Operational Envelopes for Batch Processes

Nouri J. Samsatli, Mona Sharif, and Nilay Shah
Centre for Process Systems Engineering, Imperial College of Science, Technology and Medicine,
London SW7 2BY, U.K.

Lazaros G. Papageorgiou
Dept. of Chemical Engineering, University College London, London WC1E 7JE, U.K.

Batch processes are often subjected to a number of sources of variability, which are frequently overlooked due to operating policy implementation. In many pharmaceutical and speciality chemical processes, which are predominantly batch in nature, a high proportion of the operating procedure is implemented manually, which can lead to significant variability. The effect of this variability is typically neglected at all stages in process development and even when full-scale production begins. The result of any variability can have a dramatic effect on some process stages (particularly sensitive ones such as crystallization), and it is therefore important to understand how variability in any of the operating variables affects the overall process. Ideally, one would like to design processes that are less sensitive to variability and specify precisely the limits of any allowable variability. A method for designing batch processes using "operating envelopes" that specify process operation in terms of a region within which the operating variables should be maintained to guarantee feasible and profitable operation of the process is presented. The problem is formulated as a multiscenario dynamic optimization problem, and an efficient hierarchical solution procedure is presented. Finally, the applicability of the procedure is demonstrated by using two example problems.

Introduction

The majority of work in process operations has focused on methods that can determine the best operating policy and/or control strategy for a process. When control aspects are not taken into account, it is usually assumed (implicitly) that the operating policies generated from these methods can be precisely implemented in the plant. When the process is designed for controllability, too, various types of uncertainty and disturbances are considered and can be accommodated with the appropriate control configuration (Hansen et al., 1998).

These methods are suitable, and appear to work well, for conventional processes (usually steady-state) where automated control is widely used. In the fine and speciality chemicals industry, however, processes are usually implemented in multipurpose batch plants with little or no automated control. In these cases, because most of the operating policy is implemented manually, it is neither possible to implement the operating policy precisely as specified nor to account for any disturbances. This leads to variability, which will more than likely have a negative effect on the performance of the plant. In industry and academia alike, the effect of this variability is rarely considered.

In this article, we consider a method to account for the inaccurate nature of operating policies that are implemented in the batch chemicals industry. The approach is to determine a range of operating policies (the union of which is called the *operating envelope*) during the design stage over which the plant performance can be guaranteed to meet certain targets. We restrict ourselves to a simple (hyperrectangular) form of operating envelope that corresponds to maintaining control variables within ranges that are determined in the design optimization. This can be thought of as a single, ideal operating policy with an allowable tolerance for imperfect control or implementation.

It is important that this envelope concept be applicable to multistep processes. It may then be possible to derive multiple decoupled envelopes that ensure that the overall process meets the performance requirements as long as the constituent steps are operated within their envelopes. This would

Correspondence concerning this article should be addressed to N. Shah.

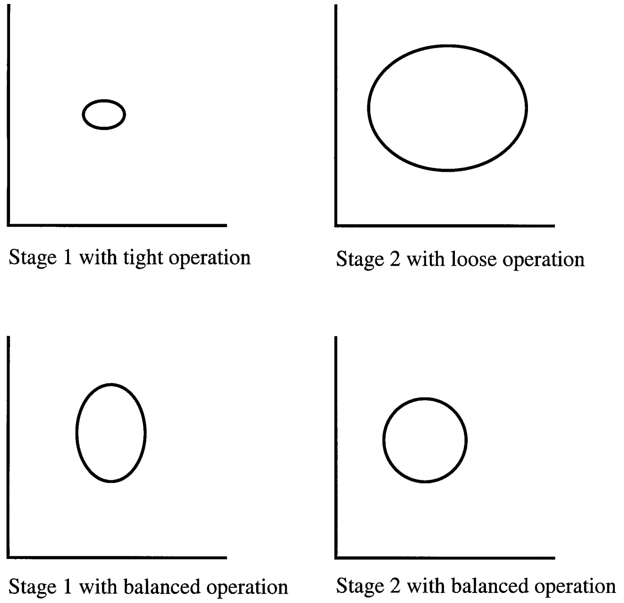
Stage 1 with tight operation
Stage 2 with loose operation
Stage 1 with balanced operation
Stage 2 with balanced operation

Figure 1. Operational envelopes for two coupled stages.

introduce a form of "robustness" into the process operation, since adaptation of downstream operating policies will no longer be required to account for deviations in upstream operation.

It is interesting to note that there is, in fact, a coupling between the dimensions of the different envelopes that is effected by the intensive and extensive properties of the materials that link adjacent stages. Trade-offs also exist between the relative dimensions of the individual envelopes for each stage and between the sizes of the envelopes of different stages. This is illustrated in Figure 1, where two possible combinations of operational envelopes are shown for a two-stage process. In the first case (upper two envelopes), stage 1 is operated using a small envelope so that stage 2 can be operated loosely; in the second case, a more balanced operating policy is adopted. Hence some specifications of desirable envelope properties should be made to resolve these potential ambiguities.

Using a hyperrectangular form, the envelope concept bears some resemblance to that of operational flexibility indices as first introduced by Swaney and Grossmann (1985). They developed a procedure to design continuous processes that are better able to cope with uncertain demands (such as market conditions, conditions of feed streams, and desired throughput). Based on the given lower and upper bounds on the uncertain parameters, the procedure maximizes the ability of the process to operate within the uncertain parameter space by selecting the most appropriate design with an operating policy that can adapt to the prevailing conditions (the actual values of the uncertain parameters at a particular time). The problem is solved by considering a nominal set of the uncertain parameters and defining a family of hyperrectangles centered on this point that have the same aspect ratio as the boundary of the uncertain parameter space. The largest of these hyperrectangles that can be enclosed by the feasible region of the process defines the flexibility index. A flexibility index of 1 corresponds to the largest hyperrectangular coinciding with the boundary of the uncertain parameter space. Assuming certain convexity conditions, and because the hyperrectangle has a fixed aspect ratio and center, each vertex can be examined in turn to find the vertices that limit the size of the hyperrectangle. A similar hyperrectangular representation has been used by Saraiva and Stephanopoulos (1998) to describe different operating regions extracted from historical data.

Much of the work in the area of operational flexibility is based on the approach of Swaney and Grossmann. Grossmann and Floudas (1987) proposed an alternative solution procedure based on an active-set strategy. Ostrovsky et al. (1994) presented several algorithms to solve the problem including an approach that approximates the flexibility index by formulating two simpler problems that give upper and lower bounds. Dimitriadis and Pistikopoulos (1995) extended the concept to cover the case of dynamic systems. They used an active-set strategy and orthogonal collocation on finite elements to discretize time in order to generate a mixed-integer nonlinear programming formulation. Their procedure automatically determines the dynamic profiles of the uncertain parameters and associated control actions that limit plant flexibility.

A different approach was used by Bahri et al. (1997), who applied a two-level optimization procedure to determine the feasible region with respect to the uncertain parameters. An outer problem optimizes performance at the nominal point, subject to constraints at a number of critical parameter scenarios. The inner problem generates additional critical parameter sets that maximize the constraint violation for a given design. The solution procedure terminates when the inner problem can no longer produce any constraint violation.

More recently, Bernardo and Saraiva (1998) considered both operational and stochastic flexibility (e.g., Pistikopoulos and Mazzuchi, 1990) simultaneously using a sampling-based optimization procedure. By treating operating variables as random distributions (whose characteristics are degrees of freedom), an operating envelope can be generated to account for the uncertainty in other parameters. The procedure minimizes an expected cost function, which consists of terms for equipment costs, operating costs ($f(u)$) and control costs $[f(\mu_u, \sigma_u)]$ as well as a Taguchi quality loss term. The algorithm comprises a stochastic simulation at the inner level, to generate function evaluations and gradients, and an outer-level NLP optimization to obtain the operating and design parameters.

Of the preceding techniques, those of Swaney and Grossmann and Bernardo and Saraiva are most closely related to this work. The approach used by Bernardo and Saraiva is particularly interesting, because it simultaneously accounts for both imperfect control and imperfect knowledge of the process. The solution approach has the advantage that large numbers of uncertain parameters and controls can be handled relatively easily with little effect on the computational costs; two disadvantages are that computational costs for small problems would be higher than in other approaches, and that the sensitivities may not be as reliable as in solution procedures that use analytical gradients. The use of control and quality costs in the objective function makes this approach somewhat difficult to apply to fine chemical processes. This is because most of the products in the fine chemicals and pharmaceuticals industries must satisfy hard quality requirements; failure to do so results in an unsalable product, so associating a cost with quality loss is both difficult and inappropriate. Also, the reliance on manual control in these industries means that there is no real relationship between control performance and cost.

The flexibility-index problem of Swaney and Grossmann aims to find the largest hyperrectangle that can be encompassed entirely by the feasible region. Although their problem definition is different from ours (considering uncertain demands on the process and finding operating policies that can *adapt* to them), the concept of determining the largest hyperrectangle is identical. However, to simplify the problem and to remove degenerate solutions, they restrict the hyperrectangle by centering it on a nominal point and fixing its aspect ratio based on some lower and upper bounds on the uncertain parameters. This removes the possibility of exploring the trade-offs between various levels of control of the individual operating variables; an aspect of the problem that is of particular interest. Furthermore, the solution of the problem depends on the nominal point and the lower and upper bounds of the uncertain-parameter space; in some cases, the nominal point could lie on the boundary of the feasible region, and the resulting flexibility index would be *inaccurately* determined to be zero. From this, we conclude that a different solution approach will be necessary to solve the problems that we pose.

The remainder of the article is structured as follows. In the next two sections, the problem formulation and solution procedure, respectively, are described. In the subsequent section, two example problems are presented: the first is a simple problem to illustrate some of the features of the approach, and the second example is a more representative problem to demonstrate the applicability of the approach to large-scale problems. In the final section, we discuss some of the key observations and conclusions of this work.

Mathematical Formulation

The formulation is based on a typical optimal control and design formulation, as illustrated below:

$$\max_{\boldsymbol{u},\,\boldsymbol{\nu},\,\tau} \Phi[\dot{\boldsymbol{x}}(\tau),\, \boldsymbol{x}(\tau),\, \boldsymbol{y}(\tau),\, \boldsymbol{u}(\tau),\, \boldsymbol{\nu},\, \tau]$$

subject to

$$\begin{aligned} &\boldsymbol{J}_0[\dot{\boldsymbol{x}}(0),\, \boldsymbol{x}(0),\, \boldsymbol{y}(0),\, \boldsymbol{u}(0),\, \boldsymbol{\nu}] = \boldsymbol{0}, \\ &\qquad \boldsymbol{h}(\dot{\boldsymbol{x}},\, \boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{u},\, \boldsymbol{\nu},\, t) = \boldsymbol{0} \qquad \forall t \in (0,\tau], \\ &\qquad \boldsymbol{g}(\dot{\boldsymbol{x}},\, \boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{u},\, \boldsymbol{\nu},\, t) \leq \boldsymbol{0} \qquad \forall t \in [0,\tau]. \qquad \text{(M1)} \end{aligned}$$

In the preceding model (M1), $\boldsymbol{x}$ are differential state variables, $\dot{\boldsymbol{x}}$ their derivatives with respect to time, $\boldsymbol{y}$ are algebraic state variables, $\boldsymbol{u}$ are time-varying controls, $\boldsymbol{\nu}$ are time-invariant optimization parameters (that may represent both design decision variables and time-invariant control variables), τ is the processing time, and t time. The constraints $\boldsymbol{J}_0$ represent the initial conditions of the system, whereas $\boldsymbol{h}$ and $\boldsymbol{g}$ are, respectively, general equality and inequality constraints. The latter may be active throughout the time horizon (path constraints) or at any particular point in time (point constraints). The majority of the equality constraints are due to the model equations; a small number will be used for various product and process specifications. The inequality constraints typically represent constraints on safety, operational limitations, product specifications, and so on.

The solution of the preceding problem results in unique profiles and values for $\boldsymbol{u}$ and $\boldsymbol{\nu}$, respectively, that maximize the objective function, Φ. However, since it will not be possible to control some of the $\boldsymbol{u}$ and $\boldsymbol{\nu}$ variables perfectly throughout the operation of the process, or from batch to batch, our approach is to find the largest region in which the process can be operated such that a certain level of performance can be guaranteed (i.e., Φ is always above a certain value). If this region in the $\boldsymbol{u}$ and $\boldsymbol{\nu}$ variables is denoted by E (the operational envelope), then the formulation can be written as follows:

$$\begin{aligned} &\max f(E) \\ &\text{subject to} \\ &\boldsymbol{J}_0[\dot{\boldsymbol{x}}(0),\, \boldsymbol{x}(0),\, \boldsymbol{y}(0),\, \boldsymbol{u}(0),\, \boldsymbol{\nu}] = \boldsymbol{0} \qquad \forall \boldsymbol{u},\, \boldsymbol{\nu},\, \tau \in E, \\ &\boldsymbol{h}(\dot{\boldsymbol{x}},\, \boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{u},\, \boldsymbol{\nu},\, t) = \boldsymbol{0} \qquad \forall \boldsymbol{u},\, \boldsymbol{\nu},\, \tau \in E,\, t \in [0,\tau], \\ &\boldsymbol{g}(\dot{\boldsymbol{x}},\, \boldsymbol{x},\, \boldsymbol{y},\, \boldsymbol{u},\, \boldsymbol{\nu},\, t) \leq \boldsymbol{0} \qquad \forall \boldsymbol{u},\, \boldsymbol{\nu},\, \tau \in E,\, t \in [0,\tau], \\ &\Phi[\dot{\boldsymbol{x}}(\tau),\, \boldsymbol{x}(\tau),\, \boldsymbol{y}(\tau),\, \boldsymbol{u}(\tau),\, \boldsymbol{\nu},\, \tau] \geq \Phi^* \\ &\qquad\qquad\qquad\qquad \forall \boldsymbol{u},\, \boldsymbol{\nu},\, \tau \in E; \quad \text{(M2)} \end{aligned}$$

where Φ^* is a specified lower bound.

In order to solve this problem, it is necessary to specify a geometric form for the envelope. Before doing this, it is more convenient to partition the optimization parameters into those that cannot be controlled adequately and those that can. The former will be denoted by the vector $\boldsymbol{b}$ and will be called "bounded" variables (i.e., we wish to have some operational bounds for flexibility in these), the latter will be denoted by the vector $\boldsymbol{a}$ and will be called "nonbounded" variables (i.e., point values for these will suffice). To make the formulation less cumbersome, the $\boldsymbol{a}$ and $\boldsymbol{b}$ variables will represent both time-invariant optimization parameters and time-dependent controls (it makes little difference since $\boldsymbol{\nu}$ is just a special case of $\boldsymbol{u}$). The envelope, E, will therefore be a region only in the space of the $\boldsymbol{b}$ variables.

A simple hyperrectangular form for the envelope is chosen for a variety of reasons. From a numerical point of view, this form of envelope means that the solution of the preceding model will be relatively easy. From a physical perspective, it makes sense to specify the envelope in terms of allowable ranges of the bounded parameters; anything more complex would be difficult to implement in practice and would require unusually complicated operating policies. A significant benefit of selecting a hyperrectangular form is that it decouples the control objectives, so that each parameter can be individually maintained within its bounds without knowing precisely what the values of the other parameters are. As a consequence, for multistage processes, each stage can be operated without any information about how previous stages were operated (except that they were operated within their envelopes); no adaptation of the operating policy (or, rather, envelope) will be required.

The envelope is now represented by the region $E \equiv \{\boldsymbol{b} | \boldsymbol{b}^{\min} \leq \boldsymbol{b} \leq \boldsymbol{b}^{\max}\}$, where $\boldsymbol{b}^{\min}$ and $\boldsymbol{b}^{\max}$ are vectors of the lower and upper limits of the ranges of the "bounded" parameters. The function $f(E)$ can then be written as a function of these lower and upper bounds. A suitable function may be:

$$f(E) \equiv \prod_{i=1}^{N_b} b_i^{\max} - b_i^{\min}, \tag{1}$$

where N_b is the number of bounded variables $[= \dim(\boldsymbol{b})]$. However, this function is highly nonconvex and could easily be substituted with a linear one. Defining Δb_i as $\Delta b_i \overset{\Delta}{=} b_i^{\max} - b_i^{\min}$, the objective function (Eq. 1) can be replaced by

$$f(E) \equiv \frac{1}{N_b} \sum_{i=1}^{N_b} \frac{\Delta b_i - \Delta b_i^{\min}}{\Delta b_i^{\max} - \Delta b_i^{\min}}, \tag{2}$$

where $\Delta b_i^{\min}$ and $\Delta b_i^{\max}$ are lower and upper bounds on the *sizes* of the ranges. The function therefore takes the form of a scaled perimeter. Lower bounds on the sizes of the ranges are required because, in practice, there will be a limit to how tightly an operating variable can be controllable. Upper bounds are applied to prevent the solution containing unrealistically large ranges. If there is a level of control (that is, the width of the range) that is trivial to maintain, this is set as the upper bound on a range so that any remaining flexibility in the process can be used to widen the range of a different parameter.

The preceding objective function therefore lies in the range [0,1], with 0 representing the smallest possible envelope (that is, $\Delta b_i = \Delta b_i^{\min}\ \forall i$) and 1 representing the largest ($\Delta b_i = \Delta b_i^{\max}\ \forall i$). The overall formulation is shown in the model below (M3).

$$\max_{\boldsymbol{a}, \boldsymbol{b}^{\min}, \boldsymbol{b}^{\max}} \quad f \equiv \frac{1}{N_b} \sum_{i=1}^{N_b} \frac{\Delta b_i - \Delta b_i^{\min}}{\Delta b_i^{\max} - \Delta b_i^{\min}}$$

subject to

$$\boldsymbol{J}_0[\dot{\boldsymbol{x}}(0), \boldsymbol{x}(0), \boldsymbol{y}(0), \boldsymbol{a}(0), \boldsymbol{b}(0)] = \boldsymbol{0} \qquad \forall \boldsymbol{b} \in [\boldsymbol{b}^{\min}, \boldsymbol{b}^{\max}],$$

$$\boldsymbol{h}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}, t) = \boldsymbol{0} \qquad \forall \boldsymbol{b} \in [\boldsymbol{b}^{\min}, \boldsymbol{b}^{\max}], \qquad t \in (0, \tau], \qquad \tau \in \boldsymbol{b},$$

$$\boldsymbol{g}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}, t) \leq \boldsymbol{0} \qquad \forall \boldsymbol{b} \in [\boldsymbol{b}^{\min}, \boldsymbol{b}^{\max}], \qquad t \in (0, \tau], \qquad \tau \in \boldsymbol{b},$$

$$\Phi[\dot{\boldsymbol{x}}(\tau), \boldsymbol{x}(\tau), \boldsymbol{y}(\tau), \boldsymbol{a}[\tau], \boldsymbol{b}(\tau), \tau] \geq \Phi^* \qquad \forall \boldsymbol{b} \in [\boldsymbol{b}^{\min}, \boldsymbol{b}^{\max}], \qquad \tau \in \boldsymbol{b},$$

$$\Delta \boldsymbol{b} = \boldsymbol{b}^{\max} - \boldsymbol{b}^{\min},$$

$$\Delta \boldsymbol{b}^{\min} \leq \Delta \boldsymbol{b} \leq \Delta \boldsymbol{b}^{\max}. \tag{M3}$$

Note that τ may be a member of the envelope variables, $\boldsymbol{b}$; that is, we may be interested in finding lower and upper bounds on the processing time (such as how important it may be to quench a reaction at a precise time).

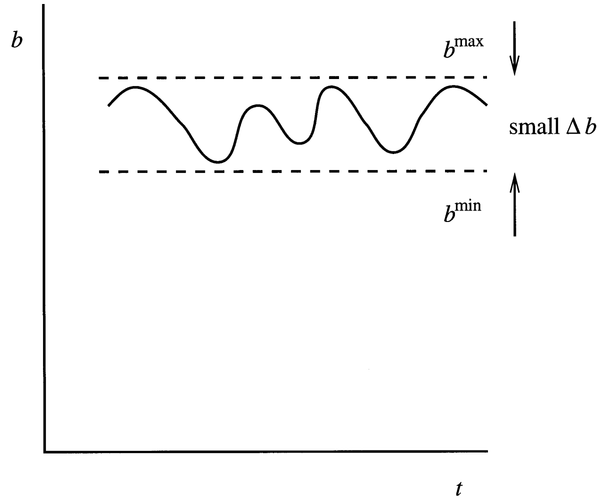

Figure 2. Example of a control variable that is held approximately constant if the bounds are also constant and Δb_i is small.

Treatment of time-varying bounded variables

In the derivation of the preceding model, no restriction was placed on the type of variable that can be "bounded." However, although both time-invariant parameters and controls can be "bounded" variables, the formulation is not well suited to controls. With a tightly bounded control variable (that is, if it has a small Δb_i), the variable may be so tightly constrained that it is essentially constant over time (see Figure 2). A more desirable approach is to allow the bounds also to vary with time, as shown in Figure 3. This can be represented mathematically as

$$\Delta \boldsymbol{b} = \boldsymbol{b}^{\max}(t) - \boldsymbol{b}^{\min}(t), \tag{3}$$

where $\Delta \boldsymbol{b}$ is bounded as before and is *constant* over time.

Clearly, to allow $\Delta \boldsymbol{b}$ to vary with time would result in better envelopes, but this level of complexity would probably not

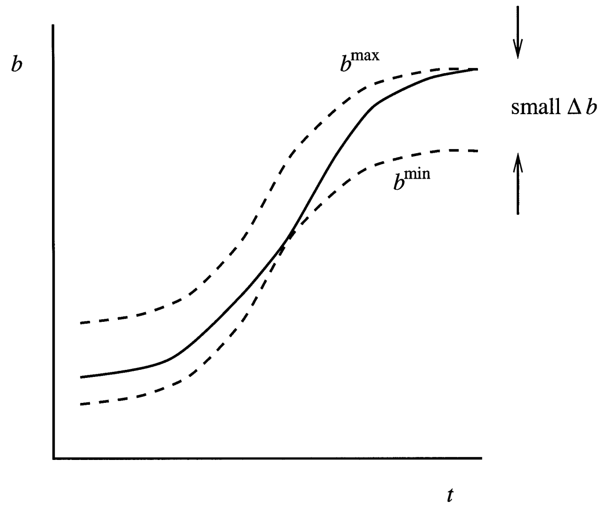

Figure 3. Example of a control variable bounded by appropriate time-varying bounds.

be adapted for use in pharmaceutical processes. For this reason, the examples considered here will only use static envelopes. However, the formulation can easily be extended to cover transient envelopes (the $\Delta \boldsymbol{b}$ variables would simply change from time-invariant parameters to controls).

This formulation essentially allows the operating envelope to be specified in terms of values or profiles (one of $\boldsymbol{b}^{\min}$ and $\boldsymbol{b}^{\max}$), for the operating variables, and associated tolerances ($\Delta \boldsymbol{b}$).

Solution Procedure

In this section, we will use the following terminology:

Scenario. Any realization of the $\boldsymbol{b}$ variables (for a given $\boldsymbol{a}$).

Vertex Scenario. Any scenario at a vertex of the envelope (that is, consisting of any combination of $b_i^{\max}$ and $b_i^{\min}$).

Interior Scenario. Any scenario within the envelope other than at a vertex (that is, at least one b_i is not at its bound);

Critical Scenario. Any scenario (usually a vertex) that is on the border of the feasible region, and hence a scenario that limits the flexibility of the system. Critical scenarios are determined by searching for points in the envelope that are outside the feasible region.

The formation presented in the previous section involves constraints that must be satisfied throughout the region (a hyperrectangle) enclosed by the bounded variables. A typical approach to solving such problems is to apply a two-level iterative procedure. At one level, one needs to consider only the vertices of the hyperrectangle plus any interior scenarios (initially none are included) in a multiscenario optimization problem (later we will show that only a very small number of vertices needs to be considered explicitly). In the second level, a single-scenario optimization problem is solved to determine an additional interior scenario by determining the point within the hyperrectangle with the highest constraint violation. The procedure converges when no more scenarios with constraint violation can be found at the second level.

The preceding procedure can be simplified to a single multivertex optimization problem if the feasible region can be shown (or is assumed to be) convex. In the case of the flexibility-index problem, Swaney and Grossmann (1985) showed that, under certain conditions, the points that limit flexibility always lie at the vertices. The multivertex optimization may also be simplified: since their hyperrectangle is always scaled about a nominal point, the multivertex optimization problem can be reduced to solving all of the vertices individually (maximizing the distance from the nominal point while remaining feasible) and finding the one that is closest to the nominal point. Finally, by employing a heuristic search strategy, only a small fraction of the total number of vertices needs to be examined.

In our case, however, it is vital that the hyperrectangle is neither centered about a nominal point nor restricted to a constant aspect ratio. Since these are the properties that allow the previous simplifications to be made, we must solve both levels of the procedure without being able to simplify the multivertex optimization (or multiscenario, if interior scenarios are considered) as just described.

Since the overall procedure of iterating between solving a multiscenario problem and finding additional interior scenarios has been described in a number of earlier publications (Mohideen et al., 1996; Bahri et al., 1997), we shall not discuss it further here. One point worth noting, however, is that the interior scenarios must be specified *relative* to the vertices of the hyperrectangle, rather than as *absolute* points in the bounded-variable space (as is usually the case, as in, for example, robust optimization procedures). This is because specifying an interior scenario, which by definition is outside the feasible region, in absolute terms will force the optimization to move the feasible region (through selection of the design and operating parameters) until the interior scenario is also feasible. Assuming this is possible (if not, the optimization will be infeasible and the overall solution procedure will fail), it will result in a smaller envelope (because the system is now more tightly constrained) and in no way addresses the original problem of a nonconvex feasible region. This is illustrated in Figure 4 where (a), (b) and (c) represent the first three iterations of a solution procedure using relative scenarios, and (a) and (d) represent the first two iterations of the same problem solved using absolute scenarios.

It is more important to discuss the means for solving the multivertex optimization problem, since this will have the greatest computational impact on the overall procedure. Clearly, since there will be 2^{N_b} vertices in the optimization problem (where N_b is the number of bounded variables), most problems of industrial significance will become prohibitively expensive to solve, if they are at all tractable. Naturally, some form of decomposition will be needed in these cases.

The decomposition method applied here is based on the observation that very few of the 2^{N_b} vertices will be critical scenarios. The basis of the decomposition is to find these critical vertex scenarios so that only the minimum number of vertices need to be considered in the optimization. The problem is solved using an iterative procedure involving a reduced-vertex optimization and a full-vertex simulation (that is, each vertex is simulated) to identify the critical vertices.

The procedure begins by solving an optimization problem consisting of any pair of opposite vertices. This is the minimum number of vertices that can be included in the optimization. Since there are only two vertices, they must also be opposite vertices of the envelope. This is because each of the lower and upper bounds (which are degrees of freedom and of which the objective function is composed) of the bounded variables must be represented in at least one vertex; if not, any unrepresented bounds would have no influence on the constraints in the system (that is, that each vertex should lie within the feasible region) and would therefore result in an overestimation of the envelope size. Then a full-vertex simulation is performed to identify a second pair of vertices to replace the first. At this stage, the aim is to identify the pair of vertices that leads to the smallest envelope (that is, the best upper bound on the envelope size). The process is then repeated with each simulation identifying *additional* vertices to be included in the optimizations. When no further vertices can be identified in the simulation step (that is, none of the vertices are violated), the procedure is terminated and the solution is found. The overall procedure is described in more detail below and illustrated in Figure 5.

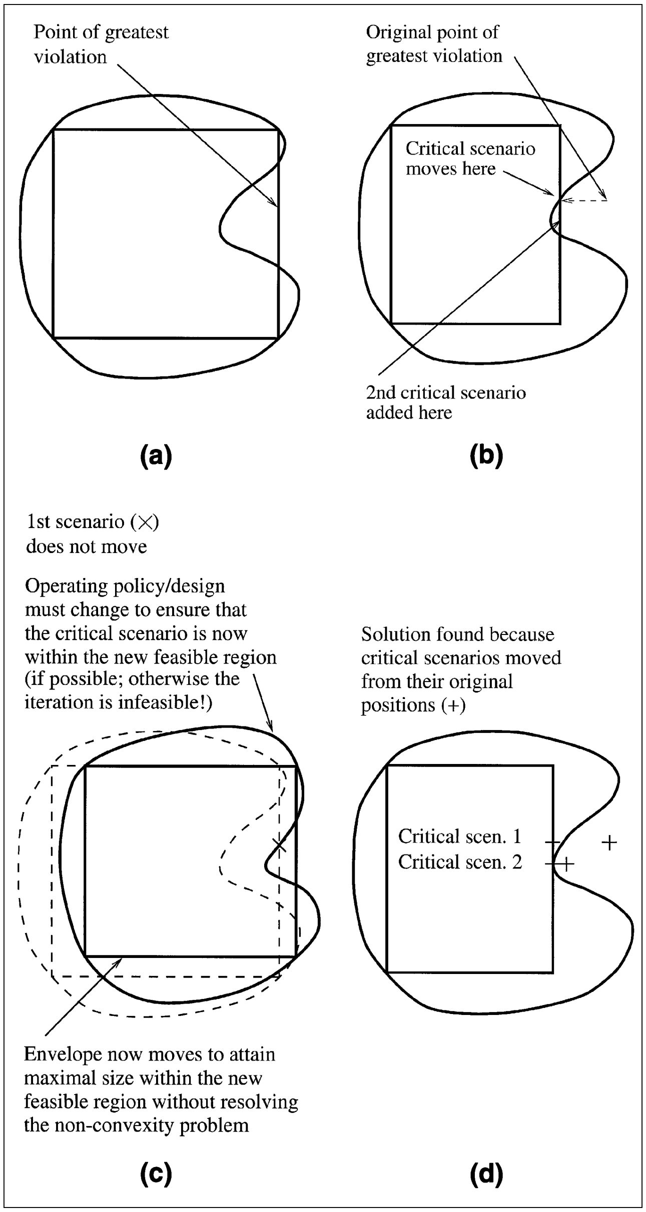
Point of greatest violation
Original point of greatest violation
Critical scenario moves here
2nd critical scenario added here
(a)
(b)
1st scenario (×) does not move
Operating policy/design must change to ensure that the critical scenario is now within the new feasible region (if possible; otherwise the iteration is infeasible!)
Solution found because critical scenarios moved from their original positions (+)
Critical scen. 1
Critical scen. 2
Envelope now moves to attain maximal size within the new feasible region without resolving the non-convexity problem
(c)
(d)

Figure 4. Importance of adding internal scenarios *relative* to vertices of the envelope.

(a) First iteration: solve only for vertices; (b) second iteration: add first critical scenario as relative points in b space; (c) second iteration: add first critical scenario as absolute point in b space; (d) third iteration: add second critical scenario relative to vertices.

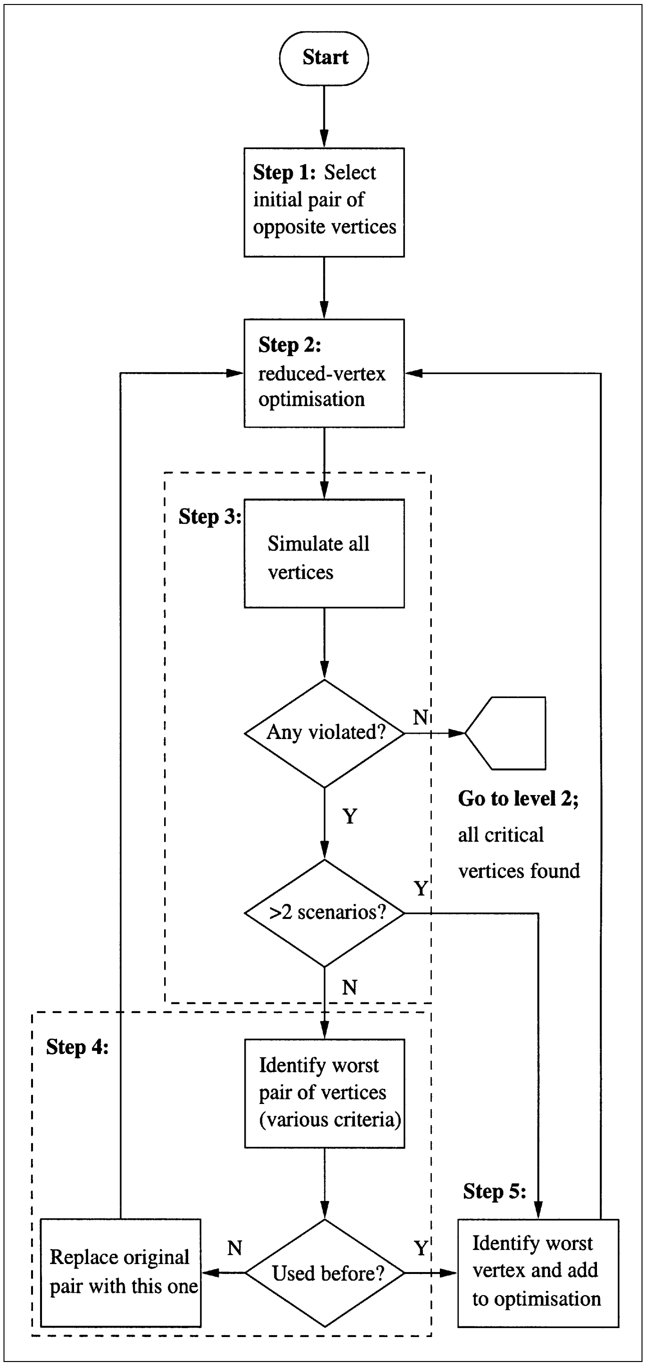
Start
Step 1: Select initial pair of opposite vertices
Step 2: reduced-vertex optimisation
Step 3:
Simulate all vertices
Any violated?
N
Go to level 2; all critical vertices found
Y
>2 scenarios?
Y
N
Step 4:
Identify worst pair of vertices (various criteria)
Step 5:
Replace original pair with this one
N
Used before?
Y
Identify worst vertex and add to optimisation

Figure 5. Solution procedure level 1.

1. Select an initial pair of opposite vertices.

2. Solve a reduced-vertex optimization to obtain an estimate of the envelope.

3. Simulate each vertex of the envelope just generated, sequentially or in parallel, to identify all vertices with violation. Stop or go to level 2 if there are no vertices with violation; otherwise, go to step 5 if there are currently more than two scenarios in the reduced-optimization problem.

4. Identify a pair of vertices to be used within the model, with a view to replacing the current pair. This is either the pair consisting of the vertex with the worst violation and its opposite or the pair of opposite vertices with the worst combined (summed) violation. If the selected pair has already been used in a previous iteration, then go to step 5; otherwise, replace the current pair with the selected pair and go to step 2.

5. Select the vertex with the highest violation and add it as an additional vertex to the current optimization model. Go to step 2.

Various criteria may be used to select vertices for inclusion in the reduced-vertex optimization. In the example problems presented later, we selected the initial pair corresponding to one vertex with all upper bounds and its opposite vertex (all lower bounds). After the first simulation, the next pair was selected by taking the vertex with the worst violation along with its opposite vertex. An alternative criterion would have been to select the pair with the worst combined violation. After the second iteration, individual vertices were added based on those with the worst violation. A final note is that once there are more than two vertices in the optimization model, it may no longer be necessary to include a pair of opposite vertices. However, for convenience, we chose to retain at least one pair of opposite vertices even if one of them was not critical.

The procedure just described essentially represents the first multivertex optimization. When this has been solved, the next step is to identify interior scenarios (relative to the vertices) to add to the multivertex optimization. Because all of the critical vertices will have been identified in the first iteration, subsequent iterations (that also include interior scenarios) will not require the full-scenario simulation step. If any new critical vertices arise due to the inclusion of the interior scenarios, then they should be identified in the single-scenario optimization that is used to identify the interior scenarios. This is illustrated in Figure 6.

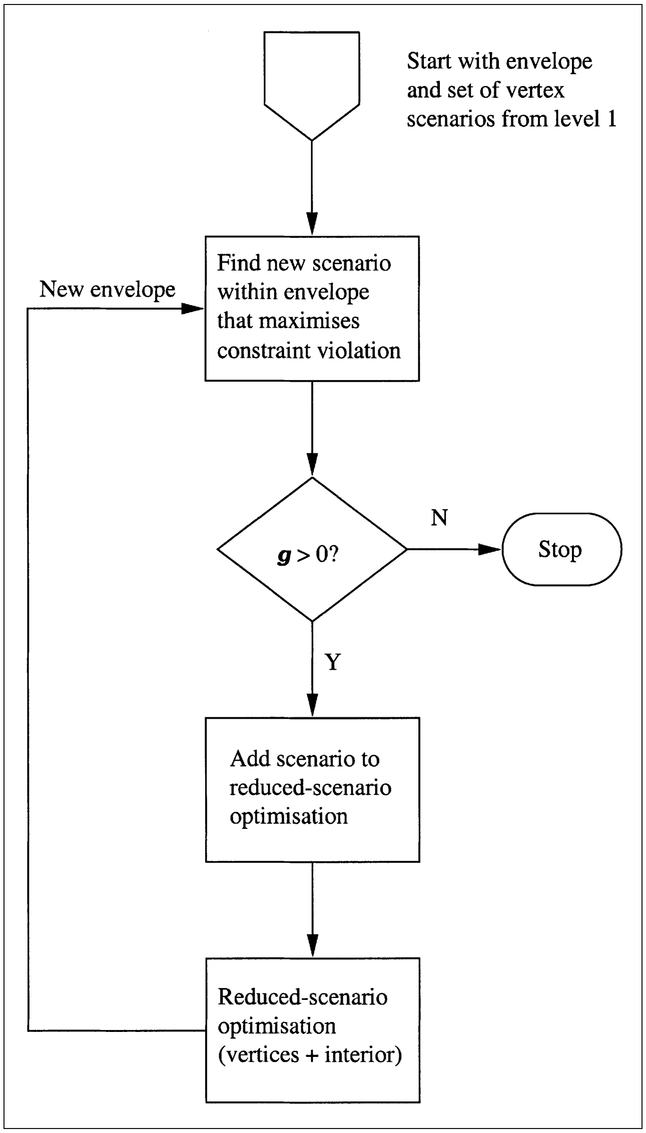

Figure 6. Solution procedure level 2.

An alternative solution procedure

The solution procedure described earlier relies on a series of large-scale simulation problems in combination with smaller optimization problems to solve the first multivertex optimization problem. For problems with a large number of bounded variables, the simulation step can take longer than the reduced-vertex optimizations. A quicker approach may be to identify the critical vertices using the same optimization problem that identifies interior critical scenarios.

The procedure would start with a two-vertex optimization and then use optimization to determine the point within the envelope with the greatest violation. This is likely to be at a vertex, and so would be used to replace the previous pair (as was described earlier); if not, the interior scenario would simply be added to the original two vertex scenarios. In subsequent iterations, critical scenarios (vertex or interior) would be identified and added to the reduced-scenario optimization. The procedure would terminate when no violation can be found within the envelope. This algorithm would be very similar to the second step in the previous approach (Figure 6).

The advantage of this approach is that the overall procedure may be quicker than the procedure described before. This depends on how quickly the single-scenario optimization can be solved in comparison with the full-scenario simulation. One possible disadvantage is the reliance on *global* optimality in identifying critical scenarios. For this reason, we chose not to compare this approach with the one described earlier.

Computational Results

The process modeling tool *gPROMS* (Process Systems Enterprise Ltd., 1997) was used to implement the envelope procedure described previously. The *gPROMS* modeling language allows existing models of processes to be converted to the envelope form quickly and easily. The optimizations were performed using the *gOPT* dynamic optimizer within *gPROMS*.

The problems were run on a variety of Sun workstations. Ultra-1 workstations (with a 143-MHz UltraSPARC processor and 96 MB or 256 MB RAM) were primarily used for smaller problems, whereas Ultra-60 workstations (2×360-MHz UltraSPARC-II CPU, 512 MB RAM) were used for the more computationally intensive problem.

Two examples are presented here: the first is a reactor model that is simple enough to illustrate the key features of the approach; the second is a more realistic case (involving three processing steps) to demonstrate the reduced-scenario solution procedure. The details of the two example problems are described in the following two subsections.

Example problem 1

This first example problem is a very simple, hypothetical batch region process and is used purely for illustrative purposes. In the next subsection, we consider a more realistic example and apply the reduced-scenario approach to solve the problem.

The following reactions are to take place in a reactor of fixed volume with a given initial charge of a solution of A:

$$2A \rightarrow B \rightarrow C.$$

The temperature (T) and processing time (τ) can be adjusted to maximize the production rate of B, the desired product, subject to a lower-bound constraint on the purity of B. In a traditional optimal control problem (as in model M1), the optimization parameters might be temperature and processing time, for which *point* values (or a profile for T) will be determined. In our case, we are interested in *ranges* of these quantities. More specifically, we want to find the largest envelope in T and τ such that the production rate is guaranteed to meet a minimum specification and that the purity is guaranteed to meet the lower-bound constraint.

To make the visualization of the results simple, we use the *average* temperature as a bounded variable, relating it to the control variable T. This is, of course, not particularly realistic or useful in practice, but it reduces the envelope to a static 2-dimensional (2-D) rectangle (rather than a dynamic 2-D envelope that would result if the time-varying T was a bounded variable). The feasible region may also be plotted in the 2-dimensional space of the average temperature and the processing time.

The solution procedure is as follows. The first optimization problem consists of the two vertices corresponding to T and τ at their lower bounds ("LL') and both at their upper bounds ("UU'). Once this has been solved, all four vertices are checked using simulation. This shows that both other vertices (LU and UL) fail to meet one or more constraints. The next iteration is to solve a new optimization problem with the two vertices LU and UL. Finally, the new envelope is tested using simulation, which in this case shows that all four vertices are now within the feasible region. The final stage is to check for interior critical vertices, but since a plot of the feasible region is available for this case, it can be shown graphically that there will be no interior critical scenarios. For larger problems, it will not be possible to obtain a plot of the feasible region, so the second step of the algorithm will need to be performed. This was done in this example and, as would be expected, no constraint violation could be found within the envelope.

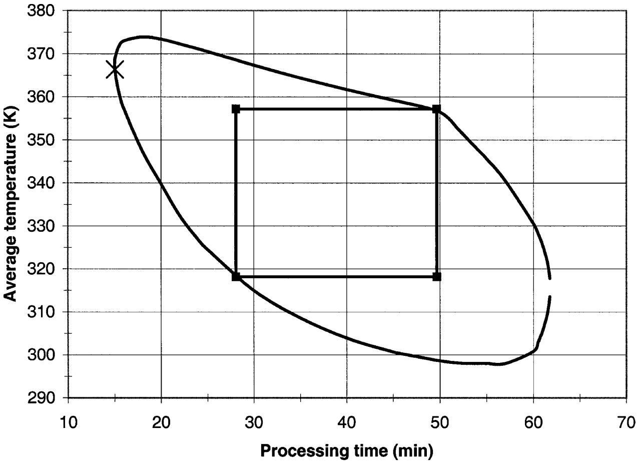

Figure 7. Results of envelope and standard optimizations, compared with feasible region.

The results of two optimization problems are shown in Figure 7 along with the feasible region for this problem. The graph shows the result of a standard optimization problem (indicated by the cross on the graph), with the objective of maximizing the production rate, in comparison with the envelope generated using the formulation presented earlier. The envelope and standard optimization were solved in 21.4 s and 4.0 s, respectively, on an Ultra-1 workstation.

As can be seen, the envelope expands to fill as much of the feasible region as possible without the need of a nominal point. Since the feasible region is very nearly convex, there would be no need for a further optimization to determine any critical points within the envelope that result in constraint violations. However, because the average temperature is used as a bounded variable, rather than the temperature profile, it is possible to violate constraints even if the average temperature is within the envelope. This shows that controls would need to be treated more carefully in a real case.

One final point to note about the results of this example is that the optimal operating *point* lies on the border of the feasible region. If this point were used as a nominal point about which to expand our envelope, then by using the flexibility-index analysis, the result would be an envelope of zero size (that is, a point operating policy), when clearly there is a great deal of flexibility in this system.

It is also possible to perform a series of envelope optimizations in order to generate a trade-off curve comparing envelope size with guaranteed process performance (Φ^*). The graph in Figure 8 shows such a curve, produced by using the same process as used earlier, but now with four bounded variables: T and τ as before, plus the initial charge and concentration of the solution of A. Now, the temperature is treated as a time-varying bounded variable with constant bounds. It has been verified, with these results, that no operating policy exists within the envelope that violates any constraints. A profit-rate function, instead of the production rate, is used as the measure of process performance, and the con-

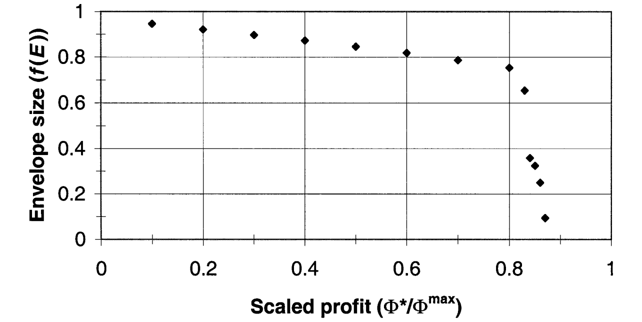

Figure 8. Result of a parametric optimization to determine the trade-off between envelope size and guaranteed performance.

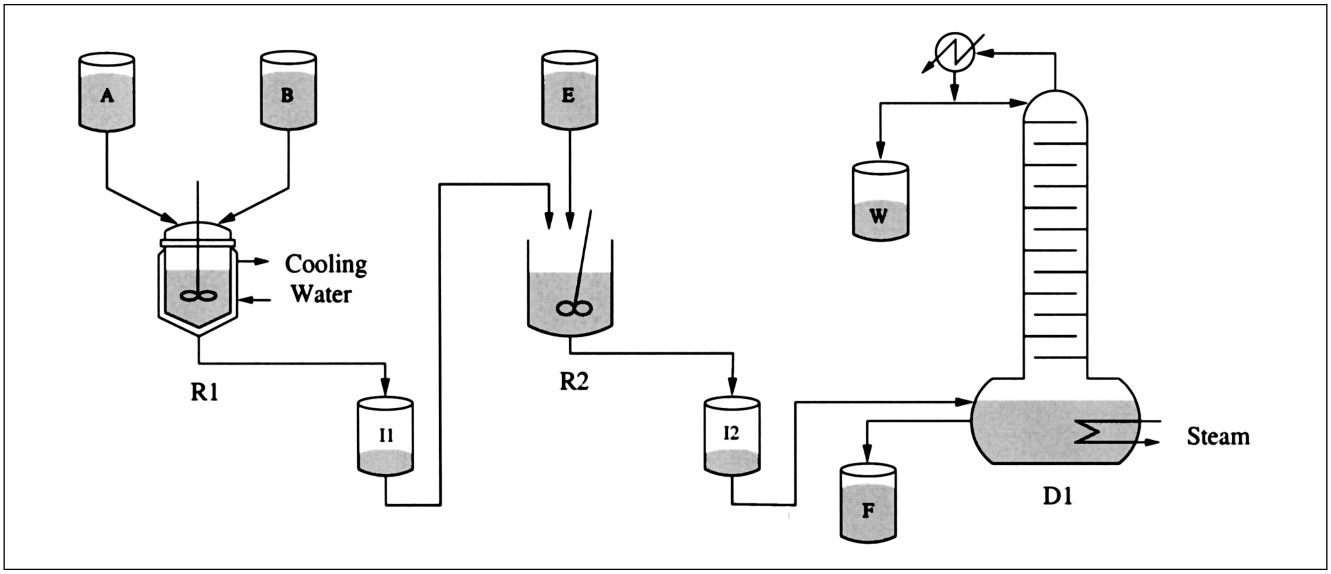

Figure 9. Process of Example 2.

straints on this are expressed as a fraction of the maximum profit rate attained from a point operating policy ($\Phi^{\max}$).

The results are quite interesting and show that with quite a loose envelope the process can be "guaranteed" to generate a profit rate of up to 80% of the maximum possible profit. Achieving anything more than this requires far tighter envelopes, and the maximum guaranteed profit rate would seem to be slightly less than 90%. This is to be expected because, as defined here, an envelope size of zero relates to the smallest envelope achievable (that is, the lower bounds on the ranges of the bounded variables are all active, and these are usually nonzero), not an operating point.

Example problem 2

The second example problem, taken from Sharif (1999), consists of two batch reaction tasks and a batch distillation task. The process is shown in Figure 9. The first reactor (R1) which is fitted with a cooling jacket, receives initial charges of reactants A and B and, potentially, an additional flow of B during the reaction. At the end of the first task, the contents are transferred to the second (adiabatic) reactor (R2) along with an initial charge of reactant E. Finally, the mixture resulting from the second reactor is separated in the batch distillation unit (D1) to produce the desired product F at a purity of no less than 96% and a waste state W. The reaction scheme is shown below.

$$A + B \rightarrow 2C$$

$$C \rightarrow D$$

$$C + E \rightarrow F.$$

The degrees of freedom and constraints on the process are summarized in Table 1.

The objective of the problem is to ensure that the process can achieve a profit rate (which comprises terms for the product value, raw material and utility consumption, and

Table 1. Summary of Degrees of Freedom and Constraints for Example 2

	Degrees of Freedom		
Unit	Bounded	Nonbounded	Constraints
R1	Initial charge of A Initial charge of B Feed profile of B Cooling-water flow rate Processing time	Volume Idle time	Lower and upper bound path constraints: • on liquid volume • on temperature
R2	Initial charge of E Processing time	Volume Idle time	As R1
D1	Reflux ratio profile Reboiler heat-load profile Processing time	Reboiler volume Idle time	Final mole fraction of F ≥ 0.96
Overall			Production rate between 5 and 8 kmol/h Profit rate $\geq$ 60% of max. profit rate obtained using a point operating policy

Table 2. Statistics for the Second Example Problem (all on an Ultra-60)

Iter.	Optimi. time	Scenario	Simul. time	Envelope Size	Violation
1	78,020 s (21.6 h)	2	50,940 s (14.2 h)	0.860752	719
2	44,470 s (12.4 h)	2	51,250 s (14.2 h)	0.678705	3
3	53,220 s (14.8 h)	3	54,190 s (15.1 h)	0.712289	2

equipment capital cost) of at least 60% of the maximum attainable profit through perfect operation of the process (a constraint of the form $\Phi \geq \Phi^*$).

Since there are ten bounded variables, the resulting formulation will consist of $2^{10} = 1024$ scenarios. Furthermore, because the single-scenario simulation consists of 598 variables (both algebraic and differential), an optimization containing all 1024 scenarios would consist of more than 600,000 variables. A problem of this size is, at best, very difficult to solve. Using the solution procedure outlined previously, it is possible to solve this problem with a much smaller number of scenarios.

The overall problem took three major iterations to converge. The pair of vertices for the first iteration was selected arbitrarily (the vertex with all bounded variables at their lower bounds, along with its opposite vertex). As is summarized in Table 2, the optimization problem took nearly 22 h to solve on a Sun Ultra-60 workstation and results in a solution that has 719 violated scenarios. Here, we only consider the main constraints of profit rate, production rate, and purity; there were many violations in the temperature and volume constraints, but their magnitudes were not significant at this stage. The number of violated vertices was determined through the simulation step.

The pair of vertices for the second iteration was selected simply by considering the vertex with the worst profit rate along with its opposite vertex; the violations in the profit rate were proportionately greater than any other. After the second iteration, it can be seen that the number of violated vertices has been reduced dramatically (see Table 2). In fact, the remaining violations are within the accuracy of the optimization. For good measure, it was decided that one extra vertex should be added to account for violations in the other constraints.

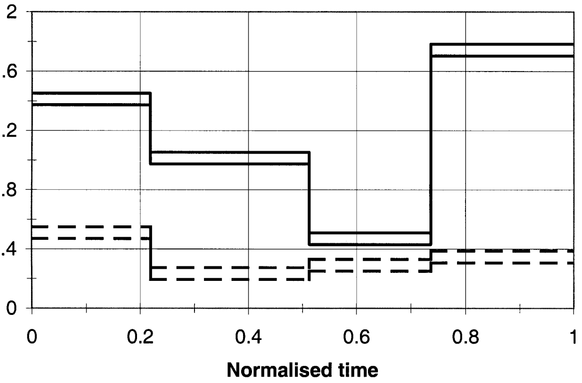

Figure 10. Optimal lower and upper bounds (b_i^{min} and b_i^{max}) for the cooling-water flow rate in reactor 1 (solid lines [tonnes/h]) and the feed flow rate of B to reactor 1 (dashed lines [kmol/h]).

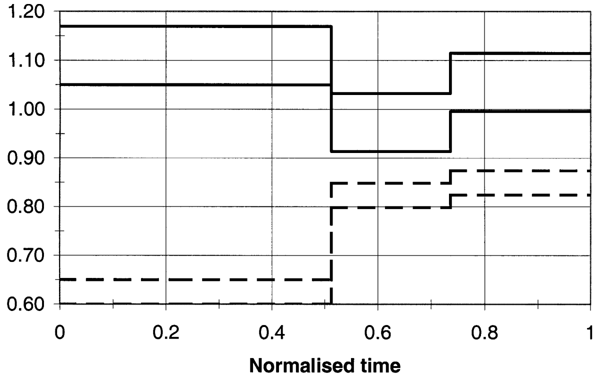

Figure 11. Optimal lower and upper bounds (b_i^{min} and b_i^{max}) for the heat-load profile in the reboiler (solid lines [MJ/h]) and reflux ratio (dashed lines).

After the final iteration, none of the vertices is violated beyond the tolerance of the optimization. The two remaining violations shown in Table 2 involve the purity constraint. Since the purity is no lower than 95.97% (see Figure 13), it is considered that the problem has converged satisfactorily.

A number of single-level optimizations were performed to check for any violation of the constraint *within* the envelope (as depicted in Figure 6). Since none of these optimizations were able to locate a critical scenario (point within the envelope with constraint violation), no further iterations were required. It is again worth noting that, since only a local-optimization algorithm was used, there is no guarantee that the process is entirely feasible within the envelope. Another consequence of the use of a local-optimization algorithm is that, in this example, the envelope size did not decrease monotonically with each iteration, as would be expected.

In total, the problem took about 92 h to solve (this compares with 1.3 for the single-scenario optimization). Each simulation stage took around 14 h to solve, and the optimizations took between roughly 12 and 22 h. Although the first and second optimizations were the same size, the second required nearly half the time to solve. This is probably due to better starting points being used in the second iteration. Similarly, the third optimization problem took considerably less time than the first despite being larger (three scenarios rather than two).

It might be expected that as the number of bounded variables increases, the simulation stages would represent the greatest computational burden. For very large numbers of bounded variables, it may then be worth considering the alternative solution procedure described previously. However, this of course relies on the ability of the optimization algorithm to obtain the global optimum. For this example, the optimization software used was not suitable for the alternative solution procedure.

The solution to the problem is shown in Tables 3 and 4, and Figures 10 to 12. Table 3 shows the optimal values of all of the time-invariant parameters, including both ***a*** and ***b***^{min}

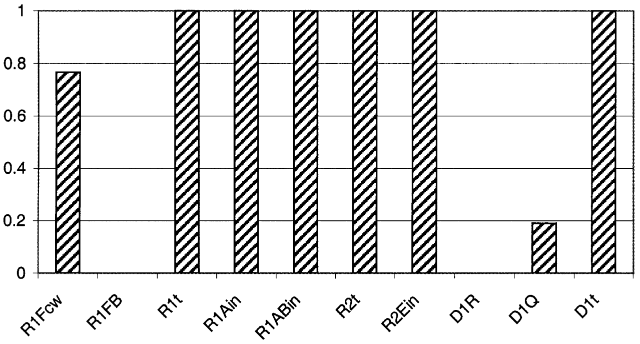

Figure 12. Contributions of the individual ranges to the objective function $[(\Delta b_i - \Delta b_i^{min})/(\Delta b_i^{max} - \Delta b_i^{min})]$.

The variables shown are, from left to right: for reaction 1: cooling-water flow rate, flow rate of B, processing time, initial charge of A, initial charge of B; for reaction 2: processing time, initial charge of E; for the distillation: reflux ratio, reboiler heat duty, processing time.

Table 3. Optimal Values and Ranges for the Time-Invariant Optimization Parameters for Example 2

Unit	Deg. of Freedom	Optimal Value/Range
R1	Initial charge of A	14.5–14.6 kmol
	Initial charge of B	9.31–9.41 kmol
	Processing time	2.93–3.03 h
	Volume	2.26 m^3
	Idle time	0.496 h
R2	Initial charge of E	13.2–13.3 kmol
	Processing time	3.07–3.17 h
	Volume	3.34 m^3
	Idle time	0
D1	Processing time	0.778–0.785 h
	Reboiler volume	1.31 m^3
	Idle time	0.0532 h

variables. Table 4 and Figure 12 show the optimal values of the $\Delta \boldsymbol{b}$ variables, along with their lower and upper bounds. Figures 10 and 11 show the control profiles ($\boldsymbol{b}^{min}$ and $\boldsymbol{b}^{max}$).

Table 4. Optimal Values of the Δb_i Variables, with Their Lower and Upper Bounds

Deg. of Freedom	Δb_i^{min}	Δb_i	Δb_i^{max}
Reactor 1			
Cooling-water flow rate (tonnes/h)	0.01	**0.08**	0.1
Flow rate of B (kmol/h)	0.08	**0.08**	0.8
Processing time (h)	0.02	**0.1**	0.1
Initial charge of A (kmol)	0.02	**0.1**	0.1
Initial charge of B (kmol)	0.02	**0.1**	0.1
Reactor 2			
Processing time (h)	0.02	**0.1**	0.1
Initial charge of E (kmol)	0.02	**0.1**	0.1
Distillation			
Reflux ratio	0.05	**0.05**	0.1
Reboiler heat duty (MJ/h)	0.1	**0.119**	0.2
Processing time (min)	0.18	**1.02**	1.02

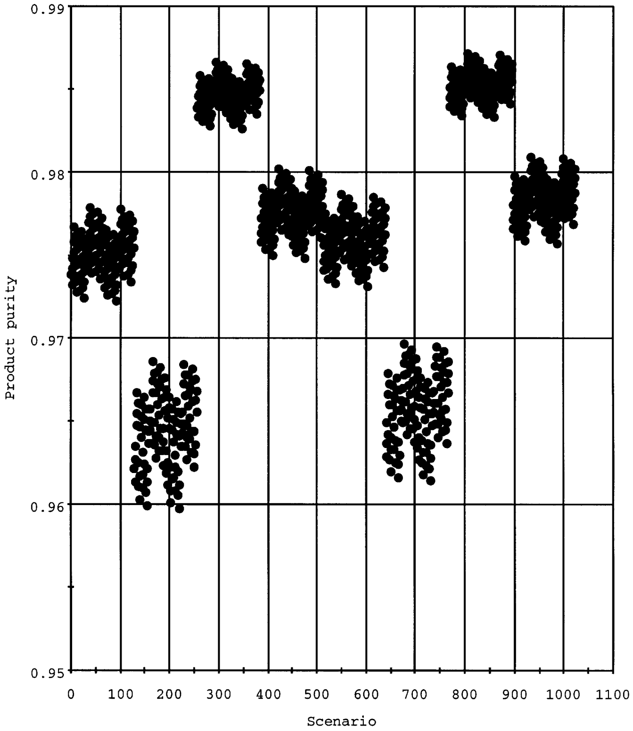

Figure 13. Product purity for all vertices.

Figure 10 shows the flow rates of cooling water in the jacket of reactor 1 and B fed to reactor 1. Figure 11 shows the reflux ratio profile and the heat-load profile for the column reboiler.

Table 4 and Figure 12 are particularly interesting because they show that a number of $\Delta \boldsymbol{b}$ variables are on their bounds. This does not indicate that the problem is too tightly bounded, and that the bounds should be relaxed. As was discussed earlier, the lower bounds represent the best level of control that can be achieved, and therefore relaxing them would be unrealistic; whereas, the upper bounds represent a level of control that can be achieved with little effort, and there would be no point in relaxing them. What these figures show is that, because many variables are on their upper bounds, the desired level of performance can be guarateed with a relatively large envelope. This can also be seen by examining the constraints for each scenario (not shown). Rather than relaxing any of the bounds, it would be more useful to solve the problem with a higher performance requirement. If this were done a number of times, then a graph similar to Figure 8 could be generated and the trade-off between guaranteed performance and control effort could be evaluated.

Finally, Figure 13 shows a typical result of a full-vertex simulation. In this case, the product purity is shown for each vertex (represented by a dot in the diagram), and it can be seen that the constraint on purity is satisfied (to within the accuracy of the model and the tolerance of the optimization) for all vertices. Similar graphs were obtained for the other constraints, which were satisfied relatively easily.

Concluding Remarks

In this article, we have outlined an approach to process design where the operational aspects are treated in terms of ranges rather than points. Recognizing that it is not possible to operate batch chemical plants at a precise point, we aim to find the range of operating parameters within which the process can be operated successfully (that is, feasibly and profitably). This facilitates the design of processes that are less sensitive to control variability; a property that is particularly suitable for pharmaceutical and speciality chemical processes, which are normally operated manually.

The overall problem has been formulated as a multistage dynamic optimization problem. A novel solution procedure has been presented that efficiently solves the problem by iterating between a reduced-vertex optimization and a full-vertex simulation. Nonconvexities in the feasible region can be accounted for by using a single-scenario optimization to identify critical interior scenarios. The efficiency of the approach has been demonstrated successfully by using two examples.

Acknowledgments

This research forms part of a wider project for improving process development in the fine, speciality, and pharmaceutical industries. Details can be found at the web site: http://www.britest.co.uk.

We are particularly grateful to the U.K. Engineering and Physical Sciences Research Council, who partly supported this work (grant GR/L63020).

Literature Cited

Bahri, P. A., J. A. Bandoni, and J. A. Romagnoli, "Integrated Flexibility and Controllability Analysis in Design of Chemical Processes," *AIChE J.*, **43**, 997 (1997).

Bernardo, F. P., and P. M. Saraiva, "Robust Optimization Framework for Process Parameter and Tolerance Design," *AIChE J.*, **44**, 2007 (1998).

Dimitriadis, V. D., and E. N. Pistikopoulos, "Flexibility Analysis of Dynamic Systems," *Ind. Eng. Chem. Res.*, **34**, 4451 (1995).

Grossmann, I. E., and C. A. Floudas, "Active Constraint Strategy for Flexibility Analysis in Chemical Processes," *Comput. Chem. Eng.*, **16**, 675 (1987).

Hansen, J. E., S. B. Jorgensen, J. Heath, and J. D. Perkins, "Control Structure Selection for Energy Integrated Distillation Columns," *J. Proc. Control*, **8**, 185 (1998).

Mohideen, M. J., J. D. Perkins, and E. N. Pistikopoulos, "Optimal Design of Dynamic Systems Under Uncertainty," *AIChE J.*, **42**, 2251 (1996).

Ostrovsky, G. M., Y. M. Volin, E. I. Barit, and M. M. Senyavin, "Flexibility Analysis and Optimization of Chemical Plants with Uncertain Parameters," *Comput. Chem. Eng.*, **18**, 755 (1994).

Pistikopoulos, E. N., and T. A. Mazzuchi, "A Novel Fleixibility Analysis Approach for Processes with Stochastic Parameters," *Comput. Chem. Eng.*, **14**, 991 (1990).

Process Systems Enterprise Ltd., "*gPROMS* Introductory User Guide," Tech. Rep., London, U.K. (1997).

Saraiva, P. M., and G. Stephanopoulos, "Process Improvement: An Exploratory Data Analysis Approach Within an Interval-Based Optimization Framework," *Prod. Oper. Manage.*, **7**, 19 (1998).

Sharif, M., "Design of Integrated Batch Processes," PhD Thesis, The Univ. of London, London, U.K. (1999).

Swaney, R. E., and I. E. Grossmann, "An Index for Operational Flexibility in Chemical Process Design," *AIChE J.*, **31**, 621 (1985).

Manuscript received Mar. 27, 2000, and revision received Dec. 20, 2000.