AIAA JOURNAL
Vol. 41, No. 4, April 2003

Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling

Yew S. Ong,* Prasanth B. Nair,† and Andrew J. Keane‡
University of Southampton, Southampton, England SO17 1BJ, United Kingdom

We present a parallel evolutionary optimization algorithm that leverages surrogate models for solving computationally expensive design problems with general constraints, on a limited computational budget. The essential backbone of our framework is an evolutionary algorithm coupled with a feasible sequential quadratic programming solver in the spirit of Lamarckian learning. We employ a trust-region approach for interleaving use of exact models for the objective and constraint functions with computationally cheap surrogate models during local search. In contrast to earlier work, we construct local surrogate models using radial basis functions motivated by the principle of transductive inference. Further, the present approach retains the intrinsic parallelism of evolutionary algorithms and can hence be readily implemented on grid computing infrastructures. Experimental results are presented for some benchmark test functions and an aerodynamic wing design problem to demonstrate that our algorithm converges to good designs on a limited computational budget.

Nomenclature

d	=	number of design variables
$f(\boldsymbol{x})$	=	exact analysis function
$\hat{f}(\boldsymbol{x})$	=	approximate analysis function
$g(\boldsymbol{x})$	=	exact constraint function
$\hat{g}(\boldsymbol{x})$	=	approximate constraint function
$\boldsymbol{K}$	=	Gram matrix
k	=	trust-region iteration number
$k_{\max}$	=	maximum trust-region iterations allowed
m	=	number of nearest design points employed
$m_{\max}$	=	maximum number of nearest design points specified
$m_{\min}$	=	minimum number of nearest design points specified
n	=	size of training dataset
P_q	=	polynomial of order $q-1$
p	=	number of inequality constraints
t_c	=	current time spent
t_t	=	computational time budget allocated
$\boldsymbol{x}$	=	design variable vector
$\boldsymbol{x}_l$	=	lower bound of design variable vector
$\boldsymbol{x}_u$	=	upper bound of design variable vector
$\boldsymbol{x}_c^k$	=	initial guess at the kth trust-region iteration
$\boldsymbol{x}_{\text{lo}}^k$	=	local optimum at the kth trust-region iteration
y	=	exact function value
$\hat{y}$	=	approximate function value
$\boldsymbol{\alpha}$	=	weight vector
Δ^k	=	trust-region radius at iteration k
ϵ	=	approximation error
$\boldsymbol{\theta}$	=	undetermined coefficients of polynomial P
ξ	=	update factor for trust region radius
ρ^k	=	approximation figure of merit

Received 6 May 2002; revision received 28 October 2002; accepted for publication 1 November 2002.

*Ph.D. Student, Computational Engineering and Design Center, School of Engineering Sciences; currently Assistant Professor, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Republic of Singapore; asysong@ntu.edu.sg.

†Senior Research Fellow, Computational Engineering and Design Center, School of Engineering Sciences, Highfield; P.B.Nair@soton.ac.uk. Member AIAA.

‡Professor of Computational Engineering, Director Computational Engineering and Design Center, School of Engineering Sciences, Highfield; Andy.Keane@soton.ac.uk.

I. Introduction

HIGH computational costs associated with the use of high-fidelity simulation models pose a serious impediment to the successful application of evolutionary algorithms (EAs) to engineering design optimization. A motivating example for us is aerodynamic wing design, where one function evaluation involving the solution of the Navier–Stokes equations can take many hours of computer time. Such computationally expensive problems also arise in other areas such as structural design, electromagnetics, and design of coupled multidisciplinary systems. In such complex engineering design problems EAs typically require thousands of function evaluations to locate a near-optimal solution. Hence, when computationally expensive high-fidelity simulation models are used for predicting design improvements, the use of EAs can be computationally prohibitive.

In gradient search it is now standard practice for computationally cheap surrogate models to be used in lieu of exact models to reduce computational cost. Because gradient-based optimization algorithms make use of line searches to locate a new iterate, the issue of range of validity of the approximation models or the control of approximation errors can be directly addressed by using ad hoc move limits or a trust-region framework. As shown by Alexandrov et al.,[1] the trust-region strategy for adaptively controlling the move limits guarantees global convergence under some mild assumptions on the accuracy of the surrogate model. More general frameworks for managing the use of approximation models in pattern search algorithms have also been proposed in the literature, for example, see Booker et al.[2] and Serafini.[3] A more detailed survey of the state of the art can be found in Simpson et al.[4]

In contrast, because EAs make use of probabilistic recombination operators controlling the step size of design changes (to control the accuracy of approximate fitness predictions) is not straightforward as in gradient-based optimization algorithms. This difficulty becomes particularly severe when local approximation models are employed during search. In principle, global models can be employed to circumvent this problem. However, in practice, because of the curse of dimensionality, such models become increasingly difficult to construct for problems with large number of variables.

Robinson and Keane[5] presented a case for employing variable-fidelity analysis models and approximation techniques to improve the efficiency of evolutionary optimization for complex design tasks. Computational frameworks for integrating a class of single-point approximation models with EAs were proposed by Nair and Keane.[6] However, such frameworks are restricted to a special class of approximation models that are domain specific. For more general approximation models Ratle[7] examined a strategy for integrating evolutionary search with Kriging models. This problem was revisited

by El-Beltagy et al.,[8] where it is argued that the issue of balancing the concerns of optimization with that of design of experiments must be addressed. Numerical studies were presented for certain pathological cases to show that the idea of constructing an accurate global surrogate model might be fundamentally flawed because of the curse of dimensionality. Liang et al.[9] proposed a strategy for coupling EAs with local search and quadratic response surface methods. However, when working with multimodal problems the accuracy of quadratic models can become questionable. Jin et al.[10] presented a framework for coupling EAs and neural-network-based surrogate models. This approach uses both the expensive and approximate models throughout the search, with an empirical criterion to decide the frequency at which each model should be used.

In spite of extensive work on this topic, existing strategies for integrating approximation models with EAs have met with limited success in applications to real-world problems. Some of the key factors responsible for this are as follows:

1) The first factor is the curse of dimensionality that causes significant difficulties in constructing a global surrogate model which is capable of accurately predicting fitness improvements during the search. This fundamental difficulty arises from the fact that the number of hypercubes required to fill out a compact region of a d-dimensional space grows exponentially with d.

2) The inability of most frameworks to handle problems with general nonlinear inequality and equality constraints is the second factor.

3) Most of the proposed strategies for managing the interplay between the exact and approximate models tend to compromise on the intrinsic parallelism of traditional EAs.

The objective of the present paper is to develop a general approach for integrating surrogate models with EAs, which addresses the limitations of existing strategies just outlined. The proposed algorithm leverages well-established notions in the literature on hybrid evolutionary optimization techniques, radial basis functions, and trust-region frameworks. The essential backbone of our approach is an EA hybridized with a feasible sequential quadratic programming (SQP) solver. The rationale behind using a feasible SQP solver is to exploit its well-known ability to efficiently locate the local optima of optimization problems with general constraints.[11] Each individual in an EA generation is used as an initial guess for local search in the spirit of Lamarckian learning. We employ a trust-region framework to manage the interplay between the original objective and constraint functions and computationally cheap surrogate models during local search.

We propose the idea of employing local surrogate models that are constructed using data points that lie in the vicinity of an initial guess. This local learning technique is an instance of the transductive inference paradigm, which has recently been the focus of recent research in statistical learning theory.[12,13] Traditionally, surrogate models are constructed using inductive inference, which involves using a training dataset to estimate a functional dependency and then using the computed model to predict the outputs at the points of interest. However, when constructing surrogate models for optimization we are specifically interested in ensuring that the models predict the objective and constraint function values accurately at the sequence of iterates generated during the search; how well the model performs at other points in the parameter space is of no concern in this specific context. Transductive inference offers an elegant solution to this problem by directly estimating the outputs at the point of interest in one step; the reader is referred to Vapnik's text[12] (see Chapter 8) for a detailed theoretical analysis of its superior generalization capabilities over standard inductive inference.

In the present work we implement transduction by constructing radial basis networks using data points in the local neighborhood of an optimization iterate. In other words, instead of constructing a global surrogate model a local model is created on the fly whenever the objective and constraint functions must be estimated at a design point during local search. This idea of constructing local models is similar in spirit to the multipoint approximation technique proposed by Toropov et al.[14] and the moving least-squares approximation technique.[15] It is shown that localized training data can be readily selected from a search engine database containing previous iterates, which is continuously updated as the search progresses. Further, the proposed algorithm can be efficiently parallelized on grid computing architectures because it does not compromise on the intrinsic parallelism offered by EAs. Extensive numerical studies are presented for some benchmark test functions and an aerodynamic wing design problem. We show that the present framework allows for the possibility of converging to good designs on a limited computational budget.

This paper is organized as follows: Section II outlines surrogate model construction using radial basis functions. Section III presents the proposed algorithm for integrating local surrogate models and trust-region methods with EAs. The grid infrastructure employed to achieve parallelism is also briefly discussed. In Sec. IV, we summarize experimental studies on some benchmark test functions and an aerodynamic wing design problem. Section V summarizes the main conclusions.

II. Surrogate Modeling

Surrogate models or metamodels are (statistical) models that are built to approximate computationally expensive simulation codes. Surrogate models are orders of magnitude cheaper to run and can be used in lieu of exact analysis during evolutionary search. Further, the surrogate model can also yield insights into the functional relationship between the input $\boldsymbol{x}$ and the output y. If the true nature of a computer analysis code is represented as

$$y = f(\boldsymbol{x}) \tag{1}$$

then a surrogate model is an approximation of the form

$$\hat{y} = \hat{f}(\boldsymbol{x}) \tag{2}$$

such that $y = \hat{y} + \epsilon$.

There exist a variety of techniques for constructing surrogate models, for example, see the texts by Vapnik[12] and Bishop[16] for an excellent exposition of this area. One popular approach in the design optimization literature is least-squares regression using low-order polynomials, also known as response surface methods. A statistically sound alternative for constructing surrogate models of deterministic computer models is Kriging, which is referred to as design and analysis of computer experiments models in the statistics literature[17] and Gaussian process regression in the neural-network literature.[18] A comparison of some surrogate modeling techniques has been presented by Giunta and Watson[19] and Jin et al.[20]

As mentioned earlier, in the present study the use of local surrogate models in the spirit of transductive inference is proposed. In particular, a surrogate model is built on the fly when the objective and constraint functions at an optimization iterate are to be estimated. This local model is built using only a small set of data points that lie in the local neighborhood of the design point of interest. Because surrogate models will probably be built thousands of times during the search in this fashion, computational efficiency is a major concern. This consideration motivates the use of radial basis function networks, which can be efficiently applied to approximate multiple-input multiple-output data, particularly when a few hundred data points are used for training.

Let $\{\boldsymbol{x}_i, y_i, i = 1, 2, \ldots, n\}$ denote the training dataset, where $\boldsymbol{x} \in \mathbb{R}^d$ is the input vector and $y \in \mathbb{R}$ is the output. Because we are interested in cases where the training data are generated by running deterministic computer models, we will focus on interpolating radial basis function models of the form

$$\hat{y} = \sum_{i=1}^{n} \alpha_i K(\|\boldsymbol{x} - \boldsymbol{x}_i\|) \tag{3}$$

where $K(\|\boldsymbol{x} - \boldsymbol{x}_i\|)$: $\mathbb{R}^d \to \mathbb{R}$ is a radial basis kernel and $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \in \mathbb{R}^n$ denotes the vector of weights.

Typical choices for the kernel include linear splines, cubic splines, multiquadrics, thin-plate splines, and Gaussian functions.[16] The structure of some commonly used radial basis kernels and their parameterization are shown in Table 1. Given a suitable kernel, the weight vector can be computed by solving the linear algebraic system of equations $\boldsymbol{K}\boldsymbol{\alpha} = \boldsymbol{y}$, where $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\} \in \mathbb{R}^n$ denotes

Table 1 Radial basis kernels

Kernel	Parameterization
Linear splines	$\|x - c_i\|$
Thin-plate splines	$\|x - c_i\|^k \ln\|x - c_i\|^k$
Cubic splines	$\|x - c_i\|^3$
Gaussian	$\exp[-(\|x - c_i\|^2)/\beta_i]$
Multiquadrics	$\sqrt{1 + (\|x - c_i\|^2)/\beta_i}$
Inverse multiquadrics	$[1 + (\|x - c_i\|^2)/\beta_i]^{-1/2}$

the vector of outputs and $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ denotes the Gram matrix formed using the training inputs (i.e., the ijth element of $\boldsymbol{K}$ is computed as $K(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|)$).

Micchelli[21] proved that nonsingularity of the Gram matrix $\boldsymbol{K}$ can be theoretically guaranteed for a class of kernels only when the set of input vectors in the training dataset is distinct. In many papers in the radial basis function literature, a polynomial term P is often appended to Eq. (3) along with some constraints. In other words, if K is a conditionally positive definite function of order q then to ensure a unique solution for the weight vector Eq. (3) is rewritten as

$$\hat{y} = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + P_q(\boldsymbol{x}) \tag{4}$$

where P_q is a polynomial of order $q - 1$ and the following homogeneous constraint equations are imposed:

$$\sum_{i=1}^{n} \alpha_i P_k(\boldsymbol{x}_i) = 0, \qquad 1 \le k \le q \tag{5}$$

Then the weight vector can be computed by solving a linear algebraic system of equations of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{P} \\ \boldsymbol{P}^T & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{x} = \{\boldsymbol{\alpha}, \boldsymbol{\theta}\}^T, \qquad \boldsymbol{b} = \{\boldsymbol{y}, \boldsymbol{0}\}^T \tag{6}$$

where $\boldsymbol{P}$ is a matrix that arises by substituting the input vectors in the training dataset into the polynomial term P. In practice, good approximations can be obtained by using a constant instead of a full-order polynomial. Here, the coefficient matrix $\boldsymbol{A}$ becomes

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{1} \\ \boldsymbol{1}^T & 0 \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)} \tag{7}$$

where $\boldsymbol{1} \in \mathbb{R}^n$ is a vector of ones.

For problems with multiple outputs, the weight vector can be efficiently computed for all of the outputs of interest once the matrix $\boldsymbol{K}$ is decomposed. For a typical dataset with 500 training points, 20 inputs, and five outputs, surrogate model construction using linear splines takes a fraction of a second on one processor of an SGI Power Challenge. When dealing with computationally expensive problems that cost more than a few minutes of cpu time per function evaluation, this training cost is negligible.

In the present study we use linear splines to construct surrogate models because experimental studies in the literature[20] suggest that this kernel is capable of providing models with good generalization capability at a low computational cost. We present next an algorithm that integrates a local version of such surrogates in hybrid evolutionary search.

III. Present Framework

In this section we present the essential ingredients of the proposed local surrogate modeling algorithm for parallel evolutionary optimization of computationally expensive problems. In particular, we consider a general nonlinear programming problem of the following form.

Minimize:

$$f(\boldsymbol{x})$$

Subject to:

$$g_i(\boldsymbol{x}) \le 0, \qquad i = 1, 2, \ldots, p$$

$$\boldsymbol{x}_l \le \boldsymbol{x} \le \boldsymbol{x}_u \tag{8}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the vector of design variables.

In this paper we are interested in cases where the evaluation of $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ is computationally expensive, and it desired to obtain a near-optimal solution on a limited computational budget. The basic steps of the proposed algorithm are outlined here:

BEGIN
Initialize: Generate a database containing a population of designs. (Optional: upload a historical database if one exists)
While (computational budget not exhausted)
Evaluate all individuals in the population using the exact models.
For each nonduplicated individual in the population
1) Apply trust-region enabled feasible SQP solver to each individual in the population by interleaving the exact and local surrogate models for the objective and constraint functions.
2) Update the database with any new design points generated during the trust-region iterations and their exact objective and constraint function values.
3) Replace the individuals in the population with the locally improved solution in the spirit of Lamarckian learning.
End For
Apply standard EA operators to create a new population.
End While
END

In the first step we initialize a database using a population of designs, either randomly or using design of experiments techniques such as Latin hypercube sampling. All of the design points thus generated and the associated exact values of the objective and constraint functions are archived in the database that will be used later for constructing local surrogate models. Alternatively, one could use a database containing the results of a previous search on the problem or a combination of the two.

Subsequently, with ample design points in the database a hybrid EA is employed, where for each nonduplicated design point or chromosome in the population a local search is conducted using surrogates. The local strategy used here embeds the feasible SQP optimizer within a trust-region framework.[22] However, instead of adopting an augmented Lagrangian approach we handle the objective and constraint functions separately using the approach of Giunta and Eldred.[23] More specifically, during local search for each chromosome in an EA generation we solve a sequence of trust-region subproblems of the following form.

Minimize:

$$\hat{f}^k\left(\boldsymbol{x} + \boldsymbol{x}_c^k\right)$$

Subject to:

$$\hat{g}_i^k\left(\boldsymbol{x} + \boldsymbol{x}_c^k\right) \le 0, \qquad i = 1, 2, \ldots, p$$

$$\|\boldsymbol{x}\| \le \Delta^k \tag{9}$$

where $k = 0, 1, 2, \ldots, k_{\max}$. In practice, the L_∞ norm can be employed to impose the second constraint in Eq. (9). Hence, this constraint can be transformed into appropriate bounds on the design variables, which is updated at each trust-region iteration based on the value of Δ^k.

For each subproblem (or during each trust-region iteration) surrogate models of the objective and constraint functions, namely $\hat{f}^k(\boldsymbol{x})$ and $\hat{g}_i^k(\boldsymbol{x})$, are created dynamically. The m nearest neighbors of the initial guess $\boldsymbol{x}_c^k$ are first extracted from the archived database of design points evaluated so far using the exact analysis code. The criterion used to determine the similarity between design points is the simple Euclidean distance metric. These points are then used to construct local surrogate models of the objective and constraint functions. Care has to be taken to ensure that repetitions do not occur in the training dataset because this might lead to a singular Gram matrix $\boldsymbol{K}$.

The surrogate models thus created are used to facilitate the necessary objective and constraint function estimations in the local searches. During local search, we initialize the trust region Δ using the minimum and maximum values of the design points used to

construct the surrogate model. We found this initialization strategy worked well for the problems considered in this paper. After each iteration the trust region radius Δ^k is updated based on a measure that indicates the accuracy of the surrogate model at the kth local optimum $\boldsymbol{x}_{\text{lo}}^k$. After computing the exact values of the objective and constraint functions at this point, the figure of merit ρ^k is calculated as

$$\rho^k = \min\left(\rho_f^k, \rho_{gi}^k\right) \qquad \text{for} \qquad i = 1, 2, \ldots, p \tag{10}$$

where

$$\rho_f^k = \frac{f\left(\boldsymbol{x}_c^k\right) - f\left(\boldsymbol{x}_{\text{lo}}^k\right)}{\hat{f}\left(\boldsymbol{x}_c^k\right) - \hat{f}\left(\boldsymbol{x}_{\text{lo}}^k\right)}, \qquad \rho_{gi}^k = \frac{g_i\left(\boldsymbol{x}_c^k\right) - g_i\left(\boldsymbol{x}_{\text{lo}}^k\right)}{\hat{g}_i\left(\boldsymbol{x}_c^k\right) - \hat{g}_i\left(\boldsymbol{x}_{\text{lo}}^k\right)} \tag{11}$$

The preceding equations provide a measure of the actual vs predicted change in the objective and constraint function values at the kth local optimum. The value of ρ^k is then used to update the trust region radius as follows[1]:

$$\begin{aligned} \Delta^{k+1} &= 0.25\Delta^k \qquad \text{if} \qquad \rho^k \le 0.25 \\ &= \Delta^k \qquad \text{if} \qquad 0.25 < \rho^k \le 0.75 \\ &= \xi\Delta^k \qquad \text{if} \qquad \rho^k \ge 0.75 \end{aligned} \tag{12}$$

where $\xi = 2$ if $\|\boldsymbol{x}_{\text{lo}}^k - \boldsymbol{x}_c^k\|_\infty = \Delta^k$, or $\xi = 1$ if $\|\boldsymbol{x}_{\text{lo}}^k - \boldsymbol{x}_c^k\|_\infty < \Delta^k$.

The trust region radius Δ^k is reduced if the accuracy of the surrogate, measured by ρ^k, is low. Δ^k is doubled if the surrogate is found to be accurate and the kth local optimum $\boldsymbol{x}_{\text{lo}}^k$ lies on the trust-region bounds; otherwise, the trust-region radius remains unchanged.

The exact values of the objective and constraint functions at the optimal solution of the kth subproblem are combined with the m nearest neighboring design points to generate a new surrogate model for the next iteration. In addition, the initial guess for the $(k+1)$th iteration within each local search is determined by

$$\begin{aligned} \boldsymbol{x}_c^{k+1} &= \boldsymbol{x}_{\text{lo}}^k \qquad \text{if} \qquad \rho^k > 0 \\ &= \boldsymbol{x}_c^k \qquad \text{if} \qquad \rho^k \le 0 \end{aligned} \tag{13}$$

The trust-region iterations (for each chromosome) are terminated when $k \ge k_{\max}$. At the end of $k_{\max}$ trust-region iterations for a chromosome, the exact fitness of the locally optimized design point is determined. If it is found to be better than that of the initial guess, then Lamarckian learning proceeds. Lamarckian learning forces the genotype to reflect the result of improvement by placing the locally improved individual back into the population to compete for reproductive opportunities. In addition, the locally optimized design point and its corresponding objective and constraint function values are added to the database. This process of hybrid EA search is continued until the computational budget is exhausted or a user-specified termination criterion is met.

Apart from the parameters used in standard EAs, our algorithm has two additional user-specified parameters: $k_{\max}$ and m. In Sec. IV we present experimental studies to investigate the effect of these parameters on the convergence trends.

A. Some Remarks on Global Convergence

Global convergence is defined in the optimization literature as the mathematical assurance that the iterates produced by an algorithm, started from an arbitrary initial guess, will converge to a stationary point or local optima of the original high-fidelity expensive analysis code. An approach based on the classical trust region idea from nonlinear programming is shown by Alexandrov et al.[1] to be probably convergent to a local optima of the original problem.

Global convergence results for EAs that make use of approximation models in the search have not appeared in the literature. Nevertheless, it is possible to design EAs that inherit the global convergence properties of existing algorithms. The work by Hart[24] has shown one such possibility where a provably convergent evolutionary pattern search algorithm was proposed that inherits the existing theory for traditional pattern search.

To prove global convergence for trust-region frameworks that embed surrogate models in the local search, Alexandrov et al.[1] showed that zero-order and first-order consistency conditions have to be imposed at the initial guess, that is,

$$\hat{f}\left(\boldsymbol{x}_c^k\right) = f\left(\boldsymbol{x}_c^k\right) \tag{14}$$

$$\nabla\hat{f}\left(\boldsymbol{x}_c^k\right) = \nabla f\left(\boldsymbol{x}_c^k\right) \tag{15}$$

Because we use an interpolating surrogate model in the present approach, only the zero-order consistency condition is satisfied at the initial guess. To satisfy Eq. (15), the exact sensitivities of the objective and constraint functions are required, which would be computationally prohibitive for many problems. Convergence analysis of trust-region algorithms when only inexact gradient information is available has been considered by Carter[25] and Toint.[26] Leveraging these results, Arian et al.[27] presented a theoretical analysis for unconstrained optimization using surrogates to show that under mild assumptions convergence can still be guaranteed. In particular, the condition the surrogate model needs to satisfy is that the predicted direction of descent approximates the "true" direction sufficiently well in the limit. This result can be readily extended to nonlinear programming problems with general constraints by adopting an augmented Lagrangian formulation on the lines of that presented by Rodriguez et al.[22] In summary, global convergence can be guaranteed only when some assumptions are made regarding the descent direction computed using the surrogate model.

The observations made here are of theoretical interest alone because 1) we only carry out a few trust-region iterations during local search for each chromosome and 2) we do not have a theoretical upper bound on the degree to which the descent direction computed using the surrogate model approximates the actual direction of descent.

B. Parallel Implementation

In engineering design optimization evaluation of the objective and constraint functions takes up the overwhelming bulk of the computation. Therefore, a sublinear improvement in design search efficiency can be achieved via global parallelism, where all design points within a single population are evaluated simultaneously across multiple machines. Parallelism is thus considered a desirable feature of any framework for optimization of computationally expensive problems. In the present algorithm it is relatively straightforward to achieve parallelism because local search for each individual in an EA generation can be conducted independently. To ensure load balancing, we only need to specify that the number of trust-region iterations be kept the same for each individual.

In the present implementation of our algorithm, we employed NetSolve,[28] a computational platform that facilitates grid-based heterogenous computing[29] in a transparent and efficient manner. Parallelism is achieved by wrapping the local search and surrogate modeling routines on a NetSolve server. The analysis codes are also wrapped on NetSolve servers, which can be invoked by the local search and the client routines. Hence, using the farming client application programming interface local search for each chromosome in an EA generation can be readily conducted in parallel on remote servers. Even though we used a centralized database, NetSolve has capabilities for distributed data storage on remote servers. This approach does not parallelize the SQP steps and thus is only suitable where the SQP updates can be offered as serial processes.

IV. Numerical Studies

In this section we present numerical results obtained by implementing the proposed approach within a standard binary coded genetic algorithm (GA). We employed a population size of 50 and uniform crossover and mutation operators at probabilities 0.6 and 0.01, respectively. A linear ranking algorithm is used for selection. The codes implementing the objective and constraint functions were wrapped on NetSolve servers running Red Hat Linux on Pentium III processors. The GA code is linked to the NetSolve client library so that the objective function and constraint evaluation modules, and the local search routines can be invoked remotely.

The feasible SQP implementation that is used here is the FFSQP code developed by Lawrence and Tits.[11] When started from an infeasible point (one that violates at least one of the linear or nonlinear constraints), FFSQP first does an optimization in which it minimizes the maximum of the constraint violations. Subsequently, FFSQP minimizes the objective function, while maintaining feasibility of the iterates.

A. Results for Benchmark Test Functions

Two benchmark problems commonly used in the global optimization literature are adopted here for testing the proposed algorithm. They represent classes of unconstrained and constrained multimodal test problems. These problems make it possible to study whether the proposed approach would bring any increase in efficiency or computational cost reduction when used on complex problems. The first example considered is minimization of the Rastrigin function defined here.[30]

Minimize:

$$10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i) \right]$$

Subject to:

$$-5.12 \leq x_i \leq 5.12, \qquad i = 1, 2, \ldots, d \tag{16}$$

The second problem considered is maximization of the bump test function, which is very hard for most search methods to handle.[31] It is quite smooth but contains many peaks, all of similar heights. Its main purpose is to test how methods cope with optima that occur hard up against the constraint boundaries commonly found in engineering design. These properties make it suitable for the study of GA performance as well as optimizing control parameters of evolutionary optimization methods. The function is defined as follows.

Maximize:

$$\text{abs}\left[\sum_{i=1}^{d} \cos^4(x_i) - 2\prod_{i=1}^{d} \cos^2(x_i) \right] \Bigg/ \sqrt{\sum_{i=1}^{d} i x_i^2}$$

Subject to:

$$\prod_{i=1}^{d} x_i > 0.75, \qquad \sum_{i=1}^{d} x_i < \frac{15d}{2}$$

$$0 \leq x_i \leq 10, \qquad i = 1, 2, \ldots, d \tag{17}$$

The objective function in the preceding problem gives a highly bumpy surface where the true global optimum is usually defined by the product constraint. Figure 1 shows both the Rastrigin and the bump objective function for $d = 2$. A 20-dimensional ($d = 20$) version of the test functions is used here for numerical studies. For the first problem involving minimization of the Rastrigin function, there is a unique global optima at which the function value is zero. For the bump test function, even though the global optima is not precisely known for $d = 20$, a value of 0.81 can be obtained after around 100,000 function evaluations using a GA.

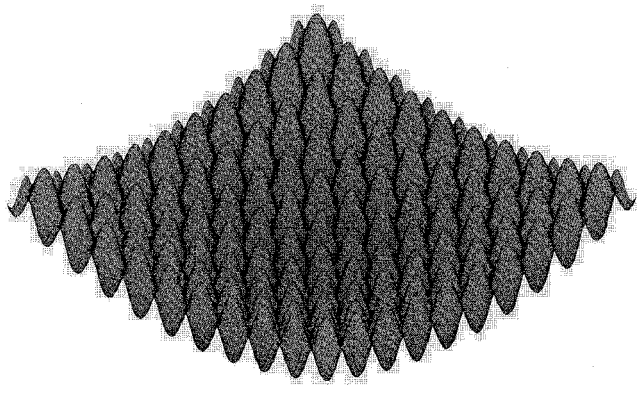

Fig. 1a Surface plot for $d = 2$ of Rastrigin function.

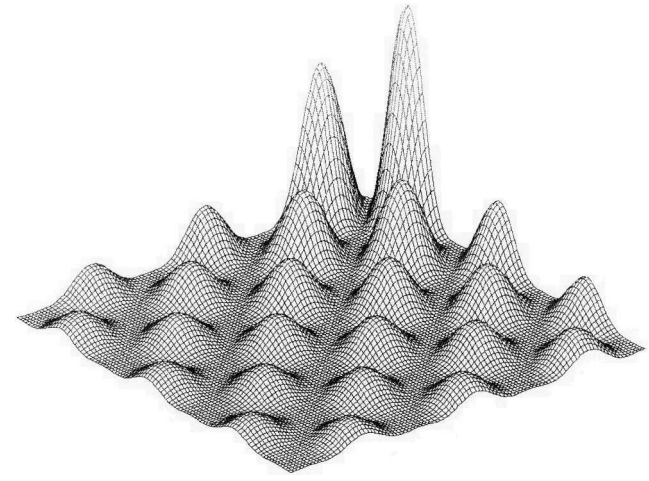

Fig. 1b Surface plot for $d = 2$ of bump function.

The averaged convergence trends obtained by applying the present algorithm to the benchmark test problems as a function of the total number of function evaluations are shown in Figs. 2–4. The results presented here were averaged over 20 runs for each test function. Also shown in the figures are averaged convergence trends obtained using a standard GA and a global surrogate modeling strategy. The algorithm based on global surrogate models employed in our numerical studies is based on the approach proposed by Ratle[7]; see the following for an outline of the steps involved in this algorithm:

```
BEGIN
Initialize: Generate a database containing a population
of designs.
(Optional: upload a historical database if one exists)
Construct surrogate model using all available design points
in the database.
Set fitness function := Surrogate model
While (computational budget not exhausted)
   Evaluate all individuals in the population using the fitness
   function.
   Apply standard EA operators to create a new population.
   If (fitness function := Exact model)
      Update database with any new designs generated using
      the exact model.
      Update surrogate model using all designs in the database.
   End If
   If (convergence over surrogate model)
      fitness function := Exact model
   Else
      fitness function := Surrogate model
   End If
End While
END
```

The results obtained for the test functions show that the global surrogate framework displays early sign of stalling. This is consistent with previous studies in the literature,[7,8,10] which suggest that when global surrogate models are applied to high-dimensional and multimodal test functions the search generally tends to stall early on. Such an effect is a result of the curse of dimensionality, which often leads to early convergence at false global optima of the surrogate model. In contrast, the results obtained using the proposed algorithm clearly demonstrate that solutions close to the global optima can be obtained on a limited computational budget. As surrogates are used only for local searches, that is, as the exact model is used for all analysis conducted at the EA level, the chances for convergence to false global optima are greatly reduced. In addition, the use of the trust-region framework maintains convergence close to the local optima of the original problem during the SQP steps.

For these benchmark problems we also studied the effect of increasing the maximum number of trust-region iterations and the

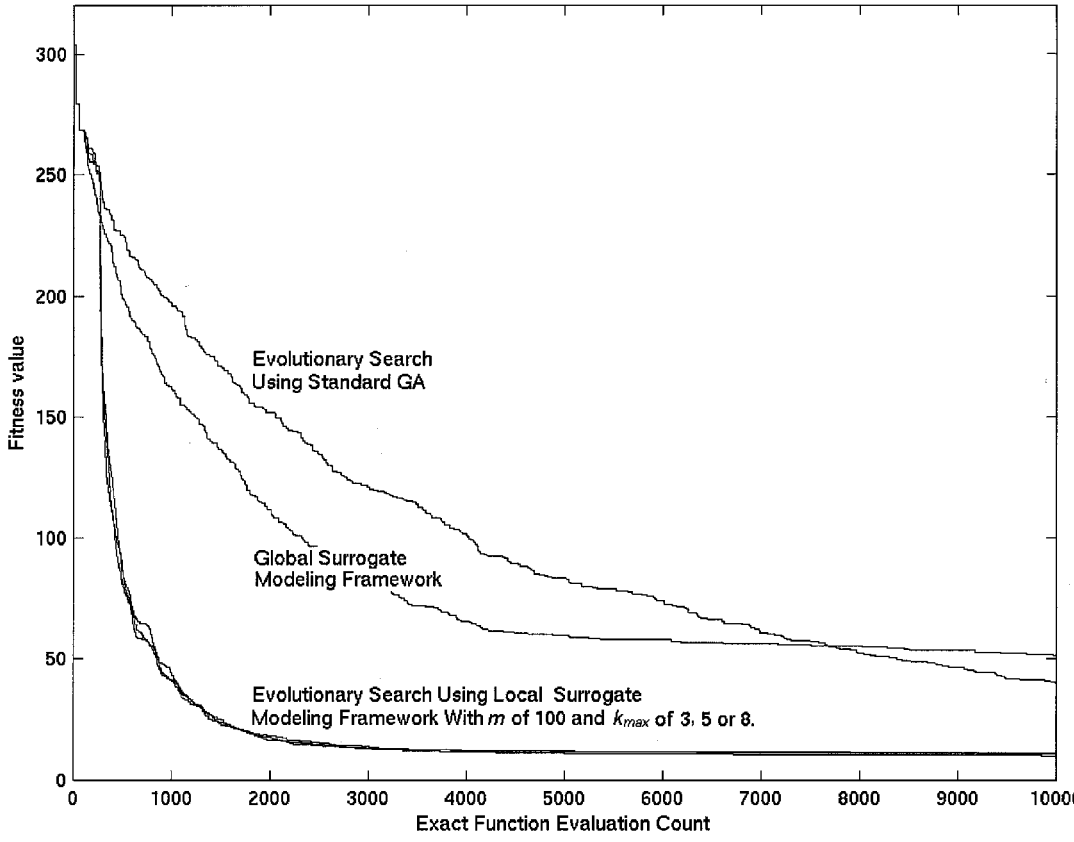

Fig. 2 Averaged convergence trends for m = 100 and various values of $k_{\max}$ (3, 5, and 8) compared with standard GA and global surrogate modeling framework for the 20-dimensional Rastrigin function.

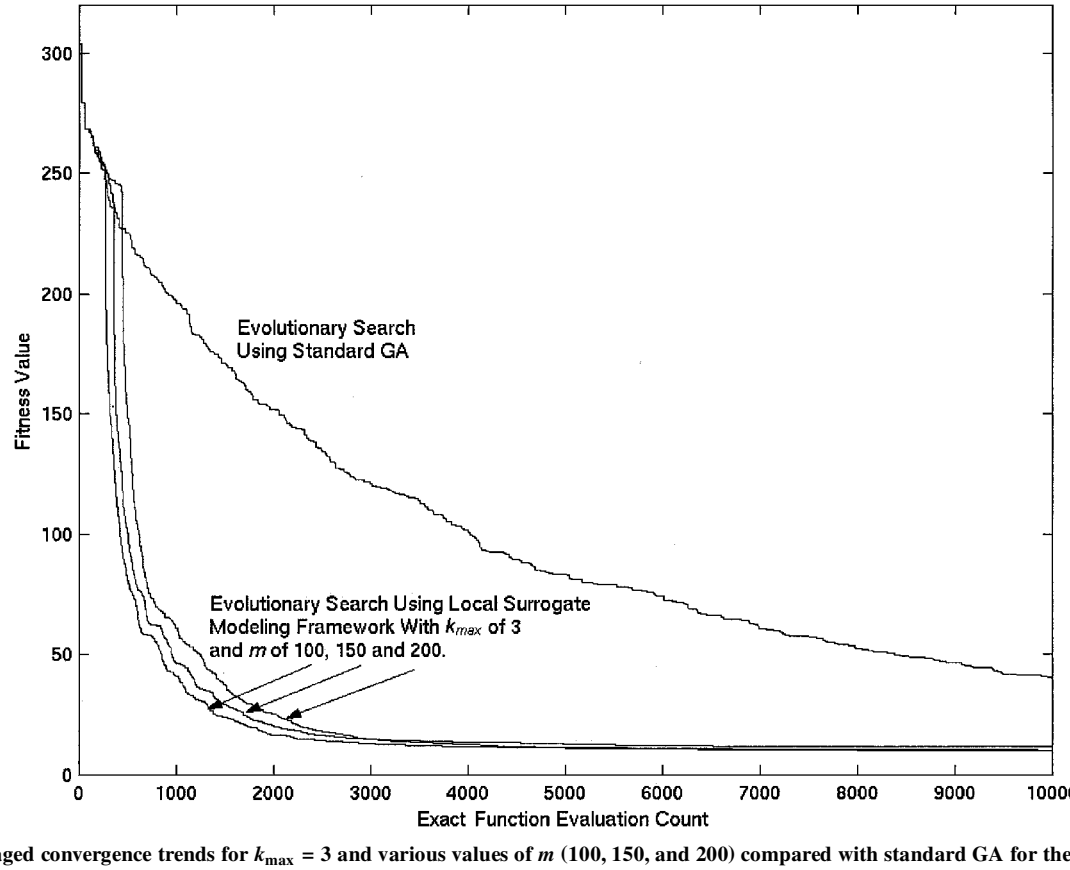

Fig. 3 Averaged convergence trends for $k_{\max}$ = 3 and various values of m (100, 150, and 200) compared with standard GA for the 20-dimensional Rastrigin function.

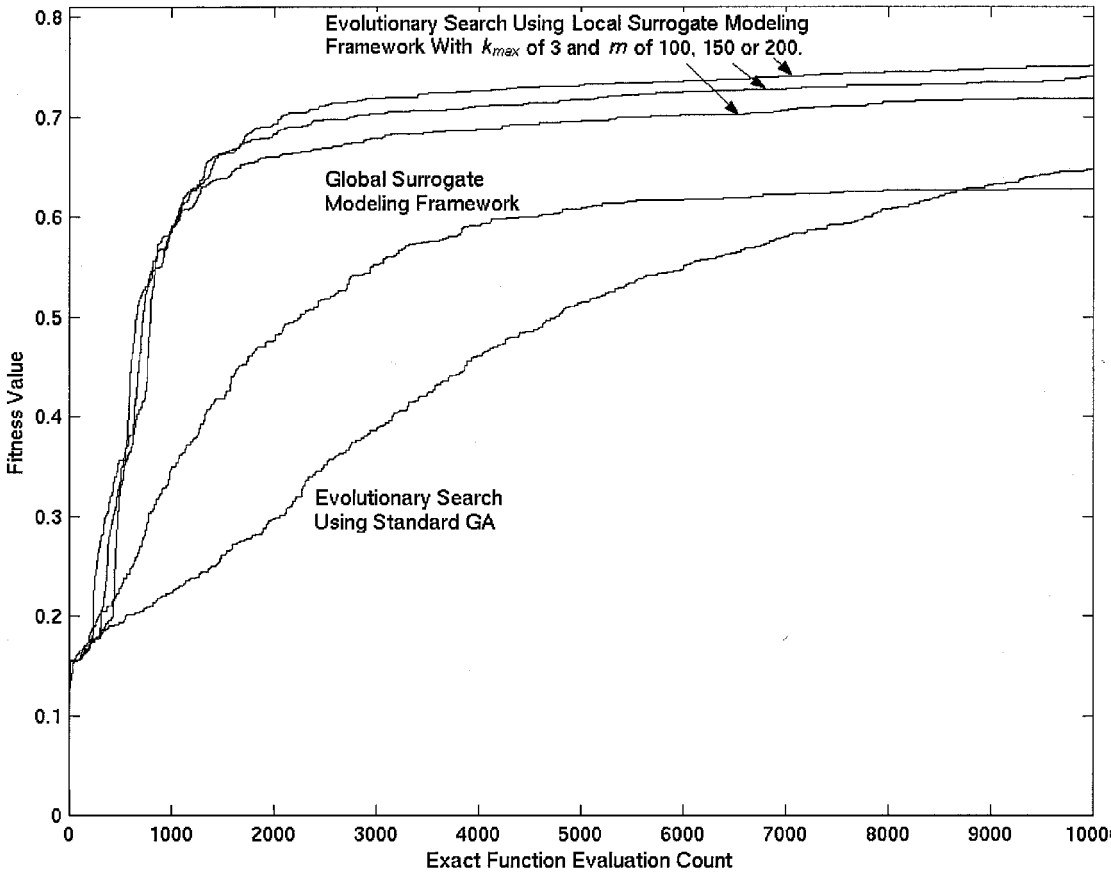

Fig. 4 Averaged convergence trends for $k_{\max} = 3$ and various values of m (100, 150, and 200) compared with standard GA for the 20-dimensional bump function.

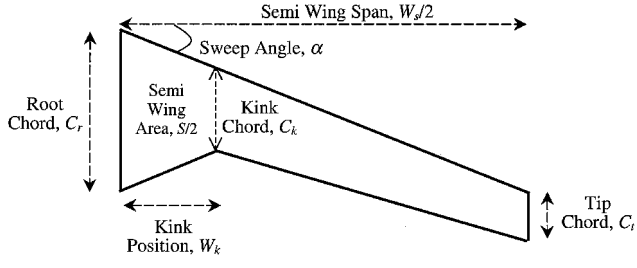

Fig. 5 Aircraft wing planform geometry.

number of nearest neighbors (employed to construct the local surrogate model) on the convergence behavior (Figs. 2–4). A number of observations can be made from the convergence trends. First, it appears that there is not much to be gained by increasing $k_{\max}$ beyond three. Second, it appears that smaller values of m generally lead to faster convergence during the early stages of search, but there is a tendency to stall at later stages. The converse is true for increases in m. This suggests the possibility of adaptively selecting m during the search. We propose the following simple strategy:

$$m = (m_{\min} + m_{\max})(t_c/t_t) \tag{18}$$

where $m_{\min} =$ population size and $m_{\max}$, which limits the design point size used for local surrogate modeling, is set to 400 here.

B. Aerodynamic Wing Design

In this section we present the application of the proposed algorithm to the transonic civil transport aircraft wing design problem considered in Keane and Petruzzelli.[32] Aerodynamic wing design is an extremely complex task, which is normally undertaken over an extended time period and at different levels of complexity. The parameters used to describe the wing design problem considered here consist of the freestream velocity and coefficient of lift of the wing together with a small number of overall wing geometry variables. The geometry is characterized by the planform shape of the wing together with several span-wise functions such as twist and thickness to chord ratio. The planform geometry and design variable definitions are shown in Fig. 5 and Table 2, respectively. To prevent the optimizer from driving the designs to unworkable extremes, several constraints are placed on the wings designed. These are the undercarriage bay length (which must be accommodated within the root to kink section of the wing), the fuel tank volume (which must be accommodated between the main spars within the wing), the wing weight, and the pitch-up margin. These four nonlinear inequality constraints are also listed in Table 2.

Table 2 Optimization conditions for wing design parameters, constraints, and respective limits

Lower limit	Upper limit	Quantity (unit)
11 wing design variable definitions		
100	250	Wing area (m^2), S
6	12	Aspect ratio, W_s^2/S
0.2	0.45	Kink position, $2W_k/W_s$
25	45	Sweep angle (deg), α
0.4	0.7	Inboard taper ratio, C_k/C_r
0.2	0.6	Outboard taper ratio, C_t/C_r
0.1	0.18	Root thickness/chord, T_r/C_r
0.06	0.14	Kink thickness/chord, T_k/C_k
0.06	0.14	Tip thickness/chord, T_t/C_t
4.0	5.0	Tip wash (deg)
0.65	0.85	Kink washout fraction
4 design constraints		
2.5	——	Undercarriage bay length
——	135,000	Wing weight (N)
40.0	——	Wing volume (m^3)
——	5.4	Pitch-up margin

In the present study we considered the optimization of a civil transport aircraft wing for operation at Mach 0.785 and a Reynolds

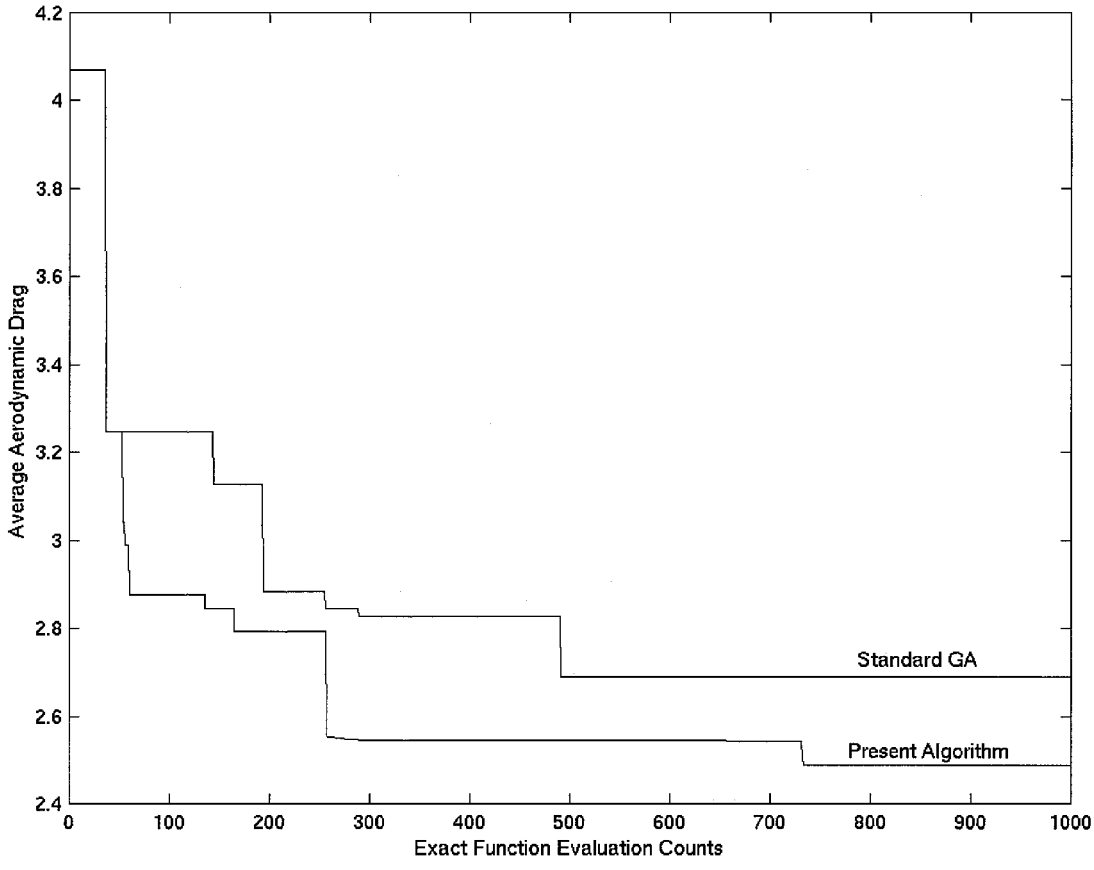

Fig. 6 Summary of minimum drag values (in counts) using the VSAERO code and surrogate models. Results are average of three runs.

Table 3 Summary of minimum drag values (in counts) using the TADPOLE code and surrogate models

	Mean wing drag values (D/q, m^2) at evaluation counts of					
Approach	250	500	600	800	1200	1500
$m = 100$	2.823	2.781	2.788	2.788	2.788	2.788
$m = 200$	3.218	2.788	2.775	2.771	2.768	2.768
$m = 300$	3.235	3.049	3.049	2.771	2.764	2.761
$m = 400$	3.265	3.064	3.049	3.020	2.772	2.758
Adaptive m	2.919	2.771	2.771	2.768	2.760	2.758
Standard GA	3.213	3.004	2.983	2.978	2.974	2.961

number of 7.3×10^6. The objective is minimization of wing D/q m^2 as calculated by using an empirical drag estimation tool TADPOLE and the linearized potential code VSAERO, with target lift, wing weight, volume, pitch-up margin, and root triangle layout chosen to be representative of a 220-seat wide-body airliner. Both codes return the total drag coefficient defined by the wave drag as a result of the presence of shocks, viscous wake, or profile drag caused by the boundary layer and vortex or induced drag caused by the tip vortex of the three-dimensional wing. A common approach to drag recovery is also implemented in the two codes. TADPOLE takes only some 6 s to run and returns drag values based on curve fits to previously analyzed wings making assumptions about the kinds of roof-top pressure profiles now commonly achieved in transonic wing design. VSAERO is a linearized potential code with coupled viscous boundary layer and, as employed here, with added correction for compressibility.[33] It is computationally more expensive than TADPOLE and requires approximately 11 min of compute time per drag evaluation. However, it has the advantage of providing more accurate drag predictions provided Mach numbers are not too high.

A summary of the search results obtained for the aerodynamic wing design problem is shown in Table 3 and Figs. 6 and 7. Table 3 shows the search results when using TADPOLE for drag estimation. Once again, it is observed from these results that smaller values of m lead to faster convergence of the wing drag values during the early stages of search, but result in stalling at later stages, and the converse is true for increases in m. These results are in line with those observed earlier for the test functions. Hence, for the computationally expensive VSAERO code m is chosen adaptively using Eq. (18). We also set the maximum number of trust region iterations ($k_{\max}$) to three. During local search, we constructed surrogate models for the objective function and the four inequality constraints.

For this problem it has been observed from previous studies[32] using the TADPOLE code that a design with a drag value of 2.758 counts can be obtained after 10,000 evaluations. In comparison, using the proposed approach we are able to converge to this solution on an average after 1500 exact evaluations, when m is chosen adaptively during the search.

In Fig. 6, the averaged convergence trends using VSAERO for wing drag estimation also clearly illustrate the ability of the proposed algorithm to arrive at good designs on a limited computational budget. The convergence trends of the best run for the aerodynamic wing design problem using VSAERO are also plotted as a function of wall time in Fig. 7. Because of the availability of only eight licenses for the VSAERO code, the timing plot obtained in Fig. 7 is based on a total of eight processors being used for parallel computations. Previous studies using the VSAERO code for wing drag estimation have revealed that the best design that can be obtained using the standard GA with only the exact analysis model has a drag value of 2.63 after 1800 evaluations. In comparison, using the present approach we were able to obtain this solution on average after 250 exact evaluations. After 733 exact evaluations the best design obtained using our algorithm had a drag value of 2.404, which is the lowest value obtained to date for this problem using various optimization algorithms. The optimal solutions reported here satisfied all four of the inequality constraints.

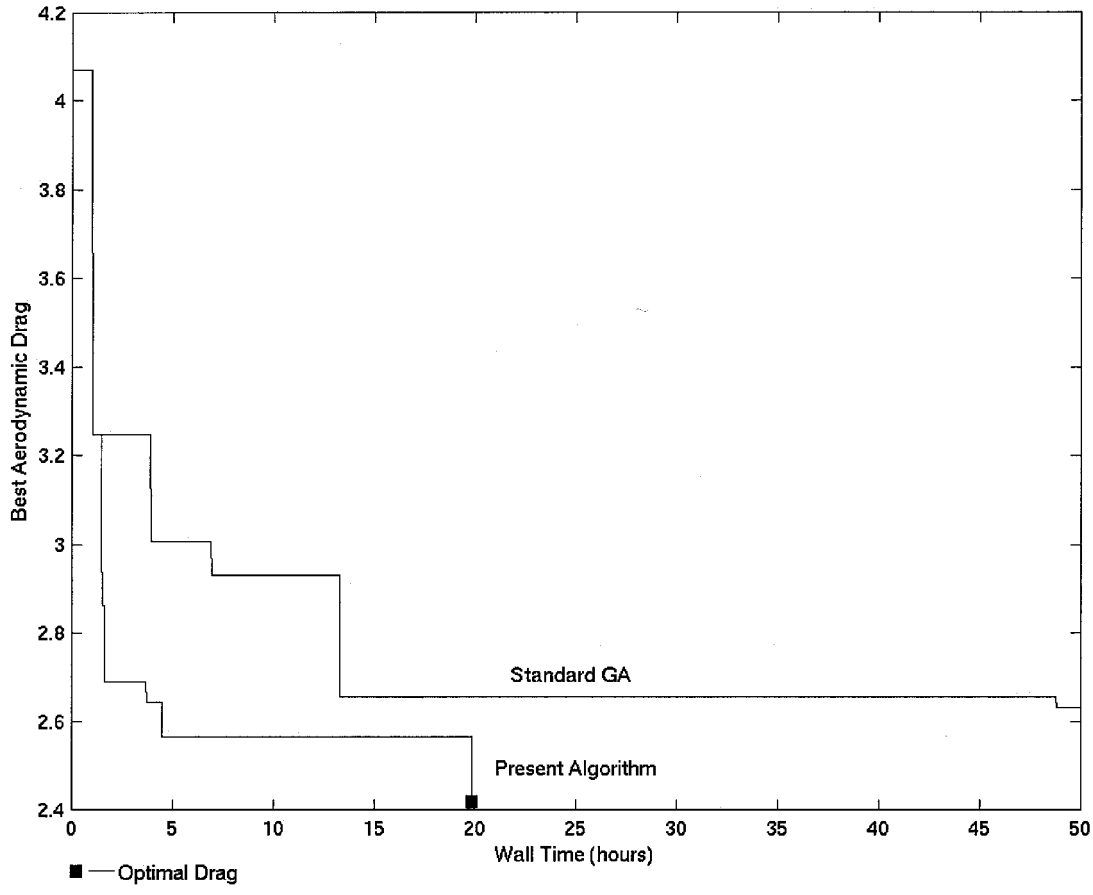

Fig. 7 Comparison of best convergence trends as a function of wall time for the aerodynamic wing design problem using the VSAERO code and surrogate models.

V. Conclusions

In this paper we present a hybrid algorithm that leverages surrogate models for evolutionary optimization of computationally expensive constrained design problems. It is argued that for such complex design problems constructing an accurate global surrogate model is fraught with fundamental difficulties because of the curse of dimensionality. A local learning approach in the spirit of transductive inference is employed to construct surrogate models. We show that such local approximation models can be readily incorporated into hybrid evolutionary-gradient optimization algorithms. Because our local search strategy employs a trust-region framework to interleave the exact and approximate models, convergence to the optima of the original expensive problem can be guaranteed under some mild assumptions. Further, it is shown that the present approach retains the intrinsic parallelism of traditional evolutionary algorithms.

We presented extensive numerical studies on some benchmark test functions to demonstrate the competitiveness of the proposed algorithm. The results were compared with those obtained using a standard genetic algorithm and a global surrogate modeling strategy. Experimental results are also presented for an aerodynamic wing design problem. These studies indicate that the present approach allows for the possibility of arriving at near-optimal solutions on a limited computational budget.

Acknowledgments

This research was supported by the University Technology Partnership for Design, a partnership between BAe Systems, Rolls–Royce Plc., and the Universities of Southampton, Cambridge, and Sheffield, England, United Kingdom. The authors thank the anonymous referees and editors for their constructive comments on an earlier draft of this paper.

References

[1]Alexandrov, N., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," *Structural Optimization*, Vol. 15, No. 1, 1998, pp. 16–23.

[2]Booker, A. J., Dennis, J. E., Jr., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," *Structural Optimization*, Vol. 17, No. 1, 1998, pp. 1–13.

[3]Serafini, D. B., "A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions," Ph.D. Dissertation, Computational and Applied Mathematics Dept., Rice Univ., Houston, TX, Nov. 1998.

[4]Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch, P. N., and Yang, R.-J.,"Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion," *Proceedings of the Third ISSMO/AIAA Internet Conference on Approximations in Optimization* [CD-ROM], Oct. 2002.

[5]Robinson, G. M., and Keane, A. J., "A Case for Multi-Level Optimization in Aeronautical Design," *Aeronautical Journal*, Vol. 103, No. 1028, 1999, pp. 481–485.

[6]Nair, P. B., and Keane, A. J., "Passive Vibration Suppression of Flexible Space Structures via Optimal Geometric Redesign," *AIAA Journal*, Vol. 39, No. 7, 2001, pp. 1338–1346.

[7]Ratle, A., "Kriging as a Surrogate Fitness Landscape in Evolutionary Optimization," *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, Vol. 15, No. 1, 2001, pp. 37–49.

[8]El-Beltagy, M. A., Nair, P. B., and Keane, A. J., "Metamodelling Techniques for Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO99)*, edited by W. Banshaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Morgan Kaufman, San Mateo, CA, 1999, pp. 196–203.

[9]Liang, K.-H., Yao, X., and Newton, C., "Evolutionary Search of Approximated N-dimensional Landscapes," *International Journal of Knowledge-Based Intelligent Engineering Systems*, Vol. 4, No. 3, 2000, pp. 172–183.

[10]Jin, Y., Olhofer, M., and Sendhoff, B., "A Framework for Evolutionary Optimization with Approximate Fitness Functions," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 5, 2002, pp. 481–494.

[11] Lawrence, C. T., and Tits, A. L., "A Computationally Efficient Feasible Sequential Quadratic Programming Algorithm," *SIAM Journal on Optimization*, Vol. 11, No. 4, 2001, pp. 1092–1118.

[12] Vapnik, V., *Statistical Learning Theory*, Wiley, New York, 1998.

[13] Chapelle, O., Vapnik, V., and Weston, J., "Transductive Inference for Estimating Values of Functions," *Advances in Neural Information Processing Systems*, Vol. 12, 1999.

[14] Toropov, V. V., Filatov, A. A., and Polynkin, A. A., "Multiparameter Structural Optimization Using FEM and Multipoint Explicit Approximations," *Structural Optimization*, Vol. 6, No. 1, 1993, pp. 7–14.

[15] Levin, D., "The Approximation Power of Moving Least-Squares," *Mathematics of Computation*, Vol. 67, No. 224, 1998, pp. 1517–1532.

[16] Bishop, C., *Neural Networks for Pattern Recognition*, Oxford Univ. Press, Oxford, 1995.

[17] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–435.

[18] Williams, C. K. I., and Rasmussen, C. E., "Gaussian Processes for Regression," *Advances in Neural Information Processing Systems 8*, edited by D. S. Touretsky, M. C. Mozer, and M. E. Hasselmo, MIT Press, Cambridge, MA, 1996, pp. 514–520.

[19] Giunta, A. A., and Watson, L. T., "A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," AIAA Paper 98-4758, Sept. 1998.

[20] Jin, R., Chen, W., and Simpson, T. W., "Comparative Studies of Metamodeling Techniques Under Multiple Modeling Criteria," *Structural and Multidisciplinary Optimization*, Vol. 23, No. 1, 2001, pp. 1–13.

[21] Micchelli, C. A., "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions," *Constructive Approximation*, Vol. 2, No. 1, 1986, pp. 11–22.

[22] Rodriguez, J. F., Renaud, J. E., and Watson, L. T., "Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data," *Structural Optimization*, Vol. 15, No. 3–4, 1998, pp. 141–156.

[23] Giunta, A. A., and Eldred, M. S., "Implementation of a Trust Region Model Management Strategy in the DAKOTA Optimization Toolkit," *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* [CD-ROM], 2000.

[24] Hart, W. E., "A Convergence Analysis of Unconstrained and Bound Constrained Evolutionary Pattern Search," *Evolutionary Computation*, Vol. 9, No. 1, 2000, pp. 1–23.

[25] Carter, R. G., "On the Global Convergence of Trust-Region Algorithms Using Inexact Gradient Information," *SIAM Journal of Numerical Analysis*, Vol. 28, No. 1, 1991, pp. 251–265.

[26] Toint, P. L., "Global Convergence of a Class of Trust Region Methods for Nonconvex Minimization in Hilbert Space," *IMA Journal of Numerical Analysis*, Vol. 8, No. 2, 1988, pp. 231–252.

[27] Arian, E., Fahl, M., and Sachs, E. W., "Trust-Region Proper Orthogonal Decomposition for Optimal Flow Control," NASA CR-2000-210124, May 2000.

[28] Casanova, H., and Dongarra, J., "NetSolve: A Network Server for Solving Computational Science Problems," *International Journal of Supercomputer Applications and High Performance Computing*, Vol. 11, No. 3, 1997, pp. 212–223.

[29] Foster, C., and Kesselman, C. (eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Mateo, CA, 1999.

[30] Liang, K. H., Yao, X., and Newton, C., "Evolutionary Search of Approximated N-Dimensional Landscapes," *International Journal of Knowledge-Based Intelligent Engineering Systems*, Vol. 4, No. 3, 2000, pp. 172–183.

[31] Keane, A. J., "Genetic Algorithm Optimization of Multi-Peak Problems: Studies in Convergence and Robustness," *Artificial Intelligence in Engineering*, Vol. 9, No. 2, 1995, pp. 75–83.

[32] Keane, A. J., and Petruzzelli, N., "Aircraft Wing Design Using GA-Based Multi-Level Strategies," *Proceedings of the 8th AIAA/USAF/NASSA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* [CD-ROM], 2000.

[33] Petruzzelli, N., and Keane, A. J., "Wave Drag Estimation for Use with Panel Codes," *Journal of Aircraft*, Vol. 38, No. 4, 2001, pp. 778–782.

A. Messac
Associate Editor