Single- and Multiple-Objective Optimization with Differential Evolution and Neural Networks

Man Mohan Rai*
NASA Ames Research Center, Moffett Field, CA-94035, USA

INTRODUCTION

Genetic and evolutionary algorithms[1] have been applied to solve numerous problems in engineering design where they have been used primarily as optimization procedures. These methods have an advantage over conventional gradient-based search procedures because they are capable of finding global optima of multi-modal functions (not guaranteed) and searching design spaces with disjoint feasible regions. They are also robust in the presence of noisy data. Another desirable feature of these methods is that they can efficiently use distributed and parallel computing resources since multiple function evaluations (flow simulations in aerodynamics design) can be performed simultaneously and independently on multiple processors. For these reasons genetic and evolutionary algorithms are being used more frequently in design optimization. Examples include airfoil and wing design[2 - 3] and compressor and turbine airfoil design. They are also finding increasing use in multiple-objective and multidisciplinary optimization.[4] The references cited here represent a very small sample of the literature.

One problem with genetic and evolutionary algorithms is that they often require many more function evaluations than other optimization schemes to obtain the optimum. In fact they are not the preferred method when a purely local search of a smooth landscape is required. Rai[5] presents an evolutionary method, based on the method of Differential Evolution[6] (DE), and investigates its strengths in the context of some test problems as well as nozzle and turbine airfoil design. The results of applying a neural network-based response surface method (RSM [7]) to the same design problems are also presented in this study. It was found that DE required about an order of magnitude more computing time than the neural network-based design method. In a more recent article Madavan[8] has explored the possibility of combining DE with local search methods and, utilized the resulting hybrid method in airfoil inverse design. The best variant of these combined methods required 420 function evaluations for this inverse design. In contrast, this inverse design problem required about 50 simulations with a neural-network based algorithm[9]. Here again, the computational cost is about an order of magnitude less. In general where applicable, significant cost reductions can be achieved by using gradient- and RSM-based methods instead of evolutionary algorithms. However, the latter approach is preferred for multi-modal functions and design spaces with disjoint feasible regions. One of the pioneers of evolutionary algorithms (EAs), Scwefel[10], writes with regard to choosing between optimization methods (in particular EAs and local search methods) "…there cannot exist but one method that solves all problems effectively as well as efficiently. These goals are contradictory."

Multiple-objective design optimization is an area where the cost effectiveness and utility of evolutionary algorithms (relative to local search methods) needs to be explored. Deb[11] presents numerous evolutionary algorithms and some of the basic concepts and theory of multi-objective optimization. Mietinnen[12] also presents an excellent survey of the state of the art in multiple-objective optimization. Both these authors provide a large number of references for the interested reader.

The objective here is to introduce a relatively new evolutionary method, Differential Evolution (DE), developed by Price and Storn [6]. In its original version [6], DE was developed for single objective optimization. DE is a population-based method for finding global optima. It is easy to program and use and requires relatively few user-specified constants. These constants are easily determined for a wide class of problems. Fine-tuning the constants will yield the solution to the optimization problem at hand more rapidly. The method can be efficiently implemented on parallel computers and can be used for

*Senior Scientist, Exploration Technology Directorate

continuous, discrete and mixed discrete/continuous optimization problems. It does not require the objective function to be continuous and is noise tolerant. Additionally, the method does not require the transformation of continuous variables into binary integers. The basic method is presented later in the text. Although DE is an effective and efficient global optimization method compared to other evolutionary and genetic algorithms, in general, powerful local search methods continue to be the best choice for locating local optima. Here we also explore the possibility of integrating DE with response surface methodology; the objective being a hybrid design procedure that has the strengths of both methods [5].

Differential evolution can also be used effectively in multiple-objective optimization. Abbas et al.[13] first proposed an extension to DE (PDE) to handle multiple objectives. PDE is a Pareto-based approach that uses non-dominated ranking and selection procedures to compute several Pareto-optimal solutions simultaneously. Madavan[14] presents a different extension to DE to handle multiple objectives. This method is also a Pareto-based approach that uses non-dominated ranking and selection procedures to compute several Pareto-optimal solutions simultaneously. It combines the features of DE and the NSGA-II method of Deb et al.[15].

In more recent articles, Rai[16-17] presents an evolutionary algorithm, based on the method of DE, for multiple-objective design optimization. One goal of this developmental effort was a method that required a very small population of parameter vectors to solve complex multiple-objective problems involving several Pareto fronts (global and local) and nonlinear constraints. Applications of this evolutionary method to some difficult model problems involving the complexities mentioned above are also presented in these articles. The computed Pareto-optimal solutions closely approximate the global Pareto-front and exhibit good solution diversity. Many of these solutions were obtained with small population sizes. Here we present Rai's extension of DE to multiple-objective optimization and apply it to numerous model problems.

Achieving solution diversity and accurate convergence to the exact Pareto front usually requires a significant computational effort with evolutionary algorithms. Here we explore the possibility of using neural networks to obtain estimates of the Pareto optimal front using non-dominated solutions generated by DE as training data. Neural network estimators have the potential advantage of reducing the number of function evaluations required to obtain solution accuracy and diversity, thus reducing cost to design. The estimating curve or surface can be used to generate any desired distribution of Pareto optimal solutions.

SINGLE-OBJECTIVE DIFFERENTIAL EVOLUTION

The single-objective evolutionary algorithm proposed by Rai[5] draws upon ideas from several genetic algorithms and evolutionary methods. One of them is a relatively new member to the general class of evolutionary methods called differential evolution [6]. As with other evolutionary methods and genetic algorithms, DE is a population based method for finding global optima. The three main ingredients are mutation, recombination and selection. Much of the power of this method is derived from a very effective mutation operator that is simple and elegant. Mutations are obtained by computing the difference between two randomly chosen parameter vectors in the population and adding a portion of this difference to a third randomly chosen parameter vector to obtain a candidate vector. The resulting magnitude of the mutation in each of the parameters is different and close to optimal. For example, in the case of an elliptical objective function in two dimensions, the set of all possible mutation vectors would be longer in the direction of the major axis and shorter in the direction of the minor axis. Thus, the mutation operator adapts to the particular objective function and this results in rapid convergence to the optimal value. In addition, this approach automatically reduces the magnitude of mutation as the optimization process converges.

To describe one version of single-objective DE [6], we consider the set of parameter vectors at the nth generation, $\mathbf{X}_{j,n}$. The subscript j refers to the jth member in a population of N parameter vectors and,

$$\mathbf{X}_{j,n} = \left[x_{1,j,n}, x_{2,j,n}, \ldots x_{D,j,n} \right] \qquad \textbf{(1)}$$