Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 345–351
Progress in Natural Science

Special report

A novel evolutionary algorithm for global numerical optimization with continuous variables

Wenhong Zhao [a,b], Wei Wang [a], Yuping Wang [a,*]

[a] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[b] Faculty of Science, Xidian University, Xi'an 710071, China

Received 29 October 2007; received in revised form 19 November 2007; accepted 19 November 2007

Abstract

Evolutionary algorithms (EAs) are a class of general optimization algorithms which are applicable to functions that are multimodal, non-differentiable, or even discontinuous. In this paper, a novel evolutionary algorithm is proposed to solve global numerical optimization with continuous variables. In order to make the algorithm more robust, the initial population is generated by combining determinate factors with random ones, and a decent scale function is designed to tailor the crossover operator so that it can not only find the decent direction quickly but also keep scanning evenly in the whole feasible space. In addition, to improve the performance of the algorithm, a mutation operator which increases the convergence-rate and ensures the convergence of the proposed algorithm is designed. Then, the global convergence of the presented algorithm is proved in detail. Finally, the presented algorithm is executed to solve 24 benchmark problems, and the results show that the convergence-rate of the proposed algorithm is much faster than that of the compared algorithms.

Keywords: Evolutionary algorithm; Decent scale function; Global numerical optimization; Global convergence

1. Introduction

During the past three decades, global optimization problems have been intensively studied in various areas. A number of algorithms for the global optimization problem arise and all these can be divided roughly into two main classes: the determinate method [1,2] and stochastic modeling method [3,4]. In global optimization problems, algorithms may tend to get stuck in local minima, and the convergence-rates of them are usually very low when there are numerous local optima [5].

Evolutionary algorithm (EA) is a kind of global random search methods based on life evolution mechanisms. It contains Genetic Algorithm (GA) [6], Evolutionary Programming (EP) [7], Evolutionary Strategy (ES) [8] and Genetic Programming (GP) [9]. The main features of the EA are swarm exploration and global performance. EA is suitable for the problems with both discrete variables and continuous variables, and does not need to get exact priori knowledge on the problems. In the existing algorithms, EA has received considerable attention regarding its potential to solve complex global optimization problems. However, low convergence-rate and prematurity are also challenging problems for EA.

The crossover operator and mutation operator are the main components to improve the EA's behavior [7,10]. Improvements have been sought in the optimal crossover rates, mutation rates and a more powerful alternative crossover or mutation [11]. In this paper, to enhance the algorithm, a crossover operator which keeps global search when finding descent directions and a mutation operator which balances global exploration and local search are

* Corresponding author. Tel.: +86 15829630075.
E-mail address: ywang@xidian.edu.cn (Y. Wang).

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2007.11.008

designed. Furthermore, to speed up the convergence, when producing the initial population, both determinate method and random method are employed. Based on these operations, a novel evolutionary algorithm for global numerical optimization with continuous variables is proposed.

2. Novel evolutionary algorithm for global optimization

Consider the following global optimization problem:

$$\min_{L\leqslant X\leqslant U} f(X) \tag{1}$$

where $X=(x_1, x_2,\ldots,x_N)^T$ is a variable vector in R^N, N is the dimension of the problem, $f(X)$ is the objective function, and $L=(l_1,\ldots,l_n)^T$ and $U=(u_1,\ldots,u_N)^T$ define the feasible solution space. We denote the domain of x_i by $[l_i, u_i]$, and the feasible solution space by $[L, U]$.

2.1. Initial population

Here each individual is taken as a vector of floating-point numbers, with the same length as that of decision variables. The initial population is generated as follows: $(x_1, x_2,\ldots,x_N)^T$ represents a solution to the optimization problem. *pop* individuals, where *pop* is population size, are produced by the following two algorithms. $1/m$ initial population, where m is a pre-specified number, is produced by Algorithm 1; others are generated by Algorithm 2.

Algorithm 1.

Step 1: Produce a random vector ra in $[0, 1]^N$ uniformly. Let $k=1$.
Step 2: Compute $X=L+(U-L)\times k/\lceil pop/m\rceil+(U-L)\cdot ra/\lceil pop/m\rceil$, $k=k+1$.
Step 3: If $k\leqslant\lceil pop/m\rceil$, go to step 2.

Remark 1. Algorithm 1 divides the domain $[L, U]$ into $\lceil pop/m\rceil$ equal parts, and generates one individual in every part uniformly. Thus, some determinate factors are added into the initial population.

Algorithm 2.

Step 1: Produce a random vector ra in $[0, 1]^N$ uniformly. Let $k=1$.
Step 2: Compute $X=L+(U-L)\cdot ra$, $k=k+1$.
Step 3: If $k\leqslant pop-\lceil pop/m\rceil$, go to step 2.

Remark 2. In Algorithm 2, $(pop-\lceil pop/m\rceil)$ individuals are produced randomly in the searching space.

2.2. Crossover operator

In order to find the decent direction quickly, a decent scale function is introduced in this subsection. And according to the relationship between the population and the decent scale function, a crossover operator is constructed.

The decent scale function is defined as $\hat{y}=f(\widetilde{X})-\delta$, where $\widetilde{X}$ denotes the optimal solution in the population of the current generation. In numerical experiments of Section 4, $\delta=\lambda\mid f(\widetilde{X})\mid$, $\lambda=1/10000$.

Definition 1. For problem (1) and points X, $Y\in[L, U]$, X is better than Y if $f(X)<f(Y)$.

Let $X=(x_1, x_2,\ldots,x_N)^T$ and $Y=(y_1, y_2,\ldots,y_N)^T$ be the crossover parents, t the current generation, $E_{n,1}=(1, 1,\ldots,1)^T$ the unit vector, and g_0 a parameter for generating the temporary offspring. The crossover offspring is produced by Algorithm 3.

Algorithm 3.

Step 1: Let $Z_1=(z_{11}, z_{12},\ldots,z_{1N})^T=\alpha_1X+\beta_1Y$, $Z_2=(z_{21}, z_{22},\ldots,z_{2N})^T=\alpha_2X+\beta_2Y$, where α_1, β_1, α_2, β_2 are real numbers such that z_1 and z_2 are in the feasible solution space. Compute $f(Z_1)$, $f(Z_2)$, $f(X)$, $f(Y)$ and add Z_1, Z_2, X, Y to the temporary offspring set. Let $g=1$, $j=4$, where j denotes the size of the temporary offspring set.
Step 2: Denote the better one between X and Y as $V=(v_1, v_2,\ldots,v_N)^T$. For k=1 to 2,

For $1\leqslant i\leqslant N$, denote the line passing through $(v_i, f(V))$ and $(z_{ki}, f(Z_k))$ as L_1, the line presenting $\hat{y}=f(\widetilde{X})-\delta$ as L_2. If L_1 and L_2 intersect at point $p_{int}=(p_{int}.x, p_{int}.y)$, let $\widetilde{Z}_{j+k,i}=p_{int}.x$. Otherwise, $\widetilde{Z}_{j+k,i}=v_i$. Let the $(j+k)$th temporary offspring $Z_{j+k}=(\widetilde{Z}_{j+k,1},\ldots,\widetilde{Z}_{j+k,N})^T$ and add Z_{j+k} to the temporary offspring set. $j=j+2$.

Step 3: Select crossover offspring in the temporary offspring set. Denote the temporary offsprings better than $\widetilde{X}$ as $Z_1, Z_2,\ldots,Z_q$.

If $q<2$ and $g<g_0$, let $\hat{y}=\hat{y}-\delta$, $g=g+1$, go to step 2; otherwise, choose the best two individuals in the temporary offspring set as the crossover offspring. Stop.

From the above algorithm, the crossover offsprings are always better than the parents.

2.3. Mutation operator

The non-uniform mutation operator is introduced in Ref. [12] is as follows:

$$x'_k=\begin{cases}x_k+(u_k-x_k)r[1-t/T]^b & \text{if random}(0,1)=0\\ x_k-(x_k-l_k)r[1-t/T]^b & \text{if random}(0,1)=1\end{cases}$$

where $X=\{x_1,\ldots,x_N\}^T$ is the mutation parent, $k=1\sim n, X'=(x'_1,\ldots,x'_N)^T$ is the resulting offspring, t is the current generation, T is the maximal number of generations,r is a random number in [0, 1] uniformly, b is a system parameter which determines the dependence degree of