ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220405681

A Robust Archived Differential Evolution Algorithm for Global Optimization Problems

Article *in* Journal of Computers · February 2009

DOI: 10.4304/jcp.4.2.160-167 · Source: DBLP

CITATIONS
3

READS
49

3 authors, including:

Chengen Wang
Northeastern University (Shenyang, China)

105 PUBLICATIONS **966** CITATIONS

SEE PROFILE

All in-text references underlined in blue are linked to publications on ResearchGate, letting you access and read them immediately.

Available from: Chengen Wang
Retrieved on: 09 May 2016

A Robust Archived Differential Evolution Algorithm for Global Optimization Problems

Zhangjun Huang, Cheng-en Wang and Mingxu Ma
MOE Key Laboratory of Process Industry Automation, Northeastern University, Shenyang, China
Email: {zhangjunhuang2000@gmail.com, wangc@mail.neu.edu.cn, mamingxu@ise.neu.edu.cn}

***Abstract*—A robust archived differential evolution algorithm is put forward by means of embedding a flexibility processing operator and an efficiency processing operator based on original DE and ADE. A special constraint-handling mechanism based on dynamic penalty functions and fitness calculation of individuals is adopted in the proposed method to deal with various constraints effectively, which is further extended by means of a flexibility processing operator so as to make it suitable for different type problems, including those with or without constraint(s) and those with continuous, discrete or mixed discrete-continuous variables. Furthermore, an archive of solutions is maintained during the evolutionary process so as to keep the useful information of previous solutions and local optima for the estimation of new solutions. Based on the archive of solutions, an iterative control operator and an efficiency processing operator are designed in the algorithm. The former guides the evolutionary process towards a promising search space, avoiding unnecessary and worthless search. The latter improves the local searching efficiency and the final searching quality. Experimental results based on a suite of six well-known optimization problems reveal that the proposed algorithm is robust, effective, efficient and suitable for different type global optimization problems.**

***Index Terms*—Global optimization, differential evolution, constraint handling, archived solutions, iterative control**

I. INTRODUCTION

Evolutionary computation technique is one of the most important optimization techniques. It has been successfully applied to a wide range of engineering optimization problems. As more and more difficult engineering optimization problems appear, always with objective functions being non-differentiable, non-continuous, non-linear, noisy, and multi-dimensional or having many local minima and complex constraints because of various practical requirements, practicable and effective approaches to solve such problems are becoming unsatisfactory and insufficient. Therefore, more valuable work and research on evolutionary computation techniques for constrained optimization problems are urgent and significant.

Because differential evolution (DE) [1,2] is a simple and powerful population-based stochastic search technique for solving global optimization problems over continuous spaces, and its effectiveness and efficiency have been successfully demonstrated in the last few years through a vast amount of applications [3], it becomes one of the most satisfying methods for solving such engineering problems [4,5]. However, DE method was originally proposed, in principle, to solve unconstrained optimization problems, hence we present a modified differential evolution with constraints handling for constrained optimization problems, i.e. the archived differential evolution (ADE) [6].

Dynamic penalty functions and fitness calculation of individuals for handling linear and non-linear constraints are adopted in ADE, and an archive of solutions is maintained in order that the best information of previous local optimums can be kept for the quality estimate of new solutions in the evolutionary process. Besides, an iterative control operator is designed in ADE based on the archive of solutions, which can make the search process adjusted and guided towards a promising search space.

In our latest work and research on ADE, two operators are designed and introduced into the technique to make it more flexible and efficient for optimization tasks. The one for flexibility processing can make the approach suitable for different type optimization problems, including those with or without constraint(s) and those with continuous, discrete or mixed discrete-continuous variables. The other one for efficiency processing can reduce the quantity of total calculation and improve the quality of final solutions.

This paper is organized as follows. In Section II, a review of related works on optimization and DE techniques is provided. In Section III, a brief description of the original DE algorithm is given. In Section IV, the description of our approach is presented in detail. The experimental design and obtained results are provided in Section V, and some conclusions are established in Section VI.

II. RELATED WORK

Generally, an optimization problem can be formulated as

$$\min f(x)$$
$$s.t.\begin{cases} g_j(X) \le 0, & j = 1,\dots,m, \\ h_k(X) = 0, & k = 1,\dots,n, \\ l_i \le x_i \le u_i, & i = 1,\dots,D, \end{cases} \qquad (1)$$

© 2009 ACADEMY PUBLISHER

where D is the number of design variables, $X=(x_1,x_2,\ldots,x_D)\in R^D$ is the vector of solution, f is the objective function, m and n are the number of inequality and that of equality constraints respectively. g_j and h_k are linear or nonlinear real-value functions respectively. l_i and u_i are the lower and upper bounds of x_i respectively, and they define the whole search space $S\subseteq R^D$.

The optimization problem formulated above is a constrained problem if $m+n$=0, otherwise the problem is an unconstrained one. Considering the constraint(s) of a constrained optimization problem, the feasible region can be defined as

$$F=\left\{X\in S \middle| g_j(X)\le 0 \wedge h_k(X)=0\right\}. \tag{2}$$

Thus solutions of the problem concerned are separated into feasible ones in the feasible region and infeasible ones out the region.

Many methods were originally proposed for unconstrained optimization problems, and were improved later by means of constraint-handling techniques for more difficult constrained optimization problems [7]. Original DE is one of those methods, which has been proposed and generally considered as a reliable, accurate, robust and fast optimization method for unconstrained continuous optimization problems [5] and since then it has attracted much attention and many new versions of it have been proposed and applied to practical optimization problems. Liu and Lampinen [8] reported that the effectiveness, efficiency and robustness of the DE algorithm are sensitive to the settings of the control parameters, and hence introduced a fuzzy adaptive differential evolution algorithm by using fuzzy logic controllers to adapt the search parameters for the mutation operator and crossover operator. Ali and Törn [9] introduced new versions of DE algorithm and suggested some modifications to the classical DE in order to improve its efficiency and robustness. They introduced an auxiliary population of individuals alongside the original population. Sun et al. [10] proposed a combination of DE algorithm and the estimation of distribution algorithm (EDA), which tries to guide the search towards a promising area by sampling new solutions from a probability model. Based on experimental results it has been demonstrated that the DE/EDA algorithm outperforms both DE and EDA algorithms.

In order to simplify the parameter setting, a self-adaptive DE algorithm was proposed by Qin and Suganthan [11], where the choice of learning strategy and the two control parameters F and CR are not required to be pre-defined. During evolution, the suitable learning strategy and parameter settings are gradually self-adapted according to the learning experience. Teo [12] introduced a DE algorithm with a dynamic population sizing strategy called DESAP based on self-adaptation, where two versions of DESAP were implemented using absolute and relative encoding methodologies respectively for dynamically self-adapting the population size parameter. Becerra and Coello [13] proposed a cultural algorithm with a differential evolution population by using different knowledge sources to influence the variation operator of the differential evolution algorithm, and obtained their reported results at a relatively low computational cost with their proposed approach on solving constrained optimization problems.

On constraint-handling techniques, Coello [7] provided a comprehensive survey of the most popular constraint-handling techniques used with evolutionary algorithms (EAs), and each of these approaches is briefly described and discussed, indicating their main advantages and disadvantages. Takahama and Sakai [14] classified EAs for constrained optimization into several categories by the way the constraints are treated: (1) Constraints are only used to see whether a search point is feasible or not; (2) The constraint violation, which is the sum of the violation of all constraint functions, is combined with the objective function; (3) The constraint violation and the objective function are used separately and are optimized separately [15]; (4) The constraints and the objective function are optimized by multi-objective optimization methods [16]. Storn [2] proposed constraint adaptation, in which all constraints of the problem at hand are relaxed so that all individuals in the initial population become feasible, but the approach was not suitable for handling equality constraints. Lampinen [17] proposed another constraint-handling technique as an extension for the DE algorithm for handling nonlinear constraint functions, and stated some rules for the replacement made during the selection procedure, which can be summarized as follows: (1) If both the compared solutions are feasible, the one with lower objective function value is better; (2) Feasible solution is better than infeasible; (3) If both compared solutions are infeasible, the parent is replaced if the new one has lower or equal violation for all the constraints.

In 2000, Runarsson and Yao [18] introduced a stochastic ranking approach as a new constraint-handling technique to balance objective and penalty functions stochastically, and reported that the stochastic ranking approach is capable of improving the search performance significantly. In 2002, Lin et al. [19] used an augmented Lagrangian function with a multiplier updating method to solve constrained problems, where the penalty parameters could be automatically updated so as to obtain a near identical minimum solution despite wide variation in the initial penalty parameters. In 2006, Montes et al. [20] proposed a DE-based approach by allowing each parent to generate more than one offspring and using three selection criteria based on feasibility to deal with the constraints, but the approach was not able to solve problems with a dimensionality higher than 22 and more than 11 nonlinear equality constraints. Zielinski and Laur [21] handled constraints with a modified selection procedure based on a modified selection procedure employed for multi-objective optimization, which does not require additional parameters, but the method failed to reach the best known solutions for four functions of the given 24 test problems.

In summary, all the methods above aimed principally at one special type of constrained or unconstrained optimization problems with continuous or discrete variables. Comparing to the existing research, our

© 2009 ACADEMY PUBLISHER

approach can solve different type problems, including those with or without constraint(s) and those with continuous, discrete or mixed discrete-continuous variables. Furthermore, our approach can find efficiently the global optima with good quality by means of archived solutions and newly designed operators.

III. Differential Evolution

Differential Evolution (DE) [1] is a floating-point encoding evolutionary algorithm for global optimization over continuous spaces. As with all evolutionary optimization algorithms, DE maintains and operates on a population of constant size. It creates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces the parent only if it has better fitness value.

Mutation and crossover operators are two important operators by which DE algorithm with different versions is differentiated and identified. A most popular model of basic differential evolution algorithm is the model "DE/rand/1/bin", which is used throughout this work, where "DE" means differential evolution, "rand" indicates that individuals selected to compute the mutation values are chosen at random, "1" is the number of pairs of solutions chosen and finally "bin" means that a binomial recombination is used.

The model "DE/rand/1/bin" is shown in Fig. 1, where G is the current generation number, $G_{\max}$ is the maximum number of iterations that the algorithm may run, D is the number of decision variables, NP is the population size, CR is the real-valued crossover rate in [0,1], F is the mutation factor in [0,2], and $x_{i,j}$ is the ith decision variable of the jth individual in the population, $rand$[0,1] denotes a uniformly distributed random value in [0,1].

There are three basic components in this approach besides initialization, i.e. mutation, crossover and selection operators. Initial values are selected for control parameters NP, CR and F in the foremost initialization. In addition, the upper and lower bounds for each design variable are also defined, and a random value is selected within its boundaries for each design variable in an individual of the initial population P_0 as follows,

```
Begin
  G=0
  Initialization
  For G=1 to Gmax
    For j=1 to NP
      Select random integers r1≠r2≠r3≠j∈(1,NP)
      Generate a random integer ir∈(1,D)
      For i=1 to D
        If (rand_ij[0,1]<CR or i=ir)
          v_i,j,G+1 = x_i,r1,G + F·(x_i,r2,G - x_i,r3,G)
        Else
          v_i,j,G+1 = x_i,r1,G
        End If
      End For
      If V_j,G+1 is better than X_j,G
        X_j,G+1 = V_j,G+1
      Else
        X_j,G+1 = X_j,G
      End If
    End For
  End For
End
```

Figure 1. Differential evolution algorithm (model "DE/rand/1/bin").

$$x_{i,j,0} = l_i + rand_i[0,1] \times (u_i - l_i), \tag{3}$$

where i=1,…,D, j=1,…,NP, and $rand_i$[0,1] denotes a uniformly distributed random value in [0,1].

A feature of original DE algorithm is that three control parameters NP, CR and F are fixed during the optimization process. However, there still exists a lack of knowledge of how to find reasonably good values for the control parameters of DE for a given function [8]. Another important feature of the DE algorithm is the local criterion of the selection operator, which is efficient and fast.

The efficiency and robustness of the DE algorithm are much more sensitive to the setting of control parameters F and CR than to the setting of NP. The parameter F controls the amplification of differential variations, and the parameter CR controls the probability that a trial vector will be selected form the randomly chosen mutated vector instead of from the current vector. Generally, both F and CR affect the convergence rate and robustness of the search process. Their optimal values are dependent both on objective function characteristics and on NP. Usually, suitable values for F, CR and NP can be found by experimentation after a few tests using different values. Practical advice on how to select values for the three control parameters can be found in [1].

IV. Our Approach

We have put forward an archived differential evolution (ADE) for constrained optimization problems based on original DE, using a special dynamic penalty and fitness function for handling constraints and evaluating individuals, a modified selection and an archive of solutions for keeping the best solutions at any generation, and an iterative control operator for regulating and guiding the evolutionary process [6].

In order to improve the flexibility and efficiency of ADE, we propose a robust archived differential evolution (RADE) algorithm by means of embedding a flexibility processing operator and an efficiency processing operator in ADE algorithm. The flowchart of the improved approach, namely the robust archived differential evolution (RADE) algorithm, is illustrated in Fig. 2.

A. Flexibility Processing Operator

Since that it is effective to use a dynamic penalty and fitness function for handling constraints and evaluating individuals in ADE, we modified this method as little as possible in order to fit it for different type problems without weakening its effectiveness and efficiency.

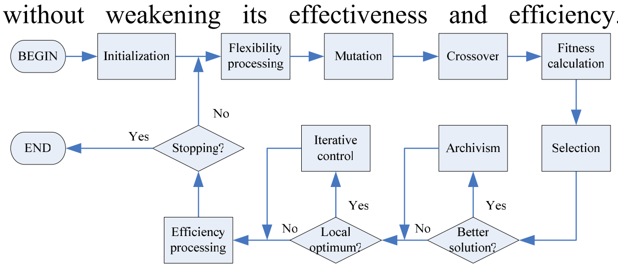

Figure 2. Flowchart of the robust archived differential evolution algorithm.

© 2009 ACADEMY PUBLISHER

So we consider all solutions as feasible ones when the optimization problem under consideration is an unconstrained optimization problem, and then the dynamic penalty function can be formulated as

$$p(X)=\begin{cases}0, & \text{if } m+n=0,\\ \left(C\times G\right)^{\alpha}\times\sum_{j=1}^{m+n}\left(\max\left\{0,g_j(X)\right\}\right)^{\beta}, & \text{otherwise.}\end{cases} \quad (4)$$

where C is a adaptable penalty factor, G is the current generation number, m and n are the number of inequality and equality constraints respectively, α and β are dynamic penalty exponents, g_j denotes inequality constraint(s) for constrained optimization, including those transformed from equality constraints by

$$\left|h_k(X)\right|-\varepsilon\le 0, \quad (5)$$

where $k=\{1,\ldots,n\}$, ε is the tolerance allowed for equality constraints h_k.

The fitness of each individual is calculated in the same way using the following formulation,

$$fitness(X)=\begin{cases}-I_{\mathrm{V}}-\dfrac{f(X)}{f(X)+1}, & \text{if } p(X)>0,\\ -\dfrac{f(X)}{\left|f(X)\right|_{\max}+1}, & \text{otherwise,}\end{cases} \quad (6)$$

where p is the dynamic penalty function, f is the objective function, $|f|_{\max}$ is the maximal objective value for all individuals in the current population, I_{V} is the number of constraints violated for constrained optimization.

As the value of p is equal to zero for all solutions of unconstrained optimization and feasible solutions of constrained optimization, the fitness of relevant individual in the evolutionary population is calculated all the same. As a result, the solutions for unconstrained optimization can be evaluated and selected well, like the feasible solutions for constrained optimization in ADE. Therefore, individuals in the evolutionary populations for constrained and unconstrained optimization can be evaluated and compared with each other by the criterion: The individual with a higher fitness value is better than another one.

Considering discrete or mixed discrete-continuous variables in some optimization problems, a relevant processing module is embedded as a part of the flexibility processing operator, which goes into effect only if there is discrete variable(s). Supposing that x_i is a discrete variable with multiples value of a constant $c(c\neq 0)$, it will be trimmed by the processing module as follows,

$$x_i^{'}=\begin{cases}c\cdot\left\lfloor\dfrac{1}{c}\left(x_i+\left(1+\left\lfloor\dfrac{l_i-x_i}{u_i-l_i}\right\rfloor\right)(u_i-l_i)\right)\right\rfloor, & \text{if } x_i<l_i,\\ c\cdot\left\lceil\dfrac{1}{c}\left(x_i-\left(1+\left\lfloor\dfrac{x_i-u_i}{u_i-l_i}\right\rfloor\right)(u_i-l_i)\right)\right\rceil, & \text{if } x_i>u_i,\\ c\cdot round\left(\dfrac{x_i}{c}\right), & \text{otherwise}\end{cases} \quad (7)$$

where l_i and u_i are the lower and upper bounds of x_i respectively, $x^{'}_i$ is the value of trimmed x_i, *round* denotes a function that returns the nearest integer of a given real number.

Thus, continuous, discrete or mixed discrete-continuous variable(s) can be efficiently generated and calculated altogether.

B. Selection, Archivism and Iterative Control

The selection, archivism and Iterative control operators are same as those in ADE [6], so they are just described briefly here.

The best individual at current generation is found out by means of comparison between individuals according to their fitness value, and then it is compared with the best one in the archive of solutions chosen and stored during the evolutionary process. If it wins, it will be stored as a new best solution in the archive. As a result, the archive keeps all best solutions in previous evolution for the following iterative control.

Iterative control operator is realized based on the selection and archivism operators. When the Euclidian distance between the best individual and the mean individual at current generation is small than a given tolerance ε_d, the best solution of current generation is then considered as one near a local optimum. If solutions are considered as ones near local optimums continuously for a given times, or the archive of solutions is not renewed continuously for another given times, the iterative control operator is performed once.

C. Efficiency Processing Operator

The efficiency processing operator is in fact an efficient local search operator, which takes effect when the best local optimum is obtained. When the operator begins work, the evolutionary population with the best local optimum, denoted by P_{best}, is utilized as the initial population, and the best individual in the current population is protected against mutation.

The pseudo code of the efficiency processing operator is shown in Fig. 3, where *Flag* is a binary variable denoting whether the operator is put into action, P denotes the evolutionary population, I_{best} denotes the best individual in the current population.

D. Parameter Setting

Parameter setting in the improved approach (RADE) is similar to that in ADE, which is listed with some

```
While (G<G_max) and (Flag=Ture)
  P_G = P_best
  For j=1 to NP
    If (rand_j[0,1]<CR) and (j≠I_best)
      Select random integers r1,r2,r3∈(1,NP)
      V_j,G+1 = X_r1,G + F·(X_r2,G - X_r3,G)
    Else
      V_j,G+1 = X_j,G
    End If
    If V_j,G+1 is better than X_j,G
      X_j,G+1 = V_j,G+1
    Else
      X_j,G+1 = X_j,G
    End If
  End For
  G=G+1
End While
```

Figure 3. Pseudo code of the efficiency processing operator.

© 2009 ACADEMY PUBLISHER

suggestions as follows:

(1) Population size *NP* takes a default value of 10×*D*. Larger sizes of population may be adopted, but such values are recommended only for very hard problems if one can afford the extra computational cost.

(2) Maximum number of iterations G_{max} may be set to 100 or 200 generations for easy problems. We suggest setting this parameter to 500 for most problems.

(3) *F* and *CR* can be set following the suggestions in [1]. Their default values in RADE are *F*=0.8 and *CR*=0.9.

(4) Default values for *C*, *α* and *β* in the dynamic penalty function are *C*=0.5, *α*=2 and *β*=2, and the default values for ε and ε_d are both set to 0.001.

V. Numerical Experiments

To validate our approach, we adopted six well-known optimization problems, including two unconstrained problems [22], two constrained problems with continuous variables and two constrained problems with mixed discrete-continuous variables [23]. The main characteristics of the six optimization problems are summarized in Table I, where LI is the number of linear constraints, NI is the number of nonlinear constraints, and ND is the number of discrete variables.

Since our approach is expected to be robust, there is no need for us to spend too much effort in performing a very thorough parameters fine-tuning, and therefore we set the parameters used by our approach for all selected problems as follows: *F*=0.8, *CR*=0.9, *C*=0.5, *α*=2, *β*=2, $\varepsilon=\varepsilon_d$=0.0001, *NP*=10×*D*.

A. Unconstrained Problems

In order to investigate the performance of our approach (RADE) on unconstrained optimization problems and compare it with the dynamic clustering based differential evolution (DCDE) [24], differential evolution (DE) [1], particle swarm optimization (PSO) [25], improved PSO (IPSO) [22] and genetic algorithm (GA) [26] in terms of the accuracy and the frequency of finding optimal solutions within 2000 function evaluations (FES), we performed 30 independent runs for the unconstrained problems P1 and P2 with the maximum number of iterations set as G_{max}=100. Thus the maximum FES is $NP\times G_{max}$=2000 for the two unconstrained problems solved with our approach. The average best function values (AB) found and the success rate (SR) are shown in Table II, and the average function values (Avg. Val.) and their standard deviations (Std. Dev.) over generations are illustrated in Fig. 4. The best, the mean and the worst objective values for P1 are all -2.00000, and for P1 they are all -186.7309. The standard deviations of the best objective values in 30 runs are zeros for both P1 and P2.

It can be seen from Table II that RADE and DCDE outperform any other method in terms of the average best function values and success rate in 30 runs within 2000 function evaluations. Furthermore, Fig. 4 shows that the average best function values found by the proposed method is approaching steadily theoretical global minima along with the increasing generations, and the final optima obtained are very close to theoretical global minima. That is to say, the proposed approach can find solution of unconstrained optimization problems with higher quality than the traditional DE, PSO, IPSO and GA.

TABLE I.
SUMMARY OF SIX OPTIMIZATION PROBLEMS

Prob.	Number of variables	Objective function	LI	NI	ND	Prob. no. in ref.
P1	2	nonlinear	0	0	0	RA in [22]
P2	2	nonlinear	0	0	0	SH in [22]
P3	4	nonlinear	2	5	0	g07 in [23]
P4	3	nonlinear	1	3	0	g08 in [23]
P5	6	quadratic	0	1	3	g06 in [23]
P6	4	nonlinear	3	1	2	Ex.9 in [23]

TABLE II.
THE AVERAGE BEST FUNCTION VALUES AND THE SUCCESS RATE FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

Method	P1		P2	
	AB	***SR***	***AB***	***SR***
RADE	-2.0000	100%	-186.7309	100%
DCDE	-2.0000	100%	-186.7309	100%
DE	-2.0000	98%	-180.7100	78%
PSO	-1.9702	100%	-180.3265	98%
IPSO	-1.9940	98%	-186.7274	100%
GA	-1.9645	84%	-182.1840	98%

B. Constrained Problems with Continuous Variables

The problems P3 and P4 are constrained problems with continuous variables, and 30 independent runs for each of them were performed with G_{max}=200. The results obtained by our approach (RADE) are compared with that obtained by other five state-of-the-art approaches: the archived differential evolution (ADE) [6], the co-evolutionary differential evolution (CDE) [23], GA with

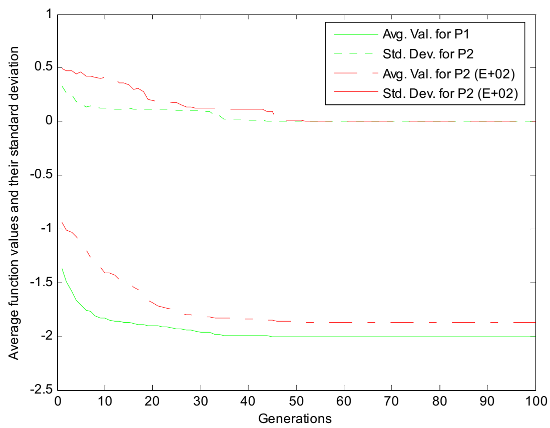

Figure 4. The average function values and their standard deviation over generations for unconstrained optimization problems

© 2009 ACADEMY PUBLISHER

TABLE III.
THE RESULTS BY COMPARISON FOR CONSTRAINED OPTIMIZATION PROBLEMS WITH CONTINUOUS VARIABLES

Method	P3				P4			
	Best	***Mean***	***Worst***	***Std. Dev.***	***Best***	***Mean***	***Worst***	***Std. Dev.***
RADE	1.724852	1.724852	1.724853	1.8711E-07	0.012665	0.012667	0.012680	3.7033E-06
ADE	1.724852	1.724852	1.724856	6.5732E-07	0.012665	0.012665	0.012666	6.9647E-08
CDE	1.733461	1.768158	1.824105	0.022194	0.0126702	0.012703	0.012790	0.000027
GACO	1.748309	1.771973	1.785835	0.011220	0.0127048	0.012769	0.012822	0.000039
GADTS	1.728226	1.792654	1.993408	0.074713	0.0126810	0.0127420	0.012973	0.000059
CPSO	1.728024	1.748831	1.782143	0.012926	0.0126747	0.012730	0.012924	0.000052

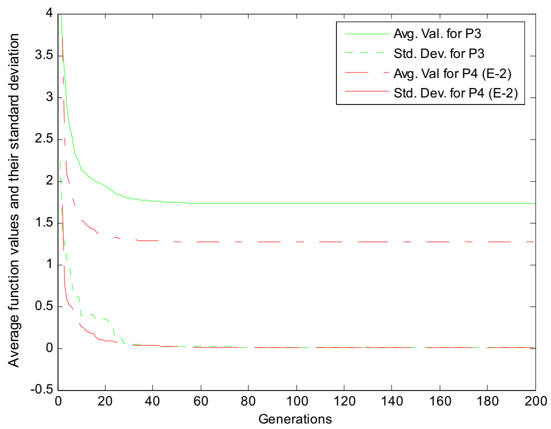

Figure 5. The average function values and their standard deviation over generations for constrained problems with continuous variables.

co-evolution model (GACO) [27], GA with dominance-based tournament selection (GADTS) [28] and co-evolutionary particle swarm optimization (CPSO) [29]. The results by comparison in terms of the standard deviation (Std. Dev.), the best, the mean and the worst objective values are summarized in Table III, and the average function values and their standard deviation over generations are illustrated in Fig. 5.

As shown in Table III, the worst solution and the standard deviation found by RADE for P4 are little worse than that obtained by ADE, but the reverse are true for P3. Therefore, the RADE is similar to ADE on the performance of solving constrained optimization problems with continuous variables, but better than any other method in all terms listed. In addition, it can be seen from Fig. 5 that the average function values decrease rapid at foremost 40 generations, and the standard deviations for both P3 and P4 decrease to almost zero soon afterwards, which means that the proposed method performs well in convergence speed and accuracy for constrained optimization problems with continuous variables.

C. Constrained Problems with Mixed Variables

The problems P5 and P6 are constrained problems with mixed discrete-continuous variables, and 30 independent runs for each of them were carried out with G_{max}=300. The results obtained by our approach (RADE) are compared with that obtained by other six state-of-the-art approaches: the co-evolutionary differential evolution (CDE) [23], the homomorphous mapping (HM) [30], stochastic ranking (SR) [18], GA with co-evolution model (GACO) [27], GA with dominance-based tournament selection (GADTS) [28] and co-evolutionary particle swarm optimization (CPSO) [29]. The results by comparison in terms of the standard deviation (Std. Dev.), the best, the mean and the worst objective values are listed in Table IV, form which it can be seen that the results obtained by RADE is almost perfect for P5, similar to the solutions obtained by CDE, SR and GADTS, but better than that found by HM. The best, the mean and the worst solutions found by RADE for P6 is slightly worse than that by CDE, but obviously better than that by GACO, GADTS and CPSO. Besides, the standard deviation of final solutions obtained by RADE

TABLE IV.
THE RESULTS BY COMPARISON FOR CONSTRAINED OPTIMIZATION PROBLEMS WITH MIXED DISCRETE-CONTINUOUS VARIABLES. NA=NOT AVAILABLE.

Method	P5				P6			
	Best	***Mean***	***Worst***	***Std. Dev.***	***Best***	***Mean***	***Worst***	***Std. Dev.***
RADE	-1.000000	-1.000000	-1.000000	0.000000	6059.714343	6062.795598	6090.526263	9.401573
CDE	-1.000000	-1.000000	-1.000000	0.000000	6059.7340	6058.2303	6371.0455	43.0130
HM	-0.999999	-0.999135	-0.991950	NA	NA	NA	NA	NA
SR	-1.000000	-1.000000	-1.000000	0.000000	NA	NA	NA	NA
GACO	NA	NA	NA	NA	6288.7445	6293.8432	6308.1497	7.4133
GADTS	-1.000000	-1.000000	-1.000000	0.000000	6059.946341	6177.253268	6469.322010	130.929702
CPSO	NA	NA	NA	NA	6061.0777	6147.1332	6363.8041	86.4545

© 2009 ACADEMY PUBLISHER

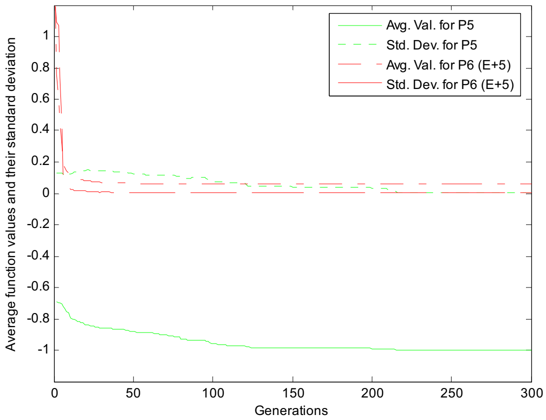

Figure 6. The average function values and their standard deviation over generations for constrained problems with mixed variables.

for P6 in 30 runs is better than that obtained by CDE, GADTS and CPSO, which means the proposed method outperforms the three approaches in stability.

The evolving process of the average function values (Avg. Val.) and their standard deviation (Std. Dev.) over generations are illustrated in Fig. 6, from which it can be seen that RADE can converge to the global optima quickly and steadily for the two constrained optimization problems with mixed discrete-continuous variables.

Moreover, the maximum FES in RADE is $NP \times G_{max}$, which varies with the settings of the population size and the maximum number of generations. Thus the maximum FES is 6×10×300=18000 for P5 and 4×10×300=12000 for P6 by our proposed approach, while the number of FES is 204800 for P5 by CDE in [23] and 200000 for P6 by CPSO in [29].

Based on the above simulation results and comparisons, it can be concluded that RADE is efficient, robust, and suitable for unconstrained or constrained problems with continuous, discrete or mixed discrete-continuous variables.

VI. Conclusion

A Robust archived differential evolution (RADE) for optimization problems was proposed based on the original DE and ADE. The main features of RADE can be summarized as follows: 1) using a new constraint-handling technique based on the dynamic penalty function and the fitness function of individuals to deal with various constraints effectively; 2) maintaining an archive of solutions to make use of the information of previous solutions and local optima for the estimation of new solutions; 3) avoiding unnecessary and worthless search and speeding up the total convergence by taking advantage of an iterative control operation based on the archive of solutions; 4) extending the applicability of this method by means of a flexibility processing operator, which makes the approved method suitable for different type problems, without weakening its effectiveness and efficiency; 5) enhancing the local searching efficiency and improving the final searching quality by means of an efficiency processing operator.

Simulation results based on a set of six well-known optimization problems and comparisons with previously reported results demonstrated the effectiveness, efficiency, applicability and robustness of the proposed RADE. Because of the contribution of archived solutions, the flexibility processing operator, the iterative control operator and efficiency processing operator, RADE performed better than or similar to any other state-of-the-art approaches referred in terms of the searching ability, quality and efficiency.

Despite of the encouraging results with the current test problems, only a few conclusions are justified concerning the effectiveness, efficiency and robustness of the proposed method due to the limited size of the test problem set. As part of our future work, we are considering the application of our method to more engineering problems.

Acknowledgment

This work was supported by a grant from the National Defense Pre-Research Foundation of China (No. 9140A18010106LN0101) and the Doctor Start-up Fund Item of Scientific Office of Liaoning Province of China (No. 20071022).

References

[1] R. Storn, “Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[2] R. Storn, “System design by constraint adaptation and differential evolution,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 1, pp. 22-34, 1999.

[3] A. C. Nearchou and S. L. Omirou, “Differential evolution for sequencing and scheduling optimization,” *Journal of Heuristics*, vol. 12, no. 6, pp. 395-411, 2006.

[4] J. Rönkkönen, S. Kukkonen and K. V. Price, “Real-parameter optimization with differential evolution,” *2005 IEEE Congress on Evolutionary Computation*, Edinburgh, 2005, vol. 1, pp. 506-513.

[5] A. Salman, A. P. Engelbrecht and M. G. H. Omran, “Empirical analysis of self-adaptive differential evolution,” *European Journal of Operational Research*, vol. 183, no. 2, pp. 785-804, 2007.

[6] Zhangjun Huang, Mingxu Ma and Chengen Wang, “A robust archived differential evolution algorithm for constrained global optimization,” *International Conference on Smart Manufacturing Application*, Gyeonggi-do, Korea, 2008, pp. 255-260.

[7] C. A. C. Coello, “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, nos. 11-12, pp. 1245-1287, 2002.

[8] J. Liu and J. Lampinen, “A fuzzy adaptive differential evolution algorithm,” *Soft Computing*, vol. 9, no. 6, pp. 448-462, 2005.

[9] M. M. Ali and A. Törn, “Population set-based global optimization algorithms: some modifications and numerical studies,” *Computers & Operations Research*, vol. 31, no. 10, pp. 1703-1725, 2004.

[10] J. Sun, Q. Zhang and E. P. K. Tsang, “DE/EDA: a new evolutionary algorithm for global optimization,” *Information Sciences*, vol. 169, nos. 3-4, pp. 249-262, 2005.

© 2009 ACADEMY PUBLISHER

[11] A. K. Qin and P. N. Suganthan, “Self-adaptive differential evolution algorithm for numerical optimization,” *2005 IEEE Congress on Evolutionary Computation*, 2005, pp. 1785-1791.
[12] J. Teo, “Exploring dynamic self-adaptive populations in differential evolution,” *Soft Computing*, vol. 10, no. 8, pp. 673-686, 2006.
[13] R. L. Becerra and C. A. C. Coello, “Cultured differential evolution for constrained optimization,” *Computation Methods in Applied Mechanics and Engineering*, vol. 195, nos. 33-36, pp. 4303-4322, 2006.
[14] T. Takahama and S. Sakai, “Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites,” *IEEE Congress on Evolutionary Computation*, Vancouver, 2006, pp. 1-8.
[15] E. Mezura-Montes and C. A. C. Coello, “A simple multimembered evolution strategy to solve constrained optimization problems,” *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 1, pp. 1-17, Feb. 2005.
[16] T. P. Runarsson and X. Yao, “Evolutionary search and constraint violations,” *2003 Congress on Evolutionary Computation*, Piscataway, New Jersey, Dec. 2003, pp. 1414-1419.
[17] J. Lampinen, “A constraint handling approach for the differential evolution algorithm,” *2002 IEEE Congress on Evolutionary Computation*, Piscataway, New Jersey, May 2002, pp. 1468-1473.
[18] T. P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284-294, 2000.
[19] Y. C. Lin, K. S. Hwang and F. S. Wang, “Hybrid differential evolution with multiplier updating method for nonlinear constrained optimization,” *Congress on Evolutionary Computation 2002*, Piscataway, New Jersey, May 2002, pp. 872-877.
[20] E. M. Montes, J. V. Reyes and C. A. C. Coello, “Modified differential evolution for constrained optimization,” *2006 IEEE Congress on Evolutionary Computation*, Vancouver, Canada, July 2006, pp. 25-32.
[21] K. Zielinski and R. Laur, “Constrained single-objective optimization using differential evolution,” *2006 IEEE Congress on Evolutionary Computation*, Vancouver, Canada, July 2006, pp. 223-230.
[22] B. Liu, L. Wang, Y. H. Jin, F. Tang and D. X. Huang, “Improved particle swarm optimization combined with chaos,” *Chaos, Solitons & Fractals*, vol. 25, pp. 1261-1271, 2005.
[23] F. Huang, L. Wang and Q. He, “An effective co-evolutionary differential evolution for constrained optimization,” *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 340-356, 2007.
[24] Y. J. Wang, J. S. Zhang and G. Y. Zhang, “A dynamic clustering based differential evolution algorithm for global optimization,” *European Journal of Operation Research*, vol. 183, pp. 56-73, 2007.
[25] J. Knnedy, R. C. Eberhart and Y. Shi, *Swarm Intelligence*, San Francisco: Morgan Kaufmann Publisher, 2001.
[26] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine learning*, MA: Addison-Wesley, 1989.
[27] C. A. C. Coello, “Use of a self-adaptive penalty approach for engineering optimization problems,” *Computers in Industry*, vol. 41, no. 2, pp. 113-127, 2000.
[28] C. A. C. Coello and E. M. Montes, “Constraint handling in genetic algorithms through the use of dominance-based tournament selection,” *Advanced Engineering Informatics*, vol. 16, no. 3, pp. 193-203, 2002.
[29] Q. He and L. Wang, “An effective co-evolutionary particle swarm optimization for constrained engineering design problems,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 1, pp. 89-99, 2007.
[30] S. Koziel and Z. Michalewicz, “Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 19-44, 1999.

Zhangjun Huang is currently a Ph.D. candidate in the MOE Key Laboratory of Process Industry Automation, Northeastern University, Shenyang, China. He received his B.S. degree in Marine Engineering in 2000 from Dalian Maritime University, Dalian, China, and received his M.S. degree in Computer Science and Technology in 2006 from Yanshan University, Qinhuangdao, China. His current research interests focus on system engineering, multidisciplinary design optimization, optimal algorithms, computer aided design and computer aided engineering.

Cheng-en Wang received his Ph.D. degree in 1991 from Nanjing University of Aeronautics and Aerospace, Nanjing, China. He formerly worked at Shenyang Institute of Automation, Chinese Academy of Sciences. Now, he is a professor at Northeastern University, Shenyang, China, and his research interests include advanced manufacturing technologies, information integration, and enterprise management.

Mingxu Ma received Ph.D. degree from Shenyang Institute of Automation, Chinese Academy of Sciences in 2006 and now is a lecturer at Northeastern University, Shenyang, China. His current research interests include multidisciplinary design optimization and product lifecycle modeling.

© 2009 ACADEMY PUBLISHER