AIChE Wiley InterScience DISCOVER SOMETHING GREAT

A Probability Distribution Estimation Based Method for Dynamic Optimization

Jie Xiao and Yinlun Huang
Dept. of Chemical Engineering and Materials Science, Wayne State University, Detroit, MI 48202

Helen H. Lou
Dept. of Chemical Engineering, Lamar University, Beaumont, TX 77710

DOI 10.1002/aic.11209
Published online June 1, 2007 in Wiley InterScience (www.interscience.wiley.com).

Engineering optimization of a highly nonlinear complex system is always a challenge methodologically and computationally. This is especially true when multistage dynamic optimization is involved. While significant progress has been made in rigorous deterministic algorithms for dynamic optimization, meta-heuristic-based optimization may offer an attractive alternative. This paper introduces a general mathematical framework, called the Population-based Probability Distribution Estimation (PPDE) method, for tackling constrained multistage complex process dynamic optimization problems. Solution identification is accomplished through probability distribution estimation based search in a continuous space, where special solution migration and penalty assignment techniques are integrated. Besides an optimal parameter estimation problem for a reactor system, an automotive coating curing optimization problem is also investigated, where the PPDE successfully minimizes oven energy consumption under various process/product constraints. Optimization results demonstrate superiorities of the method over the Ant Colony System (ACS) based dynamic optimization method. © 2007 American Institute of Chemical Engineers *AIChE J,* 53: 1805–1816, 2007

Keywords: probability density function, probability distribution estimation, multistage constrained dynamic optimization, paint shop, polymeric coating curing

Introduction

A generalized engineering optimization problem can be stated as follows: given a set of initial conditions and necessary parameter values, find an optimal solution of a set of decision variables in an effective and efficient way so that the specified objective function can be extremized while all constraints are satisfied.[1] In solution search, complexity can arise from large-scale multistage dynamic processing and time-dependent constraints, which may be expressed by sets of linear/nonlinear differential, integral, and algebraic equations. It is always a challenge in solving such problems.[2] Over the past years, significant progress has been made in developing deterministic global optimization methods for solving nonconvex nonlinear programming (NLP) and mixed-integer nonlinear programming (MINLP) problems.[3–7] Efforts have been also devoted to global optimization for dynamic systems.[8–10] Papamichail and Adjiman and Chachuat et al.[11–14] proposed deterministic global optimization algorithms for a type of dynamic systems, which involve a set of first-order differential equations in the constraint set. Meanwhile, meta-heuristics-based approaches, such as evolutionary algorithm (EA) and particle swarm optimization (PSO), which are based upon the principles of natural biological evolution or artificial life, have also drawn considerable attention.[15,16]

Recently, Xiao et al. introduced a mathematical framework for solving complex process dynamic optimization problems,[17] which is an extension of a relatively new meta-heu-

Correspondence concerning this article should be addressed to Y. Huang at yhuang@wayne.edu.

ristic approach called ant colony system (ACS) algorithm.[18,19] In that approach, a solution identification process is expressed by a set of search trees and all the (artificial) ants will work on this set of trees cooperatively. Each tree corresponds to a decision variable, which is optimized jointly by a set of ants. The solution search utilizes three classical rules, i.e., the state transition rule, the local pheromone updating rule, and the global pheromone updating rule. In addition, a novel sequence determination rule is introduced to improve algorithm performance. The methodology is general and has been successfully used to solve a coating quality constrained oven energy minimization problem and a cure-window-based coating quality optimization problem.[17,20] However, the solution search in an intrinsic discretized space may restrict solution precision. Further, its computational efficiency is yet to be improved.

In discretized space search, solution quality in most cases is heavily affected by the fineness of discretization. Continuous space search, on the other hand, avoids the discretization step and has a better chance to obtain a globally optimal solution. Note that extending meta-heuristics to continuous solution space search in ant colony optimization (ACO) is not straightforward. Hitherto, only a few ant approaches for continuous optimization have been proposed in the literature, such as continuous ACO and continuous interacting ant colony (IAC).[21,22] The latest approach, which is at the same time closest to the spirit of ACO for combinatorial problems, is proposed by Socha and coworkers.[23,24] Its principal idea is to construct solutions by sampling probability distribution over the search space. However, all the above approaches are functionally restricted to solving unconstrained static optimization problems.

A general mathematical framework, called the population-based probability distribution estimation (PPDE) method, is developed in this work to tackle complex multistage constrained dynamic optimization problems. By comparing with the ACS approach by Xiao et al.[17] PPDE advances the solution identification process from search-tree-based discretized space search to probability distribution estimation based continuous space iterative search. A special penalty assignment technique is applied to take care of various constraints, and a solution migration step is introduced to prevent premature convergence commonly occurred in constrained optimization problems. Solution quality and computational efficiency can be greatly improved. The efficacy of the method will be demonstrated by tackling a parameter estimation problem as well as a complex industrial dynamic optimization problem.

General Dynamic Optimization Formulation and Analysis

A class of dynamic optimization problems addressed in this work are formulated below:

$$\min_{\boldsymbol{X}(t)} J = \sum_{i=1}^{N^{\mathrm{st}}} \Phi_i \left(\boldsymbol{Y}(t,z); \boldsymbol{X}(t); t; \boldsymbol{U}(\vec{z}) \right) \tag{1}$$

s.t.

$$F_i \left(\frac{\partial \boldsymbol{Y}}{\partial t}; \frac{\partial \boldsymbol{Y}}{\partial z}; \int \boldsymbol{Y} dz; \boldsymbol{Y}(t,z); \boldsymbol{X}(t); t; \boldsymbol{U}(\vec{z}) \right) = 0 \qquad i = 1, \ldots, N^{\mathrm{st}} \tag{2}$$

$$G_i \left(\boldsymbol{Y}\left(t_i^0, z\right); \boldsymbol{Y}\left(t_i^{\mathrm{e}}, z\right); \boldsymbol{Y}\left(t, z_i^{\mathrm{b1}}\right); \frac{\partial \boldsymbol{Y}}{\partial z}\left(t, z_i^{\mathrm{b2}}\right) \right) = 0 \qquad i = 1, \ldots, N^{\mathrm{st}} \tag{3}$$

$$H_i \left(\frac{\partial \boldsymbol{Y}}{\partial t}; \boldsymbol{Y}(t,z); \boldsymbol{U}\left(\vec{z}\right); \Phi_i \right) \leq 0 \quad i = 1, \ldots, N^{\mathrm{st}} \tag{4}$$

$$\boldsymbol{X}_i^{\min} \leq \boldsymbol{X}_i(t) \leq \boldsymbol{X}_i^{\max} \quad i = 1, \ldots, N^{\mathrm{st}} \tag{5}$$

where Φ's are scalar sub-objective functions; F's are integrated process-product system models; G's are initial and boundary conditions for the system models; H's are series of inequality constraints for the optimization problem; subscript i represents the stage index; N^{st} is the total number of stages; $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the system input and output variables, respectively; z is the generalized coordinate (e.g., one coordinate in Cartesian coordinates); $\vec{z}$ is the generalized position vector; $\boldsymbol{U}$ includes all time-independent parameters, some of which are dependent on positions; t_i^0 and t_i^{e} are, respectively, the starting and ending times of the ith stage; z^{b1} and z^{b2} are two boundaries; and $\boldsymbol{X}^{\min}$ and $\boldsymbol{X}^{\max}$ are, respectively, the lower and upper bounds for input variables.

This is a large-scale multistage dynamic optimization problem with a mixed set of sparse linear and nonlinear constraints. As shown in Eq. 1, the objective is time dependent and is a sum of interdependent sub-objective functions. It is clear that the stage-wised local optimization dose not guarantee the global optimization for the overall system.

Individual sub-model shown in Eq. 2 is considered in each dynamic stage. These sub-models are interacted among operational stages, and between the process and the product. Further, each sub-model consists of more detailed models describing the dynamics of a number of process-product variables. Since system outputs depend on both time and position, ordinary or partial nonlinear differential equations and integral equations are involved. Moreover, the model takes the transitions between adjacent stages into account. For example, the continuity of physically meaningful variables during transition has been included in Eq. 3. Equation 4 encapsulates process-product related and solution feasibility related constraints. Due to the multistage nature of the problem, different operational requirements are imposed on system inputs during different stages (see Eq. 5).

The complexities described earlier render tremendous difficulties in solution identification using conventional dynamic optimization approaches (e.g., variational methods, dynamic programming, and direct sequential methods); it is very likely that a variety of local optima in solution space exist. To identify optimal solution effectively and efficiently, a PPDE method is proposed in this work. As it is a population-based iterative approach, a population of solutions, rather than just one solution, is needed in each iteration for solution identification. It possesses the following merits in dealing with complex multistage nonlinear constrained dynamic optimizations.

(1) Only objective function value is used to drive the search process. This avoids obtaining gradient information required in gradient-based methods or rigorous mathematical formulations appeared in some indirect search methods (e.g., variational method), which may be very difficult to obtain when the optimization model is complex.

(2) The system inputs in all time stages can be optimized simultaneously. Compared with sequential stage-wised optimization approaches (e.g., dynamic programming), it may be more effective for multistage dynamic optimization problems.

(3) A population of solution provides the information of a good search region, instead of only a good point in the search space. Consequently, for complex problems with many local optima, it has a better chance to obtain global optimal solution than point-by-point approaches (e.g., most gradient-based techniques).

(4) Presence of multiple solutions can be beneficial in handling constrained optimization problems.[25] It is not a prerequisite for the algorithm to start from feasible solutions. On the other hand, estimating probability distribution of promising solutions to guide the exploration of the search space is a new optimization concept for chemical engineers, although it is recognized to be valuable for improving the performance of the conventional evolutionary algorithms (e.g., GA) in many computer science related applications.[26] This concept also enables the continuous solution space search, and consequently contributes to the precision of the solution. Note that applying PPDE for simpler dynamic and static optimization problems is straightforward.

Basics of Population-Based Probabilistic Search

The PPDE optimization framework can be classified into the research on population-based probabilistic search algorithms (PBPS),[26] which are based on modeling promising solutions by estimating their probability distribution and using the constructed model to guide the exploration of the search space. The main procedure of these algorithms is similar to that of GA. An initial population is randomly generated, and then new solutions will be constructed. The constructed better solutions will replace the old ones in the original solution population. The same procedure will be repeatedly executed until a termination criterion is met. The difference between PBPS and GA is the approach of constructing new solutions. In GA, a new solution is generated by directly operating on the solutions in the current generation through several operators (e.g., selection, crossover, and mutation). However, in PBPS, a new solution is generated by the following two steps: (a) to construct a model to estimate the real distribution based on the solutions in the current iteration, and (b) to generate new solutions according to the constructed model.

The principal idea in PBPS is that a population of solutions can be viewed as a sample drawn from an unknown probability distribution; iteratively refining the estimated distribution would allow the optimization algorithm to generate new better solutions. The algorithms have shown satisfactory performance in solving a variety of problems (mainly in the computer science realm).[26] However, a general framework of incorporating the PBPS principal idea for chemical engineering optimization problems (especially, the complex multistage nonlinear constrained dynamic optimization problems formulated in the previous section) is unavailable. Developing such a framework becomes the major focus of this work.

PPDE Method

The PPDE method delineated in this section is to provide answers to the following essential questions: What is the approach by which an algorithm is used to deal with dynamic optimization problems? How to construct a probabilistic model to estimate the probability distribution of promising solutions based on the solutions in the current iteration and how to generate new solutions according to the probabilistic model? How various constraints can be taken care of and how to prevent premature convergence?

Time horizon discretization for dynamic optimization

To solve the problems formulated in Eqs. 1–5, the continuous system input in each stage should be approximated by a sequence of constant inputs. Assume that there are n^{u} input variables in $\boldsymbol{X}$, the time duration of each stage $[t_i^0, t_i^{\mathrm{e}}]$ is divided into n_i^{d} intervals, and the system input in each time interval is constant. Note that all time intervals have the same time duration Δt^{d}. After time horizon discretization, the optimization objective is to identify the optimal values for $\sum_{i=1}^{N^{\mathrm{st}}} n^{\mathrm{u}} \times n_i^{\mathrm{d}}$ discretized system inputs (decision variables), such that the objective function can be minimized, and all the constraints are obeyed.

Since PPDE generates a population of complete solutions before any solution quality evaluation and probabilistic model construction, the systems inputs in all time stages are actually simultaneously optimized. Given a complete solution, the calculated objective function and constraints evaluation information will be used for constructing a probabilistic model.

Probabilistic model construction

A probabilistic model needs to be constructed in each iteration for new solution generation. This model should be capable of estimating the probability distribution of promising solutions so that the new solutions obtained from this model has a potential to be better ones. Further, this model should be constructed based on the solutions in the current iteration, which can be treated as a sample drawn from an unknown probability distribution. The task here is to estimate this probability distribution from a given sample.

Estimation of probability distribution from a given sample is extensively studied in the particle physics community, where the kernel probability density estimation is a fast growing technique.[27,28] It is based on the premise that continuous, differentiable functions can be exactly modeled by the infinite sum of some other, appropriately chosen, "kernel" functions. Gaussian function is chosen to be the kernel function and it is the probability density function (PDF). According to this technique, an estimated probability distribution (of promising solutions) can be represented by the sum of Gaussian kernels centered at the sample points (individual solutions). The idea of assigning a weight to each Gaussian kernel according to the solution quality is adopted here to construct the probabilistic model, which is a weighted sum of Gaussian kernels.[23,24]

Solution Population Matrix. In matrix $\tilde{\boldsymbol{X}} = [\tilde{x}_{i,j}]_{K \times N}$, the elements in each row form a solution for the optimization problem (called an individual solution, either feasible or infeasible). Element $\tilde{x}_{i,j}$ is the value for the jth decision variable, x_j, in the ith individual solution. Note that there are K solutions in this population matrix and each solution contains N decision variable values. In other words, a sample of K values is given for each decision variable.

Kernel Probability Density Functions. As discussed, each decision variable has a sample of K values, and K kernel probability density functions can be constructed for each decision variable. The kernel probability density function matrix, $\boldsymbol{P}([p_{i,j}]_{K \times N})$, is constructed from $\tilde{\boldsymbol{X}}$. Each element $p_{i,j}$ (called a kernel probability density function or simply KPDF) is a Gaussian (or normal) function centered at $\tilde{x}_{i,j}$, i.e.,

$$p_{i,j}(x_j, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_j - \mu)^2}{2\sigma^2} \right) \tag{6}$$

where

$$\mu = \tilde{x}_{i,j} \tag{7}$$

$$\sigma = \frac{\rho_1}{\sqrt{K-1}} \sqrt{\sum_{l=1}^{K} (\tilde{x}_{l,j} - \tilde{x}_{i,j})^2} \tag{8}$$

where $\rho_1 \in (0, 1)$ is a scaling parameter.[24,27] Note that for each element $\tilde{x}_{i,j}$ in solution population, a normal function $p_{i,j}$ can be constructed according to Eqs. 6–8. There are K Gaussian functions (i.e., $p_{i,j}$, $i = 1, \ldots, K$) corresponding to one decision variable (i.e., x_j).

Probability Distribution Estimation. The weighted sum of KPDFs (i.e., $p_{i,j}$, $i = 1, \ldots, K$) is used as the probabilistic model to estimate the probability distribution of promising values for decision variable x_j. There are N decision variables, and thus N models are needed.

In individual preference vector $\Omega = [\omega_i]_{K \times 1}$, element $\omega_i \in (0, 1)$ gives a weight for the ith KPDF (there are K KPDFs for the decision variable x_j). This vector also shows the solution quality. The larger the value of ω_i, the better the ith solution.

The population probability density function vector, $\bar{\boldsymbol{P}} = [\bar{p}_j]_{1 \times N}$, can be constructed from $\boldsymbol{P}$ and Ω, i.e.,

$$\bar{\boldsymbol{P}} = \Omega^T \boldsymbol{P} \tag{9}$$

Note that element $\bar{p}_j$ (called a population probability density function or simply PPDF) is a weighted sum of kernel probability density functions for decision variable x_j, i.e.,

$$\bar{p}_j = \sum_{i=1}^{K} (\omega_i \times p_{i,j}) \quad j = 1, \ldots, N \tag{10}$$

The element serves as the probabilistic model for decision variable x_j and it may not be a PDF according to the classical definition. It is constructed from a set of KPDFs, and the set of KPDFs comes from the solution population.

Solution construction

Solutions in the initial iteration are randomly created. New solutions are then generated according to the probabilistic models. To prevent premature convergence, a solution migration technique is introduced to increase the diversity of solutions in the population. Solutions generated in the migration case are also discussed.

Initial Solutions. Elements in the initial solution population are randomly generated between the upper and lower bounds of decision variables according to the uniform distribution. It is understandable that generating initial solutions uniformly distributed in the solution space contributes to global optimization, which requires exploring complete solution spaces.

New Solution Construction in Cases Without Migration. As discussed earlier, $\bar{p}_j$ serves as the probabilistic model for decision variable x_j; it estimates probability distribution of promising values for x_j. Sampling the distribution expressed by $\bar{p}_j$ (or simply called sampling $\bar{p}_j$) once gives a new value for x_j. It can be accomplished by following the next two successive steps.[23,24]

(i) Select one KPDF from the KPDFs that construct $\bar{p}_j$ according to the following probability distribution:

$$\hat{p}_i = \frac{\omega_i}{\sum_{l=1}^{K} \omega_l} \quad i = 1, \ldots, K \tag{11}$$

where $\hat{p}_i$ is the probability of choosing $p_{i,j}$, the ith KPDF for the jth decision variable. Remember that a better solution has a larger value of the weight, ω. Consequently, the KPDF centered at that solution has a better chance to be selected for generating new solutions.

(ii) Generate a random number according to the selected KPDF, $p_{i,j}$. If this number is between the upper and lower bounds of decision variable x_j, it is the new value for x_j. Otherwise, repeat this step until a satisfactory value is obtained.

By sampling $\bar{p}_1$ through $\bar{p}_N$ successively a complete new solution will be generated. Repeat this process will give another new solution.

New Solution Construction in Migration Cases. The objective of solution migration is to increase solution diversity in the solution population, thereby ensuring a thorough search of the entire space. New solutions constructed in these cases will be added into the solution population and should have a certain level of variance.

Assume that the i^*th solution in the current solution population is the best solution. New solutions will be generated around this solution. Thus, it is needed to introduce a PDF vector in the migration case, i.e., $\bar{\boldsymbol{P}}^{\mathrm{m}} = [\bar{p}_j^{\mathrm{m}}]_{1 \times N}$, using the following equation:

$$\bar{p}_j^{\mathrm{m}} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x_j - \mu)^2}{2\sigma^2} \right) \quad j = 1, \ldots, N \tag{12}$$

where

$$\mu = \tilde{x}_{i^*,j} \tag{13}$$

$$\sigma = \max\left((x_j^{\max} - \tilde{x}_{i^*,j}),\ (\tilde{x}_{i^*,j} - x_j^{\min}) \right) \tag{14}$$

where $\bar{p}_j^{\mathrm{m}}$ is the PDF in the solution migration case (or simply called MPDF) for the jth decision variable; $\tilde{x}_{i^*,j}$ is the value for the jth decision variable in the i^*th solution (the best solution in the current solution population); $x_j^{\min}$ and $x_j^{\max}$ are the lower and upper bounds for the jth decision variable, respectively.

Matrix $\bar{\boldsymbol{P}}^{\mathrm{m}}$ gives probability distribution of migrated solutions. Sampling $\bar{p}_j^{\mathrm{m}}$ will lead to a new value for the jth decision variable. This can be accomplished by generating a random number according to the normal distribution, $\bar{p}_j^{\mathrm{m}}$. Keep in mind that the bounds for the decision variables should be satisfied. Sampling $\bar{p}_1^{\mathrm{m}}$ through $\bar{p}_N^{\mathrm{m}}$ successively generates a complete new solution.

Note that new solutions are generated by sampling a set of PDFs, i.e., KPDFs in normal cases and MPDFs in solution migration cases. By comparing the standard deviation of KPDF (see Eq. 8), the standard deviation of MPDF (see Eq. 14) is much larger in most cases. It is known that in probability theory, standard deviation is the most common measure of statistical dispersion. Thus, sampling MPDFs can give rise to a greater diversity of the generated new solution alternatives and, consequently, a more thorough search of the entire space. Therefore, solution migration contributes to the prevention of premature convergence.

Solution evaluation and constraints consideration

Recall that the individual preference vector Ω shows the quality of solutions in the corresponding solution population and gives a relative importance of each solution in constructing probabilistic models (PPDFs in this work). The objective function value together with the constraints compliance conditions of each solution should be considered to determine this vector.

To consider various constraints in the optimization problems, the most common way is application of penalty techniques. Usually, it is accomplished by introducing penalized objective function. Once some constraints are violated, penalty terms will be added into the original objective function. However, the primary deficiency of this method is that the penalty terms are usually set roughly (at the same or higher order than the order of the original objective function). In this work, ranks are assigned to the solutions in the solution population and penalties for infeasible solutions are designed into the solution ranks, which consequently affect the possibility of utilizing certain search spaces for generating new solutions. In this way, adding penalty terms into the objective function can be avoided. Since objective function value needs to be calculated only for feasible solutions, computational time in calculating objective function value for infeasible solutions can be saved.

In the ranking vector, $\boldsymbol{R} = [r_i]_{K\times 1}$, element r_i is a positive integer, showing the rank of the ith solution in the solution population. The rank is evaluated according to the objective function value and constraint compliance conditions of each solution. The smaller the value, the better the corresponding solution.

For the feasible solutions, ranks are assigned according to their objective function values. Assign 1 to r_i if the ith solution is the best solution, assign 2 if it is the second best, and so on. If two solutions give the same objective function value, they will obtain the same rank. For an infeasible solution, the rank is calculated as follows:

$$r_i = \mathrm{ceil}\left[K\times\left(1+\frac{n_i^{\mathrm{v}}}{n^{\mathrm{c}}}\right)\right], \quad \text{if the } i\text{th solution is infeasible} \tag{15}$$

where K is the number of solutions in one solution population; n^{c} is the number of inequality constraints in the optimization problem (constraints shown in Eq. 4); n_i^{v} is the number of violated constraints when using the ith solution; ceil is an operator that rounds the element to the nearest integer towards infinity. Note that, if the rank is smaller or equal to K, the corresponding solution is a feasible solution; otherwise, it is an infeasible solution.

According to $\boldsymbol{R}$, the individual preference vector, Ω, can be determined. Element $\omega_i \in (0, 1)$ gives the preference of using the ith KPDF $p_{i,j}$ (there are K KPDFs for the decision variable x_j) to generate a new value for decision variable x_j (see Eq. 11). It can be calculated as follows:

$$\omega_i = \frac{1}{\rho_2 K\sqrt{2\pi}}\exp\left(-\frac{(r_i-1)^2}{2(\rho_2 K)^2}\right) \quad i = 1,\ldots,K \tag{16}$$

where $\rho_2 \in (0, 1)$ is another scaling parameter. Since ω is calculated from a Gaussian function, it is a value between 0 and 1. Note that r_i is a positive integer (always larger than or equal to 1). Consequently, a better solution with a better rank (a lower r_i value) will gain a greater preference value (ω_i). If ρ_2 is small, the standard deviation is small, which means the best-ranked solutions are strongly preferred. On the other hand, if ρ_2 is large, different solutions will have a more uniform preference. The preference for infeasible solutions will be very low due to the large rank value (larger than the population size K).

Note that scaling parameters, ρ_1 and ρ_2 (all in the range of 0 to 1) affect both the algorithm convergence and solution quality. Generally speaking, the smaller the value, the faster the convergence but the greater the possibility of being trapped in local optima. A criterion used for choosing parameter values is that appealing solutions can be obtained within reasonable computational time. Consequently, a medium value (i.e., between 0.4 and 0.6) is suggested to test first. However, trial and error of other values is always recommended.

Solution identification procedure

The proposed procedure is an iterative process with $L^{\max}$ iterations. Assume that the population size is K, and in each iteration (except the initial one), M ($<K$) new solutions are generated. Let $\tilde{\boldsymbol{X}}^{q}$ be the solution population at the qth iteration, $\tilde{\boldsymbol{X}}^{\mathrm{opt}}$ be the best solution, $\tilde{\boldsymbol{X}}^{q,\mathrm{opt}}$ be the best solution at the qth iteration, and J ($\tilde{\boldsymbol{X}}^{q,\mathrm{opt}}$) be the objective function value using this solution. The search approach contains the following major steps.

Step 1. Generate an initial solution population $\tilde{\boldsymbol{X}}^{0}$ containing K individual solutions.

Step 2. Evaluate all K solutions in the solution population. If it is a feasible solution, the objective function value will be recorded; otherwise, record the number of violated constraints.

Step 3. Generate ranking values in $\boldsymbol{R}$.

Step 4. Based on the known $\tilde{\boldsymbol{X}}^{q}$ (iteration number q = 0, 1,. . ., $L^{\max}$) and its ranking vector, update $\tilde{\boldsymbol{X}}^{\mathrm{opt}}$, if the best solution in this iteration, $\tilde{\boldsymbol{X}}^{q,\mathrm{opt}}$, is better than the existing best solution. Then calculate individual preference vector Ω from ranking vector $\boldsymbol{R}$.

Step 5. Generate KPDF matrix $\boldsymbol{P}$ using Eqs. 6–8. PPDF vector $\bar{\boldsymbol{P}}$ can be constructed from $\boldsymbol{P}$ and Ω using Eq. 9.

Table 1. Results in Numerical Explanation Example

		$\tilde{X}^q$					
Iteration Index	Solution Index	$\tilde{x}_{i,1}$ [10, 50]	$\tilde{x}_{i,2}$ [0, 20]	$\tilde{x}_{i,3}$ [60, 90]	$\boldsymbol{R}$	Ω	$\hat{\boldsymbol{p}}$ (%)
Initial iteration (q = 0)	Solution 1	24.2	12.3	66.5	2	0.2349	41.3
	Solution 2	30.4	8.1	77.2	5	0.0013	0.23
	Solution 3	12.2	5.4	85.7	1	0.3325	58.47
First iteration (q = 1)	Solution 1	24.2	12.3	66.5	3	0.0829	12.75
	Solution 2	13.1	12.5	86.1	1	0.3325	51.12
	Solution 3	12.2	5.4	85.7	2	0.2349	36.13

Step 6. Generate M new solutions (either feasible or infeasible) and then construct solution population for the next iteration. Solution migration is designed here to prevent premature convergence. Executing this step has two options, i.e.,

$$I_o = \begin{cases} 1 & \text{if } \mathrm{mod}(q, N^m) = 0 \text{ and } q > 0 \\ 2 & \text{otherwise} \end{cases} \quad (17)$$

where I_o is the option index; q is the iteration number; N^m is a positive integer number; mod is an operator which gives the remainder after dividing q by N^m. Since option 1 is for solution migration, Eq. 17 suggests that to prevent the optimization process from being trapped in local optima, a solution migration will be introduced once in every N^m iterations. The two options will be delineated later in this section.

Step 7. Go to *Step 4* to initiate another iteration, if the maximum number of iteration has not been exceeded. Otherwise, optimal solution $\tilde{\boldsymbol{X}}^{\mathrm{opt}}$ is obtained and $J(\tilde{\boldsymbol{X}}^{\mathrm{opt}})$ is the minimum value, and the optimization process can be terminated.

New Solution Construction Options. There are two options (see Eq. 17) in constructing solution population for the next iteration. They are detailed below.

(a) *Option 1* (I_o= 1). This is a solution migration step; it is to generate M new solutions in migration cases. In this option, the worse M solutions in $\tilde{\boldsymbol{X}}^q$ are directly replaced by M new solutions to form a new solution population, $\tilde{\boldsymbol{X}}^{q+1}$, for the next iteration. Then, evaluate M new solutions. If a solution is feasible, the objective function value will be recorded. Otherwise, the number of violated constraints needs to be recorded. Then, the K solutions in $\tilde{\boldsymbol{X}}^{q+1}$need to be ranked.

(b) *Option 2* (I_o= 2). M new solutions are generated by sampling PPDFs, all of which need to be evaluated. Then all (K + M) solutions (K solutions in $\tilde{\boldsymbol{X}}^q$, and M new solutions) should be ranked using the same method described in the previous section. After that, the first K best solutions need to be chosen to form a new solution population $\tilde{\boldsymbol{X}}^{q+1}$ for the next iteration. The corresponding ranking vector $\boldsymbol{R}$ (the first best K ranks in total K+ M ranks) should be recorded.

Numerical explanation on major search steps

A simplified numerical example is given to explain major steps in solution search. Suppose we attempt to solve an optimization problem with three decision variables (N = 3) and two constraints (n^c = 2). Also assume that there are three solutions (K = 3) in one solution population and one new solution (M = 1) is generated in each iteration. The scaling parameters ρ_1 and ρ_2 are both set to 0.4 and the solution migration step is introduced once every 100 iterations (N^m = 100).

Steps 1 to 3. The upper and lower bounds for each decision variable and the initial solution population are shown in Table 1 (see the columns with the heading of “$\tilde{\boldsymbol{X}}^q$”). Assume that feasible solution 3 gives a smaller objective function value as compared with feasible solution 1, thus solution 1 receives rank “2” and solution 3 obtains rank “1". Solution 2 is infeasible, as it violates one constraint. According to Eq. 15, solution 2 is given rank “5.”

Step 4. Solution 3, which is (12.2, 5.4, 85.7), is used to update the optimal solution $\tilde{\boldsymbol{X}}^{\mathrm{opt}}$. The elements in the individual preference vector Ω are calculated from Eq. 16 and has been shown in Table 1 and Figure 1a. Note that in Figure 1a, two cases with different ρ_2 values are shown to illustrate the function that ρ_2 has (see the comments on ρ_2, in the paragraph after Eq. 16).

Step 5. KPDFs and PPDFs can be obtained using Eqs. 6–9. They are shown in Figures 1b–d. Each decision variable has three KPDFs (K = 3) and one PPDF.

Steps 6 and *7.* Since this is the initial iteration (q = 0), option 2 is selected. The first new solution is constructed by sampling $\bar{p}_1$, $\bar{p}_2$, and $\bar{p}_3$. Sampling each PPDF can be accomplished in two successive steps. First, according the probability distribution calculated by Eq. 11 (values are shown in Table 1), one KPDF is chosen. Then, sample that KPDF. Assume $p_{3,1}$, $p_{1,2}$, and $p_{3,3}$ (Figure 1) are chosen and one new solution (13.1, 12.5, 86.1) is generated. In Figure 1, $\bar{\boldsymbol{P}}^m$ is also shown to illustrate the solution migration effect. As indicated in this figure, for decision variable x_j, MPDF (i.e. $\bar{p}_j^m$) has a much larger standard deviation than any KPDF (i.e., $p_{1,j}$, $p_{2,j}$, or $p_{3,j}$) has. It is understandable that generating new solutions according to $\bar{\boldsymbol{P}}^m$ (almost a uniform distribution in this case) can increase the solution diversity and help the process jump out of the local optima.

Assume that this new solution is better than any solutions in $\tilde{\boldsymbol{X}}^0$. The new solution population and ranking vector are shown in Table 1. The maximum iteration has not been reached. Thus, the solution search returns to *Step 4*.

Solution 2, which is (13.1, 12.5, 86.1), is used to update the optimal solution $\tilde{\boldsymbol{X}}^{\mathrm{opt}}$. KPDFs and PPDFs will be updated. Some results in this new iteration are shown in Table 1 and Figure 2.

Case Studies

To demonstrate the efficacy of the PPDE method, an optimal parameter estimation problem for a set of dynamic models and a multistage nonlinear constrained dynamic optimization problem are solved in this section.

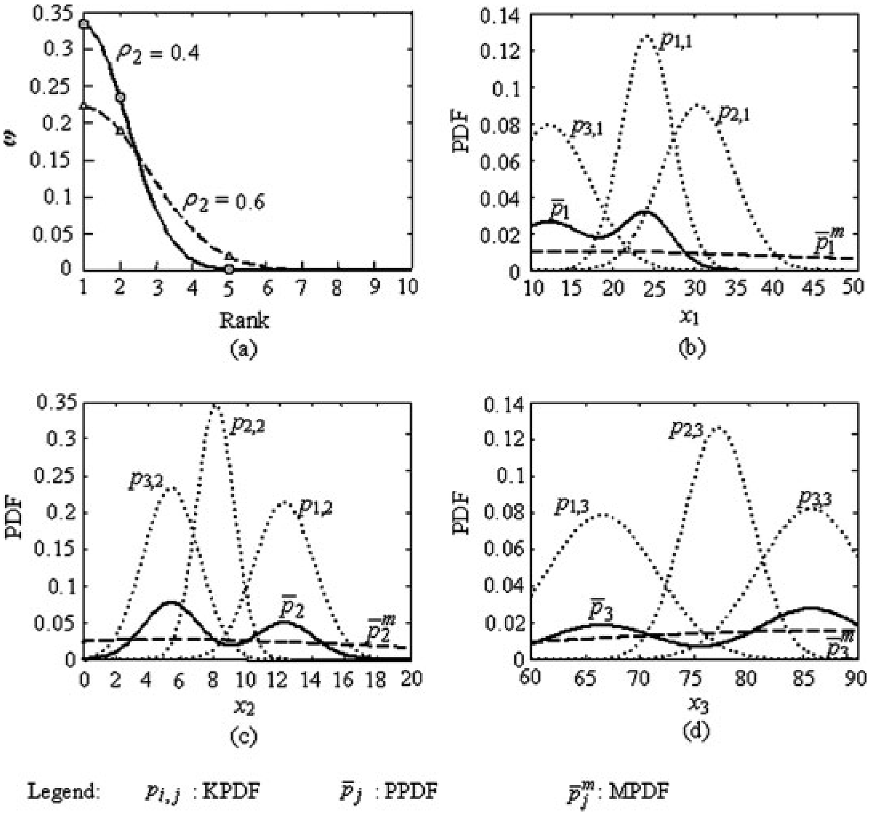

Figure 1. Initial iteration results: (a) individual preference ω vs. rank for two different ρ_2 cases, (b) kernel probability density functions (KPDFs), population probability density function (PPDF), and probability density function in solution migration cases (MPDF) for decision variable x_1, (c) KPDFs, PPDF, and MPDF for decision variable x_2, and (d) KPDFs, PPDF, and MPDF for decision variable x_3.

Case I: parameter estimation example

Floudas et al. and then Papamichail and Adjiman solved an interesting parameter estimation problem using deterministic global optimization techniques.[12,29] The dynamic optimization problem involves a first-order irreversible isothermal liquid-phase chain reaction: $A \xrightarrow{x_1} B \xrightarrow{x_2} C$, and it is formulated as follows:

$$\min_{x_1,x_2} J = \sum_{j=1}^{10}\sum_{i=1}^{2}\left(y_i(t_j) - y_i^{\exp}(t_j)\right)^2 \tag{18}$$

s.t.

$$\frac{dy_1}{dt} = -x_1 y_1 \quad \forall t \in [0,1] \tag{19}$$

$$\frac{dy_2}{dt} = x_1 y_1 - x_2 y_2 \quad \forall t \in [0,1] \tag{20}$$

$$y_1(0) = 1 \tag{21}$$

$$y_2(0) = 0 \tag{22}$$

$$0 \le x_1 \le 10 \tag{23}$$

$$0 \le x_2 \le 10 \tag{24}$$

where y_1 and y_2 are the mole fractions of components A and B, respectively; x_1 and x_2 are, respectively, the rate constants of the first and the second reactions that are to be estimated; $y_i^{\exp}(t_j)$ is the experimental point for state variable i at time t_j. The experimental points used in this work are taken from Floudas et al. (Table 2).[29]

In using the PPDE algorithm, the population size, K, is set to 20, and the scaling parameters, ρ_1 and ρ_2, are set to 0.6 and 0.5, respectively. In each iteration, 18 new solutions are generated ($M = 18$). The maximum iteration number is set to 500 ($L^{\max} = 500$) and the solution migration step is introduced after every 100 iterations ($N^{\mathrm{m}} = 100$). The MATLAB function "ode45" is used to solve the initial value problem described by Eqs. 19–22. The maximum integration time-step is set to 0.01 and the state variable value at the specified time instant (i.e., $y_i(t_j)$) is obtained through the cubic spline interpolation based on the numerical solution generated by "ode45." The PPDE method identifies the optimal parameters x_1 and x_2 at the values of 5.0034864 and 0.9999998, respectively, and the objective function value is 1.18584×10^{-6}. The experimental points and the state variable dynamics for the optimal parameters identified by PPDE are shown in Figure 3, which demonstrates an almost perfect fit of the experimental data.

The identified parameter values by the PPDE method are the same as those by Floudas et al. as well as by Papamichail and Adjiman (i.e., $x_1 = 5.0035$ and $x_2 = 1.0000$).[12,29] The objective function value identified by the PPDE is

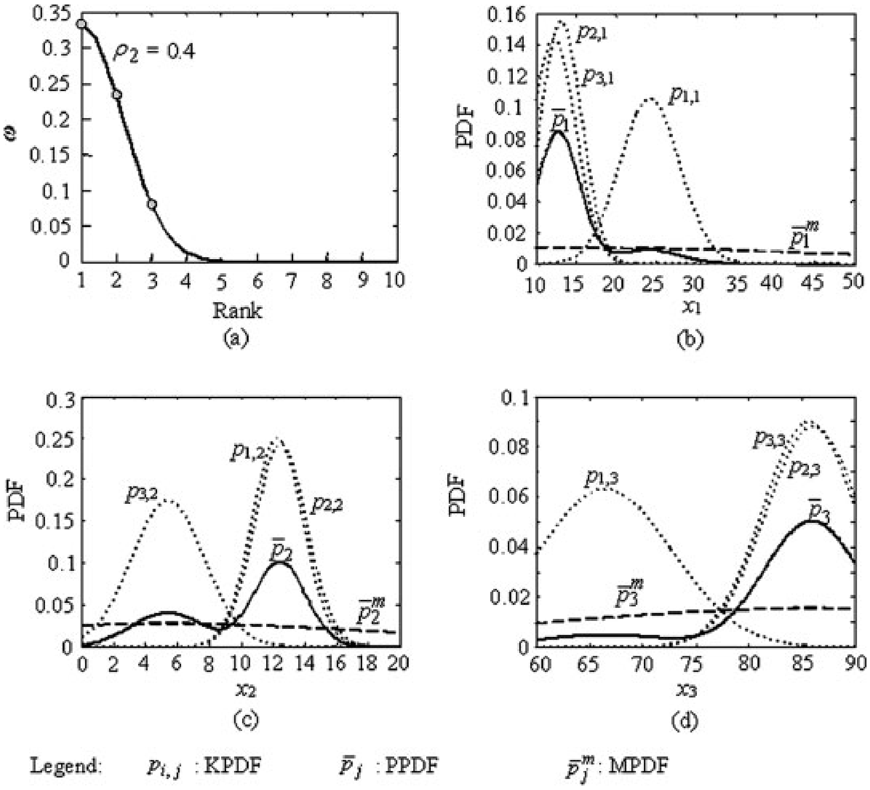

Figure 2. First iteration results: (a) individual preference ω vs. rank, (b) kernel probability density functions (KPDFs), population probability density function (PPDF), and probability density function in solution migration cases (MPDF) for decision variable x_1, (c) KPDFs, PPDF, and MPDF for decision variable x_2, and (d) KPDFs, PPDF, and MPDF for decision variable x_3.

1.18584×10^{-6}, which is the same as that by Floudas et al.[29] and Papamichail and Adjiman.[12] (Note that the objective function value in Papamichail and Adjiman was 1.18562×10^{-6}, which is erroneous because the objective function value, using their parameter values, should be 1.18585×10^{-6}).

One advantage of using the PPDE method is its computational efficiency. In this work, a single CPU desktop computer (2.39 GHz, 512 MB RAM) was used. The optimal solution (i.e., the objective function at the value of 1.18584×10^{-6}) was identified after 39 s of computation (after the 40 computational iterations) (Figure 4). Papamichail and Adjiman reported their computational time using an Ultra-60 workstation (2 × 360 MHz UltraSPARC-II CPU, 512 MB RAM), which was between 801 and 22,959 s, depending on the utilized underestimation schemes and optimality margins.[12] While a direct comparison of computational time is difficult because of different computers used, it is clear that the PPDE method is computationally very efficient.

Case II: polymeric coating curing optimization

In automotive coating, the film-build of the basecoat and clearcoat, usually known as the topcoat, is of utmost importance to the appearance and durability of the vehicle coating.[30] Three successive processes are involved in operation, i.e., a spray process for paint application, an ambient flash process for film topology finalization, and an oven baking process for solvent removal and crosslinking reactions completion. It is known that oven operation needs to be improved for coating quality assurance and energy reduction.

The case study by Xiao et al. is further investigated in this work.[17] The goal of the optimization problem is to find the optimal oven zone-based heating policy (oven wall temperature, circulated air temperature, and air velocity) for each vehicle passing through the oven so that the energy consumption is minimized, while coating quality can be guaranteed. In this case, the oven is divided into seven zones: two radiation/convection zones followed by five convection-only zones. There are seven stages in this problem ($N^{st} = 7$, see the general optimization formulation) and in each stage, the settings for the wall temperature, air temperature and air velocity are constants ($n^u = 3$, $n_i^d = 1$). Note that the wall temperatures in the five convection-only zones are not controlled in the paint shop. Consequently, there are 16 decision variables: two wall temperature in radiation/convection zones, and

Table 2. Experimental Data in Case I[29]

t_j	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$y_1^{exp}(t_j)$	0.606	0.368	0.223	0.135	0.082	0.050	0.030	0.018	0.011	0.007
$y_2^{exp}(t_j)$	0.373	0.564	0.647	0.669	0.656	0.624	0.583	0.539	0.494	0.451

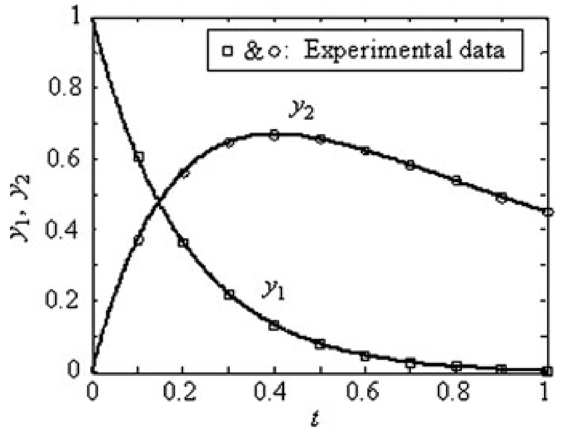

Figure 3. Experimental data and state variable dynamics for the optimal parameters in Case I.

seven air temperatures and seven air velocities in all seven zones.

Optimization Formulation. The optimization model by Xiao et al. is directly adopted in this work.[17]

Objective Function. The optimization task is defined below:

$$\text{Min} \quad Q = \sum_{i=1}^{N_\mathrm{r}} Q_i^\mathrm{r} + \sum_{i=1}^{N_\mathrm{z}-1} Q_i^\mathrm{v} \tag{25}$$

where N_r and N_z are the numbers of radiation zones and convection zones, respectively; Q_i^r and Q_i^v are, respectively, the radiation energy and the convection energy needed in the ith zone; they are evaluated by the following formulas:

$$Q_i^\mathrm{r} = \sum_{j=1}^{N_\mathrm{p}} \left[\int_{t_i^0}^{t_i^\mathrm{e}} \frac{A_j \Im\, \sigma\, \varepsilon\, e_\mathrm{r}^{-1} \left(\left(T_i^\mathrm{w} \right)^4 - T_{i,j}^4 \right)}{t_i^\mathrm{e} - t_i^0}\, dt \right] \qquad i = 1, 2, \ldots, N_\mathrm{r} \tag{26}$$

$$Q_i^\mathrm{v} = \sum_{j=1}^{N_\mathrm{p}} \left[\int_{t_i^0}^{t_i^\mathrm{e}} \frac{h_{i,j} A_j\, e_\mathrm{v}^{-1} \left(T_i^\mathrm{a} - T_{i,j} \right)}{t_i^\mathrm{e} - t_i^0}\, dt \right] \qquad i = 1, 2, \ldots, N_\mathrm{z} - 1 \tag{27}$$

where T_i^w and T_i^a are the wall and air temperature in the ith zone, respectively; e_r and e_v are the thermal efficiency of the radiation and the air convection, respectively; $T_{i,j}$ is the temperature of the jth panel in the ith zone; A_j is the surface area of the jth panel; $h_{i,j}$ is the heat transfer coefficient for the jth panel in the ith zone; t_i^0 and t_i^e are, respectively, the starting and ending times when a vehicle passes through the ith zone.

Constraints. The optimization is subject to three types of constraints: an integrated process dynamic model, coating quality, and solution feasibility.

Integrated Reactive Drying Model. The model by Lou and Huang is adopted in this work (see Appendix for detailed information about model development).[31]

Coating Quality Related Constraints. Various operation parameters are either directly or indirectly related to the eventual coating quality. These parameters are classified as follows.

(a) *Oven wall and convection air*. In any zone, the wall temperature setting must be restricted to avoid the loss of control of panel heating. The convection air temperature and air velocity should also be in their specific ranges. Thus, the constraints are

$$T_i^\mathrm{w} \le T_\mathrm{w}^\mathrm{u} \quad i = 1, 2, \ldots, N_\mathrm{r} \tag{28}$$

$$T_{\mathrm{a}_i}^\mathrm{l} \le T_i^\mathrm{a} \le T_{\mathrm{a}_i}^\mathrm{u} \quad i = 1, 2, \ldots, N_\mathrm{z} \tag{29}$$

$$V_{\mathrm{a}_i}^\mathrm{l} \le V_i^\mathrm{a} \le V_{\mathrm{a}_i}^\mathrm{u} \quad i = 1, 2, \ldots, N_\mathrm{z} \tag{30}$$

where T_w^u is the upper limit of the wall temperature; T_i^w is the wall temperature in the ith zone; $T_{\mathrm{a}_i}^\mathrm{l}$ and $T_{\mathrm{a}_i}^\mathrm{u}$ are the lower and upper bounds of the air temperature in the ith zone, respectively; $V_{\mathrm{a}_i}^\mathrm{l}$ and $V_{\mathrm{a}_i}^\mathrm{u}$ are the lower and upper limits of the air velocity in the ith zone, respectively.

(b) *Panel temperature*. In production, panel temperatures in different zones should be within specific ranges as well, i.e.,

$$T_i^\mathrm{l} \le T_{i,j} \le T_i^\mathrm{u} \tag{31}$$

where T_i^l and T_i^u are, respectively, the lower and upper bounds of the panel temperature in the ith zone.

(c) *Panel heating rate*. The temperature gradient of each panel at any time should not exceed the upper limit in each zone, i.e.,

$$\frac{dT_{i,j}}{dt} \le \mu_{i,j}^T \tag{32}$$

where $\mu_{i,j}^T$ is the upper limit of temperature gradient of the jth panel in the ith zone. Similarly, the temperature change when a vehicle moves from one zone to the next should also be restricted, i.e.,

$$\left| \frac{dT_{i,j}\left(t_i^\mathrm{e}\right)}{dt} - \frac{dT_{i+1,j}\left(t_{i+1}^0\right)}{dt} \right| \le \mu_{\Delta t} \tag{33}$$

where t_i^0 and t_i^e are, respectively, the starting and ending times when a vehicle passes through the ith zone; $\mu_{\Delta t}$ is the upper bound of temperature change rate between the two adjacent zones. When the vehicle moves into the last zone (i.e., the cooling zone), this constraint does not apply.

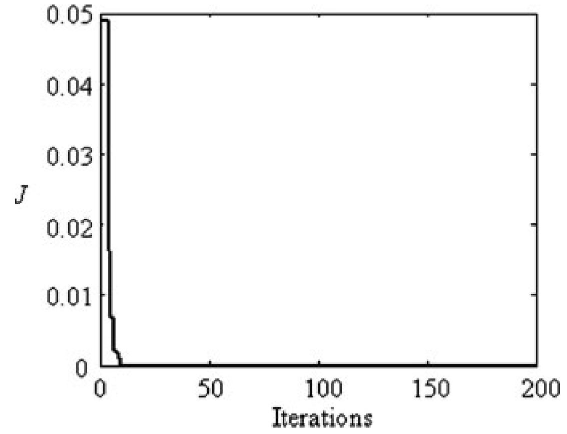

Figure 4. Solution revolution result for Case I (the first 200 iterations).

Table 3. Solution Comparison for the Polymeric Coating Curing Optimization Problem

Zone No.	Zone Length (m)	Original		Optimal by ACS[17]		Optimal by GA[17]		Optimal by PPDE	
		V^a (m/s)	T^w, T^a (K)	V^a (m/s)	T^w, T^a (K)	V^a (m/s)	T^w, T^a (K)	V^a (m/s)	T^w, T^a (K)
1	20.73	0.18	473, 403	0.29	500, 397	0.181	500.00, 404.22	0.219	499.84, 404.33
2	13.41	0.18	478, 478	0.16	500, 435	0.161	500.00, 462.30	0.025	499.96, 431.24
3	23.67	1.8	–, 428	1.53	–, 430	1.802	–, 427.42	1.009	–, 429.23
4	23.67	1.8	–, 423	1.47	–, 426	1.791	–, 422.54	2.584	–, 427.20
5	23.67	1.8	–, 418	2.27	–, 415	2.130	–, 414.51	1.555	–, 416.38
6	10.54	1.8	–, 418	2.25	–, 414	1.125	–, 396.14	2.371	–, 405.08
7	9.14	3.5	–, 300	3.5	–, 300	3.402	–, 292.34	3.618	–, 296.55
Energy (kWh)		289.92		258.87		261.32		249.02	

(d) *Crosslinking conversion percentage and time window.* The final crosslinking conversion percentage should reach a desired value and last for a certain time period. This gives,

$$r^{\mathrm{l}} \leq c_{i,j}(t) \tag{34}$$

$$t^{\mathrm{e}} - \Delta t_{\mathrm{x}} \leq t \leq t^{\mathrm{e}} \tag{35}$$

where r^{l} is the lower bound of the final conversion percentage; t^{e} is the time when the vehicle leaves the oven; Δt_{x} is the minimum time required for holding the final crosslinking conversion percentage.

(e) *Drying rate.* The drying rate should be restricted to prevent the occurrence of certain types of defects. That is,

$$-\frac{d\overline{w}_{i,j}}{dt} \leq \mu_i^w \tag{36}$$

where $\overline{w}_{i,j}$ is the mean solvent content of the film on the jth panel in the ith zone, μ_i^w is the upper limit of drying rate in the ith zone. Moreover, the solvent residue in the final dry film must be below a certain value and the thickness should be in a specific range. This means,

$$\overline{w}_j^{\mathrm{e}} \leq w^{\mathrm{e,u}} \tag{37}$$

$$z^{\mathrm{e,l}} \leq z_j^{\mathrm{e}} \leq z^{\mathrm{e,u}} \tag{38}$$

where $w^{\mathrm{e,u}}$ is the maximum final mean solvent residue; $z^{\mathrm{e,l}}$ and $z^{\mathrm{e,u}}$ are the lower and upper limits of the final film thickness, respectively.

Solution Feasibility Requirement. Non-negative energy consumption, solvent residue, film thickness and crosslinking conversion percentage should be ensured.

Solution Derivation and Analysis. Three types of parameters should be specified before optimization. The parameters of the integrated dynamic model and the constraints are set exactly the same as those in Xiao et al., except that the initial oven settings are not required when using the PPDE.[17] In the PPDE, the population size, K, is set to 20. The scaling parameters, ρ_1 and ρ_2, are both set to 0.4. In each iteration, 18 new solutions are generated ($M = 18$). The maximum iteration number is set to 500 ($L^{\max} = 500$) and the solution migration step is introduced once every 100 iterations ($N^{\mathrm{m}} = 100$).

Table 3 lists the optimal temperature and air velocity settings for each zone derived from the ACS,[17] the GA,[17] and the PPDE algorithms. Figure 5 shows the solution revolution results. These results clearly demonstrate the superiority of the PPDE algorithm to the ACS-based dynamic optimization method and GA in the following aspects.

(a) Improvement of solution quality. By comparing with the base case, the new settings obtained from the ACS, the GA and the PPDE reduce the overall energy consumption in the oven to 258.87 kW h (10.74% reduction), 261.32 kW h (9.8% reduction), and 249.02 kW h (14.11% reduction), respectively.

(b) Significant improvement of computational efficiency. As observed from Figure 5b, the PPDE identifies an optimal solution within 6 min, while it costs ACS more than 20 min to converge to a local optimal solution. If a feasible initial solution is given, the PPDE can obtain that local optimal solution derived by ACS in only about 20 s. It is also shown by this figure that the convergence of PPDE is faster than that of GA.

(c) Improvement of solution precision. By comparing with ACS, the PPDE advances solution identification process from discretized space search to continuous space search. Consequently, solution precision has been improved effectively.

(d) Applicability enhancement. Different from ACS, a feasible initial solution is not required by the PPDE. Figure 5a shows that a feasible solution can be identified by the PPDE effectively (in this case, within 100 iterations). This enables the PPDE to be applicable to the cases where only limited system information is available.

Concluding Remarks

A PPDE framework has been developed to solve engineering optimization problems, which in most cases deal with highly constrained and nonlinear multistage complex dynamic processes. By comparing with the ACS-based dynamic optimization approach by Xiao et al.,[17] the solution identification process is advanced from search-tree-based discretized-space search to probability distribution estimation based continuous space iterative search. The PPDFs, which represent probability distribution, are refined iteration after iteration to find out the promising spaces containing high quality solutions. A special penalty assignment technique is applied to tackle various constraints. Premature convergence is effectively prevented by the integrated solution migration method.

The optimal parameter estimation example and the dynamic engineering optimization problem are used to evaluate the efficacy of the PPDE. The superior algorithm performance in terms of solution quality and computational efficiency is demonstrated through comparing results with other methods (e.g., deterministic method, GA and ACS). The

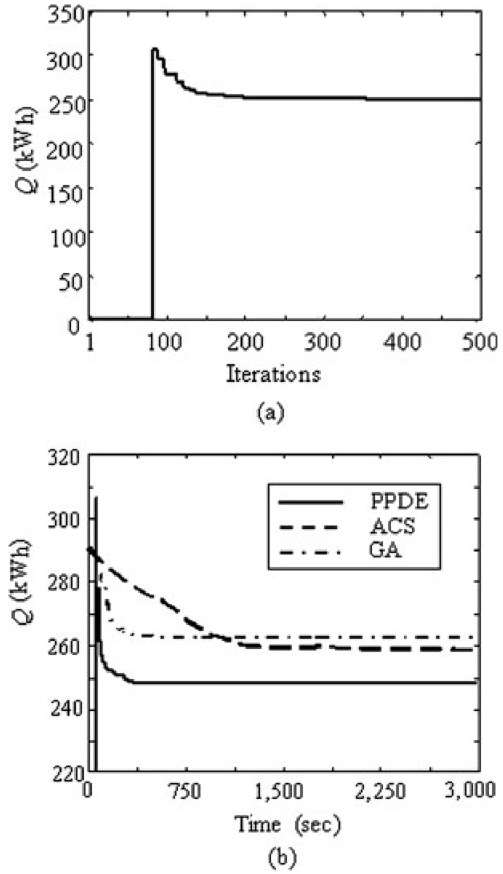

Figure 5. Solution revolution results for polymeric coating curing optimization problem: (a) PPDE result (objective function value vs. iteration), and (b) PPDE, ACS, and GA results comparison (objective function value vs. computational time).

PPDE method is, in general, applicable to a variety of industrial dynamic optimization problems, where product and process performance can be simultaneously considered.

Acknowledgments

This work is in part supported by NSF (CTS 0091398 and CTS 0407494) and the Institute of Manufacturing Research of Wayne State University. Technical assistance from Ford Advanced Manufacturing Technology Development Center is also gratefully acknowledged.

Notation

c = polymerization conversion degree
$I_{\rm o}$ = option index
J = objective function
K = number of solutions in the solution population matrix
$L^{\max}$ = total number of iterations
M = number of new solutions generated in each iteration
N = number of decision variables in each solution
$N^{\rm st}$ = number of stages
$n^{\rm c}$ = number of inequality constraints
$n^{\rm u}$ = number of system input variables
$n_i^{\rm d}$ = number of time intervals divided in the ith stage
$n_i^{\rm v}$ = number of violated constraints when using the ith solution
$\boldsymbol{P}$ = kernel probability density function (KPDF) matrix
$p_{i,j}$ = ith KPDF for the jth decision variable
$\bar{\boldsymbol{P}}$ = population probability density function (PPDF) vector
$\bar{p}_j$ = PPDF for the jth decision variable
$\bar{\boldsymbol{P}}^{\rm m}$ = probability density function vector in the solution migration case (MPDF vector)
$\bar{p}_j^{\rm m}$ = MPDF for the jth decision variable
$\hat{\boldsymbol{p}}$ = probability distribution used for selecting KPDF for the new solution construction
$\hat{p}_i$ = probability of choosing the ith KPDF
Q = total energy consumption, kWh
$\boldsymbol{R}$ = ranking vector
r_i = rank of the ith solution in the solution population
T = panel and film temperature, K
$T^{\rm a}$ = convection air temperature, K
$T^{\rm w}$ = oven wall temperature, K
$V^{\rm a}$ = convection air velocity, m/s
w = solvent content in film, kg solvent/kg solids in paint
$\tilde{X}$ = solution population matrix
$\tilde{X}^q$ = solution population at the qth iteration
$\tilde{X}^{\rm opt}$ = the best solution
x_j = jth decision variable
$\tilde{x}_{i,j}$ = value for the jth decision variable in the ith solution of the solution population

Greek letters

ρ = scaling factor
Ω = individual preference vector
ω_i = weight/preference for the ith solution

Literature Cited

1. Rao SS. Engineering Optimization: Theory and Practice. New York: Wiley, 1996.
2. Edgar TF, Himmelblau DM, Lasdon LS. Optimization of Chemical Processes. New York: Wiley, 1989.
3. Horst R, Tuy H. Global Optimization: Deterministic Approaches. Berlin: Springer-Verlag, 1996.
4. Floudas CA. Deterministic Global Optimization: Theory, Methods and Applications. Dordrecht: Kluwer Academic Publishers, 2000.
5. Grossmann IE. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optim Eng*. 2002;3:227–252.
6. Sahinidis NV. BARON: a general purpose global optimization software. *J Global Optim*. 1996;8:201–205.
7. Adjiman CS, Androulakis IP, Floudas CA. Global optimization of mixed-integer nonlinear problems. *AIChE J*. 2000;46:1769–1797.
8. Biegler LT, Grossmann IE. Retrospective on Optimization. *Comput Chem Eng*. 2004;28:1169–1192.
9. Gaghunathan AU, Diaz MS, Biegler LT. An MPEC formulation for dynamic optimization of distillation operations. *Comput Chem Eng*. 2004;28:2037–2052.
10. Flores-Tlacuahuac A, Biegler LT, Saldivar-Guerra E. Dynamic optimization of HIPS open-loop unstable polymerization reactors. *Ind Eng Chem Res*. 2005;44:2659–2674.
11. Papamichail I, Adjiman CS. A rigorous global optimization algorithm for problems with ordinary differential equations. *J Global Optim*. 2002;24:1–33.
12. Papamichail I, Adjiman CS. Global optimization of dynamic systems. *Comput Chem Eng*. 2004;28:403–415.
13. Chachuat B, Singer AB, Barton PI. Global mixed-integer dynamic optimization. *AIChE J*. 2005;51:2235–2253.
14. Chachuat B, Singer AB, Barton PI. Global methods for dynamic optimization and mixed-integer dynamic optimization. *Ind Eng Chem Res*. 2006;45:8373–8392.
15. Chiou J, Wang F. Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process. *Comput Chem Eng*. 1999;23:1277–1291.
16. Hu X, Eberhart RC, Shi Y. Engineering optimization with particle swarm. Proceedings of the IEEE Swarm Intelligence Symposium. Indianapolis, IN, 2003;53–57.
17. Xiao J, Li J, Xu Q, Huang Y, Lou HH. ACS-based dynamic optimization for curing of polymeric coating. *AIChE J*. 2006;52:1410–1422.

18. Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans n Evol Comput*. 1997;1:53–66.
19. Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern*. 1996; 26:29–41.
20. Xiao J, Li J, Lou HH, Huang Y. Cure-window-based proactive quality control in topcoat curing. *Ind Eng Chem Res*. 2006;45:2351–2360.
21. Bilchev B, Parmee IC. The ant colony metaphor for searching continuous design spaces. In: Fogarty TC, Editor, Proceedings of the AISB Workshop on Evolutionary Computation. Lecture Notes in Computer Science, Vol. 993, Springer, Berlin, 1995:25–39.
22. Dreo J, Siarry P. A new ant colony algorithm using the heterarchical concept aimed at optimization of multi minima continuous functions. In: Dorigo M, Di Caro G, Sampels M, Editors, Ant algorithms—Proceedings of ANTS 2002—Third international workshop. Lecture Notes in Computer Science. Vol. 2463, Springer, Berlin, 2002:216–221.
23. Socha K. ACO for continuous and mixed-variable optimization. In: Dorigo M, Birattari M, Blum C, Gambardella LM, Mondada F, Stützle T, Editors, Proceedings of ANTS 2004—Fourth international workshop on Ant colony optimization and swarm intelligence. Lecture Notes in Computer Science. Vol. 3172, Springer, Berlin, 2004:25–36.
24. Blum C, Socha K. Training feed-forward neural networks with ant colony optimization: an application to pattern classification. Proceedings of the 5th international conference on hybrid intelligent systems (HIS). IEEE Press, Piscataway, NJ, 2005;233–238.
25. Deb K. A population-based algorithm-generator for real-parameter optimization. *Soft Comput*. 2005;9:236–253.
26. Pelikan M, Goldberg DE, Lobo FG. A survey of optimization by building and using probabilistic models. *Comput Optim Appl*. 2002; 21:5–20.
27. Towers S. Kernel probability density estimation methods. Available at http://www.ippp.dur.ac.uk/Workshops/02/statistics/proceedings/towers1. pdf.
28. Knuteson B, Miettinen HE, Holmstrom L. αPDE: a new multivariate technique for parameter estimation. *Comput Phys Commun*. 2002; 145:351–356.
29. Floudas CA, Pardalos PM, Adjiman CS, Esposito WR, Gumus ZH, Harding ST, Klepeis JL, Meyer CA, Schweiger CA. Handbook of Test Problems in Local and Global Optimization. Dordrecht: Kluwer, 1999.
30. Cohen ED, Gutoff EB. Modern Coating and Drying Technology. New York: VCH, 1992.
31. Lou HH, Huang YL. Integrated modeling and simulation for improved reactive drying of clearcoat. *Ind Eng Chem Res*. 2000; 39:500–507.
32. Dickie RA, Bauer DR, Ward SM, Wagner DA. Modeling paint and adhesive cure in automotive applications. *Prog Org Coat*. 1997; 31:209–216.

Appendix: Integrated Reactive Drying Model

The integrated reactive drying model by Lou and Huang adopted in this work is briefly presented below.[31]

(a) *Panel heating model*. The panel temperature dynamics can be described as follows.

$$\rho_j^{\mathrm{m}} C_{p_j}^{\mathrm{m}} Z_j^{\mathrm{m}} \frac{dT_{i,j}}{dt} = \Im\sigma\varepsilon\left(\left(T_i^{\mathrm{w}}\right)^4 - T_{i,j}^4\right) + h_{i,j}\left(T_i^{\mathrm{a}} - T_{i,j}\right) \quad i = 1,2,\ldots,N_{\mathrm{r}} \qquad \text{(A1)}$$

$$h_{i,j} = \beta_{i,j}\left(V_{i,j}^{\mathrm{a}}\right)^{0.7} \qquad \text{(A2)}$$

$$\rho_j^{\mathrm{m}} C_{p_j}^{\mathrm{m}} Z_j^{\mathrm{m}} \frac{dT_{i,j}}{dt} = h_{i,j}\left(T_i^{\mathrm{a}} - T_{i,j}\right) \quad i = (N_{\mathrm{r}}+1),2,\ldots,N_{\mathrm{z}} \qquad \text{(A3)}$$

where ρ^{m}, C_p^{m}, and Z^{m} are the density, heat capacity, and the thickness of the metal substrate, respectively; $\Im$ is the viewing factor for radiation; σ is the Stefen-Boltzman constant; ε is the emissivity; h is the heat transfer coefficient, that is a function of convection air velocity (V^{a}), the distance between the panel and the convection air nozzles, and others (expressed by β).

(b) *Solvent removal model*. To facilitate the modeling, the clearcoat film is divided into a number of very thin slices, each of which has a fixed thickness (Δz). The slices are numbered starting from the one just above the basecoat surface. The following models can reveal the solvent (or water) removal process occurred on the surface and within the clearcoat:

$$\Delta z \frac{\partial w_{i,j,k}}{\partial t} = D_{f_{i,j,k}} \frac{\partial w_{i,j,k}}{\partial z} \quad k = 1 \qquad \text{(A4)}$$

$$D_{f_{i,j,k}} = \eta e^{-\left((\gamma/w_{i,j,k}) + (E_d/RT_{i,j})\right)} \quad k = 1,2,\ldots,N_j^{\mathrm{s}} \qquad \text{(A5)}$$

$$\frac{\partial w_{i,j,k}}{\partial t} = \frac{\partial}{\partial z}\left(D_{f_{i,j,k}} \frac{\partial w_{i,j,k}}{\partial z}\right) \quad k = 2,3,\ldots,(N_j^{\mathrm{s}} - 1) \qquad \text{(A6)}$$

$$\rho_j^{\mathrm{s}} z_j^{\mathrm{s}} \frac{\partial w_{i,j,k}}{\partial t} = K^{\mathrm{t}} \frac{(P_{\mathrm{a}} - \alpha P_{\mathrm{s}})}{P} + \rho_j^{\mathrm{s}} D_{f_{i,j,k}} \frac{\partial w_{i,j,k}}{\partial z} \quad k = N_j^{\mathrm{s}} \qquad \text{(A7)}$$

where $w_{i,,j,k}$ is the solvent content in the kth slice of the film on the jth panel in the ith zone; N_j^{s} is the total number of slices; D_f is the diffusivity; η is a pre-exponential constant for diffusion; γ is a constant; E_{d} is the activation energy for diffusion; ρ_j^{s} and z_j^{s} are the density and thickness of unreducible components in the film of the jth panel, respectively; K^{t} is the mass transfer coefficient.

(c) *Film thickness model*. Different from the solvent removal model, this model contains the following integral equations for each panel:

$$\rho_j^{\mathrm{l}}\left(z_j(t) - z_j(t+\Delta t)\right) = \rho_j^{\mathrm{s}}\left(\int_0^{z_j(t)} w_j(t)\mathrm{d}z - \int_0^{z_j(t+\Delta t)} w_j(t+\Delta t)\mathrm{d}z\right) \qquad \text{(A8)}$$

where ρ_j^{l} and ρ_j^{s} are densities of solvent and unreducible components in the film of the jth panel, respectively.

(d) *Polymerization model*. According to Dickie et al.,[32] the following first-order kinetic model can capture the major phenomena of the coating of each panel when it is baked:

$$\frac{dc_{i,j}}{dt} = \zeta e^{-E_{\mathrm{r}}/RT_{i,j}}\left(1 - c_{i,j}\right) \qquad \text{(A9)}$$

where $c_{i,j}$ is the polymerization conversion of the film on the jth panel in the ith zone, E_{r} is the reaction activation energy, and ζ is the reaction frequency factor.

Manuscript received Apr. 9, 2007.