Journal of Global Optimization (2005) 31: 153–171

Constrained Global Optimization of Expensive Black Box Functions Using Radial Basis Functions

ROMMEL G. REGIS[1] and CHRISTINE A. SHOEMAKER[2]
[1]*School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: rregis@orie.cornell.edu)*
[2]*School of Civil and Environmental Engineering, Cornell University, Ithaca, NY 14853, USA (e-mail: cas12@cornell.edu)*

(Received 24 March 2003; accepted in revised form 13 February 2004)

Abstract. We present a new strategy for the constrained global optimization of expensive black box functions using response surface models. A *response surface model* is simply a multivariate approximation of a continuous black box function which is used as a surrogate model for optimization in situations where function evaluations are computationally expensive. Prior global optimization methods that utilize response surface models were limited to box-constrained problems, but the new method can easily incorporate general nonlinear constraints. In the proposed method, which we refer to as the *Constrained Optimization using Response Surfaces* (CORS) Method, the next point for costly function evaluation is chosen to be the one that minimizes the current response surface model subject to the given constraints and to additional constraints that the point be of some distance from previously evaluated points. The distance requirement is allowed to cycle, starting from a high value (global search) and ending with a low value (local search). The purpose of the constraint is to drive the method towards unexplored regions of the domain and to prevent the premature convergence of the method to some point which may not even be a local minimizer of the black box function. The new method can be shown to converge to the global minimizer of any continuous function on a compact set regardless of the response surface model that is used. Finally, we considered two particular implementations of the CORS method which utilize a radial basis function model (CORS-RBF) and applied it on the box-constrained Dixon–Szegö test functions and on a simple nonlinearly constrained test function. The results indicate that the CORS-RBF algorithms are competitive with existing global optimization algorithms for costly functions on the box-constrained test problems. The results also show that the CORS-RBF algorithms are better than other algorithms for constrained global optimization on the nonlinearly constrained test problem.

Key words: Black box function, Costly function, Global optimization, Metamodel, Radial basis function, Response surface, Surrogate model

1. Introduction and Motivation

Global optimization of continuous black box functions that are costly to evaluate is a computationally challenging problem in engineering design. A single simulation performed to evaluate the costly function may require the solution of large systems of partial differential equations, and hence, may take a few minutes to many hours depending on the particular application.

Because of the enormous computational cost involved, an analyst is typically willing to perform only a small number of function evaluations when optimizing such costly functions. Our goal, then, is to develop global optimization algorithms that produce reasonably good solutions with a very limited number of function evaluations.

We now state our problem in more precise terms. Let $\mathcal{D}$ be a compact subset of R^d and let $f : \mathcal{D} \rightarrow R$ be a deterministic continuous function. The *global optimization problem* (GOP) is to find $x^* \in \mathcal{D}$ such that $f(x^*) \leqslant f(x)$ for all $x \in \mathcal{D}$. Note that under the given conditions, f attains its global minimum value on $\mathcal{D}$. In this investigation, we would like to focus on GOPs where f is a black box function that is costly to evaluate. For simplicity, we first assume that the domain $\mathcal{D}$ is a hypercube in R^d, i.e. the problem is box-constrained. Later, in Section 5, we will consider the situation where $\mathcal{D}$ defined by general nonlinear constraints. Furthermore, we also assume that the derivatives of f are unavailable and finite-difference approximations are too expensive to perform. Since f is costly to evaluate, we wish to find a point $\tilde{x} \in \mathcal{D}$ such that $f(\tilde{x})$ is close to $\min_{x \in \mathcal{D}} f(x)$ using only a relatively small number of function evaluations.

There are shortcomings with most of the existing optimization methods for costly black box functions. Gradient-based algorithms cannot be used in many cases simply because derivatives are not available and finite-difference approximations are too expensive to perform. Automatic differentiation techniques sometimes do not produce accurate derivatives because of truncation error in functions involving the solutions of PDEs or because of the presence of branching in the code for the black box function (Nocedal and Wright, 1999). In addition, automatic differentiation cannot be used in cases where the source code for the objective function is not available. A simple alternative is to use direct search methods like the simplex reflection algorithm by Nelder and Mead (1965), the DIRECT method by Jones et al. (1993), the Parallel Direct Search algorithm by Dennis and Torczon (1991), or the more general class of pattern search algorithms (Torczon, 1997). Direct search methods are derivative-free optimization methods. However, they generally require a large number of function evaluations since they do not take advantage of the inherent smoothness of some objective functions. Moreover, with the exception of the DIRECT global optimization method, the direct search methods mentioned above are only designed to find stationary points. Hence, global optimization will generally involve several restarts, requiring even more function evaluations. Finally, heuristic methods like evolutionary algorithms and simulated annealing also require a very large number of function evaluations to obtain adequately good solutions for GOPs.

A more practical type of optimization method for computationally expensive functions is one that is based on a *response surface model* (also known

as a *metamodel* or a *surrogate model*). By a *response surface model*, we simply mean a multivariate approximation of the underlying continuous black box function. The purpose of the response surface model is to serve as an inexpensive approximation to the costly black box function that can help identify promising points for costly function evaluation. The most popular of these methods is traditional response surface methodology (Myers and Montgomery, 1995; Box and Draper, 1987; Khuri and Cornell, 1987) which generally involves low-order polynomial regression. Another class of methods are the derivative-free optimization methods by Powell (1994, 2000, 2002) and by Conn et al. (1997) which utilize multivariate polynomial interpolation models within a trust-region framework. These methods are meant for unconstrained optimization problems but they can be easily tailored to deal with box constraints. They are also designed to find stationary points but they are generally more efficient than direct search methods. Other response surface methods for costly optimization are those that rely on kriging models (Jones et al., 1998; Simpson et al., 1998; Booker et al., 1999; Jones, 2001a) and radial basis functions (Ishikawa and Matsunami, 1997; Ishikawa et al., 1999; Björkman and Holmström, 2000; Gutmann, 2001b).

Response surface methods for optimization operate by maintaining an approximate model of the underlying function to be optimized. The approximate model may be local (i.e., restricted to a specific subregion of $\mathcal{D}$), as in the case of the derivative-free trust-region methods, or it may be global, as in the EGO method of Jones et al. (1998) or in the radial basis function method of Gutmann (2001b), or it may be a combination of both. In the case of methods that utilize a local response surface model, the region of exploration is periodically shifted and its size adjusted based on the information provided by newly evaluated points. In the case of methods that utilize a global response surface model, the global minimum in the approximate model does not usually correspond to a global minimum of the actual surface. Hence, the approximating global surface is periodically refitted upon the addition of newly evaluated points. However, a naive implementation of these methods, where the global minimizer of the current approximating surface is always selected for function evaluation may converge to some point which may not even be a local minimizer of the actual function (Gutmann, 2001b; Jones, 2001a).

We will focus our attention on optimization methods that utilize global response surface models. Jones (1996) proposed a general response surface method that requires a measure of "bumpiness" for the response surface model. Suppose $x_1, \ldots, x_n$ are previously evaluated points in $\mathcal{D}$. In each iteration of this method, we choose a target value f^* which represents a guess of the global minimum value of the black box function f and the next evaluation point y is chosen to be one that minimizes the bumpiness

of a response surface model that interpolates the data points $(x_1, f(x_1))$, $\ldots (x_n, f(x_n))$ and the additional data point (y, f^*) Gutmann (2001a, b) found a suitable measure of bumpiness for radial basis functions and developed a radial basis function method which he proved converges to the global minimum of any continuous function provided the target values are selected in a particular manner. Jones et al. (1998) also developed a kriging-based response surface method called *Efficient Global Optimization* (EGO) where the next evaluation point is chosen to be the one that maximizes the *expected improvement* in the objective function value. However, it remains a conjecture whether such a method converges to the global minimum of any continuous function (Jones et al., 1998).

The methods described in the previous paragraph are limited to box-constrained problems. In this paper, we introduce a new response surface method for global optimization which also works on nonlinearly constrained problems. We refer to the new method as the *Constrained Optimization using Response Surfaces* (CORS) method. In the new method, the next point for costly function evaluation is chosen to be a point that minimizes the current response surface model subject to the given constraints that define $\mathcal{D}$ and to additional constraints that it should be of some distance from previously evaluated points. The purpose of the constraint is to drive the algorithm towards unexplored regions and to prevent the algorithm from prematurely converging to some possibly undesirable point. To be able to perform both local and global search in this scheme we allow the distance requirement to cycle between high values (global search) and low values (local search). Moreover, we also prove that this new method converges to the global minimum of any continuous function. Finally, we implemented CORS using a radial basis function model (CORS-RBF) and applied it on the box-constrained Dixon–Szegö test functions (Dixon and Szegö, 1978) and on a nonlinearly constrained test function used by Gomez and Levy (1982). The results indicate that the CORS-RBF approach is competitive with existing global optimization methods for costly functions on the box-constrained problems. The results also show that the CORS-RBF approach is better than other algorithms for constrained global optimization on the nonlinearly constrained test problem.

2. A New Strategy for Global Optimization using Response Surfaces

2.1. GENERAL FRAMEWORK

We now provide a description of the CORS method for the constrained global optimization of costly functions using response surfaces. The new strategy is iterative and, in each iteration, the response surface model

is updated and exactly one point is selected for costly function evaluation. The evaluation point is selected to be one that minimizes the current response surface model subject to the given constraints (as specified by $\mathcal{D}$) and to some constraints on the distance from previously evaluated points. The guiding principle behind this method is that the selection of points for costly function evaluation has the dual goals of: (a) finding new points that have a low objective function value, and (b) improving the future response surface model by sampling regions of $\mathcal{D}$ for which little information exists. Hence, the selection of the next point for costly function evaluation is based on the minimization of current response surface model subject to constraints on how close the next point evaluated can be to previously evaluated points. Of course, there is a limit on how far a point can be from a previously evaluated point. If $x_1, \ldots, x_n$ are the previously evaluated points, then this limit is given by

$$\Delta = \max_{\tilde{x} \in \mathcal{D}} \min_{1 \leqslant j \leqslant n} \|\tilde{x} - x_j\|$$

Clearly, it makes no sense to require the distance of the next iterate from the previously evaluated points to be more than this distance, since this is impossible. Hence, we will require the next evaluation point to be at least of distance $\beta\Delta$ from all previously evaluated points, where $0 \leqslant \beta \leqslant 1$. A general framework for the CORS approach is given below.

Step 1 (*Select initial points*). Set $i := 1$ and select a finite initial set of points $S_1 = \{x_1, \ldots, x_k\} \subseteq \mathcal{D}$ for costly function evaluation.

Step 2 (*Do costly function evaluation*). Evaluate the function f at the points in S_1 and update the best function value encountered at every function evaluation.

Step 3 (*Iterate*). While termination condition is not satisfied do

Step 3.1 (*Fit or update response surface*). Fit or update a response surface model $\hat{f}_i$ using the data points $D_i = \{(x, f(x)): x \in S_i\}$.

Step 3.2 (*Select candidate point*). Select the candidate point x_{k+i} for function evaluation to be a point x that solves the following constrained optimization problem:

$$\begin{aligned} &\text{Minimize } \hat{f}_i(x) \\ &\text{Subject to} \\ &\quad \|x - x_j\| \geqslant \beta_i \Delta_i, \quad j = 1, \ldots, k + i - 1 \\ &\quad x \in \mathcal{D} \end{aligned} \tag{1}$$

where

$$\Delta_i = \max_{\tilde{x} \in \mathcal{D}} \min_{1 \leqslant j \leqslant k+i-1} \|\tilde{x} - x_j\| \tag{2}$$

and $0 \leqslant \beta_i \leqslant 1$ is a parameter to be set by the user (see discussion below for details).

Step 3.3 (*Do costly function evaluation*). Evaluate the function f at x_{k+i} and update the best function value encountered so far.

Step 3.4 (*Update information*). $S_{i+1} := S_i \cup \{x_{k+i}\}; D_{i+1} := D_i \cup \{(x_{k+i}, f(x_{k+i}))\}$.

Reset $i := i + 1$.

End.

In the above notation, k is the number of initial evaluation points, i denotes the iteration number, S_i is the set of previously evaluated points in iteration i, and $\hat{f}_i$ is the response surface model in iteration i. The parameters β_i are set by performing cycles of $N+1$ iterations where each cycle employs a range of values for β_i, starting with a high value close to 1 (global search) and ending with $\beta_i = 0$ (local search). More precisely, $\beta_i = \beta_{i+N+1}$ for all $i \geqslant 1$ and $1 \geqslant \beta_1 \geqslant \beta_2 \geqslant \cdots \geqslant \beta_{N+1} = 0$. We refer to N as the *cycle length* and we refer to the sequence $\langle \beta_1, \beta_2, \ldots, \beta_{N+1} = 0 \rangle$ as the *search pattern*. We also refer to the constrained minimization problem (1) in Step 3.2 as the *CORS auxiliary problem* (CORS-AP) or simply the *auxiliary problem*.

For simplicity in the discussion below, we use the term *maximin point* to refer to the point in $\mathcal{D}$ which is as far away as possible from any previously evaluated point. The expression Δ_i in (2) represents the distance of the maximin point from the closest previously evaluated point. In iteration i, we are requiring the candidate evaluation point to be of distance at least $\beta_i \Delta_i$ from the closest previously evaluated point. Solving the auxiliary problem with $\beta_i = 1$ is equivalent to finding a maximin point. On the other hand, solving the auxiliary problem with $\beta_i = 0$ is equivalent to simply minimizing $\hat{f}_i$ over $\mathcal{D}$. A search pattern of the form $\langle 0 \rangle$ represents *pure greedy search* whereas a search pattern of the form $\langle 1 \rangle$ represents *pure exploratory search*. A search pattern that includes a range of values between 0 and 1 such as $\langle 0.90, 0.75, 0.25, 0.05, 0.03, 0 \rangle$ balances global and local search and is generally more desirable than the extremes of pure greedy and pure exploratory search.

In any implementation of the CORS method, we have stipulated that the end of a search pattern be 0. The purpose of this requirement is to ensure that we are doing the most natural thing of minimizing the response sur-

face model subject to the constraints in $\mathcal{D}$ every $N+1$ iterations. Note that the user may specify a search pattern with more zeros. In fact, we can even do a search pattern with all zeros (pure greedy search). However, such a procedure is not recommended since it is prone to prematurely converging to a point that may not even be a local minimizer of the original function $f(x)$ (Gutmann, 2001b; Jones, 2001a, b). This happens when the minimizer of $\hat{f}_i$, in $\mathcal{D}$ (which is the next evaluation point) is a previously evaluated point. We address this issue below (in Section 2.2).

As we will see later in Section 3, the minimization of $\hat{f}_i$ in each iteration is not really necessary for the convergence of the method to a global minimum point. What is more important is that the candidate point for costly function evaluation satisfies the constraint in Step 3.2 above for some strictly positive β_i for infinitely many i. In addition, the requirement that each search pattern is a nonincreasing finite sequence ending with 0 is also not necessary for convergence. Rather these requirements are simply heuristics that are meant to speed up the process of finding a global minimum point for the original objective function. In fact, the only requirement for convergence is to have a search pattern with at least one nonzero entry.

2.2. IMPLEMENTATION ISSUES

Any algorithm twidehat follows the CORS framework requires two components: (a) a scheme for selecting an initial set of points for costly function evaluation, and (b) a procedure for globally approximating the unknown costly black box function in any iteration (i.e. a response surface model). The first component can be provided by various experimental design techniques ranging from simple grids to Latin hypercubes (McKay et al., 1979) and orthogonal arrays. The paper by Koehler and Owen (1996) describes various experimental design techniques. For the second component, we can use various multivariate approximation schemes including polynomial regression, kriging (Sacks et al., 1989; Cressie, 1993), radial basis functions (Powell, 1992, 1999), multivariate adaptive regression splines (Friedman, 1991), and neural networks. As will be seen below, the main convergence result for the CORS method does not depend on either the initial evaluation points or on the particular response surface model being used.

Another important issue is the computation of Δ_i as defined in (2). Gutmann (2001a) showed that, in the case where $\mathcal{D}$ is defined by box constraints, the computation of Δ_i may be converted into a concave minimization problem and may be solved via an outer approximation algorithm as described in Horst et al. (1995). In practice, we can approximately solve (2) by maintaining a set of points that "cover" $\mathcal{D}$ (i.e. that are spread all throughout $\mathcal{D}$) and selecting the farthest from any previously evaluated points.

Note that the auxiliary optimization problem described above (in Step 3.2) is generally nonconvex. Fortunately, its objective function and its constraint functions are cheap to evaluate. Moreover, the gradients of the objective function and constraint functions of the auxiliary problem are also easy to obtain and evaluate. Hence, we can take advantage of state-of-the-art software for gradient-based optimization to solve the auxiliary problem. Since the problem is nonconvex, there is no guarantee of finding a global minimizer for the auxiliary problem. Hence, we typically perform several runs of a nonlinear programming (NLP) solver with different starting points and the evaluation point is selected to be the local minimizer that has the lowest function value among the local minimizers that were obtained. Another option is to run a global optimization method such as Constrained DIRECT (Jones, 2001b) and refine its solution by starting a NLP solver from that point.

One problem that arises when using the CORS method is that it could happen that the best local minimizer for the auxiliary problem (i.e. the candidate evaluation point) in some iteration is a previously evaluated point. Note that this can only occur if $\beta_i = 0$. When it does occur, we simply reset $\beta_i = 0.01$ and solve the resulting auxiliary problem.

3. Convergence

Let $\mathcal{D} \subseteq R^d$ be compact and let $f : \mathcal{D} \to R$ be a continuous function. Also, let $\mathcal{A}$ be an optimization algorithm whose sequence of iterates is $\{x_k\}_{k \geqslant 1}$. We say that $\mathcal{A}$ *converges* to the global minimum of f if $\min_{1 \leqslant i \leqslant k} f(x_i) \downarrow \min_{x \in \mathcal{D}} f(x)$ as $k \uparrow \infty$. Our starting point is the following theorem:

THEOREM 1 (Torn and Zilinskas, 1989). *Let $\mathcal{D}$ be a compact set. Then an algorithm converges to the global minimum of every continuous function on $\mathcal{D}$ if and only if its sequence of iterates is everywhere dense in $\mathcal{D}$.*

To prove the convergence of the new method, Theorem 1 states that we only need to ensure that the sequence of trial points is dense in $\mathcal{D}$. Below is our result that naturally leads to a proof of the convergence of the CORS method.

THEOREM 2. *Let $\mathcal{D}$ be a compact set and let $\{x_k\}_{k \geqslant 1} \subseteq R^d$ be the sequence of iterates generated by an algorithm $\mathcal{A}$. Suppose there exists a strictly increasing sequence $\{n_t\}_{t \geqslant 1}$ of positive integers such that X_{n_t} satisfies the following condition for some $0 < \alpha \leqslant 1$:*

$$\min_{1 \leqslant k \leqslant n_t - 1} \|x_{n_t} - x_k\| \geqslant \alpha \max_{y \in \mathcal{D}} \min_{1 \leqslant k \leqslant n_t - 1} \|y - x_k\| \quad \forall t \geqslant 1, \tag{3}$$

then $\mathcal{A}$ converges to the global minimum of any continuous function *on $\mathcal{D}$.*

Proof. Define the sequence $\{s_t\}_{t\geqslant 1}$ by

$$s_t = \min_{1\leqslant k\leqslant n_t-1} \|x_{n_t} - x_k\|. \tag{4}$$

Suppose $\{x_k\}$ is not dense in $\mathcal{D}$. Then there exists $\bar{x}\in\mathcal{D}$ and $\delta > 0$ such that the open ball centered at $\bar{x}$ with radius δ does not contain any element of the sequence $\{x_k\}$. This implies that

$$\|\bar{x} - x_k\| \geqslant \delta \quad \forall k \geqslant 1, \tag{5}$$

and so, from (3)–(5)

$$s_t \geqslant \alpha \max_{y\in\mathcal{D}} \min_{1\leqslant k\leqslant n_t-1} \|y - x_k\| \geqslant \alpha\delta > 0 \quad \forall t \geqslant 1$$

Let $\alpha\delta = \epsilon$. Since $s_t \geqslant \epsilon$ for all $t\geqslant 1$, we have $\|x_{n_i} - x_{n_j}\| \geqslant \epsilon$ for any $i > j$. Since $\mathcal{D}$ is compact, it follows that it is bounded. Let $\mathcal{B}$ be a hypercube that contains $\mathcal{D}$ whose side length is $r\epsilon/(2\sqrt{d})$ for some positive integer r. Partition $\mathcal{B}$ into r^d equal-sized hypercubes where each hypercube has side length $\epsilon/(2\sqrt{d})$. Now the condition on $\{x_{n_t}\}_{t\geqslant 1}$ implies that no two of these points can belong to a single sub-hypercube. But this is a contradiction since $\{x_{n_t}\}_{t\geqslant 1}$ is an infinite set of distinct points. Thus, $\{x_k\}_{k\geqslant 1}$ must be dense in $\mathcal{D}$. □

COROLLARY 3. *Any CORS method where the search pattern contains at least one nonzero entry converges to the global minimum of any continuous function for any choice of response surface model and for any choice of initial evaluation points.*

Proof. Let N be the cycle length. Suppose the jth entry of the search pattern is nonzero. Let $n_t = j + (t-1)(N+1)$ for all $t\geqslant 1$. By assumption, we have $\beta_{n_t} = \alpha > 0$ for all $t\geqslant 1$. Now the constraints in Step 3.2 of Section 2.1 show that the candidate evaluation point satisfies condition (3) above. □

The advantage of this result is that it is independent of the choice of the initial evaluation points and it is also independent of the particular response surface model that is being used. In fact, it is not even necessary that we solve the auxiliary optimization problem in Step 3.2 above in order to guarantee the convergence of the method. All we need is that the candidate point for function evaluation satisfy the constraints in Step 3.2. However, for practical purposes, it should be intuitively clear that the rate of convergence is somehow dependent on how well the response surface

Table 1. The Dixon–Szegö test functions (Dixon and Szego, 1978)

Test function	Dimension	Domain	No. of local min	No. of global min	Global min value
Branin	2	$[-5, 10] \times [0, 15]$	3	3	0.398
Goldstein–Price	2	$[-2, 2]^2$	4	1	3
Hartman3	3	$[0, 1]^3$	4	1	−3.86
Shekel5	4	$[0, 10]^4$	5	1	−10.1532
Shekel7	4	$[0,10]^4$	7	1	−10.4029
Shekel10	4	$[0, 10]^4$	10	1	−10.5364
Hartman6	6	$[0, 1]^6$	4	1	−3.32

model approximates the underlying costly function and also on how well we are solving the CORS-AP in Step 3.2.

4. Computational Experiments

4.1. DESCRIPTION

To test the performance of the CORS method, computational experiments were performed on some benchmark box-constrained test functions using a radial basis function model initialized using the corners of each hypercube domain. We refer to the resulting CORS method as CORS-RBF. The new algorithm was tested on the Dixon–Szegö test functions (Dixon and Szegö, 1978) for global optimization. These functions are not really costly to evaluate but their shapes are complex and multimodal, and hence, the relative performance of algorithms on these test functions is expected to mimic performance on costly functions. Table 1 shows the characteristics of the Dixon-Szegö test functions. The actual functional expressions can be found in Dixon and Szegö (1978).

To assess the significance of the proposed method, it is necessary to compare its performance against the existing derivative-free methods for the global optimization of costly functions. Our new method was compared with the RBF method developed by Gutmann as implemented by Gutmann (2001b) and as implemented by Björkman and Holmström (2000), the EGO method by Jones et al. (1998), and the DIRECT method by Jones et al. (1993). The performance of the different methods were compared on the Dixon–Szegö test functions.

4.2. RADIAL BASIS FUNCTION MODEL

We now discuss the response surface model that was used in our implementation of the CORS method. The interpolation model that will be described below was extensively studied by Powell (1992, 1999) and was used as the basis of the RBF method by Gutmann (2001b).

Assume that we are given n distinct points $x_1, \ldots, x_n \in R^d$ where the function values are known. In this method, we use an interpolant of the form

$$s(x) = \sum_{i=1}^{n} \lambda_i \phi(\|x - x_i\|) + p(x), \quad x \in R^d \tag{6}$$

where $\|\cdot\|$ is the Euclidean norm in R^d, $\lambda_i \in R$ for $i = 1, \ldots, n, p$ is in Π_m^d (the space of polynomials in d variables of degree less than or equal to m), and ϕ is one of the following forms:

$$\begin{aligned}
&\phi(r) = r && \text{(linear)},\\
&\phi(r) = r^3 && \text{(cubic)},\\
&\phi(r) = r^2 \log r && \text{(thin plate spline)},\\
&\phi(r) = \sqrt{r^2 + \gamma^2} && \text{(multiquadric)},\\
&\phi(r) = \mathrm{e}^{-\gamma r^2} && \text{(Gaussian)},
\end{aligned}$$

where γ is a positive constant.

Fix ϕ. Define the matrix $\Phi \in R^{n \times n}$ by:

$$(\Phi)_{ij} := \phi(\|x_i - x_j\|), \quad i, j = 1, \ldots, n.$$

Moreover, define

$$m_\phi = \begin{cases} -1 & \text{if } \phi \text{ is Gaussian} \\ 0 & \text{if } \phi \text{ is linear or multiquadric} \\ 1 & \text{if } \phi \text{ is cubic or the thin plate spline} \end{cases}$$

and let $m \geqslant m_\phi$. Let $\hat{m}$ be the dimension of the linear space Π_m^d, let $p_1, \ldots, p_{\hat{m}}$, be a basis of this linear space, and define the matrix P as follows:

$$P = \begin{pmatrix} p_1(x_1) & \ldots & p_{\hat{m}}(x_1) \\ \vdots & & \vdots \\ p_1(x_n) & \ldots & p_{\hat{m}}(x_n) \end{pmatrix}.$$

In this model, the RBF that interpolates the points $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$ is obtained by solving the system

$$\begin{pmatrix} \Phi & P \\ P^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0_{\hat{m}} \end{pmatrix},$$

where $F = (f(x_1)), \ldots, f(x_n))^{\mathrm{T}}$, $\lambda = (\lambda_1, \ldots, \lambda_n)^{\mathrm{T}} \in R^n$ and $c = (c_1, \ldots, c_{\hat{m}})^{\mathrm{T}} \in R^{\hat{m}}$. Powell (1992) showed that the matrix

$$A = \begin{pmatrix} \Phi & P \\ P^{\mathrm{T}} & 0 \end{pmatrix} \in R^{(n+\hat{m})\times(n+\hat{m})}$$

is nonsingular if and only if $x_1, \ldots, x_n$ satisfy the property:

$$q \in \Pi_m^d \quad \text{and} \quad q(x_i) = 0, \quad i = 1, \ldots, n, \quad \Rightarrow \quad q \equiv 0.$$

Hence, in this case, the resulting RBF interpolant $s(x)$ is unique.

4.3. EXPERIMENTAL SETUP

Two implementations of the CORS-RBF algorithm with different search patterns were applied to each of the Dixon–Szegö test functions. We used a particular radial basis function model of the form (6) where ϕ is a thin plate spline and $p(x)$ is a linear polynomial. The initial evaluation points were chosen to be the corners of the hypercube domain of each test function. The reason for these choices of response surface model and initial evaluation points is that these were the ones used by Gutmann (2001b) in his computational experiments with his RBF method. It is necessary to achieve fair comparison with Gutmann's method since it is among the most recent methods proposed for the global optimization of costly functions. For the implementations of CORS-RBF we used a cycle length of 4 with a search pattern of $\mathrm{SP1} = \langle 0.95, 0.25, 0.05, 0.03, 0\rangle$ and a slightly longer cycle of length 5 with a search pattern of $\mathrm{SP2} = \langle 0.9, 0.75, 0.25, 0.05, 0.03, 0\rangle$. The high values for the parameter β_i are responsible for the global aspect of the search while the low values are responsible for local search. Values of β_i that are close to 0 are essential for the success of the method since these allow the algorithm to explore points near the vicinity of some good previously evaluated points. Finally, we also adopted a strategy used by Gutmann (2001b) of replacing large function values by the median of all available function values. The purpose of this transformation is to prevent oscillations in the RBF interpolant that are due to the large differences in function values.

In solving the auxiliary problem in Step 3.2 of the algorithm, we need to compute the maximin distance. As noted earlier, we could convert this into a concave minimization problem and solve it using an outer approximation algorithm. In this investigation, we solved this problem approximately by maintaining a set of points in a relatively fine grid that "covers" the entire hypercube domain. We shall refer to these set of points as *cover points*.

Before the first iteration, after we have specified the initial evaluation points, we determine the distance between each cover point and each initial evaluation point. For each cover point, we compute its minimum distance from the initial evaluation points. We store these minimum distances in a vector of length equal to the number of cover points. Then in each iteration, we compute the distances between the newly evaluated point and the cover points and update the vector of minimum distances. Note that the vector of minimum distances allows us to obtain an approximate maximin point by simply selecting the cover point whose distance is as far away as possible from previously evaluated points. This estimate of the maximin point is further refined by performing a local greedy search starting at the current approximate maximin point.

Once we have an estimate of the maximin distance, we solve the auxiliary problem using the DIRECT global optimization method (Jones et al., 1993; Jones, 2001b). The solution obtained by DIRECT is refined by starting a NLP solver from that point. Moreover, we also run the NLP solver from multiple randomly generated starting points near the vicinity of the solution found by DIRECT. Note that most NLP solvers will work with infeasible starting points so it is not a problem if a starting point violates some of the distance constraints. The best solution obtained in any of these optimization runs is taken to be the solution to the auxiliary problem. Note that there is no guarantee that we are really finding an optimal solution to the auxiliary problem.

However, recall that the convergence result only requires that each iterate satisfy the constraint in the auxiliary problem. Hence, any method that only approximately solves the auxiliary problem will still converge to the global minimum point.

All numerical computations were performed in Matlab R13. The auxiliary problems were solved using the glcFast (for $\beta_i \neq 0$) and glbFast (for $\beta_i = 0$) routines of Tomlab (Holmström, 1999) which implement DIRECT (Jones et al., 1993) and Constrained DIRECT (Jones, 2001b), respectively. The NLP solver used is the fmincon subroutine of the Matlab Optimization Toolbox (2000). For efficient optimization using fmincon, we supplied the gradients of the RBF model and the distance constraints since these were easy to compute.

4.4. RESULTS

In Table 2, we recorded the number of function evaluations performed by each of the CORS-RBF methods to get a relative error of $<1\%$ for each test function. If f^* is the global minimum value and f_{best} is the best value obtained by an algorithm, then the *relative error* is given by $|f_{\text{best}} - f^*|/|f^*|$ provided that $f^* \neq 0$. For comparison purposes, we included the results of the radial basis function method by Gutmann (2001a, b) as implemented

Table 2. Comparison of global optimization algorithms on the Dixon–Szegö test functions

Test function	CORS-RBF (SP1)	CORS-RBF (SP2)	RBF-G	rbfSolve	DIRECT	EGO
Branin	34	40	44	26	63	28
Goldstein–Price	49	64	63	27	101	32
Hartman3	25	61	43	22	83	35
Shekel5	41	52	76	96	103	–
Shekel7	46	64	76	72	97	–
Shekel10	51	64	51	76	97	–
Hartman6	108	104	112	87	213	121

The values in the table indicate the number of function evaluations to get a relative error of $< 1\%$.

by Gutmann (2001b) (RBF-G) and as implemented by Björkman and Holmström (2000) (rbf Solve). Björkman and Holmström (2000) experimented with several variations of how to implement Gutmann's RBF method. We only report their most successful result where ϕ is cubic, the search space was transformed to the unit hypercube, and large function values were replaced by the median of all available function values. They did not have much success with the case where ϕ is a thin plate spline with some runs being terminated before getting a solution with relative error $<1\%$. We also included the results of the EGO method by Jones et al. (1998), and the DIRECT method by Jones et al. (1993) as presented in Gutmann's paper.

The results on Table 2 indicate that CORS-RBF (SP1) is consistently better than RBF-G on all Dixon–Szegö test functions except on Shekell0 where the two algorithms have the same performance. The results for CORS-RBF (SP2) are consistently worse than those for CORS-RBF (SP1) except on Hartman6 where CORS-RBF (SP2) is slightly better than CORS-RBF (SP1). However, CORS-RBF (SP2) is very much competitive with RBF-G. It is better than RBF-G on the Branin, Shekel5, Shekel7, and Hartman6 test functions and it is worse than RBF-G on the Hartman3 and Shekell0 test functions. On the Goldstein–Price test function, the performance of CORS-RBF (SP2) is only slightly worse than RBF-G.

The CORS-RBF algorithms are not as good as rbfSolve on the Branin, Goldstein–Price and Hartman6 test functions. However, it is much better than rbfSolve on the Shekel5, Shekel7 and Shekel10 test functions. Moreover, CORS-RBF (SP1) is competitive with rbfSolve on the Hartman3 test function. The CORS-RBF algorithms are also consistently much better than DIRECT on all of the Dixon–Szegö test functions. EGO is better than the CORS-RBF algorithms on the Branin and Goldstein–Price test functions. However, CORS-RBF (SP1) is much better than EGO on Hartman3. Moreover, for the higher dimensional Hartman6 test function, we see that the CORS-RBF methods are better than EGO. These results demonstrate the potential of the CORS-RBF method for computationally

expensive real-world optimization problems. The CORS-RBF method is also attractive for practitioners in the sense that it is based on simpler ideas, it is easier to code than either the EGO method and Gutmann's RBF method, and yet it achieves comparable results.

5. Extension to Nonlinearly Constrained Global Optimization

While the paper has focused on box-constrained global optimization (i.e. $\mathcal{D}$ is defined by box constraints), the CORS method is easily extended to handle nonlinear constraints. The method still works in the general case where $\mathcal{D}$ is defined by nonlinear constraints provided $\mathcal{D}$ remains compact and the auxiliary problem in Step 3.2 is tractable. Moreover, the proof of convergence only requires that $\mathcal{D}$ be compact, and so, it also holds in the general case.

In the implementation of CORS for a nonlinearly constrained GOP, one has to be careful in making sure that the initial evaluation points all satisfy the constraints. One has to construct a space-filling design whose points all lie in the domain $\mathcal{D}$. Moreover, the maximin point computed in Step 3.2 (Section 2.1) of the algorithm must be in $\mathcal{D}$ (i.e. it must satisfy the nonlinear constraints). In our implementation, we find an approximate maximin point by choosing a feasible cover point (i.e. a cover point satisfying the nonlinear constraints specified by $\mathcal{D}$ which is as far away as possible from previously evaluated points.

The procedure for solving the auxiliary problem will depend on what kind of constraints define $\mathcal{D}$. Note that in a real problem, there are constraints whose violation will result in an undefined objective function value (e.g., some input values will cause the simulation that computes the objective function value to crash). We shall refer to these constraints as hard *constraints*. If $\mathcal{D}$ is defined by hard constraints, then we have to use *feasible algorithms* (i.e. algorithms that do not step on infeasible territory) to solve CORS-AP. For simplicity, we assume in this investigation that there are no hard constraints in $\mathcal{D}$. Moreover, we assume that $\mathcal{D}$ is defined by constraint functions that are computationally cheap to evaluate. In this situation, we proceed in the same manner as before and solve the auxiliary problem using Constrained DIRECT followed by a refinement of the solution by starting a NLP solver at that point.

Finally, fitting the response surface model is not much harder in the nonlinearly constrained case since we simply fit the model as though the constraints were absent. While fitting a response surface model over infeasible territory (i.e. over regions outside of $\mathcal{D}$) may sound absurd, this should be alright as long as the response surface model still provides a good approximation over the feasible region.

Table 3. Comparison of global optimization algorithms on the Gomez # 6 problem

Method	No. of evaluations
Tunneling algorithm (Gomez and Levy, 1982)	1053
Constrained DIRECT (Jones, 2001b)	89
CORS-RBF (SP1)	53
CORS-RBF (SP2)	30

The values in the table indicate the number of function evaluations to get a relative error of $<1\%$.

To illustrate how the method works on a simple constrained problem, we have applied CORS-RBF to the "Gomez #3" problem that was used by Gomez and Levy (1982) to test a tunneling algorithm and subsequently used by Jones (2001b) to test his Constrained DIRECT algorithm. The problem formulation is as follows:

$$\text{Minimize} \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

subject to

$$-\sin(4\pi x_1) + 2\sin^2(2\pi x_2) \leqslant 0 \quad -1 \leqslant x_1, x_2 \leqslant 1$$

The optimal solution to this problem is (0.109, −0.623) with an objective value of −0.9711. Note twidehat for this problem, the corners of the box defined by the upper and lower bounds on x_1 and x_2 are all feasible so we implemented CORS-RBF using these points as the initial evaluation points.

The results for different algorithms are summarized in Table 3. The values in the second column represent the number of function evaluations required by an algorithm to get a relative error of $<1\%$. The results for CORS-RBF are very encouraging and are much better than those obtained by the Tunneling Algorithm and the Constrained DIRECT algorithm. Of course, we would need to explore the algorithm's performance on a more comprehensive set of constrained test problems before we could say anything conclusive about its performance when constraints are added. But the simple example just presented, combined with the results on the box-constrained Dixon–Szegö test problems, suggest that this performance will be quite good.

6. Conclusions

We have introduced the CORS method which is a new strategy for the constrained global optimization of costly functions. We have shown that the

method converges to the global minimizer of any continuous function defined on a compact set. Moreover, this convergence result is independent of the particular response surface model being used and it is also independent of the choice of the initial evaluation points. Finally, computational experiments using radial basis functions indicate that CORS-RBF is a promising approach for constrained global optimization. CORS-RBF methods are competitive with existing global optimization methods for costly functions on the box-constrained Dixon–Szegö test problems and they are better than other methods on a nonlinearly constrained test problem.

Acknowledgements

We would like to thank the Intelligent Information Systems Institute (IISI) directed by Dr Carla Gomes for providing GRA funding for Rommel Regis (AFOSR, grant F49620-01-1-0076). We also acknowledge support from the CISE Directorate in NSF, grant ACI-0305583. Prof Shoemaker's time was supported in part by a Humboldt Research Prize. Finally, we would like to thank the anonymous references for their helpful comments and suggestions.

References

Björkman, M. and Holmström, K. (2000), Global optimization of costly nonconvex functions using radial basis functions. *Optimization Engineering* 1(4), 373–397.

Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V. and Trosset, M.W. (1999), A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* 17(1), 1–13.

Box, G.E.P. and Draper, N.R. (1987), *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, Inc., New York.

Conn, A.R., Scheinberg, K. and Toint, Ph.L. (1997), Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming* 79(3), 397–414.

Cressie, N. (1993), *Statistics for Spatial Data*, John Wiley, New York.

Dennis, J.E. and Torczon, V. (1991), Direct search methods on parallel machines. *SIAM Journal on Optimization* 1(4), 448–474.

Dixon, L.C.W. and Szegö, G. (1978), The global optimization problem: an introduction. In: Dixon, L.C.W. and Szegö, G. (eds.), *Towards Global Optimization 2*, pp. 1–15, North-Holland, Amsterdam.

Friedman, J.H. (1991), Multivariate adaptive regression splines (with discussion). *Annals of Statistics* 19, 1–141.

Gomez, S. and Levy, A. (1982), The tunneling method for solving the constrained global optimization problem with several non-connected feasible regions. In: Dold, A. and Eckmann, B. (eds.), *Numerical Analysis*, *Lecture Notes in Mathematics 909*, pp. 34–47, Springer-Verlag.

Gutmann, H.-M. (2001a), Radial basis function methods for global optimization. Ph.D. Thesis, University of Cambridge.

Gutmann, H.-M. (2001b), A radial basis function method for global optimization. *Journal of Global Optimization* 19(3), 201–227.

Homström, K. (1999), The TOMLAB optimization environment in Matlab. *Advanced Modeling and Optimization* 1(1), 47–69.

Horst, R., Pardalos, P.M. and Thoai, N.V. (1995), *Introduction to Global Optimization*, Kluwer, Dordrecht.

Ishikawa, T. and Matsunami, M. (1997), An optimization method based on radial basis functions. *IEEE Transactions on Magnetics* 33(2), 1868–1871.

Ishikawa, T., Tsukui, Y. and Matsunami, M. (1999), A combined method for the global optimization using radial basis function and deterministic approach. *IEEE Transactions on Magnetics* 35(3), 1730–1733.

Jones, D.R. (1996), Global optimization with response surfaces, presented at the Fifth SIAM Conference on Optimization, Victoria, Canada.

Jones, D.R. (2001a), A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization* 21(4), 345–383.

Jones, D.R. (2001b), The DIRECT global optimization algorithm. In: Floudas, C.A. and Pardalos, P.M. (eds.), *Encyclopedia of Optimization*, Vol. 1, pp. 431–440. Kluwer Academic Publishers,

Jones, D.R., Perttunen, C.D. and Stuckman, B.E. (1993), Lipschitz optimization without the lipschitz constant. *Journal of Optimization Theory and Applications* 78(1), 157–181.

Jones, D.R., Schonlau, M. and Welch, W.J. (1998), Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* 13(4), 455–492.

Khuri, A.I. and Cornell, J.A. (1987), *Response Surfaces*, Marcel Dekker, Inc., New York.

Koehler, J.R. and Owen, A.B. (1996), Computer experiments. In: Ghosh, S. and Rao, C.R. (eds.), *Handbook of Statistics*, *13*: *Design and Analysis of Computer Experiments*, pp. 261–308. North-Holland, Amsterdam.

McKay, M., Beckman, R. and Conover, W. (1979), A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21, 239–246.

Myers, R.H. and Montgomery, D.C. (1995), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, Inc., New York.

Nelder, J.A. and Mead, R.(1965), A simplex method for function minimization. *Computer Journal* 7, 308–313.

Nocedal, J. and Wright, S.J. (1999), *Numerical Optimization*, Springer, New York.

Powell, M.J.D. (1992), The theory of Radial Basis Function Approximation in 1990, in: Light, W. (ed.), *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions*, pp. 105–210. Oxford University Press.

Powell, M.J.D. (1994), A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Gomez, S. and Hennart, J.-P. (eds.), *Advances in Optimization and Numerical Analysis*, pp. 51–67, Kluwer Academic Publishers, Dordrecht.

Powell, M.J.D. (1999), Recent research at Cambridge on radial basis functions. In: Müller, M., Buhmann, M., Mache, D., and Felten, M. (eds.), *New Developments in Approximation Theory, International Series of Numerical Mathematics*, Vol. 132, pp. 215–232, Birkhauser Verlag, Basel.

Powell, M.J.D. (2000), UOBYQA: unconstrained optimization by quadratic approximation. *Mathematical Programming* 92, 555–582.

Powell, M.J.D. (2002), On trust region methods for unconstrained minimization without derivatives, Technical Report. Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK.

Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989), Design and analysis of computer experiments. *Statistical Science* 4(4), 409–435.

Simpson, T.W., Mauery, T.M., Korte, J.J. and Mistree, F. (1998), Comparison of response surface and kriging models for multidisciplinary design optimization. In: *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Vol. 1, pp. 381–391. St. Louis, MO.

The Mathworks, Inc. (2000), *Optimization Toolbox for use with MATLAB: User's Guide, Version 2.1*.

Torczon, V. (1997), On the convergence of pattern search algorithms, *SIAM Journal on Optimization* 7(1), 1–25.

Torn, A. and Zilinskas, A. (1989), *Global Optimization, Lecture Notes in Computer Science*, Vol. 350. Springer-Verlag, Berlin.