input of the stirrer, $kJ \cdot s^{-1}$
Q = heat transfer rate, $kJ \cdot s^{-1}$
q = volumetric flow rate, $m^3 \cdot s^{-1}$
S = supersaturation expressed as fraction
s = supersaturation expressed as difference
σ = interfacial tension, $J \cdot m^{-2}$
ν = molecular volume, m^3
v = volume of a crystal, μm^3
vol = volume of the vessel, m^3
X = specific attrition rate, s^{-1}

Hydrocyclone

A = constant of integration
a = feed inlet diameter, m
d_i = equivalent diameter of the particle, μm
Q = feed flow rate, $m^3 \cdot s^{-1}$
θ = angle for radial direction
R_0 = length of the hydrocyclone, m
R_C = radius of the cone in the cylindrical region, m
R_V = radius of the vortex finder, m
s = engagement length, m
σ = dimensionless group
U = particle velocity
U_{pw} = radial particle velocity near the wall of cyclone
U_{pv} = radial particle velocity in the vortex finder
U_{ro} = radial velocity of the liquid into the cone
V = swirl velocity at the entrance
v_i = liquid velocity in the ith direction
ψ = stream function

Literature Cited

Biegler, L. T., J. Nocedal, and C. Schmid, "A Reduced Hessian Method for Large Scale Constrained Optimization," *SIAM J. Opt.*, **5**(2), 314 (1995).

Bloor, M. I. G., and D. B. Ingham, "Flow in Industrial Cyclones," *J. Fluid Mech.*, **178**, 507 (1987).

Brayshaw, M. D., "Numerical Model for the Inviscid Flow of a Fluid in a Hydrocyclone to Demonstrate the Effects of Changes in the Vorticity Function of the Flow Field on Particle Classification," *Int. J. Miner. Process.*, **29**, 51 (1990).

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Belmont, CA (1984).

Brodley, C. E., and P. E. Utgoff, "Multivariate Decision Trees," *Mach. Learn.*, **19**, 45 (1995).

Cuthrell, J. E., and L. T. Biegler, "On the Optimization of Differential-Algebraic Process Systems," *AIChE J.*, **33**(8), 1257 (1987).

Dietz, P. W., "Collection Efficiency of Cyclone Separators," *AIChE J.*, **27**(6), 888 (1981).

Evans, L. B., "Simulation with Respect to Solid Fluid Systems," *Comp. Chem. Eng.*, **13**, 343 (1989).

Fayyad, U. M., and K. B. Irani, "On the Handling of Continuous-Valued Attributes in Decision Tree Generation," *Mach. Learn.*, **8**, 87 (1992).

Garside, J., and R. J. Davey, "Secondary Contact Nucleation: Kinetics, Growth and Scale-up," *Chem. Eng. Commun.*, **4**, 393 (1980).

Hart, M., and R. Hart, *Quantitative Methods for Quality and Productivity Improvements*, Quality Press, Ottawa, Canada (1989).

Hill, P. J., and K. M. Ng, "New Discretization Procedure for the Breakage Equation," *AIChE J.*, **41**(5), 1204 (1995).

Hill, P. J., and K. M. Ng, "New Discretization Procedure for the Agglomeration Equation," *AIChE J.*, **42**(3), 727 (1996).

Hounslow, M. J., R. L. Ryall, and V. R. Marshall, "Discretized Population Balance for Nucleation, Growth, and Aggregation," *AIChE J.*, **34**(11), 1821 (1988).

Hounslow, M. J., and E. J. W. Wynn, "Short-cut Models for Particulate Processes," *Comp. Chem. Eng.*, **17**(5/6), 505 (1993).

Jones, A. G., "Design and Performance of Crystallization Systems," *Advances in Industrial Crystallization 91*, Butterworth-Heinemann, Boston (1991).

Marchal, P., R. David, J. P. Klein, and J. Villermaux, "Crystallization and Precipitation Engineering—I: An Efficient Method for Solving Population Balance in Crystallization with Agglomeration," *Chem. Eng. Sci.*, **43**(1), 59 (1988).

Mingers, J., "An Empirical Comparison of Pruning Methods for Decision Tree Induction," *Mach. Learn.*, **4**, 227 (1989).

Quinlan, J. R., "Decision Trees and Decision Making," *IEEE Trans. Syst. Man, Cybern.*, **SMC-20**(2), 339 (1990).

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

Randolph, A. D., and M. A. Larson, *Theory of Particulate Processes*, 2nd ed., Academic Press, Orlando, FL (1988).

Rossiter, A. P., and J. M. Douglas, "Design and Optimization of Solids Processes," *Chem. Eng. Res. Dev.*, **64**, 175 (1986).

Saraiva, P. M., and G. Stephanopoulos, "Continuous Process Improvement through Inductive and Analogical Learning," *AIChE J.*, **38**(2), 161 (1992a).

Saraiva, P. M., and G. Stephanopoulos, "An Exploratory Data Analysis Robust Optimization Approach to Continuous Process Improvement," Working Paper, Dept. of Chemical Engineering, MIT, Cambridge, MA (1992b).

Saraiva, P. M., "Inductive and Analogical Learning: Data-driven Improvement of Process Operation," *Intelligent Systems in Process Engineering* (Advances in Chemical Engineering), G. Stephanopoulos and C. Han, eds., Vol. 22, p. 377 (1995).

Utgoff, P. E., "Incremental Induction of Decision Trees," *Mach. Learn.*, **4**, 161 (1989).

Manuscript received Oct. 8, 1996, and revision received Jan. 27, 1997.