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

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. 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. Over the past years, significant progress has been made in developing deterministic global optimization methods for solving non-convex nonlinear programming (NLP) and mixed-integer nonlinear programming (MINLP) problems. Efforts have been also devoted to global optimization for dynamic systems. 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.

Recently, Xiao et al. introduced a mathematical framework for solving complex process dynamic optimization problems, which is an extension of a relatively new meta-heu-
ristic approach called ant colony system (ACS) algorithm. 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. 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).

The latest approach, which is at the same time closest to the spirit of ACO for combinatorial problems, is proposed by Socha and coworkers. 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., 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:

\[
\min_{X(t)} J = \sum_{i=1}^{N_{st}} \Phi_i \left( Y(t,z); X(t); t; U(z) \right) \tag{1}
\]

s.t.

\[
F_i \left( \frac{\partial Y}{\partial t}, \frac{\partial Y}{\partial z}; Y(t,z); X(t); t; U(z) \right) = 0 \quad i = 1, \ldots, N_{id} \tag{2}
\]

\[
G_i \left( Y(t^i_0, z); Y(t^i_f, z); Y(t, z^i_1); \frac{\partial Y}{\partial z} (t, z^i_2) \right) = 0 \quad i = 1, \ldots, N_{id} \tag{3}
\]

\[
H_i \left( \frac{\partial Y}{\partial t}; Y(t,z); U \left( z \right); \Phi_i \right) \leq 0 \quad i = 1, \ldots, N_{id} \tag{4}
\]

\[
X^i_{\min} \leq X_i(t) \leq X^i_{\max} \quad i = 1, \ldots, N_{id} \tag{5}
\]

where \( \Phi_i \)’s are scalar sub-objective functions; \( F_i \)’s are integrated process-product system models; \( G_i \)’s are initial and boundary conditions for the system models; \( H_i \)’s are series of inequality constraints for the optimization problem; subscript \( i \) represents the stage index; \( N_{id} \) is the total number of stages; \( X \) and \( Y \) denote the system input and output variables, respectively; \( z \) is the generalized coordinate (e.g., one coordinate in Cartesian coordinates); \( z \) is the generalized position vector; \( U \) includes all time-independent parameters, some of which are dependent on positions; \( \ell \) and \( \ell_f \) are, respectively, the starting and ending times of the \( i \)th stage; \( z^i_1 \) and \( z^i_2 \) are two boundaries; and \( X^i_{\min} \) and \( X^i_{\max} \) 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-wise local optimization does 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.
26 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. 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. 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), 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). 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 \( n_x \) input variables in \( X \), the time duration of each stage \([t_i, t_{i+1}]\) is divided into \( n_i \) intervals, and the system input in each time interval is constant. Note that all time intervals have the same time duration \( \Delta t \). After time horizon discretization, the optimization objective is to identify the optimal values for \( \sum_{i=1}^{N} n_i \) 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. 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.
Solution Population Matrix. In matrix \( \mathbf{X} = [\tilde{x}_{ij}]_{K \times N} \), the elements in each row form a solution for the optimization problem (called an individual solution, either feasible or infeasible). Element \( \tilde{x}_{ij} \) is the value for the \( j \)th decision variable, \( x_j \), in the \( i \)th 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, \( \mathbf{P}(\mathbf{p}_{i,j} | \mathbf{x}) \), is constructed from \( \mathbf{X} \). Each element \( p_{ij} \) (called a kernel probability density function or simply KPDF) is a Gaussian (or normal) function centered at \( \tilde{x}_{ij} \), i.e.,

\[
p_{ij}(x_j; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x_j - \mu)^2}{2\sigma^2}\right) \tag{6}
\]

where

\[
\mu = \tilde{x}_{ij} \tag{7}
\]

\[
\sigma = \frac{\rho_1}{\sqrt{K - 1}} \sqrt{\frac{1}{K} \sum_{k=1}^{K} (\tilde{x}_{ki} - \tilde{x}_{ij})^2} \tag{8}
\]

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

Probability Distribution Estimation. The weighted sum of KPDFs (i.e., \( p_{ij}, i = 1, \ldots, K \)) 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 \( \mathbf{\Omega} = [\omega_i]_{1 \times K} \), element \( \omega_i \in (0, 1) \) gives a weight for the \( i \)th KPDF (there are \( K \) KPDFs for the decision variable \( x_j \)). This vector also shows the solution quality. The larger the value of \( \omega_i \), the better the \( i \)th solution.

The population probability density function vector, \( \mathbf{P} = [p_j]_{1 \times N} \), can be constructed from \( \mathbf{P} \) and \( \mathbf{\Omega} \), i.e.,

\[
\mathbf{P} = \mathbf{\Omega}^T \mathbf{P} \tag{9}
\]

Note that element \( p_j \) (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.,

\[
p_j = \sum_{i=1}^{K} (\omega_i \times p_{ij}) \quad j = 1, \ldots, N \tag{10}
\]

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, \( \tilde{p}_j \) 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 \( \tilde{p}_j \) (or simply called sampling \( \tilde{p}_j \)) 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 \( \tilde{p}_j \) according to the following probability distribution:

\[
\tilde{p}_j = \frac{p_{ij}}{\sum_{i=1}^{K} p_{ij}} \quad i = 1, \ldots, K \tag{11}
\]

where \( \tilde{p}_j \) is the probability of choosing \( p_{ij} \), the \( i \)th KPDF for the \( j \)th decision variable. Remember that a better solution has a larger value of the weight, \( \omega_i \). 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, \( p_{ij} \). 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 \( p_1 \) through \( p_N \) 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., \( \mathbf{P}^m = [\tilde{p}_{j}^m]_{1 \times N} \), using the following equation:

\[
\tilde{p}_{j}^m = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x_j - \mu)^2}{2\sigma^2}\right) \quad j = 1, \ldots, N \tag{12}
\]

where

\[
\mu = \tilde{x}_{r,j} \tag{13}
\]

\[
\sigma = \max\left(\tilde{x}_{r,j} - x_j^{\min}, x_j^{\max} - \tilde{x}_{r,j}\right) \tag{14}
\]

where \( \tilde{p}_{j}^m \) is the PDF in the solution migration case (or simply called MPDF) for the \( j \)th decision variable; \( \tilde{x}_{r,j} \) is the value for the \( j \)th 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 \( j \)th decision variable, respectively.
Matrix $P^m$ gives probability distribution of migrated solutions. Sampling $p^{m}_i$ will lead to a new value for the $i$th decision variable. This can be accomplished by generating a random number according to the normal distribution, $p^{m}_i$. Keep in mind that the bounds for the decision variables should be satisfied. Sampling $p^{m}_i$ through $p^{m}_n$ 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 $\Omega$ 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 value) together with the constraints compliance conditions. 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 identification procedure**

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

**Step 1.** Generate an initial solution population $X^0$ 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 $R$.

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

**Step 5.** Generate KPDF matrix $P$ using Eqs. 6–8. PPDF vector $P$ can be constructed from $P$ and $\Omega$ using Eq. 9.
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.,

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

where $I_o$ is the option index; $q$ is the iteration number; $N^m$ is a positive integer number; $\bmod$ 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 $\widetilde{X}^{\text{opt}}$ is obtained and $J(\widetilde{X}^{\text{opt}})$ 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 $\widetilde{X}^m$ are directly replaced by $M$ new solutions to form a new solution population, $\widetilde{X}^m + 1$, 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 $\widetilde{X}^m + 1$ need to be ranked. The corresponding ranking vector $R$ (the first $K$ ranks in total $K + M$ ranks) should be recorded.

(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 $\widetilde{X}^m$, 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 $\widetilde{X}^{m+1}$ for the next iteration. The corresponding ranking vector $R$ (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 ($m^c = 2$). 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 $\rho_1$ and $\rho_2$ 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 “$X^{\text{opt}}$”). 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 $\widetilde{X}^{\text{opt}}$. The elements in the individual preference vector $\Omega$ are calculated from Eq. 16 and has been shown in Table 1 and Figure 1a. Note that in Figure 1a, two cases with different $\rho_1$ values are shown to illustrate the function that $\rho_2$ has (see the comments on $\rho_2$ in the paragraph after Eq. 16).

**Step 5.** PPDFs and PPDFs can be obtained using Eqs. 6–9. They are shown in Figures 1b–d. Each decision variable has three PPDFs ($K = 3$) 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 $\rho_1$, $\rho_2$, and $\rho_1$. 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 $\rho_{3,1}$, $\rho_{1,2}$, and $\rho_{3,3}$ (Figure 1) are chosen and one new solution (13.1, 12.5, 86.1) is generated. In Figure 1, $\Omega$ is also shown to illustrate the solution migration effect. As indicated in this figure, for decision variable $x_j$, MPDF (i.e., $p_{j}^{\text{MPDF}}$) has a much larger standard deviation than any KPDF (i.e., $p_{j,1}$, $p_{j,2}$, or $p_{j,3}$) has. It is understandable that generating new solutions according to $\Omega$ (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 $\widetilde{X}^m$. 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 $\widetilde{X}^{\text{opt}}$. 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.
**Case I: Parameter estimation example**

Floudas et al. and then Papamichail and Adjiman solved an interesting parameter estimation problem using deterministic global optimization techniques. The dynamic optimization problem involves a first-order irreversible isothermal liquid-phase chain reaction:

\[
\begin{align*}
\text{A} & \xrightarrow{C_0} \text{B} \xrightarrow{C_1} \text{C},
\end{align*}
\]

and it is formulated as follows:

\[
\begin{align*}
\min_{x_1, x_2} J &= \sum_{j=1}^{10} \sum_{i=1}^{2} (y_i(t_j) - y_i^{exp}(t_j))^2 \\
\text{s.t.} \quad \frac{dy_1}{dt} &= -x_1 y_1 \quad \forall t \in [0, 1] \\
\frac{dy_2}{dt} &= x_1 y_1 - x_2 y_2 \quad \forall t \in [0, 1] \\
0 &= y_1(0) \quad \text{(21)} \\
0 &= y_2(0) \quad \text{(22)} \\
0 &\leq x_1 \leq 10 \quad \text{(23)} \\
0 &\leq x_2 \leq 10 \quad \text{(24)}
\end{align*}
\]

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).

In using the PPDE algorithm, the population size, \(K\), is set to 20, and the scaling parameters, \(\rho_1\) and \(\rho_2\), 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 (\(L_{\text{max}} = 500\)) and the solution migration step is introduced after every 100 iterations (\(N_{\text{m}} = 100\)). 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 \(\times 10^{-6}\). 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 \(x_2 = 1.0000\)). The objective function value identified by the PPDE is...
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 = 7, see the general optimization formulation) and in each stage, the settings for the wall temperature, air temperature and air velocity are constants (n² = 3, n³ = 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>
<thead>
<tr>
<th>t</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>0.9</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>x₁²₀₅(t₁)</td>
<td>0.606</td>
<td>0.368</td>
<td>0.223</td>
<td>0.135</td>
<td>0.082</td>
<td>0.050</td>
<td>0.030</td>
<td>0.018</td>
<td>0.011</td>
<td>0.007</td>
</tr>
<tr>
<td>x₂²₀₅(t₁)</td>
<td>0.373</td>
<td>0.564</td>
<td>0.647</td>
<td>0.669</td>
<td>0.656</td>
<td>0.624</td>
<td>0.583</td>
<td>0.539</td>
<td>0.494</td>
<td>0.451</td>
</tr>
</tbody>
</table>

Figure 2. First iteration results: (a) individual preference α 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 x₁, (c) KPDFs, PPDF, and MPDF for decision variable x₂, and (d) KPDFs, PPDF, and MPDF for decision variable x₃.
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. The objective function is defined as follows:

\[
Q = \sum_{i=1}^{N_i} Q_i + \sum_{i=1}^{N_e-1} Q_e
\]

where \(N_i\) and \(N_e\) are the numbers of radiation zones and convection zones, respectively; \(Q_i\) and \(Q_e\) are, respectively, the radiation energy and the convection energy needed in the \(i\)th zone; they are evaluated by the following formulas:

\[
Q_i = \sum_{j=1}^{N_z} \left[ \int_{t^0_i}^{t^1_i} \frac{A_j \varepsilon_s \epsilon_r \left( T_{w}^a - T_{w}^d \right)}{\epsilon_r^e - \epsilon_r^0} \right. dt
\]

\[
Q_e = \sum_{j=1}^{N_z} \left[ \int_{t^0_e}^{t^1_e} \frac{h_{ij} A_j \varepsilon_v \left( T_{a}^a - T_{a}^d \right)}{\epsilon_v^e - \epsilon_v^0} \right. dt
\]

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).

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

\[
T_w^i \leq T_w^u \quad i = 1, 2, \ldots, N_i
\]

\[
T_a^i \leq T_a^u \quad i = 1, 2, \ldots, N_e
\]

\[
V_a^i \leq V_a^u \quad i = 1, 2, \ldots, N_e
\]

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

\[
T_i^l \leq T_i \leq T_i^u
\]

where \(T_i^l\) and \(T_i^u\) are, respectively, the lower and upper bounds of the panel temperature in the \(i\)th 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.,

\[
\frac{dT_{ij}}{dt} \leq \mu_{ij}^T
\]

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

\[
\left| \frac{dT_{ij}}{dt} \right| \leq \mu_{ij}
\]

Figure 4. Solution revolution result for Case I (the first 200 iterations).
The final crosslinking conversion percentage should reach a desired value and last for a certain time period. This gives,

\[ r^f \leq c_i(t) \]  

(34)

\[ r^f - \Delta t_i \leq t \leq r^f \]  

(35)

where \( r^f \) is the lower bound of the final conversion percentage; \( r^f \) is the time when the vehicle leaves the oven; \( \Delta t_i \) 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,

\[ -\frac{d\pi_{ij}}{dt} \leq \mu_i^w \]  

(36)

where \( \pi_{ij} \) is the mean solvent content of the film on the \( j \)th panel in the \( i \)th zone, \( \mu_i^w \) is the upper limit of drying rate in the \( i \)th 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,

\[ \pi_{ij}^e \leq \pi_{ij} \]  

(37)

\[ z_{ij}^l \leq \pi_{ij} \leq z_{ij}^u \]  

(38)

where \( \pi_{ij}^e \) is the maximum final mean solvent residue; \( z_{ij}^l \) and \( z_{ij}^u \) 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, \( p_1 \) and \( p_2 \), are both set to 0.4. In each iteration, 18 new solutions are generated (\( M = 18 \)). The maximum iteration number is set to 500 (\( L_{\text{max}} = 500 \)) and the solution migration step is introduced once every 100 iterations (\( N_{100} = 100 \)).

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.,\(^{15}\) 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
Technological Development Center is also gratefully acknowledged.

University. Technical assistance from Ford Advanced Manufacturing
0407494) and the Institute of Manufacturing Research of Wayne State


ces performance can be simultaneously considered.

PPDE method is, in general, applicable to a variety of indus-

of trial dynamic optimization problems, where product and pro-

PPDE method is, in general, applicable to a variety of indus-

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).

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
\( I_o \) = 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
\( n^e \) = number of stages
\( n^o \) = number of inequality constraints
\( n^s \) = number of system input variables
\( n^t \) = number of time intervals divided in the \( j \)th stage
\( n^v \) = number of violated constraints when using the \( j \)th solution
\( P \) = kernel probability density function (KPDF) matrix
\( \rho_{ij} \) = \( j \)th KPDF for the \( i \)th decision variable
\( P = \) population probability density function (PPDF) vector
\( \rho_i = \) PPDF for the \( i \)th decision variable
\( p_m = \) probability density function vector in the solution migration case
\( \rho_j = \) MPDF for the \( j \)th decision variable
\( p = \) probability distribution used for selecting KPDF for the new
solution construction
\( \rho_i = \) probability of choosing the \( i \)th KPDF
\( Q = \) total energy consumption, kWh
\( R \) = ranking vector
\( r = \) rank of the \( j \)th solution in the solution population
\( T \) = panel and film temperature, K
\( T^a = \) convection air temperature, K
\( T^w = \) oven wall temperature, K
\( V^a = \) convection air velocity, m/s
\( w = \) weight/preference for the \( i \)th decision variable
\( x_{ij} = \) value for the \( j \)th decision variable in the \( i \)th solution of the
solution population
\( X^{opt} = \) the best solution
\( x_i = \) \( j \)th decision variable
\( \Omega = \) individual preference vector
\( \omega_i = \) weight/preference for the \( i \)th solution

Greek letters

\( \rho = \) scaling factor

Literature Cited

1. Rao SS. Engineering Optimization: Theory and Practice. New York:
2. Edgar TF, Himmelblau DM, Lasdon LS. Optimization of Chemical
Berlin: Springer-Verlag, 1996.
4. Floudas CA. Deterministic Global Optimization: Theory, Meth-
ods and Applications. Dordrecht: Kluwer Academic Publishers,
2000.
5. Grossmann IE. Review of nonlinear mixed-integer and disjunctive
6. Sahinidis NV. BARON: a general purpose global optimization soft-
7. Adjiman CS, Androulakis IP, Floudas CA. Global optimization of
8. Biegler LT, Grossmann IE. Retrospective on Optimization. Comput
9. Gagunathan AU, Diaz MS, Biegler LT. An MPEC formulation for
10. Flores-Tlacuahuac A, Biegler LT, Saldivar-Guerra E. Dynamic opti-
mization of HIPS open-loop unstable polymerization reactors. Ind
11. Papamichail I, Adjiman CS. A rigorous global optimization algo-
rithm for problems with ordinary differential equations. J Global
Optim. 2002;24:1–33.
12. Papamichail I, Adjiman CS. Global optimization of dynamic sys-
13. Chachuat B, Singer AB, Barton PL. Global mixed-integer dynamic
14. Chachuat B, Singer AB, Barton PL. Global methods for dynamic
optimization and mixed-integer dynamic optimization. Ind Eng
15. Chiu J, Wang F. Hybrid method of evolutionary algorithms for
static and dynamic optimization problems with application to a fed-
16. Hu X, Eberhart RC, Shi Y. Engineering optimization with particle
swarm. Proceedings of the IEEE Swarm Intelligence Symposium. In-
dianapolis, IN, 2003;53–57.
17. Xiao J, Li J, Xu Q, Huang Y, Lou HH. ACS-based dynamic optimiza-
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.

\[
\rho_j^m C_{p,m}^{m} \frac{dT_{i,j}}{dt} = \mp \sigma \varepsilon \left( T_{i,j}^4 - T_{i,j}^4 \right) + h_{z,j} (T_{i,j}^4 - T_{i,j}) \quad i = 1, 2, \ldots, N_t \quad (A1)
\]

\[
h_{z,j} = \beta_{z,j} \left( \frac{V_{j}}{V_{i,j}} \right)^{0.7} \quad (A2)
\]

\[
\rho_j^m C_{p,m}^{m} \frac{dT_{i,j}}{dt} = h_{j} (T_{i,j}^4 - T_{i,j}) \quad i = (N_t + 1), 2, \ldots, N_e \quad (A3)
\]

where \( \rho_j^m \), \( C_{p,m}^{m} \), and \( Z^m \) are the density, heat capacity, and the thickness of the metal substrate, respectively; \( \varepsilon \) is the wavelength factor for radiation; \( \sigma \) is the Stefan-Boltzman constant; \( \eta \) 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 \( \beta \)).

(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 \( \Delta z \). 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:

\[
\Delta z \frac{\partial w_{i,j,k}}{\partial t} = D_{j,k} \frac{\partial w_{i,j,k}}{\partial z} \quad k = 1 \quad (A4)
\]

\[
D_{j,k} = \eta e^{-\left( \gamma / w_{i,j,k} \right) + \left( E_a / RT_{i,j} \right)} \quad k = 1, 2, \ldots, N_z \quad (A5)
\]

\[
\frac{\partial w_{i,j,k}}{\partial t} = \frac{\partial}{\partial z} \left( D_{j,k} \frac{\partial w_{i,j,k}}{\partial z} \right) \quad k = 2, 3, \ldots, (N_z - 1) \quad (A6)
\]

\[
\rho_{z,j}^s \frac{\partial w_{i,j,k}}{\partial t} = K' \left( \frac{P_s - z P_s}{P} \right) + \rho_{z,j}^s D_{j,k} \frac{\partial w_{i,j,k}}{\partial z} \quad k = N_z \quad (A7)
\]

where \( w_{i,j,k} \) is the solvent content in the \( k \)th slice of the film on the \( j \)th panel in the \( i \)th zone; \( N_z \) is the total number of slices; \( D_j \) is the diffusivity; \( \eta \) is a pre-exponential constant for diffusion; \( \gamma \) is a constant; \( E_a \) is the activation energy for diffusion; \( \rho_{z,j}^s \) and \( \rho_{z,j}^u \) are the density and thickness of unreducible components in the film of the \( j \)th panel, respectively; \( K' \) 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:

\[
\rho_j^i \left( z_j(t) - z_j(t + \Delta t) \right) = \rho_{j}^i \left( \int_0^{z_j(t)} w_j(t)dz \right) - \int_0^{z_j(t+\Delta t)} w_j(t + \Delta t)dz \quad (A8)
\]

where \( \rho_j^i \) and \( \rho_{j}^i \) are densities of solvent and unreducible components in the film of the \( j \)th 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:

\[
\frac{dc_{j,i}}{dt} = \frac{\zeta e^{-E_r / RT_{i,j} \left( 1 - c_{j,i} \right)}}{1} \quad (A9)
\]

where \( c_{j,i} \) is the polymerization conversion of the film on the \( j \)th panel in the \( i \)th zone, \( E_r \) is the reaction activation energy, and \( \zeta \) is the reaction frequency factor.

---