An Adaptive Differential Evolution Algorithm and its Performance on Real World Optimization Problems

Md.Asafuddoula
School of Engineering & I.T
University of New South Wales at the
Australian Defence Force Academy,
CANBERRA BC ACT 2610
Email: Md.Asaf@student.adfa.edu.au
Telephone: (+61) 2 626 88479

Tapabrata Ray
School of Engineering & I.T
University of New South Wales at the
Australian Defence Force Academy,
CANBERRA BC ACT 2610
Email: t.ray@adfa.edu.au
Telephone: (+61) 2 626 88248
Fax: (+61) 2 626 88581

Ruhul Sarker
School of Engineering & I.T
University of New South Wales at the
Australian Defence Force Academy,
CANBERRA BC ACT 2610
Email: r.sarker@adfa.edu.au
Telephone: (+61) 2 626 88051
Fax: (+61) 2 626 88581

***Abstract*—Real world optimization problems are challenging as they often involve a large number of variables and highly nonlinear constraints and objective functions. While a number of efficient optimization algorithms and numerous mathematical benchmark test functions have been introduced in recent years, the performance of such algorithms have rarely been studied across a range of real world optimization problems. In this paper, we introduce an improved adaptive differential evolution (DE) algorithm and report its performance on the newly proposed real world optimization problems. The proposed differential evolution algorithm incorporates adaptive parameter control strategies; a center based differential exponential crossover and hybridization with local search to improve its efficiency. While comprehensive results of other algorithms on the test problems are unavailable at this stage, our preliminary comparison with published results indicates promising performance of the proposed DE across the range of problems.**

I. INTRODUCTION

Differential Evolution (DE) has proven to be a promising candidate for optimization of real- valued, multi-modal objective functions [1]. Several recent reports have highlighted excellent performance of DE on benchmark functions [2], [3]. A search made through SciVerse Scopus using the keyword *Differential Evolution Optimization* identified 1,199 relevant articles published between 1996 and 2010. The steep increase in the number of publications related to DE can be observed from Figure 1. Differential evolution has been shown to be a efficient algorithm for solving constrained optimization as well as unconstrained optimization problems [4], [5]. DE has been successfully applied to multimodal and multi-objective optimization problems in the past with adaptive crossover rate and mutation factor [6], [7]. While the performance of an algorithm is always dependent on its constituent mechanisms and their corresponding control parameters, a robust algorithm demands the need for adaptive parameter control. The proposed DE adaptively controls the crossover rate based on its probability of success. In terms of crossover, various schemes have been proposed over the years, varying both in terms of parent selection strategies and the nature of distribution of the offsprings. The present algorithm generates an offspring based on center based differential exponential crossover around the geometric center of the parent vectors while its mutation factor is controlled using the scheme proposed in [5]. The algorithm also maintains an active and an archive population to strike a balance between exploitation and exploration.

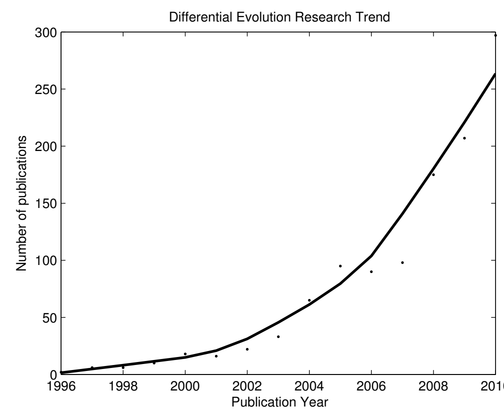


Fig. 1. Differential Evolution Research Trend

This paper is structured as follows. Section II provides an overview of the related work. The proposed DE algorithm with center based differential exponential crossover and adaptive crossover mechanism is presented in Section III. The performance of the algorithm is presented in Section IV while Section V concludes the paper with final remarks.

II. BACKGROUND

Genetic annealing proposed by K.Price [8] led to the birth of Differential Evolution(DE) as we know of it today. While in its original form, the algorithm was a means to solve combinatorial optimization problems, it attracted serious research

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

attention in 1997 when it turned out to be a winner in the Second International Contest on Evolutionary Optimization. The key distinguishing feature of DE as compared to other population-based techniques is its differential mutation [9]. The initial set of strategies realizing differential mutation was proposed by Storn and Price [9], [10]. The first attempt to guide differential mutation was presented by Price [9],where *semi directed* mutation was realized by special pre-selection operator. Several enhancements in operators have been proposed over the years which include combination of differential mutation with arithmetic crossover, triangle mutation scheme [11], alternations of weighted directed strategy [12] etc. While recommended values for control parameters (mutation and crossover parameters) have been suggested in literature [1], ample studies have shown that the performance is affected based on their choice. To alleviate the problems of apriori selection of control parameters, adaptive strategies have been introduced [13].

Ali and Torn in [13] proposed a new version of DE algorithm with an aim to improve the efficiency and robustness of the algorithm through adaptive control of crossover rate. Since then, a number of attempts have been made along similar lines such as those by Sun et al. [14] using a probability model,Qin and Suganthan in [15] using Self-adaptive Differential Evolution algorithm (SaDE) and hybridization of local search as in *Accelerating Differential Evolution* [16] and *gradient based mutation* as in [5]. A significant number of these attempts involved expensive parameter tuning (crossover rate and mutation factor) during a learning phase which is an impediment for real world applications.

The details of the proposed DE along with its components are described in the following section.

III. Improved Adaptive DE Algorithm

The proposed algorithm follows the generic structure of a differential evolution scheme. A geometric center based exponential crossover (CBDEX), adaptive parameter control strategies and a local search is embedded within the algorithm. The algorithm maintains an active population and an archive of solutions in an attempt to maintain the balance between exploitation and exploration. The algorithm creates a set of *M* individuals which is used to initialize the archive. Top *N* individuals of the archive is copied and maintained in the active population. In an attempt to maintain diversity, two parents are selected from the active population randomly, while the third parent is selected from the archive. Offspring solutions are created using CBDEX and compared with the solutions in the active population sequentially for a possible replacement. If the offspring is unable to make it into the active population, it replaces a random solution in the archive. A local search is periodically invoked from an offspring solution based on a user defined probability and the dimensionality of the problem. The local search is expected to inject good quality solutions into the active population which in turn could be selected as potential parents. The replacement scheme is analogous to steady state models of evolutionary computation.

The algorithm starts off with a predefined set of crossover rates (4 values selected in the study) with equal probability. The number of offsprings which make it to the active population serves as an indicator of the success of the corresponding crossover rate. Based on this success rate, the corresponding probability of the crossover rate is controlled whereby; promising crossover rates are preferred in subsequent generations. The pseudocode of the algorithm is presented in Algorithm 1, while the constituent components are described in sub-sections.

Algorithm 1 Improved Adaptive DE Algorithm

Require: GEN_{max}\{Number of Max Generation\}
Require: M\{Size of Archive Population\}
Require: N\{Size of Active Population\}

$Archive\_Pop = initialize()$
$Evaluate(Archive\_Pop)$
$Sort(Archive\_Pop)$
Copy Best *N* to *Active_Pop*
$CR_0 = \{0.85, 0.90, 0.95, 0.98\}$
$PR_0 = \{0.25, 0.50, 0.75, 1\}$
while $((gen \leq GEN_{max})||(feval \leq feval_{max}))$ **do**
  $CR = CR_{selection}()$
  for *i=1:N* **do**
    for *k=1:2* **do**
      $\vec{X}_i^{new} = CBDEX()$
      if $(gen\%D == 0)\&\&(rand[0, 1] \leq 0.2)$ **then**
        $Gradient_{search}(\vec{X}_i^{new})$
      end if
      if $f(\vec{X}_i^{new}) \leq f(Active\_Pop_i)$ **then**
        $Active\_Pop_i = \vec{X}_i^{new}$
      else
        $Archive\_Pop_j = \vec{X}_i^{new}$;$j = rand[N, M]$
      end if
    end for
  end for
  $PR = PR_{update}()$
end while

A. Center Based Differential Exponential Crossover(CBDEX):

Center Based Exponential Crossover (CBDEX) is the process of generating the offspring based on the geometric center of two vectors. CBDEX uses the merits of exponential crossover with the changes of center difference between two vectors. Mutation as used for offspring generation is performed as follows.

$$x_i^{new} = \vec{x_i} + F(\vec{o_w} - \vec{o}) \quad (1)$$

Where, $\vec{o_w} = \sum_{j=1}^{q} w_j \vec{x_j}$ is the weighted parent center $w_j \in [0, 1] (j = 1, 2, ..., q)$ where $q \geq 2$ and $\sum_{j=1}^{q} w_j = 1$. $\vec{o}$ can be defind as $\vec{o} = 1/q \sum_{j=1}^{q} \vec{x_j}$ and F is a random number in between 0 to 2.

In this algorithm, two parents are used to find the weighted parent center. Here, *q=2* and the parents are $\vec{x_j}$ and $\vec{x_k}$ with the defined weights *w1=0* and *w2=1*.

$$\vec{o_w} - \vec{o} = \vec{x_j} - (\vec{x_j} + \vec{x_k})/2 = (\vec{x_j} - \vec{x_k})/2$$

For the mutation process of this algorithm three parents $\vec{x_i}$,$\vec{x_j}$ and $\vec{x_k}$ are used.Thus, Equation 1 can be written as follows.

$$x_i^{new} = \vec{x_i} + F(\vec{x_j} - \vec{x_k})/2 \tag{2}$$

The pseudocode of the algorithm for CBDEX is presented in Algorithm 2.

Algorithm 2 Center Based Differential Exponential Crossover (CBDEX)

Require : x{Individual solution}

$j = rand(1, D), l = 0;$
while $(l \leq D)\&\&(rand(0,1) < CR)$ **do**
  $x_j^{new} = \vec{x_{1j}} + F(\vec{x_{2j}} - \vec{x_{3j}})/2;$
  $l = l + 1; j = (j + 1)\%D;$
end while
while $l \leq D$ **do**
  $x_j^{new} = x_j;$
  $l = l + 1; j = (j + 1)\%D;$
end while

B. Adaptive Parameter Control :

The adaptive parameter control for crossover rate is introduced here with a set of crossover rates, $CR = \{0.85, 0.90, 0.95, 0.98\}$. The probabilities of the crossover rates are set to 0.25 initially which is subsequently adapted based on their success measure. The success measure of a particular crossover rate is defined as the ratio of the offsprings making their way into the active population and the number of offspring created.

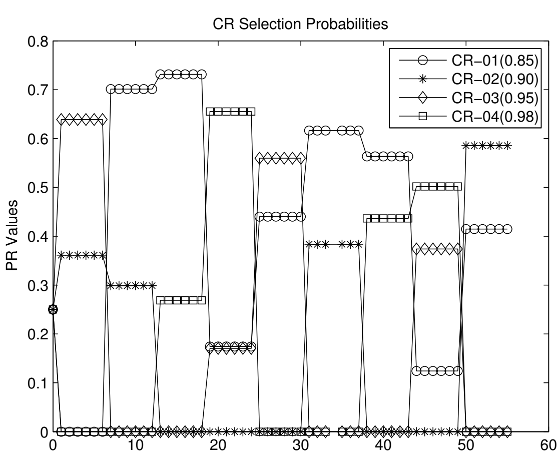


Fig. 2. CR Selection Probability of problem TP10.1 for run-1.

$$CR_{i,G+1} = \begin{cases} CR_{i,G+1}, & PR = rand(0,1), \\ & if PR > PR_t \; and \; PR \leq PR_{t+1}; \\ & Where \; t \; denotes \; the \; set \\ & of \; possible \; crossover \; rates. \\ CR_{i,G}, & \text{otherwise} \end{cases}$$

In the Figure 2 the progression of the probabilities of the prescribed crossover rates $CR = \{0.85, 0.90, 0.95, 0.98\}$ are presented for an instance of the TP10.1 (DED instance 1). While they all have same initial probabilities, the probability of each changes depending on the success of offsprings.

Algorithm 3 Algorithm for $CR_{selection}$

Require: n {Possible Crossover Rates}
Require: $CR = \{0.85, 0.90, 0.95, 0.98\}$
Require: $PR = \{0.25, 0.50, 0.75, 1\}$

$rs = rand(0,1)$
for *i=1:n* **do**
  if $rs > PR_i \&\& rs \leq PR_{i+1}$ **then**
    $CR = CR_i$
  end if
end for

Algorithm 4 Algorithm for PR_{update}

Require: t {Possible Crossover Rates}
Require: SuccessEval {Number of Success Evaluation}
Require: TotalEval {Total Number of Evaluation}

for *i=1:t* **do**
  $SuccessRate_i = SuccessEval_i / TotalEval_i$
end for
for *i=1:t* **do**
  $PR_i = SuccessRate_i / Sum(SuccessRate)$
end for

C. Gradient Local Search :

Gradient based local search used in this study is based on sequential quadratic programming (SQP). Incorporation of a local search always requires a careful consideration, as its too frequent use could turn out to be expensive while on the other hand it offers the possibility of injecting good quality of solutions into the population. In the proposed algorithm, a local search is invoked periodically based on the dimensionality of the problem and a user defined probability.

IV. Experimental Results

Of the fourteen test problems (TP01-TP14), TP01-TP03,TP05,TP06,TP09, and TP11-TP12 has bound constraints. Other 5 problems i.e. TP07, TP08, TP10, TP11, TP12 have inequality constraints and the remaining two have equality constraints. All results are reported for $50,000$, $100,000$, and $150,000$ function evaluations. Since the test problems are newly introduced in CEC-2011, there is no scope for a detailed comparison of performance between various optimization algorithms. It is also important to highlight that no external parameter tuning experiments were conducted to identify the most appropriate values for the parameters listed below.

A. Experimental Conditions

Twenty five independent runs were performed with randomly initialized populations.

Tables IV to VII summarize the experimental results on problems TP-01 to TP-14 respectively. In the tables, the best, median, worst, average function values and standard deviation are shown when FEs are 5.0×10^4, 1.0×10^5 and 1.5×10^5. Figure 3 shows the convergence plot of problems TP01,TP02,TP10.1, and TP10.2 for first run.

TABLE I
PROBLEM DESCRIPTIONS

Problem No.	Description
TP01	Parameter Estimation of FM Sound Waves
TP02	Lennard-Jones Potential
TP03	Bi-functional Catalyst Blend Optimal Control
TP04	Optimal Control of a Non-Linear Stirred Tank Reactor
TP05.1	Tersoff Potential Function Min. Problem - Si(B)
TP05.2	Tersoff Potential Function Min. Problem - Si(C)
TP06	Spread Spectrum Radar Polly phase Code Design
TP07	Transmission Network Expansion Planning
TP08	Large Scale Transmission Pricing
TP09	Circular Antenna Array Design
TP10.1	Dynamic Economic Dispatch - Instance 1
TP10.2	Dynamic Economic Dispatch - Instance 2
TP11.1	Static Economic Load Dispatch - Instance 1
TP11.2	Static Economic Load Dispatch - Instance 2
TP11.3	Static Economic Load Dispatch - Instance 3
TP11.4	Static Economic Load Dispatch - Instance 4
TP11.5	Static Economic Load Dispatch - Instance 5
TP12.1	Hydrothermal Scheduling - Instance 1
TP12.2	Hydrothermal Scheduling - Instance 2
TP12.3	Hydrothermal Scheduling - Instance 3
TP13	Messenger: Spacecraft Trajectory Optimization
TP14	Cassini 2: Spacecraft Trajectory Optimization

Parameters Setting:

a) The dynamic ranges of the control parameters are listed in table II and the corresponding parameter values used are listed in table III.

TABLE II
DYNAMIC PARAMETER RANGES

Control Parameter	Dynamic Ranges
Archive size (M)	[2D-100D]
Active Population size (N)	[1D- 4D]
Scaling factor (F0)	[0.1-0.5]
Crossover rate (CR0)	[0.8-0.98]

e) Actual parameter values are provided in table III.

TABLE III
ACTUAL PARAMETER VALUES

Control Parameter	Actual values
Archive size (M)	if (D $>$ 40) M=2D else M=100D
Active Population size (N)	if (D $>$ 40) M=1D else M=4D
Scaling factor(F)	0.5
Crossover rate(CR)	[0.85,0.90,0.95,0.98] Adaptive

V. CONCLUSIONS

In this paper, an improved adaptive differential evolution algorithm has been introduced. The proposed differential evolution algorithm incorporates adaptive parameter control strategies; a center based differential exponential crossover and hybridization with local search to improve its efficiency. The exponential crossover mechanism has been embedded within the center based crossover scheme. The performance of the algorithm is presented across the range of 14 real world optimization problems for $50,000$,$100,000$ and $150,000$ function evaluations. While comprehensive results of other algorithms on the test problems are unavailable at this stage, our preliminary comparison with published results indicates promising performance across the range of problems. It is important to highlight that although some of the problems involved constraints, the proposed DE did not make use of any special constraint handling mechanisms to deal with them in the spirit of black-box fitness functions implicitly considering the constraints. Attempts to further improve the performance of the algorithm via an ensemble of crossover variants is currently being investigated.

REFERENCES

[1] R.Storn, "On the Usage of Differential Evolution for Function Optimization," *Fuzzy Information Processing Society*, vol. 1996, NAFIPS. Biennial Conference of the North American.

[2] J. Vesterstram and R.Thomsen, "A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems," vol. 2: ISBN: 0-7803-8515-2, 19-23 June 2004.

[3] K.Hammouche, M.Diaf, and P.Siarry, "A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem," *Advances in metaheuristics for hard optimization*, vol. 23, Issue 5, pp. 676–688, August 2010.

TABLE IV
FUNCTION VALUES ACHIEVED WHEN FEs=50000,FEs=100000,AND EFs=150000 FOR TEST PROBLEM TP-01 TO TP-05

FEs		TP01	TP02	TP03	TP04	TP05.1	TP05.2
5.0×10^4	Best	**0.000000E+00**	**-2.798360E+01**	**1.151489E-05**	**1.432911E+01**	**-3.019122E+01**	**-2.285357E+01**
	Median	0.000000E+00	-2.593278E+01	1.151489E-05	2.081992E+01	-2.805588E+01	-2.000870E+01
	Worst	1.992078E+01	-1.889360E+01	1.151489E-05	2.153960E+01	-2.700942E+01	-1.814074E+01
	Mean	4.852582E+00	-2.523095E+01	1.151489E-05	1.831217E+01	-2.834573E+01	-2.020391E+01
	Std	6.690016E+00	2.109179E+00	3.804327E-19	3.254783E+00	9.381209E-01	1.323855E+00
1.0×10^5	Best	**0.000000E+00**	**-2.842253E+01**	**1.151489E-05**	**1.432911E+01**	**-3.272503E+01**	**-2.687879E+01**
	Median	0.000000E+00	-2.747767E+01	1.151489E-05	2.081992E+01	-3.145192E+01	-2.429244E+01
	Worst	1.752078E+01	-1.830315E+01	1.151489E-05	2.153960E+01	-2.900694E+01	-2.128983E+01
	Mean	4.052582E+00	-2.678212E+01	1.151489E-05	1.831217E+01	-3.125792E+01	-2.443067E+01
	Std	5.790016E+00	2.278416E+00	3.804327E-19	3.254783E+00	8.462227E-01	1.340860E+00
1.5×10^5	Best	**0.000000E+00**	**-2.842253E+01**	**1.151489E-05**	**1.432911E+01**	**-3.684537E+01**	**-2.916612E+01**
	Median	0.000000E+00	-2.719010E+01	1.151489E-05	2.081992E+01	-3.410766E+01	-2.742977E+01
	Worst	1.702078E+01	-1.820789E+01	1.151489E-05	2.153960E+01	-3.148397E+01	-1.951175E+01
	Mean	3.852582E+00	-2.678273E+01	1.151489E-05	1.831217E+01	-3.381816E+01	-2.583055E+01
	Std	5.690016E+00	1.974757E+00	3.804327E-19	2.915478E+00	1.434399E+00	2.991160E+00

TABLE V
FUNCTION VALUES ACHIEVED WHEN FEs=50000,FEs=100000,AND EFs=150000 FOR TEST PROBLEM TP-06 TO TP-10

FEs		TP06	TP07	TP08	TP09	TP10.1	TP10.2
5.0×10^4	Best	**5.000000E-01**	**2.200000E+02**	**5.710000E+02**	**-2.012749E+01**	**2.683505E+05**	**8.206118E+06**
	Median	-5.000000E-01	2.200000E+02	6.425000E+02	-1.988024E+01	3.074734E+05	9.152269E+06
	Worst	-5.000000E-01	2.200000E+02	9.920000E+02	-1.327656E+01	3.503932E+05	9.673434E+06
	Mean	-5.000000E-01	2.200000E+02	5.780000E+02	-2.011583E+01	3.066656E+05	9.119174E+06
	Std	0.000000E+00	0.000000E+00	5.690016E+00	1.711543E+00	1.941239E+04	3.838030E+05
1.0×10^5	Best	**5.000000E-01**	**2.200000E+02**	**4.628560E+01**	**-2.098991E+01**	**1.501997E+05**	**6.109888E+06**
	Median	5.000000E-01	2.200000E+02	8.500000E+01	-2.094802E+01	1.681557E+05	7.152468E+06
	Worst	5.000000E-01	2.200000E+02	9.000000E+01	-1.399656E+01	1.830202E+05	7.583435E+06
	Mean	5.000000E-01	2.200000E+02	1.092987E+02	-2.095834E+01	1.676233E+05	9.119174E+06
	Std	0.000000E+00	0.000000E+00	5.690016E+00	1.895431E+00	9.422916E+03	2.857040E+05
1.5×10^5	Best	**5.000000E-01**	**2.200000E+02**	**7.585491E+00**	**-2.180845E+01**	**5.017060E+04**	**1.078526E+06**
	Median	5.000000E-01	2.200000E+02	4.730413E+01	-2.148024E+01	5.282237E+04	1.086166E+06
	Worst	5.000000E-01	2.200000E+02	1.159561E+02	-1.497656E+01	7.127143E+04	1.094758E+06
	Mean	5.000000E-01	2.200000E+02	1.930413E+01	-2.095834E+01	5.418152E+04	1.086684E+06
	Std	0.000000E+00	0.000000E+00	5.690016E+00	1.895431E+00	4.866964E+03	4.455655E+03

TABLE VI
FUNCTION VALUES ACHIEVED WHEN FEs=50000,FEs=100000,AND EFs=150000 FOR TEST PROBLEM TP-11

FEs		TP11.1	TP11.2	TP11.3	TP11.4	TP11.5
5.0×10^4	Best	**1.544623E+04**	**1.839204E+04**	**3.281980E+04**	**1.348142E+05**	**2.437214E+06**
	Median	1.545043E+04	1.858793E+04	3.293484E+04	1.461288E+05	2.462936E+06
	Worst	1.548125E+04	1.879731E+04	3.303703E+04	1.546566E+05	3.908069E+06
	Mean	1.545222E+04	1.859426E+04	3.293601E+04	1.460195E+05	2.837523E+06
	Std	7.778997E+00	1.058295E+02	4.823777E+01	5.461530E+03	3.431656E+05
1.0×10^5	Best	**1.544555E+04**	**1.811679E+04**	**3.278082E+04**	**1.378465E+05**	**2.137214E+06**
	Median	1.545044E+04	1.836620E+04	3.287634E+04	1.438524E+05	2.517835E+06
	Worst	1.547703E+04	1.860062E+04	3.301658E+04	1.544642E+05	3.197061E+06
	Mean	1.545398E+04	1.832634E+04	3.288677E+04	1.450264E+05	2.503521E+06
	Std	7.645967E+00	1.279299E+02	5.792179E+01	4.477218E+03	3.411256E+05
1.5×10^5	Best	**1.544538E+04**	**1.822472E+04**	**3.274420E+04**	**1.240690E+05**	**1.890671E+06**
	Median	1.547957E+04	1.856815E+04	3.285861E+04	1.252703E+05	1.924213E+06
	Worst	1.556846E+04	1.878500E+04	3.305473E+04	1.274304E+05	1.982961E+06
	Mean	1.548207E+04	1.855026E+04	3.285940E+04	1.255323E+05	1.925098E+06
	Std	3.047366E+01	1.415147E+02	6.841365E+01	8.362557E+02	2.342904E+04

[4] J.Brest, S. Greiner, B.Boskovic, M.Mernik, and V.Zumer, “Self adapting Control Parameters in Differential Evolution:A Comparative study on Numerical Benchmark Problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10(6), pp. 646–657, 2006.

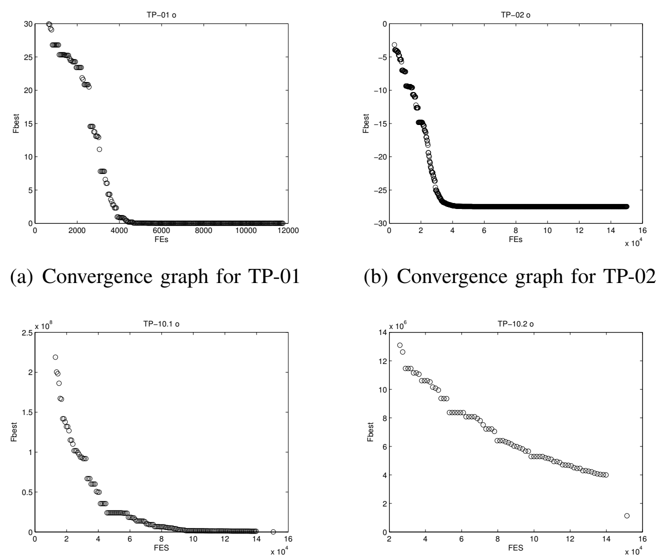
[5] T. Takahama and S. Sakai, “Constrained optimization by ϵ constrained particle swarm optimizer with ϵ level control,” *in Proc. of the 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST05)*, pp. 1019–1029, May 2005.

[6] M. Srinivas and L. M. Patnaik, “Adaptive Probabilities of Crossover and Mu tation in Genetic Algorithms,” *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETIC*, vol. 24, NO. 4, APRIL 1994.

[7] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F.

TABLE VII
FUNCTION VALUES ACHIEVED WHEN FEs=50000,FEs=100000,AND EFs=150000 FOR TEST PROBLEM TP-12 TO TP-14

FEs		TP12.1	TP12.2	TP12.3	TP13	TP14
5.0×10^4	Best	**1.480446E+06**	**1.861558E+06**	**1.544530E+06**	**1.787373E+01**	**9.117683E+00**
	Median	2.152676E+06	3.131885E+06	2.261493E+06	2.125791E+01	1.472799E+01
	Worst	3.040902E+06	4.400451E+06	2.855889E+06	2.494530E+01	2.297971E+01
	Mean	2.175455E+06	3.105775E+06	2.204170E+06	2.119566E+01	1.519891E+01
	Std	4.359290E+05	5.457313E+05	3.862200E+05	1.640632E+00	3.421207E+00
1.0×10^5	Best	**1.163269E+06**	**1.605539E+06**	**1.212969E+06**	**1.389178E+01**	**8.780862E+00**
	Median	1.512639E+06	1.958127E+06	1.452102E+06	1.621140E+01	1.349985E+01
	Worst	1.896863E+06	2.259261E+06	1.715713E+06	2.050951E+01	1.989954E+01
	Mean	1.502697E+06	1.942472E+06	1.470997E+06	1.679767E+01	1.336922E+01
	Std	1.699322E+05	1.621776E+05	1.418318E+05	1.779448E+00	3.114089E+00
1.5×10^5	Best	**9.248215E+05**	**9.283227E+05**	**9.271454E+05**	**1.239837E+01**	**8.621241E+00**
	Median	9.299678E+05	9.346668E+05	9.307335E+05	1.388321E+01	1.279869E+01
	Worst	9.428696E+05	1.784064E+06	9.397024E+05	2.236330E+01	1.640074E+01
	Mean	9.306026E+05	9.929643E+05	9.305288E+05	2.119566E+01	1.253713E+01
	Std	3.502770E+03	1.964245E+05	3.063813E+03	5.906079E+00	2.466385E+00



(a) Convergence graph for TP-01 (b) Convergence graph for TP-02

(c) Convergence graph for TP-10.1 (d) Convergence graph for TP-10.2

Fig. 3. Convergence graphs are shown for run-1.

Tasgetiren, “Differential evolution algorithm with ensemble of parameters and mutation strategies,” *Applied Soft Computing,DOI:10.1016/j.asoc.2010.04.024*, vol. 11 No. 2, pp. 1679–1696, MAR 2011.

[8] K. Price and R. Storn, “Differential Evolution: Numerical Optimization Made Easy,” *Dr. Dobb’s Journal*, pp. 18 – 24, April 97.

[9] R. Storn and K. Price, “Differential evolution A simple and efficient adaptive scheme for global optimization over continuous spaces,” *Technical report TR-95-012,International Computer Science Institute, Berkeley,CA*, 1995.

[10] R. Storn, “System Design by Constraint Adaptation and Differential Evolution. Evolutionary Computation,” *IEEE Transactions on, 3(1)*, pp. 22–34, Apr 1999.

[11] H.-Y. Fan and J. Lampinen, “A trigonometric mutation approach to differential evolution,” *In K.C. Giannakoglou,D.T.Tsahalis,J. periaux,K.D. Papailiou ,and T.Fogarty, editors, Evolutionary methods for Design Optimization and Control with applications to Industrial Problems,Athens,Grece. International Centre for Numerical Methods in Engineering (Cmine)*, pp. 65–70, 2001.

[12] ——, “A directed mutation operation for the differential evolution algorithm,” *International Journal of Industrial Engineering: Theory, Applications and practice*, vol. 1(10), pp. 6–15, 2003.

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

[14] J. Sun, Q. Zhang, and E. P. Tsang, “DE/EDA: A new evolutionary algorithm for global optimization,” *Information Sciences 169*, pp. 249–262, 2005.

[15] A.K.Qin and P.N.Suganthan, “Differential evolution algorithm with the strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13(2), pp. 398–417.

[16] N. Noman and H. Iba, “Accelerating Differential Evolution Using an Adaptive Local Search,” *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 12, NO. 1, FEBRUARY 2008.