J Glob Optim (2009) 43:191–205
DOI 10.1007/s10898-007-9217-2

A kriging based method for the solution of mixed-integer nonlinear programs containing black-box functions

Eddie Davis · Marianthi Ierapetritou

Received: 4 April 2007 / Accepted: 13 July 2007 / Published online: 4 August 2007

Abstract In this paper a new methodology is developed for the solution of mixed-integer nonlinear programs under uncertainty whose problem formulation is complicated by both noisy variables and black-box functions representing a lack of model equations. A branch-and-bound framework is employed to handle the integer complexity whereby the solution to the relaxed nonlinear program subproblem at each node is obtained using both global and local information. Global information is obtained using kriging models which are used to identify promising neighborhoods for local search. Response surface methodology (RSM) is then employed whereby local models are sequentially optimized to refine the problem's lower and upper bounds. This work extends the capabilities of a previously developed kriging-response surface method enabling a wider class of problems to be addressed containing integer decisions and black box models. The proposed algorithm is applied to several small process synthesis examples and its effectiveness is evaluated in terms of the number of function calls required, number of times the global optimum is attained, and computational time.

Keywords Black-box models · Optimization · Mathematical modeling · Kriging · Response surface

1 Introduction

Process synthesis problems are difficult to solve when the problem formulation contains black-box models for which noisy input–output sampling data are the only information available. A black-box representation describes individual process units whose design equations, such as rate expressions, are unavailable. The process behavior is described by input–output data which may be noisy due to environmental conditions, or as a result of the detail employed in a computational model such as a CFD simulation. Black-box models are also used in the

E. Davis · M. Ierapetritou (✉)
Department of Chemical and Biochemical Engineering, Rutgers – The State University of New Jersey, 98 Brett Road, Piscataway, NJ 08854, USA
e-mail: marianth@soemail.rutgers.edu

cases when legacy codes are the only information available for the description of process behavior. Synthesis and design policies for new product campaigns are often based on simulation data assuming ideal process operation. To better represent realistic scenarios, the problem addressed in this paper is the synthesis and design optimization problem containing black-box units.

This problem presents four main challenges. First, should a local or global model be used to approximate the black-box function? Local models provide a limited understanding, while global models are more expensive to build. Since there may be multiple black-box units, coarser models may need to be used in order to maintain reasonable computational expense. At the same time, how can accurate models be constructed in reasonable CPU time so that suboptimal solutions can be avoided? Second, how can the true process behavior be captured instead of the noise? The use of derivative-based optimization techniques can lead to the discovery of artificial optima due to iterates becoming trapped by the noise. This problem appears if surrogate models fit the noise instead of the actual process behavior. Third, given that sampling-based models are often built using symmetrically arranged collocation points, how can reliable models be constructed for problems whose feasible region is convex but not symmetrical? When solutions reside along feasible region boundaries, it may not be possible to obtain sampling information according to a symmetrical design in the current design space. Fourth, due to the combinatorial nature of the problem, how can the global optimal solution be obtained at reasonable computational cost?

The objective of this paper is to present a new algorithm in order to solve Mixed-Integer Nonlinear Programs (MINLP) containing black-box functions and noisy variables. This work is an extension of our previous work in which a solution methodology is presented for nonlinear problems (NLP) involving black-box models and noisy variables [10]. The new method addresses the integer complexity using a Branch-and-Bound framework, enabling the previously developed techniques to handle a larger class of problems. In the new algorithm, kriging is used to construct global models of all black-box units. At each node, a kriging predictor describing the behavior of the relaxed NLP objective is employed to identify regions of potential optima. Response surface techniques are then applied to local models in order to refine the set of candidate solutions. The global model building expense is offset by the identification of more reliable lower and upper bounds (LB/UB) at each node, improving the speed at which the global optimum to the MINLP is obtained.

In the complementary field of deterministic global optimization, a similar framework has been presented to address the solution of NLP containing nonanalytical differentiable constraints [21]. The global model applied in [21] assumes the form of a blending function from which valid over- and under-estimators can be generated in order to provide ε-guarantee of global optimality. The family of αBB algorithms (αBB, SMIN-αBB, GMIN-αBB) targets the solution of twice-differentiable nonconvex NLP and MINLP having restricted participation in the binary variables [1–4]. These methods rely on the generation of valid convex underestimators for the lower bounding problems in order to overcome the algorithmic difficulties presented by the nonconvex functions.

A variety of techniques have also been applied in the field of stochastic programming towards the solution of MINLP containing variables with uncertainty. One class of techniques determines the solution based on information obtained from complementary deterministic problems created after obtaining sampling realizations in the uncertain space. More specifically, Outer Approximation is used to solve MILP and NLP subproblems formulated using the sample average approximation method [27]. Confidence intervals on LB/UB are refined by solving a higher number of replicated subproblems created from an additional number of realizations in the uncertain space. This methodology has been extended by using

the simplicial approximation approach to describe the feasible region using a convex hull approximation [15]. These methods assume model equation availability in order to obtain linearization information, so they cannot be directly applied towards the solution of the problem involving black-box models.

In order to circumvent this problem, a second class of techniques can be used which rely on zero-order techniques to find the integer global solution. Derivative-free methods can be coupled with process simulators such as PRO-II and ASPEN, thereby enabling the uncertainty complications to be addressed outside of the simulation environment without losing the synthesis capabilities built around deterministic models. This approach is used by [8] whereby a stochastic annealing algorithm is wrapped around ASPEN in order to obtain the solution of a hydrodealkylation synthesis under uncertainty. However, since surrogate models are not constructed when using zero-order techniques such as Nelder-Mead [5], Divided Rectangles [19], multilevel coordinate search [17] and differential evolution [26], convergence to the optimum can be slow, motivating the use of gradient-based algorithms applied to data-driven models. The key problems associated with using first- and second-order techniques are asymptotic convergence and premature termination due to the artificial local optima resulting from noise. Modifications to gradient-based techniques overcome the latter problem using finite differences of large enough interval length to filter out noise [7,13]. However, the limited understanding of system behavior motivates the application of methods for which optimization is applied once surrogate models have first been built. Response surface optimization techniques belong to a class of local methods in which gradient techniques can be sequentially applied to inexpensive fitted local models known as response surfaces [22]. The response surface assumes the form of a low-order polynomial that is minimized to find the next candidate solution. A new response surface centered at the new iterate is built and minimized, and the procedure is repeated until a convergence criterion such as a prespecified decrease in the objective function has been attained. Sampling expense is the primary source of the optimization cost since the least-squares models can be built cheaply. In our recent work, an algorithm employing (1) adaptive experimental designs to retain feasibility, and (2) projection of the n-dimensional response surface onto constraints, has been successfully applied towards identifying NLP solutions in higher dimensions described by arbitrary convex feasible regions [10]. In addition to local optimization of response surfaces, global methods have also been studied for box-constrained problems which rely upon a larger set of basis functions used in response surface construction [16,18,20,23].

In order to overcome the limited understanding of system behavior obtained using response surfaces, global models can also be constructed. A kriging model is one example of a global representation that also includes a description of prediction uncertainty since normally distributed random function models are used [14]. The field of kriging originated in 1951 and was targeted at determining the spatial distribution of mineral deposits in mining applications based on physical sampling data. More recently, kriging has been increasingly applied to develop models based on the complementary field of computer experiments when simulation is used to describe complex processes [24]. Kriging is an interpolation technique whereby a prediction and its variance $(\tilde{z}_2, \tilde{\sigma}^2(\tilde{z}_2))$ at test point x_k are obtained according to a weighted sum of the nearby sampling data. Global prediction and variance mappings are generated after obtaining information from a set of points defined by feasible region discretization. Additional sampling is conducted in high-variance regions leading to improved models with reduced overall prediction uncertainty. The main limitation associated with kriging is that the model building costs are higher compared to response surface methods since a more detailed description of process behavior is obtained. Once a stopping criterion has been achieved, such as convergence in the average prediction value, response surface methodolgy (RSM)

can be applied to promising local regions in order to identify the global optimum. If the problem dimensionality increases due to the presence of multiple black-box units, kriging model building costs can significantly increase, an issue addressed in our proposed methodology. The remainder of the paper is organized as follows. In Sect. 2 the details and algorithmic features of the proposed method are presented. The proposed method is applied to several numerical examples in Sect. 3, and concluding remarks are given in Sect. 4.

2 Solution approach

The problem addressed in this work has the following form:

$$\begin{aligned} min \quad & F(x, y, z_1, z_2) \\ s.t. \quad & g(x, y, z_1, z_2) \leq 0 \\ & h(x, y, z_1) = 0 \\ & z_2(x) = \Gamma(x) + \varepsilon(x) \\ & \varepsilon(x) \in N(x|\mu, \sigma^2) \\ & x \in \Re^n, y \in \{0, 1\}^q \end{aligned} \tag{1}$$

In this formulation, x and y represent continuous and binary variables, respectively. The deterministic variables z_1 describe outputs whose modeling equations $h(x, y, z_1)$ are known. Stochastic output variables z_2 exist when the input–output functionality $\Gamma(x)$ is black-box simulated by a deterministic output perturbed with noise. Synthesis equations are given by $g(x, y, z_1, z_2)$ which include design constraints, operating specifications, and logical relations. The noise is described by a normally distributed error whose properties can change depending upon the spatial location of x.

In Sub-Section 2.1, the basic idea of the Branch-and-Bound (B&B) Kriging-RSM is presented as a technique for solving the class of problems described by (1). Sub-Section 2.2 describes how kriging is used to build global models and Sub-Section 2.3 explains how the best kriging solutions are further refined by optimizing response surfaces. Sub-Section 2.4 provides additional algorithmic details of the new B&B Kriging-RSM method.

2.1 Method outline

The central idea of the proposed algorithm for the solution of problem (1) is to use a B&B framework whereby at each node, kriging predictors built for each black-box process unit serve as basis functions in the construction of kriging predictors for the relaxed NLP objective. Using the global model of the objective, promising regions for local search are identified that serve as starting neighborhoods for refinement of the candidate solution set using sequential response surfaces. In order to reduce sampling and model building costs, RSM is applied to coarse kriging predictors during the initial stages of optimization. The kriging models are refined by incorporating RSM sampling data enabling tighter LB/UB solutions to be determined over the course of optimization.

During the early stages of the optimization computational cost is reduced by (a) use of coarse global models at early nodes and (b) use of weaker stopping tolerances for the kriging and RSM stages. More specifically, at the first node of the B&B tree, the global model is built using N_{Test} points. For each subsequent level of the B&B tree, global model accuracy is improved by using 10–25% additional test points relative to the number employed at the

previous level. For the examples presented in Sect. 3, N_{Test} is set at 1,000. A second method of reducing early computation expense relies upon a weak initial stopping criterion for global model improvement which is successively increased. This criterion is based on whether sampling data match predicted minima within a given tolerance, or whether convergence in the average value of the set of kriging predictions is observed. At the first node, the initial tolerance Tol_{Krig} might be satisfied if the average prediction value falls within 90% of the value at the previous iteration. The value of Tol_{Krig} could be increased to 95% for the second level, and to 99% for all subsequent levels. By applying these rules, a 10–15% decrease in total number of function calls is observed for the presented examples.

Kriging models are built for both the black-box units and the relaxed objective at each node. Since the relaxed NLP objective may differ from node to node depending upon binary variable assignments, a new kriging predictor may need to be constructed for each NLP subproblem. The kriging model of each black-box process describes unit-specific system behavior, whereas the kriging model of the relaxed NLP objective is created for optimization purposes in order to identify the best regions for local search. Once the global models for all black-box units have been created, this information is incorporated into the construction of any arbitrary objective while avoiding unnecessary sampling duplication. After refining the kriging solution using RSM, the optimum is classified as a lower or upper bound based on integer feasibility in y. If a stopping criterion based on the difference in the LB/UB is not met, B&B fathoming criteria are then applied to determine whether new subproblems should be formulated. Additional details are provided in [12].

2.2 Kriging methodology

The kriging methodology models the black-box system by considering each noisy output to be the realization of a random function. Based on this idea, a global covariance model is also employed to describe system behavior. The covariance model is built using limited sampling data and is used to determine weights for function values $f(x_i)$ corresponding to a set of sampling points x_i near test point x_k. A prediction at test point x_k can then be generated as the weighted sum of $f(x_i)$. An alternative interpretation of the kriging predictor is that it corresponds to an approximation based on function value differences evaluated for x_k and nearby sampling points x_i. Formally, the kriging predictor of the stochastic output $\tilde{z}_2(x_k)$ is linearly weighted sum of N function values at nearby sampling points as given below:

$$\tilde{z}_2(x_k) = \sum_{i=1}^{N_{Cluster}} f(x_i)\lambda(x_i) \tag{2}$$

where $\tilde{z}_2(x_k)$ represents the prediction value at test point x_k. The weights are determined in a manner similar to that of inverse distance methods, i.e. they are a decreasing distance function of x_k. In addition, the methodology avoids placing disproportionate weighting to clustered sampling data. From the set of N sampling data, squared function value differences F_{ij} can be calculated and plotted based on sampling-point pair distances.

$$F_{ij} = \left[f(x_i) - f(x_j)\right]^2 \quad i, j = 1 \ldots N, \quad i \neq j \tag{3}$$

Using the scatterpoint data, a model known as a semivariance function is fitted which describes how rapidly the function is expected to change in value based on increasing distance from x_i or x_k. The semivariance function is fitted from a plot of F_{ij} as a function of $x_i - x_j$ distance, also known as lag distance h. Due to the plot complexity as shown in Fig. 1a,

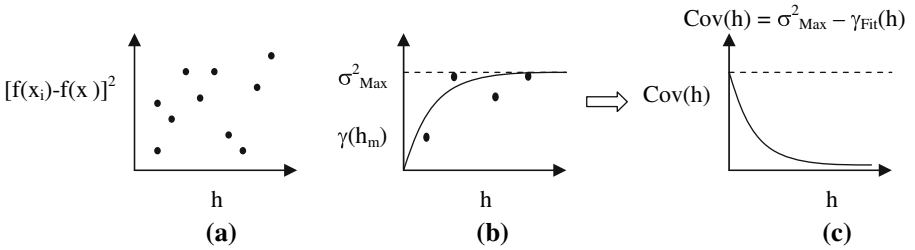

Fig. 1 Data smoothing applied to squared function differences (**a**) in order to obtain a semivariogram fit (**b**) and covariance function fit (**c**)

a resulting fit to one of the established semivariance models in the literature is not usually immediately apparent.

To alleviate this problem, data smoothing can be applied to achieve a better fit from a reduced set of scatterpoints known as semivariances $\gamma(h_m)$ as shown in Fig. 1b. Semivariances are averaged squared function differences constructed at distances h_m for P intervals:

$$\gamma(h_m) = \frac{1}{2N(h_m)} \sum_{r=1}^{N(h_m)} \left[f(x_i^r) - f(x_j^r) \right]^2 \quad m = 1 \ldots P, i \neq j \tag{4}$$

where $N(h_m)$ is the number of sampling pairs (x_i^r, x_j^r) whose separation distance lies inside the h_m interval. After obtaining the semivariance function parameters, the model is then reflected between the x-axis and its asymptotic maximum (known as the *sill*) to generate the corresponding covariance function displayed in Fig. 1c. Kriging weights are then determined by using covariance information based on nearby sampling point-test point distances as shown in the Lagrangian representation presented in (5).

$$\begin{bmatrix} \lambda(x_k) \\ \lambda'(x_k) \end{bmatrix} = \begin{bmatrix} Cov(d(x_i, x_j)) & 1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} Cov(d(x_i, x_k)) \\ 1 \end{bmatrix} \quad i, j = 1 \ldots N_{Cluster}, i \neq j \tag{5}$$

where $\Lambda = [\lambda(x_k); \lambda'(x_k)]$ represents the weight vector and Lagrange multiplier vector, respectively. Once the weights are obtained, the kriging prediction $\tilde{z}_2(x_k)$ and its expected variance $\tilde{\sigma}^2(x_k)$ are then obtained according to Eqs. (2) and (6), respectively:

$$\tilde{\sigma}^2(x_k) = \sigma_{max}^2 - \sum_{i=1}^{N_{Cluster}} \lambda(x_i) Cov\left[d(x_i, x_k)\right] - \lambda'(x_k) \tag{6}$$

After obtaining kriging and variance predictions for all test points at the current iteration m, the average prediction value $\mu_{m,Pred}$ is obtained and compared against the expected value found at the previous iteration $\mu_{m-1,Pred}$. If the value of $\mu_{m,Pred}$ is lower than some percentage Tol_{Krig} of $\mu_{m-1,Pred}$, the kriging predictor is not yet considered to be a reliable global model. Additional sampling is then performed in regions of high uncertainty characterized by either high prediction variances, or instead at points whose kriging prediction value has changed significantly between iterations [10]. It should be noted that the set of new collocation points should be ideally spaced far apart from one another thereby maximizing the global sampling information value obtained. Sampling at points taken from a single high-variance region emphasizes local model improvement and results in a limited understanding of additional global behavior. Local optimization using response surfaces is performed only after confidence in the global model has been established. However, the number of sampling

points required to build local models in high-dimensional systems can become high even when the problem dimensionality in the BB inputs is > 5. To alleviate this problem, further sampling is also conducted at points generated from refined grids of promising regions while still improving the model at the kriging level.

The kriging algorithm proceeds as follows for obtaining a prediction at x_k. First, the feasible region is described based on $g(x, y, z_1, z_2)$ and $h(x, y, z_1)$, where $g(x, y, z_1, z_2)$ is assumed to be explicitly defined in x after having obtained sampling data z_2 from upstream units. A sampling set Ω is generated whose initial size ranges between 10 and 20 dispersed data locations for problem dimensions of up to fifteen variables. As long as the starting size of Ω is not too small, the number of iterations required to achieve convergence in $\mu_{m,Pred}$ will be relatively insensitive to this number. The initial number of sampling points in Ω is small, placing emphasis on further sampling as needed during the iterative stages of predictor refinement. The location x_k is specified and $N_{Cluster}$ nearest-neighbor sampling points are chosen from Ω based on the L^2-norm relative to x_k. $N_{Cluster}$ usually varies between 5 and 10 regardless of problem dimension, although the estimate is potentially skewed if sampling information is sparse. Semivariances are then generated using all sampling data within Ω. The best semivariogram model is obtained using least squares, and the complementary covariance function is generated. The symmetrical matrix $h_{Cluster}$ is built describing sampling-pair distances among the set of $N_{Cluster}$ points. Similarly, the vector h_0 is constructed from sampling-point distances relative to x_k. The covariance matrices C and D are built using information from $h_{Cluster}$ and h_0, respectively. The kriging weights Λ are then obtained from solving the linear system of equations as presented in (5) and the prediction $\tilde{z}_2(x_k)$ and its variance $\tilde{\sigma}^2(x_k)$ are determined. A flowchart of the kriging algorithm ispresented in Fig. 2. In the next sub-section the response surface method is discussed whereby the best kriging solutions are further refined in order to improve the NLP solution.

2.3 Refinement of local optima using RSM

The main advantage of using response surfaces is that they provide inexpensive, yet accurate, approximations of local function behavior using low-dimensional polynomial approximations fitted to noisy data. These models can then be optimized using standard gradient techniques. Least-squares fitting enables models to be built which more closely describe the true functionality instead of the noise, enabling more reliable search directions to be identified. When building the response surface, the set of input points typically conforms to a stencil arrangement known as an experimental design that is centered about an iterate [6,22]. The central composite design shown in Fig. 3 requires $(1 + 2n + 2^n)$ sampling points for a problem of dimension n and is the template used at the RSM refinement stage in this paper. This design is associated with a lower sampling expense relative to the factorial design since data are not obtained at every factor-level combination. When used in combination with kriging, the starting iterates are initialized at the best solutions for the set of promising local regions.

At the start of the algorithm, the iteration index r is initialized at a value of unity. A response surface is built around a starting iterate x_r using a collocation set $x_{coll,r}$ determined by the CCD and local model radius b_r. The starting iterate x_r with objective value $f_{opt,r}$ corresponds to the optimal solution f_{act} of the kriging model for the objective. Sampling occurs at the response surface minimun $x_{opt,r+1}$ and the optimum is updated if $f_{opt,r+1}$ is now the current best objective value. If the difference between the current and previous optimum $|f_{opt,r+1} - f_{opt,r}|$ falls below Tol_{RSM}, the algorithm terminates with $[x_{opt,r+1}, f_{opt,r+1}]$ established as the RSM solution. Otherwise, another response surface is constructed at $x_{opt,r+1}$ with corresponding bounds b_{r+1} and the value of r is increased by unity. The value of b_{r+1}

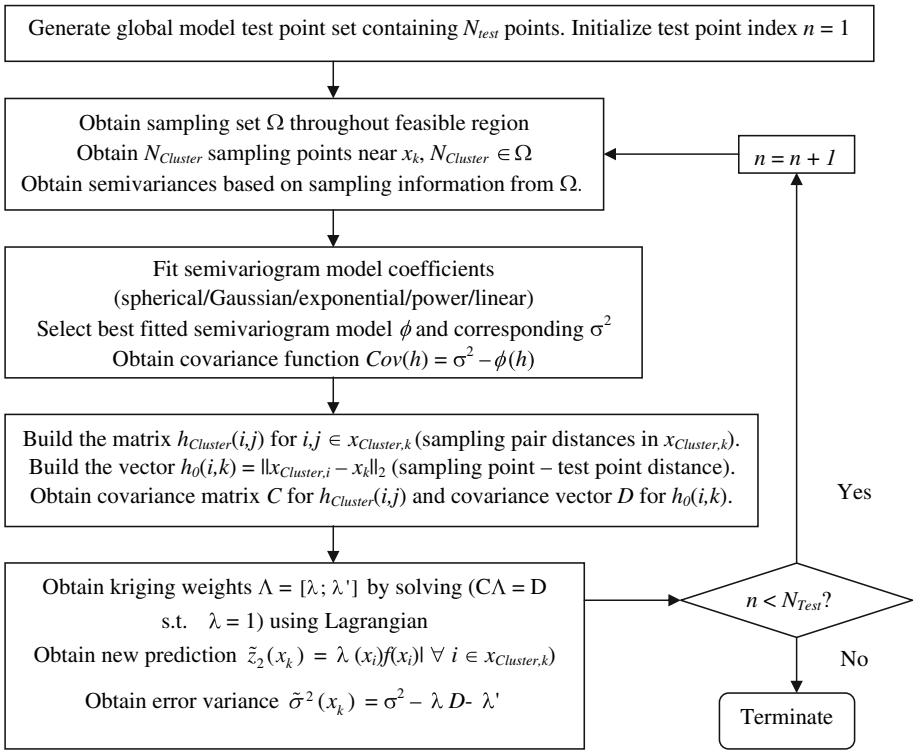

Fig. 2 Flowchart of Kriging algorithm for obtaining a prediction at test point x_{test}

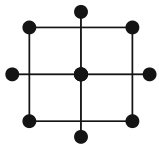

Fig. 3 Central composite design for response surface generation in 2-D

will be different from b_r if the Euclidean distance between the current and previous solution vectors is found to be lower than the value of b_r. During the later stages of the algorithm, $x_{opt,r+1}$ will be near $x_{opt,r}$ signifying that the basin of the RSM optimum has been reached. At this point, a more accurate description of the system behavior near the optimum can be attained using smaller response surfaces. Whenever iterates are close to the boundaries, lower-dimension response surfaces are created by projecting the model onto the constraint so as to prevent model generation based on an asymmetrical arrangement of the feasible sampling data. More details can be found in our earlier work [10]. A flowchart of the algorithm is presented in Fig. 4. In the next section, B&B, Kriging, and RSM are assimilated into a hybrid method targeted at solving the MINLP described by (1).

2.4 Branch-and-Bound Kriging-RSM algorithm

By combining the kriging-RSM algorithm used for obtaining NLP solutions with B&B, the integer global solution of MINLP can be efficiently found since the B&B fathoming criteria limits the number of NLP subproblems that have to be solved. The source of the computational

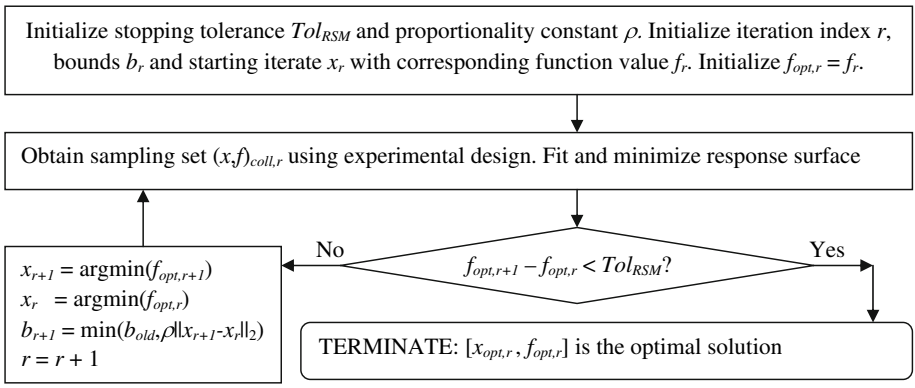

Fig. 4 Flowchart of RSM algorithm

expense for the optimization lies in generating reliable kriging models for both the black-box models and node-specific relaxed NLP objective functions. Although kriging predictors will be constructed for the black-box units over their corresponding operability ranges, additional kriging models for the black-box functions may need to be developed within subregions of the box-constrained feasible region.

Due to the noise and the presence of black-box units, global optimality cannot be guaranteed. The identification of suboptimal solutions at each node can delay search and even cause integer feasible solutions to be missed. As a result it is unlikely that local optimization using response surfaces can lead to the discovery of the global optimal solution if the correct neighborhood has not first been identified using the kriging predictor. One way of addressing this problem is to apply ideas similar to the ones used by [11,20]. Let the number of black-box units be R and let S replicate sets of the R kriging predictors be created based on different nominal sampling sets. The converged kriging mappings are not necessarily the same since they are built using different initial collocation points. For each sth set of R global models, $s = 1 \ldots S$, the corresponding mapping of the node-specific objective function is built and the optimal objective function $F_{pred,krig,s}$ is obtained. Both the mean and variance of the objective function $\{F_{pred,krig,s} | s = 1 \ldots S\}$ are then obtained and used as a point estimate and confidence interval for the global solution. If integrality in the binary variables is satisfied, the point estimate is an upper estimate of the upper bound, otherwise, it is an upper estimate of the lower bound. Next, optimization is performed at the RSM level. The best kriging solution of the S replicates is then refined using RSM T times where $T > S$. Let the set of refined RSM solutions be represented by $F_{RSM,t} | t = 1 \ldots T$. Both the mean and variance of $[F_{RSM,t} | t = 1 \ldots T]$ can then be obtained and used as a lower estimate of the solution regardless of whether it is a lower or an upper bound. The application of replicate kriging sets increases confidence in the global optimum; however, sampling costs can become prohibitive.

The Branch and Bound Kriging-RSM algorithm proceeds as follows. First, stopping tolerances are established. The Tol_{Krig} and $Tol_{Krig,Obj}$ parameters are used to terminate kriging predictor improvement for a process unit output and the relaxed NLP subproblem objective at an arbitrary node, respectively. The Tol_{BB} parameter is used to terminate further search in the Branch and Bound stage based on the difference between the lower and upper bound. This parameter is applied in order to avoid solving additional NLP subproblems whenever the improvement in the objective is expected to be low. Sampling data are then obtained in

order to create kriging predictors for the black-box units. The relaxed NLP is formulated at the first node and the kriging model of the corresponding objective is generated. The best kriging solutions are identified and sampling is performed at locations of predicted optima to confirm global model reliability for the black-box units. If the difference between the sampled and predicted objective values exceeds $Tol_{Krig,Obj}$, global improvement is considered necessary. It should be noted that a subset of the inputs x in each sampling vector may actually be noisy outputs z_2 from upstream black box units. For each black box unit, the set of sampled data $z_{2,act}$is compared against the set of corresponding kriging predictions $\tilde{z}_{2,Pred}$. Further sampling is then conducted for the subset of global models in which $|z_{2,act} - \tilde{z}_{2,Pred}| > Tol_{Krig}$ in order to improve the kriging predictors. Once the difference $|f_{act} - f_{pred}|$ falls below $Tol_{Krig,Obj}$, the best kriging solutions f_{pred} are refined by sequential optimization of response surfaces. An alternative stopping criterion Tol_{Krig} that can be applied involves determining the average value of the set of predictions for each iteration. Once convergence has been achieved in this average value for all black box units, the optimal kriging solutions for the objective are identified for refinement using RSM. Due to feasible region partitioning, the number of sampling points required for response surface building decreases as 0–1 assignments are made in the binary variables which can significantly reduce the sampling expense as evidenced by the results in the presented examples.

For a given NLP subproblem, if the best kriging solution lies within the basin of an optimum, it will be identified using RSM. In order to determine whether additional optima exist, RSM is applied to the next best kriging solution. The corresponding RSM optimal solution is compared to the one already found. If the solution is found to be inferior, further application of RSM towards additional kriging solutions terminates. Otherwise, RSM is then applied to the subsequent optimal kriging candidate in order to determine whether another minimum exists. It should be noted that if the set of points x_k over which the kriging model is built does not contain vectors located near optima, it is possible for some or all of the minima to be missed. In order to overcome this problem, the test set should be comprised of points approximating a uniform coverage of the feasible region. The optimum is determined to be a LB/UB depending upon 0–1 feasibility in the binary variables.

If the difference between the best LB/UB falls below Tol_{BB}, the algorithm terminates, otherwise additional subproblems are determined according to the Branch and Bound fathoming criteria. A new subproblem is then selected from the candidate set and the kriging model of the relaxed NLP objective for the new problem is constructed. If the set of new candidate subproblems is empty, the algorithm terminates with the best UB established as the solution to (1). A flowchart of the proposed algorithm is presented in Fig. 5 and the effectiveness of the method is demonstrated with the solution of two process synthesis examples in the next section.

3 Examples

In this section, the proposed Branch and Bound Kriging-RSM algorithm is applied to two process synthesis problems. The first example is taken from [11] and the second example is a modified propylene/propane separation synthesis study taken from [25]. For each example, a table of computational results is provided that illustrates the performance of the proposed Branch and Bound Kriging-RSM algorithm. For each algorithm, 100 trials are performed from a set of randomly selected feasible starting iterates. Using information taken from the subset of starting iterates successfully finding the global optimum, the average number of nodes visited, iterations required, function calls needed, and CPU time required are also

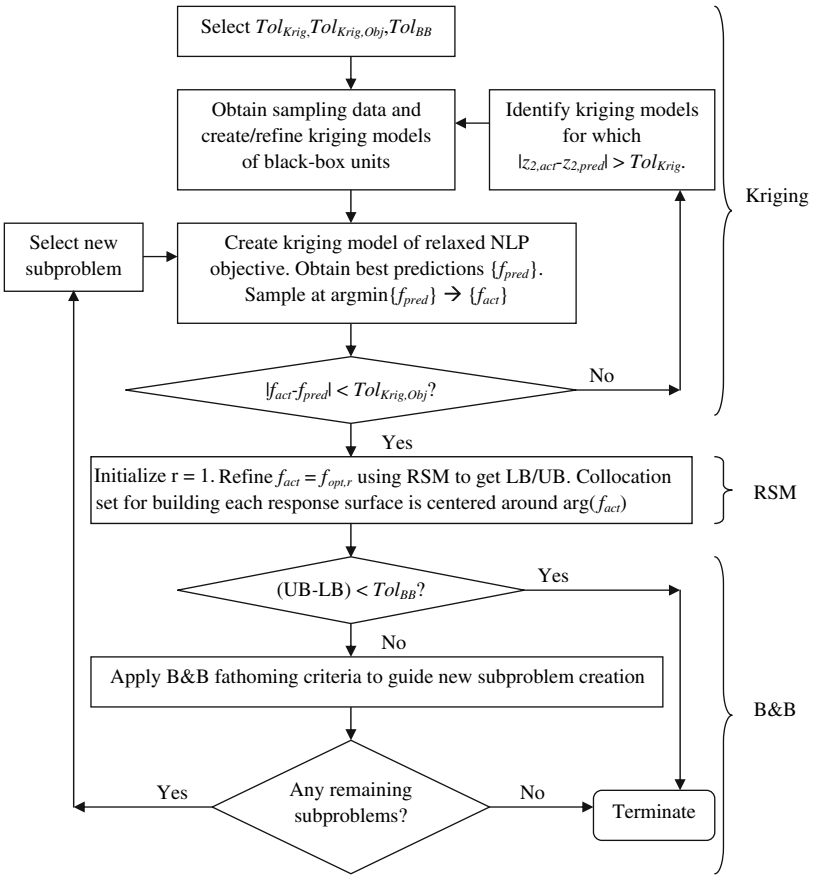

Fig. 5 B&B Kriging-RSM algorithm

reported. All computational results are obtained using an HP dv8000 CTO Notebook PC containing a 1.8 GHz AMD Turion 64 processor.

3.1 Example 1

This example involves six variables, four linear constraints, and two nonlinear constraints. The black-box variable z_2 is a function of three continuous variables and is noisy according to a normally distributed error with standard deviation 0.01. The problem is formulated as shown in (7):

$$
\begin{aligned}
min\ \ & z_2 + 5y_1 + 6y_2 + 8y_3 + 10 \\
s.t.\ \ & z_2 = -7x_6 - 18\ln(x_2+1) - 19.2\ln(x_1 - x_2 + 1) + N(0, 0.01) \\
& 0.8\ln(x_2+1) + 0.96\ln(x_1 - x_2 + 1) - 0.8x_6 \le 0 \\
& x_2 - x_1 \le 0 \\
& x_2 - Uy_1 \le 0
\end{aligned}
$$

Table 1 B&B Kriging-RSM MINLP algorithm performance to find the global optimum

	% Starting iterates finding global optimum	# Nodes	# Function calls			CPU time (s)		
			Kriging	RSM	Total	Kriging	RSM	Total
Example 1	90	6	113	161	274	36.55	0.59	37.14
Example 2	86	3	74	19	93	382	94	476

Table 2 Solution information for Example 1 using the Branch and Bound Kriging-RSM algorithm

Node	Fixed 0–1 vbls.	Optimal information		Kriging			RSM		
	(y_1,y_2,y_3)	$(x_1,x_2,x_6,y_1,y_2,y_3)$	F	Iter.	Func. calls	CPU time (s)	Iter.	Func. calls	CPU time (s)
1	(–,–,–)	(1.35,0.84,1,0.45,0.26,0)	1.427	5	43	44.27	5	75	1.02
2	(–,0,0)	(1.05,1.05,0.71,0.53,0,0)	5.205	4	34	3.06	6	47	0.19
3	(–,1,0)	(1.301,0,1,0,1,0)	6	3	25	1.58	2	13	0.14
4	(0,0,0)	(0,0,0,0,0,0)	10.02	1	1	0.03	0	0	0
5	(1,0,0)	(1.7,1.7,0.98,1,0,0)	7.258	4	34	2.14	3	21	0.08

$$
\begin{aligned}
& x_1 - x_2 - U y_2 \le 0 \\
& \ln(x_2 + 1) + 1.2 \ln(x_1 - x_2 + 1) - x_6 - U y_3 \le 0 \\
& y_1 + y_2 \le 0 \\
& y \in \{0, 1\}^3 a \le x \le b, x = (x_j : j = 1, 2, 6) \in \Re^3 \\
& a^T = (0, 0, 0), b^T = (2, 2, 1), U = 2
\end{aligned} \tag{7}
$$

The solution of the deterministic problem is $(x_1, x_2, x_6, y_1, y_2, y_3) = (1.301, 0, 1, 0, 1, 0)$. Results for this example are presented in Table 1 and Branch and Bound information for one test run is presented in Table 2.

It is observed that the algorithm is successful at finding the integer global optimum 90% of the time with the remainder converging at near-optimal solutions. At the first node, it is observed that a higher number of function calls are required at the refinement stage compared to the global model building phase. For computational simplicity, the test point set was generated from 1,000 randomly selected feasible points instead of according to a discretized grid. However, the location of the kriging optimum using the former method was inferior to the kriging solution that would have been obtained if discretization had been applied. As a result, multiple response surfaces are required in order to attain the best solution. The total number of function calls required for the RSM phase at the root node is 75 since fifteen function calls are required to build a response surface at each iteration. At subsequent nodes, the optimal vector is quickly achieved in a subset of the continuous variables, meaning that fewer function calls are needed to build lower-D response surfaces during the later stages of optimization.

3.2 Example 2 : Propylene/propane synthesis

The original case study involves the separation of a 60/40 mol% propylene/propane feed into a 99 mol% propylene distillate/95 mol% propane bottoms product, respectively using consecutive 100-trayed columns. In formulating the example as a synthesis problem, the separation is accomplished using two existing towers each having 50 and 75 trays, respectively to produce a 94% distillate. The feed can enter either one of the columns and the distillate product from the first column acts as feed to the second. A representation of the column superstructure is presented in Fig. 6a.

The propylene-rich distillate product exiting the second column is to be sold while the bottoms products from both columns is combined for use elsewhere in the plant. The separation is very difficult due to the close volatility between propylene and propane. A high reflux may be required since 37.5% fewer trays are used than as in the original study, which will increase energy costs. The problem is to determine the column sequence, operating reflux ratios, and reboiler steam inputs maximizing profit and minimizing cost. The problem is formulated in (8) where the product recoveries are simulated at 95% of their deterministic value perturbed by a normally distributed error:

$$
\begin{aligned}
max\ P &= 0.177\left(D_2 x_{Propylene,2}\right) + 0.132\left(D_2 x_{Propane,2}\right) - 4.41\left(Q_{R1}y_1 + Q_{R2}y_2\right) \\
&\quad -0.2\left(Q_{C1}y_1 + Q_{C2}y_2\right) \\
s.t.\ D_2 &= 0.95\Gamma_1\left(RR_1, RR_2, Q_{R1}, Q_{R2}, Q_{C1}, Q_{C2}, F\right) + N(0,5) \\
x_{Propylene,2} &= 0.95\Gamma_2\left(RR_1, RR_2, Q_{R1}, Q_{R2}, Q_{C1}, Q_{C2}, F\right) + N(0,0.01) \\
x_{Propane,2} &= 0.95\Gamma_3\left(RR_1, RR_2, Q_{R1}, Q_{R2}, Q_{C1}, Q_{C2}, F\right) + N(0,0.01) \\
y_1 + y_2 &= 1 \\
5 \le RR_1, RR_2 &\le 20 \\
10 \le Q_{R1}, Q_{R2} &\le 50 \quad [10^6\,\text{kJ/h}] \\
F = 11,646\,\text{kg/h},\ x_{Propylene,F} &= 0.6,\ x_{Propane,F} = 0.4
\end{aligned}
\tag{8}
$$

where the indices $i = 1,\ 2$ are referred to columns 1 and 2, respectively; y_i is the binary variable expressing the existence of column i; D_i, Q_{Ri}, Q_{Ci}, R_{Ri} are the distillate flowrates, reboiler duties, condenser duties, and reflux ratios of column i; $x_{Propylene,2}$, $x_{Propane,2}$, are the propylene and propane mole fractions of the column 2 distillate; F, $x_{Propylene,F}$, $x_{Propane,F}$ are the column 1 feed rate, propylene and propane feed mole fractions; and Γ_1, Γ_2, Γ_3 represent the black-box models, which in this case are obtained using ChemCad simulation. The objective represents a profit function where the first and second terms describe the profit obtained by selling the distillate at a price of \$0.177/kg propylene and \$0.132/kg of propane, respectively. The third and fourth terms represent heating and cooling costs, respectively. The process simulator ChemCad is used to simulate the black-box functions and is called upon as a slave program from the master driver in Matlab where the modeling and optimization tasks are carried out. Results for this example are presented in Tables 1 and 3.

The computational overhead involved for this example is higher than the previous one even though the problem dimensionality is lower due to the simulation expense. It is seen that proposed algorithm is successful at finding the global optimum as evidenced by 86% convergence. The optimal plant configuration consists of the 50-tray column preceding the 75-tray column as shown in Fig. 6b. Using this configuration, the optimal design variables are as follows. The reflux ratios for the first and second columns are 11.94 and 9.73, respectively, while the corresponding steam inputs to each reboiler are 28.72×10^6 kJ/h and 20.85×10^6 kJ/h.

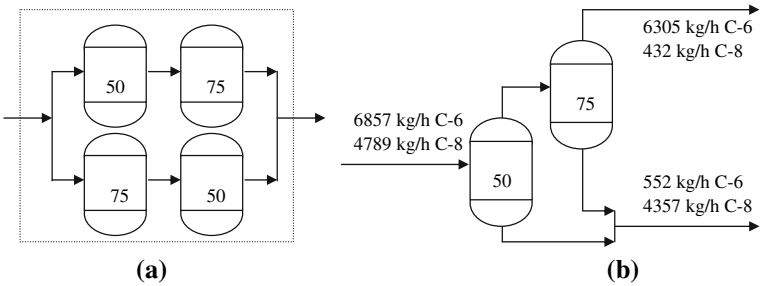

Fig. 6 Propylene/propane sequencing problem consisting of two possible configurations for two towers (**a**) and optimal configuration leading to 94% propylene distillate purity (**b**)

Table 3 Solution information for Example 2 using the Branch and Bound Kriging-RSM algorithm

Node	Fixed 0–1 vbls.	Optimal information		Kriging			RSM		
	(y_1,y_2)	(RR_1, RR_2, QR_1, QR_2)	P ($/h)	Iter.	Func. calls	CPU time (s)	Iter.	Func. calls	CPU time (s)
1	(–,–)	(9.82,10.59,13.2,17.9)	1021	5	52	176	2	12	41
2	(0,1)	(9.49,15.55,22.1,31.4)	875	4	21	89	2	12	37
3	(1,0)	(11.94,9.73, 28.72, 20.85)	931	5	26	113	3	17	43

An 85% propylene distillate purity is achieved in the first column which is improved to 94% in the second, leading to a $931/h profit. As expected, the tower that has more trays is the one which is assigned to perform the more rigorous task of achieving propylene purity. There is a high percentage of initial iterates converging to the global optimum due to the trend of improved solutions obtained at lower reflux ratios and steam consumption, with the remaining set of initial iterates converging at near-optimal values.

4 Conclusions and future work

In this work, a new Branch and Bound Kriging-RSM algorithm is presented for the solution of constrained MINLPs containing black-box functions and noisy variables. The Branch and Bound framework is used to efficiently search the 0–1 space for the integer global optimum, thereby extending the capabilities of existing work to handle process synthesis problems. Kriging is used to build global models of the black-box functions and the NLP subproblem objective at each node of the binary tree. The surrogate models are used to identify subregions where the relaxed NLP global optimum potentially resides. The set of best kriging solutions serves as starting iterates for further refinement via optimization of sequential response surfaces. The additional costs resulting from global model creation are offset by highly successful convergence to the integer global solution as shown by the represented examples. Current work focuses on generating stochastic cuts based on global model information enabling the Kriging-RSM algorithm developed for obtaining NLP solutions to be combined with other MINLP frameworks such as Generalized Benders Decomposition and Outer Approximation in order to reduce the modeling and optimization computational expenses.

Acknowledgements The authors gratefully acknowledge financial support from the National Science Foundation under the NSF CTS-0224745 and CTS 0625515 grants and also the USEPA-funded Environmental Bioinformatics and Computational Toxicology Center under the GAD R 832721-010 grant.

References

1. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global optimization of MINLP problems in process synthesis and design. Comp. Chem. Eng. **21**, S445–S450 (1997)
2. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and Computational Results. Comp. Chem. Eng. **22**, 1159–1179 (1998)
3. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: Global optimization of mixed-integer nonlinear problems. AIChE J. **46**, 1769–1797 (2000)
4. Adjiman, C.S., Androulakis, I.P., Maranas, C.D., Floudas, C.A.: A global optimization method, alpha BB, for process design. Comp. Chem. Eng. **20**, S419–S424 (1996)
5. Barton, R.R., Ivey, J.S.: Nelder-mead simplex modifications for simulation optimization. Manag. Sci. **42**, 954–973 (1996)
6. Box, G., Hunter, S., Hunter, W.G.: Statistics for Experimenters: Design, Innovation, and Discovery, 2nd edn. Wiley-Interscience, New York (2005)
7. Brekelmans, R., Driessen, L., Hamers, H., den Hertog, D.: Gradient estimation schemes for noisy functions. J. Opt. Theory Appl. **126**, 529–551 (2005)
8. Chaudhuri P., Diwekar U.: Synthesis approach to the determination of optimal waste blends under uncertainty. **45**, 1671–1687 (1999)
9. Choi, T.D., Kelley, C.T.: Superlinear convergence and implicit filtering. SIAM J. Opt. **10**, 1149–1162 (2000)
10. Davis, E., Ierapetritou, M.: AIChE J. DOI 10.1002/aic.11228 (2007, in press)
11. Duran, M., Grossmann, I.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Prog. **36**, 307–339 (1986)
12. Floudas, C.: Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications. Oxford University Press, New York (1995)
13. Gilmore, P., Kelley, C.T.: An implicit filtering algorithm for optimization of functions with many local minima. SIAM J. Opt. **5**, 269–285 (1995)
14. Goovaaerts, P.: Geostatistics for Natural Resources Evolution. Oxford University Press, New York (1997)
15. Goyal, V., Ierapetritou, M.: Stochastic MINLP optimization using simplicial approximation. Comp. Chem. Eng. In Press (2006)
16. Gutmann, H.M.: A radial basis function method for global optimization. J. Global Opt. **19**, 201–227 (2001)
17. Huyer, W., Neumaier, A.: Global optimization by multilevel coordinate search. J. Global Opt. **14**, 331–355 (1999)
18. Jones, D.: A taxonomy of global optimization methods based on response surfaces. J. Global Opt. **21**, 345–383 (2001)
19. Jones, D., Perttunen, C., Stuckman, B.: Lipschitzian optimization without the Lipschitz constant. J. Opt. Theory App. **79**, 157–181 (1993)
20. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. J. Global Opt. **13**, 455–492 (1998)
21. Meyer, C., Floudas, C., Neumaier, A.: Global optimization with nonfactorable constraints. Ind. Eng. Chem. Res. **41**, 6413–6424 (2002)
22. Myers, R., Montgomery, D.: Response Surface Methodology. Wiley, New York (2002)
23. Regis, R., Shoemaker, C.: Improved strategies for radial basis function methods for global optimization. J. Global Opt. **37**, 113–135 (2007)
24. Sacks, J., Welch, J.W., Mitchell, T.J., Wynn, H.P.: Design and analysis of computer experiments. Stat. Sci. **4**, 409–423 (1989)
25. Seader, J., Henley, E.: Separation Process Principles. Wiley, New York (2005)
26. Storn, R., Price, K.: Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. J. Global Opt. **11**, 341–359 (1997)
27. Wei, J., Realff, J.: Sample average approximation methods for stochastic MINLPs. Comp. Chem. Eng. **28**, 333–346 (2004)