DESIGN OF COMPLEX SYSTEMS UNDER UNCERTAINTY USING PROCESS SIMULATORS

A.J. Olvera and J. Acevedo[1]
Department of Chemical Engineering
Instituto Tecnologico y de Estudios Superiores de Monterrey
Monterrey, N.L., Mexico 64849

Abstract

In this work, a two-stage stochastic programming approach is implemented in a commercial simulator. A hybrid algorithm is proposed, where the first-stage decisions (existence of process units and their corresponding design parameters) are handled by a genetic algorithm, while the second-stage decisions (optimization of operational variables such as flows and temperatures) are optimized through the built-in optimization tool of Aspen Plus$^{©}$. In this way, a number of individuals (possible values of the first-stage variables) are defined, selected and combined through genetic operators, while the second stage variables are modified for each individual and different realizations of the uncertain parameters through a mathematical programming code (SQP) to minimize its expected cost. Given the complexity of the optimization problem under uncertainty, several strategies are proposed to minimize the computational requirements of the solution procedure. These strategies resulted in the reduction of up to 75% in CPU time for problems involving the optimization of complex separations systems.
.

Keywords

Optimization under uncertainty, Genetic Algorithms, Process Simulators..

Introduction

Chemical processes are usually involved with uncertain conditions during operation. Overdesign factors normally cope with this situation, but the increase in capital cost is rarely justified. During the past decades, several works have been proposed to address process synthesis and design problems under uncertainty, usually through a two-stage stochastic programming approach (Grossman et al, 1983; Acevedo and Pistikopoulos, 1998; Sahinidis, 2003) where structural decisions, which remain fixed once selected, are differentiated from operational variables, which can be adjusted during operation to achieve feasibility.

On the other hand, sequential modular simulators are now widely accepted tools that can greatly simplify the development of an accurate model that represents a chemical process. In addition to sophisticated, rigorous models of a number of unit operations and thermodynamic properties, advanced simulators, such as Aspen Plus$^{©}$, include optimization routines that have become robust for the continuous optimization of operational variables.

A number of works using process simulators have been presented where different strategies are proposed for the optimal design of chemical process. However, the computational structure of modular simulators has not allowed the complete incorporation of the latest improvements in mathematical optimization algorithms, usually based on the evaluation of the gradients of the model, information that is not directly available in the simulator. Consequently, some successful strategies are based on genetic algorithms, which have been coupled with general and dedicated simulators for the optimal design of heat exchangers (Tayal et al, 1999) and complex separations systems (Leboreiro and Acevedo, 2004).

Genetic algorithms (GA) are stochastic methods based on the idea of evolution and survival of the fittest. In a GA, a set of values of the optimization variables forms an individual; the algorithm starts generating a random set of

---

[1] To whom correspondence should be addressed. Email: jacevedo@itesm.mx

individuals to form a population and then repetitively evolving it through three basic genetic operators: selection, crossover and mutation.

In this paper, a methodology for optimization under uncertainty using process simulators and genetic algorithms is presented. A hybrid algorithm is proposed for the solution of a two-stage stochastic formulation. The first stage decisions are handle by a genetic algorithm, while the second stage decisions are optimized through the built-in optimization tool of Aspen Plus$^{©}$. In this way, a number of individuals (values of the first stage variables such as design parameters) are defined, selected and combined through genetic operators, while the second stage variables (such as flows, concentrations, etc.) are modified for each individual through a mathematical programming code to minimize its expected cost. The applicability of the proposed approach is demonstrated through the optimization of complex separation systems where extractive and integrated distillation columns are considered.

Two-Stage Optimization programing

Under the presence of uncertainty, the process synthesis and design problems can de formulated by defining an expectancy of the objective function as follows:

$$\min_{y,d} \left\{ E_q \left\{ \min_z C(y,d,z,x,q) \right\} \right\}$$

$$st.$$
$$h(y,d,z,x,q) = 0$$
$$g(y,d,z,x,q) \leq 0 \tag{1}$$
$$d \in D, z \in Z, x \in X, y \in \{0,1\}^m$$
$$q \in \Theta$$

where (y,d) are the structural and design variables and (z,x) are the operational (control and state) variables. E_q represents the expected value of the cost function, with respect to the uncertain parameters q and $\Theta = \left\{ q \middle| q \in J(q), q^l \leq q \leq q^u \right\}$ represents the uncertain parameter space where this expectancy can be evaluated according to a probabilistic distribution function $J(q)$ and, possibly, upper and lower bounds on the uncertain parameters (q^u, q^l).

The expected value can be estimated by randomly generating values of the uncertain parameters from the distribution functions and then solving the cost minimization subproblems at each of these sample points. The expected value can then be formulated as:

$$EC = \sum_{i=1}^{N_s} \overline{C}(y,d,z,x,q^i)J(q) \tag{2}$$

where N_s is the number of samples used for the evaluation.

More effective ways for approximating the expectancy can also be used (e.g. gaussian quadratures) when the number of uncertain parameters is small (Acevedo and Pistikopoulos, 1998).

A Hybrid Algorithm for Process Synthesis Under Uncertainty

The hybryd algorithm proposed here is based on the solution of the first stage optimization problem through a genetic algorithm. The algorithm starts by generating a set of different values of the design variables (individuals) to form a population. At each iteration, each individual is evaluated and the best are selected to form a new generation by combining the values of the design parameters that each individual represents (crossover). Randomly, some of these parameters are modified (mutation) to increase the search space and prevent a premature convergence. Typically, the genetic algorithm is run for a predefined number of generations, hoping to find the optimal selection (best individual) at the end of the procedure. The code used in this work (Carroll, 1996) already includes some of the most commonly accepted strategies reported in the literature to improve the performance of a basic GA, such as different alternatives for mutation (jump and creep mutation), crossover (single point and uniform) and elitism, passing the best individuals, according to their fitness, from one generation to the next without being modified by the genetic operators.

In the two-stage formulation, the evaluation of each individual requires the solution of the second-stage subproblems, that is, the operational optimization of the selected designs for a number of realizations (samples) of the uncertain parameters to estimate its expected cost.

For each individual, the values of the design parameters and the selected realizations of the uncertain parameters are transfered to an interphase which processes the information and resets the model in Aspen Plus$^{©}$. Inside the simulator, the second-stage subproblems are optimized using a built-in function, and the results are sent back to the interphase to be manipulated and evaluated through a set of decision statements with the aim to discard or improve the information.

Enhancing the Performance of the Algorithm

The performance of the proposed algorithm is affected in terms of the solution time by three critical problems. First, the large number of individuals that are needed to ensure that the GA obtains a good solution; second, the large number of samples needed to evaluate the expectancy in a typical Monte Carlo simulation, and third, the amount of flowsheet simulations performed by a sequential modular simulator to converge to an optimal solution.

A set of strategies are then implemented to improve the performance of the procedure. For the first-stage problems, a reduction of the search space is obtained through a number of logical constraints, imposing restrictions in the codification of the optimization variables for the GA. In this way, ranges of values of the design

parameters are changed dynamically for each individual depending on the selection of the main variables. A second strategy is the incorporation of a convergence criterion based on a variable mutation operator (Leboreiro and Acevedo, 2004). The main idea of this strategy is to force the convergence of the population, i.e. to have 50% of the individuals within 10% of the best solution, by reducing the mutation probability after a number of generations. Once this convergence has been achieved, the mutation probability is set to a very large value for one generation ("mutation shock"), so as to alter most of the individuals, except for those that are passed by elitism. The idea behind this step is to speed up the process of finding a better solution by introducing new genetic information to the population, thus increasing the search capacity of the algorithm. The heuristic algorithm of Leboreiro and Acevedo (2004) was adjusted to account for the confidence intervals that the estimation of the expected value implies.

To decrease the number of inner optimization subproblems a stratified sampling technique is used, where the number of samples for the evaluation of the expected cost of each individual is determined from an estimated flexibility of the design, the actual expectancy compared to that of the best individual and an the estimation of its variance.

In this way, the expectancy is first estimated from a minimum number of samples, and this value is improved using more samples only if the ratio of feasible to infeasible points is larger than a selected minimum <u>and</u> the estimated expectancy is of the same order of magnitude than the best individual <u>and</u> its variance is still too large. In any other case, the estimation is considered sufficiently good.

The idea behind this procedure is to have more accurate estimations only for those individuals that have a good chance to be selected by the GA to form the new generation and to use an optimal number of samples for this estimation as defined by the variance.

For the third problem Aspen Plus has already released an equation oriented module where continuous optimization problems can be effectively solved. Although still limited, this module greatly diminishes solution times.

Proposed Algorithm

The final algorithm considering the proposed strategies for the minimization of the computational requirements consists of the following steps.

0. Initialization
0.1 Define a flowsheet to model the process in Aspen Plus$^©$ and the constraints to reduce the search space of the GA
0.2 Define the fixed parameters for the process model, the genetic algorithm (mutation probabilities, convergence criteria etc.) and the evaluation of the second-stage subproblems (minimum flexibility, minimum number of samples, desired variance)

0.3 Initialize the best solution as an infinite value.

1. First-stage problem
1.1 If a population is not available, define randomly a first set of individuals (values of structure and design variables) and go to Step 2.
1.2 Evaluate the aptitud (total cost) of each individual and update the best solution.
1.3 If population convergence did not increase in the last five generations, reduce the mutation probability by half.
1.4 If the population converged and the best individual has not changed, then stop with the best individual as the optimal solution; else, set the mutation probability to a large value.
1.5 Perform the basic genetic operators: selection, crossover and mutation to obtain a new generation of designs to evaluate.

2. Second-stage problem
2.1 Select an individual from the new population.
2.2 Generate a minimum number of samples.
2.3 Optimize the operational cost of the individual at each sample point through Aspen Plus$^©$.
2.4 Estimate the flexibility of the design from the number of feasible solutions obtained. If the estimation is less than the minimum flexibility required, define the expected cost through a penalty function and return to Step 2.1.
2.5 Evaluate the expected cost as in (2). If the expected cost is one order of magnitude larger than that of the best individual, return to 2.1.
2.6 If the calculated variance is larger than 10% of the expected value, increase the number of samples to evaluate the individual.

Numerical Example: Extractive Distillation

In this problem, the separation of a highly non-ideal mixture is analyzed. A process stream that contains toluene and n-heptane must be separated by extractive distillation, using phenol as a solvent (Henley and Seader, 1981). This solvent is in turn recovered in a second column, where the toluene is obtained as a second product and the phenol is recycled to the first column. The uncertain variables considered are the composition of the feed stream and its molar flow. These parameters are supposed to change accordingly to normal distributions functions with mean value of 0.5 and standard deviation of 0.05 for the composition of n-heptane, and mean value of 400 lbmol/hr with a standard deviation of 40 for the total flow fed to the column. The objective of the process is to obtain n-heptane with a molar purity of at least 99% as the main product and toluene with 90% purity as a byproduct.
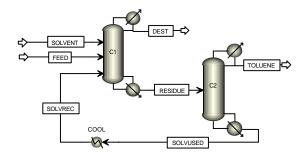
Fig. 1 Flowsheet for the Extractive Process

The first stage variables, handled by the GA, are the number of stages, column diameters, and feed locations for each column.

The second-stage subproblems are defined and evaluated through an Aspen Plus© optimization block, using RadFrac as the model for each column. The specification of the optimization subproblems includes the reflux ratios and flow of fresh solvent as optimization variables. The constraints are given by minimum requirements for the purity of both products and the range of flood factors allowed for each column to ensure adequate column performance.

The objective function is defined as the sum of the capital and operational costs. The capital cost is related to the first stage variables, and is calculated through the Equipment Module Costing technique, which involves a rigorous costing method and is widely accepted for preliminary costs in chemical plants. The operational cost is given by the utility costs and the cost of the fresh solvent.

The optimal solution obtained considers 57 stages and a diameter of 9 ft for the first column, and 16 stages and a diameter of 5.3 ft for the second. This design presents an estimated flexibility of 70%, varying the reflux ratios from 4 to 6 approximately for the first column, and from 0.5 to 1 approximately for the second. Although this flexibility could be regarded as low, it is due to the relatively large variances of the uncertain parameters and the narrow feasible region that the operation of the integrated columns presents.

From the numerical point of view, the proposed strategies reduced the computational requirements in more than 75% for some cases. The most important effect was obtained from the convergence procedure, which reduced the number of generations evaluated by the GA in more than 40% for all cases, and sometimes in more than 65%. Another important contribution was the optimization of the number of samples, which drastically reduced the number of optimization subproblems solved. For this specific

problem, however, using specialized quadratures (Acevedo and Pistikopoulos, 1998) was the most effective way to estimate the expected value.

Conclusions

A two-stage stochastic framework was presented where the advantages of a process simulator are combined with a genetic algorithm for the synthesis and design of chemical process under uncertainty.

The use of sophisticated process models and the flexibility of the optimization algorithm could allow the search of new, more efficient process, with relatively little effort form the process engineer. The computational requirements, however, are very large and, although the implementation proved to be very robust and the proposed procedure reduced drastically the solution times, more strategies to improve its performance are still required. On this line, present work includes the utilization of more efficient sampling techniques, using real codification for the genetic algorithm and the discretization of the design variables. The approach is also being applied to heterogeneous separation systems and to the optimization of complete flowsheets under uncertainty.

References

Acevedo, J. and Pistikopoulos, E. N. (1998) Stochastic optimization based algorithms for process synthesis under uncertainty. Computers and Chemical Eng. **22**, pp 647-671.

Carroll, D.L. (1996), Chemical Lasser Modeling with Genetic Algorithms. AIAA J., **34** pp. 338-346.

Grossmann, I. E., Halemane, K. P. and Swaney, R. E. (1983) Optimization strategies for flexible chemical process. Computers and Chemical Eng. **7**, 439-462.

Henley, J. E. and Seader, J. D. (1981) Equilibrium Stage Separation Operations in Chemical Engineering. John Wiley and Sons. New York, USA.

Leboreiro, J. and Acevedo, J. Process Synthesis and Design Sequences using Modular Simulators: a Genetic Algorithm Framework. In press (2004). Computers and Chemical Eng.

Sahinidis, N.V. (2003) Optimization under uncertainty: State of the Art and Opportunities. FOCAPD 2003.

Tayal, M.C., Fu, Y., and Diwekar, U.M., (1999) Optial design of Heat Exchangers: A Genetic Algorithm Framework, Ind. Eng. Chem. Res., **19** S38-S39.