Applied Soft Computing 24 (2014) 781–793

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Direct and indirect design prediction in genetic algorithm for inverse design problems

Y. Volkan Pehlivanoglu*

Turkish Air Force Academy, Istanbul 34149, Turkey

ARTICLE INFO

Article history:
Received 10 April 2013
Received in revised form 7 January 2014
Accepted 13 August 2014
Available online 23 August 2014

Keywords:
Genetic algorithm
Neural nets
Surrogate modeling
Inverse design problem

ABSTRACT

An improved genetic algorithm is proposed and tested for five different test cases: surface fittings of a wing and geographical terrain, an inverse design of an airfoil and wing shapes at subsonic flow, and an inverse design of an airfoil shape at transonic flow. The new algorithm emphasizes the use of both direct and indirect design predictions based on local surrogate models in genetic algorithm structure. Local response surface approximations are constructed by using neural networks. For all the demonstration problems considered herein, remarkable reductions in the computational times have been accomplished.

1. Introduction

An inverse design problem is a type of indirect problem and it is widely known in natural sciences. Any closed system contains three elements: these are a cause, a model, and an effect. We may call these factors as an input, a process, and an output, respectively [1]. Most of the formulations of inverse problems may proceed to the setting of an optimization problem. In general, an inverse design problem can be expressed as follows:

$$\textit{find}\ \left\{\boldsymbol{x} \in R^d\right\} \tag{1}$$

$$\min f(\boldsymbol{x}, \boldsymbol{y}) \tag{2}$$

Subject to

$$g(\boldsymbol{x}, \boldsymbol{y}) \le 0 \tag{3}$$

$$\boldsymbol{x}^L \le \boldsymbol{x} \le \boldsymbol{x}^U \tag{4}$$

where $\boldsymbol{x}$ is an input that is the design parameter vector whose values lie in the range given by upper and lower borders in Eq. (4). The objective function $f(\boldsymbol{x},\boldsymbol{y})$ in an inverse design problem is used to bring the computed response from the model as close as possible to the target output. In some problems, it may be necessary to satisfy certain inequality constraints given by $g(\boldsymbol{x},\boldsymbol{y})$. The objective function is usually a least-squares function given by

$$f(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} (y_i^c - y_i^t)^2 \tag{5}$$

where y_i^t is the ith value of the target response and y_i^c is the ith value of the computed response obtained from the simulation model.

In most engineering problems, computational methods are gradually replacing empirical methods; and design engineers are spending more time in applying computational tools instead of conducting physical experiments to design and analyze engineering components. Computational optimization efforts may be divided into Gradient-Based (GB) and non-gradient methods [2]. GB methods give accurate results; and they are usually efficient methods in terms of computational effort. However, they may have some drawbacks: (a) the search leads to a local optimum, (b) the efficiency of the process is affected by the method adopted to compute the gradient of objective and constraint functions, (c) and the gradient evaluation is highly sensitive to the presence of noise in the objective/constraint functions, thereby, compromising the effectiveness of the method [3]. The demand for a method of operations research, which is capable of escaping local optima, has led to the development of non-traditional search algorithms. Non-gradient based methodologies, such as Genetic Algorithms (GAs) or Particle Swarm Optimization (PSO) algorithms, which are less susceptible to pitfalls of convergence to local optima, suggest a good alternative to conventional optimization techniques. These algorithms are population based, and they include a lot of design candidates

* Correspondence to: Turkish Air Technical Schools Command, İzmir 35415, Turkey. Tel.: +90 232 251 16 00; fax: +90 232 251 35 89.
E-mail addresses: vpehlivanoglu@tekok.edu.tr, v.pehlivan@hho.edu.tr

http://dx.doi.org/10.1016/j.asoc.2014.08.018
1568-4946/

waiting for the objective function computations in each generation. The major weakness of population based algorithms lies in their poor computational efficiency, because the evaluation of objective function is sometimes very expensive as it can be seen in the framework of Computational Fluid Dynamics (CFD) or Computational Electromagnetics (CEM) model the evaluation of which takes sometimes hours or days [4]. Despite the considerably improved computer power over the past few decades, computational simulation can still be prohibitive for a large number of executions in practical engineering design. Therefore, improving the efficiency of evolutionary search algorithms has become a key factor in their successful applications to real-world problems.

Two categories of techniques have been proposed to tackle the efficiency issue of evolutionary search methods; the first type is focused on devising more efficient variants of the canonical algorithms, the second type involves using a surrogate model which is a kind of approximation in lieu of the exact and often expensive function evaluations [5].

In literature, there are a lot of surrogate model-based optimization algorithms. The key idea in these methods is to parameterize the space of possible solutions via a simple, computationally inexpensive model, and to use this model to generate inputs in terms of predicted objective function values for the optimization algorithm. Therefore, the whole optimization process is managed by surrogate model outputs. Such a model is often referred to as the response surface of the system to be optimized, leading to the definition of a so-called surrogate-model based optimization methodology [6]. Major issues in surrogate model-based design optimization are the approximation efficiency and accuracy. In case of the problem which has a high number of design variables, the construction of surrogate model may cause extremely high computational cost, which means inefficient approximation. The number of design points may be decreased by using Design of Experiment (DoE). However, then, this approach may lose its coherency in the application. On the other hand, it is possible to miss the global optimum, because the approximation model includes uncertainty at the predicted point, and this uncertainty may mislead the optimization process in a wrong way.

The present paper introduces the application of an improved GA to speed up the optimization process and overcome problems such as inaccuracy and premature convergence during the optimization. The principal role of a surrogate model usage in the new approach is to answer the question of which individual(s) should be merged into next population, which brings us to the use of the approximation model for the estimation of design candidate instead of predicting the objective function value. The new method also shows both an indirect and direct usage of surrogate modeling based on the nature of an inverse design problem. To demonstrate the efficiency of the proposed GA algorithm called s^{12}-GA, it is applied to six different test cases, and the results were compared with five different GAs, including Regular Genetic Algorithms (RGA-1, RGA-2, and RGA-3), Vibrational Genetic Algorithm (VGA), and augmented genetic algorithm (s^1-GA). The test bed selected herein includes curve fitting of an airfoil, surface fittings of a wing shape and geographical terrain, an inverse design of an airfoil and wing shape at subsonic flow, and an inverse design of an airfoil at transonic flow.

2. Surrogate modeling

The stages of surrogate-based modeling approach include a sampling plan for design points, numerical simulations at these design points, construction of a surrogate model based on simulations, and model validation [7]. The sampling plan contains both the samples required for surrogate model construction and a few additional samples required for the verification of a surrogate model. The DoE provides the sampling plan in design variable space. The details of DoE methods can be found in [8]. The sampling plan is model-independent, and in practice, the number of points in the sampling plan is severely limited due to computational expense. In the second stage, numerical simulations are executed in DoE specified points. There are both parametric and non-parametric alternatives to construct the surrogate model. The parametric approaches such as polynomial regression presume the global functional form between the samples and corresponding responses. The non-parametric ones such as neural networks (NN) use simple local models in different regions of a sampling plan to construct an overall model. Finally, the last stage, model validation has the purpose of judging the predictive capabilities of the surrogate model. If necessary, these stages are repeated to provide the desired model validation accuracy. After surrogate-based modeling is completed, the optimization problem is described as followsMinimize $\hat{f}(\boldsymbol{x})$Subject to

$$\hat{g}_i(\boldsymbol{x}) \leq 0,\ i = 1, 2, \ldots, I \tag{6}$$

$\boldsymbol{x}^L \leq \boldsymbol{x} \leq \boldsymbol{x}^U$

This is where the functions are the approximation models. The main purpose of constructing approximate models in this framework is to predict the value of objective and constraints. The relationship between the true response and the approximation can be expressed as follows:

$$f(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \Delta(\boldsymbol{x}) \tag{7}$$

$$\Delta(\boldsymbol{x}) = \varepsilon(\boldsymbol{x}) + \delta(\boldsymbol{x}) \tag{8}$$

The total error $\Delta(\boldsymbol{x})$ includes two types of errors: the first one is system error, $\varepsilon(\boldsymbol{x})$ which exists because of the incompleteness of the surrogate model; and the second one is random error, $\delta(\boldsymbol{x})$ which exists because of uncontrollable factors such as discretization and round off errors in computational studies. Many different surrogate-model based optimization algorithms were applied to decrease the level of $\Delta(\boldsymbol{x})$ in engineering problems. Examples are commonly from GA applications such as: an iterative response surface based optimization scheme [3], a statistical improvement criteria with Kriging surrogate modeling [9], more accurate Kriging modeling by using a dynamic multi-resolution technique [6], the use of multiple surrogates [10,11], a multistage meta-modeling approach [12], an iteratively enhanced Kriging meta-model [13], hybrid variable fidelity optimization by using a Kriging-based scaling function [14], and preference-based surrogate modeling [15].

In addition to the classical surrogate modeling approach, another methodology was also used in a few GA based studies. This group is called a hybrid approach. The main purpose of constructing approximate models in this framework is to predict the positions of new design points, rather than to make inexact computational evaluations as in the surrogate model. An example given by Ong et al. [16] presented an Evolutionary Algorithm (EA) that leverages surrogate models. The essential backbone of the framework is an EA coupled with a feasible Sequential Quadratic Programming (SQP) solver in the spirit of Lamarckian learning. Pehlivanoglu and Baysal [17] and Pehlivanoglu and Yagiz [18] have also suggested a novel usage of regression model and neural networks in GA architecture. They used a new technique to predict better solution candidates using local response surface approximation based on neural networks inside the population for the direct shape optimization of an airfoil at transonic flow conditions. Instead of predicting the design candidate, Liu [19] has suggested to predict the genes of the individuals. He proposed an intelligent GA that incorporates the fractional factorial design in the crossover operator for the determination of the best genes for children produced by the mating of a pair of parents. Moreover, this work