Journal of Universal Computer Science, vol. 18, no. 13 (2012), 1853-1870
submitted: 15/11/11, accepted: 30/3/12, appeared: 1/7/12

Two Local Search Strategies for Differential Evolution

Musrrat Ali
(Department of Computer Engineering, Sungkyunkwan University
Suwon- 440746, Republic of Korea
musrrat.iitr@gmail.com)

Millie Pant
(Department of Paper Technology, Indian Institute of Technology Roorkee
Saharanpur Campus, Saharanpur- 247001, India
millifpt@iitr.ernet.in)

Atulya K Nagar
(Department of Computer Science, Liverpool Hope University, Hope Park Liverpool
L16 9JD, UK
nagara@hope.ac.uk)

Chang Wook Ahn
(Department of Computer Engineering, Sungkyunkwan University
Suwon- 440746, Republic of Korea
cwan@skku.edu)

Abstract: Insertion of a local search technique is often considered an effective mechanism to increase the efficiency of a global optimization algorithm. In this paper we propose and analyze the effect of two local searches namely; Trigonometric Local Search (TLS) and Interpolated Local Search (ILS) on the working of basic Differential Evolution (DE). The corresponding algorithms are named as DETLS and DEILS. The performances of proposed algorithms are investigated and compared with basic DE, modified versions of DE and some other evolutionary algorithms. It is found that the proposed schemes improve the performance of DE in terms of quality of solution without compromising with the convergence rate.

Keywords: global optimization, differential evolution, trigonometric mutation, quadratic interpolation, local search
Categories: C.2.m, F.2, G.1.6, G.1.10, I.6, J.0

1 Introduction

Two competing goals that govern the design of global search methods are exploration and exploitation. Exploration is important to ensure global reliability, i.e., every part of the domain is searched enough to provide a reliable estimate of the global optimum; exploitation on the other hand is important since it concentrates the search effort around the best solutions found so far by searching their neighborhoods to produce better solutions. One of the methods by which the performance of a global optimization metaheuristics can be enhanced is their hybridization with a local search.

This may accelerate the performance of such algorithms by providing additional information about the search domain.

Local search (LS) mechanism can be applied in several ways for example the initial population may be generated by using a LS so that the search technique has some a priori knowledge about the domain. It may also be applied during the processing of the algorithm to explore its neighborhood or it may be applied in any other manner.

In the present study we have concentrated our focus on applying local search mechanism in Differential Evolution (DE), proposed by Storn and Price in 1995 [Storn and Price, 1995]. DE has emerged as a popular choice for solving global optimization problems [Storn and Price, 1997]. Using a few parameters, DE exhibits an overall excellent performance for a wide range of benchmark as well as real-world application problems [Price et al., 2005]. However, despite having several attractive features, it has been observed that the performance of DE is not completely flawless.

DE has certain unique features that distinguish it from other population based evolutionary algorithms (EA). It has a distinctive manner in which it generates the new points and performs selection of points for the next generation. In DE every individual produces a single offspring with the help of directional information and it is designed in such a manner that after the selection process, the points for the next generation are either better or as good as the points of the previous generation. Although these features are there to make DE an effective algorithm, they sometimes hinder its performance by slowing down the convergence rate. Some other drawbacks of DE [Lampinen and Zelinka, 2000] include premature convergence, where the algorithm stops without even reaching a local optimal solution and stagnation, where the algorithm accepts new points in the population but shows no improvement in fitness function value. Therefore, researchers are now concentrating on improving the performance of the classical DE algorithm by using various modifications [Pant et al., 2009, Ali et al., 2009, Brest et al., 2009, Epitropakis et al., 2009, Menchaca-Mendez and Coello Coello, 2009, Faith et al., 2009, Pant et al., 2009, Lai et al., 2009, Omran et al., 2009, Fan and Lampinen, 2003, Rahnamayan et al., 2007, Chakraborty, 2008].

Noman and Iba in their work [Noman and Iba, 2008] pointed out that the adaptive nature of the LS mechanism exploits the neighborhood quite effectively and significantly improves the convergence characteristics of the algorithm. They proposed the use of simplex crossover operator (SPX) as LS and named their algorithm as DEahcSPX, which reportedly gave a good performance on a test suite of global optimization problems. Inspired by the performance DEahcSPX, in the present study we propose two new LS methods namely Interpolated Local Search (ILS) and Trigonometric Local Search (TLS) and analyze their effect on the convergence of basic DE.

We have applied both ILS and TLS in an adaptive manner and have named the corresponding DE algorithms as DETLS and DEILS. Its comparison with basic DE and other algorithms show that the proposed schemes enhance the convergence rate besides maintaining the solution quality.

The remainder of the paper is structured as follows. Section II gives the basics DE. Section III describes the local search methods, ILS and TLS, used in this study. Section IV presents the proposed algorithms. Experimental setting and Benchmark problems are given in Section V. Section VI provides comparisons of results. Finally the paper is concluded in section VII.

2 Differential Evolution

Like other population based search heuristics, DE starts with a population of NPcandidate solutions: $X_{i,G}$, $i = 1, \ldots, NP$, where the index i denotes the *i-th* individual of the population and Gdenotes the generation to which the population belongs. There are three main operators in DE; mutation, crossover and selection. The working of these operators is defined as follows:

Mutation: The mutation operation of DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three individuals $X_{r1,G}$, $X_{r2,G}$and $X_{r3,G}$, from the population set of NP elements. The i^{th} perturbed individual, $V_{i,G+1}$, is generated from the three chosen individuals as follows:

$$V_{i,G+1} = X_{r3,G} + F * (X_{r1,G} - X_{r2,G}) \tag{1}$$

Where, $i = 1. . .NP$, $r_1, r_2, r_3 \in \{1. . . NP\}$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$,
F is the control parameter such that $F \in [0, 1]$.

Crossover: once the mutation phase is complete, the perturbed individual, $V_{i,G+1} = (v_{1,i,G+1}, \ldots, v_{n,i,G+1})$, and the current population member, $X_{i,G} = (x_{1,i,G}, \ldots, x_{n,i,G})$, are subject to the crossover operation, that finally generates the population of candidates known as “trial” vectors,$U_{i,G+1} = (u_{1,i,G+1}, \ldots, u_{n,i,G+1})$, as follows:

$$u_{j,i.G+1} = \begin{cases} v_{j,i.G+1} \; if \; rand_j \leq C_r \vee j = k \\ x_{j,i.G} \qquad otherwise \end{cases} \tag{2}$$

Where, $j = 1. . . n$, $k \in \{1, \ldots, n\}$ is a random parameter’s index, chosen once for each i. The crossover rate, $Cr \in [0, 1]$, is set by the user.

Selection: The selection scheme of DE differs from that of other EAs. Here, the population for the next generation is selected from the individual in current population and its corresponding trial vector according to the following rule:

$$X_{i.G+1} = \begin{cases} U_{i.G+1} \; if \; f(U_{i.G+1}) \leq f(X_{i.G}) \\ X_{i.G} \qquad otherwise \end{cases} \tag{3}$$

Thus, each individual of the temporary (trial) population is compared with its counterpart in the current population. The one with the lower objective function value survives the tournament selection and enters the population of the next generation. As a result, all the individuals of the next generation are as good as or better than their counterparts in the current generation. In DE trial vector is not compared against all the individuals in the current generation, but only against one individual, its counterpart, in the current generation.

3 Proposed local search methods

In this section we describe the local search methods proposed in the present study.

3.1 Trigonometric local search (TLS)

The first local search used in this study is Trigonometric Local Search (TLS) which is based on the Trigonometric Mutation Operator (TMO) proposed by [Fan and Lampinen, 2003]. The working of this operator is described as follows: Select three mutually different individuals, $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$, randomly to implement a mutation operation for the i-th individual $X_{i,G}$. Here $r1$, $r2$, $r3$ $\epsilon\{1, . . . ,NP\}$ are randomly selected and satisfy: $r_1 \neq r_2 \neq r_3 \neq i$. In Trigonometric Mutation Operation, the donor to be perturbed is taken to be the centre point of a hypergeometric triangle. Mathematically, a new point T is generated as follows:

$$T=\left(X_{r1,G}+X_{r2,G}+X_{r3,G}\right)/3+(p_2-p_1)(X_{r1,G}-X_{r2,G})+(p_3-p_2)(X_{r2,G}-X_{r3,G})+(p_1-p_3)(X_{r3,G}-X_{r1,G})$$

Such that $r1 \neq r2 \neq r3 \neq i$ where: $p_1=\left|f(X_{r1,G})\right|/p'$, $p_2=\left|f(X_{r2,G})\right|/p'$, $p_3=\left|f(X_{r3,G})\right|/p'$ and $p'=\left|f(X_{r1,G})\right|+\left|f(X_{r2,G})\right|+\left|f(X_{r3,G})\right|$

As it can be seen from the above formulation, the perturbation part in the trigonometric mutation is contributed together by the three legs of the triangle defined with $X_{ri,G}$, i = 1, 2, 3, i.e., by $(X_{r1,G}-X_{r2,G})$, $(X_{r2,G}-X_{r3,G})$ and $(X_{r3,G}-X_{r1,G})$. This perturbation can also equivalently be viewed as yielded by the donor, namely, the triangle's centre, through shifting along the directions of each leg of the triangle with different step-sizes respectively. The weight terms $(p_2 - p_1)$, $(p_3 - p_2)$ and $(p_1- p_3)$ multiplied to the vector differentials are defined along the two considerations. Firstly, these terms can make the perturbation have a tendency to produce a better individual. This can further be explained from the case that the perturbation is viewed as the result from the donor's shifts. Evidently, with these weight terms the donor is insured to move along in the direction from a vertex with a higher value of objective function towards a vertex with a lower value of objective function. Secondly, the weight terms can automatically scale the contribution magnitudes of the vector differentials to the perturbation in such a way that the greater the difference in the objective function values between the individuals that form a vector differential, the larger the contribution the corresponding vector differential offers to the perturbation. In [Fan and Lampinen, 2003], the authors applied it stochastically along with the basic mutation operation given by equation (1).

3.2 Interpolated local search (ILS)

The second LS operator is based on Quadratic Interpolation (QI) and is therefore named as Interpolated Local Search or ILS. The ILS method is one of the simplest and the oldest direct search method used for solving optimization problems that makes use of gradient in a numerical way. In this method we try to select the three distinct points randomly from the population. A parabolic curve is fitted into the selected points and the point lying at the minimum of this quadratic curve is then evaluated. Mathematically, the new point (say *T*)is produced as follows:

$$T = \frac{1}{2} \frac{(X_{r1,G}^2 - X_{r2,G}^2) f(X_{r3,G}) + (X_{r2,G}^2 - X_{r3,G}^2) f(X_{r1,G}) + (X_{r3,G}^2 - X_{r1,G}^2) f(X_{r2,G})}{(X_{r1,G} - X_{r2,G}) f(X_{r3,G}) + (X_{r2,G} - X_{r3,G}) f(X_{r1,G}) + (X_{r3,G} - X_{r1,G}) f(X_{r2,G})}$$

The symbols have the usual meaning as described in the previous section. ILS has been used in conjugation with several variants of random search/ evolutionary algorithms and has given good results. [Li et al., 2005] proposed a hybrid genetic algorithm (HGA), incorporating QI, for solving constrained optimization problems. [Zhang et al., 2009], used it in DE. Some other papers using QI approach are [Mohan and Shanker, 1994, Ali et al., 1997, Deep and Das, 2008].

4 Proposed algorithms

In the present study we have made slight changes in the ILS and TLS methods. Here, we select one best point and two random points distinct from each other and also from the best point in contrast to three distinct random points used in original versions of these operators. This is done to bias the search in the neighbourhood of the individual having the best fitness. These algorithms start like basic DE up to the selection process. Then at the end of every generation, the point having the best fitness is selected and its neighbourhood is explored with the help of local search schemes. The LS is applied in an adaptive manner as suggested in [Noman and Iba, 2008]. That is to say, the process of LS is repeated till the time there is an improvement in the fitness of the best particle (say X_{Best}). In case there is no improvementin the fitness the algorithm moves on to the next generation. The working of these algorithms is described with the help of a flowchart given in Figure 1.

5 Experimental setup and numerical problems

In order to make a fair comparison of proposed algorithms and basic DE, we have used C++ rand *( )* function to generate initial population for both the algorithms with same seed. The number of individuals in the population is taken as the dimension of the problem. Value of scaling factor F outside the range of 0.4 to 1.2 are rarely effective, so F=0.5 is usually considered a good initial choice. In general very large values of C_r may end up in premature convergence, while very small values of C_r may slow down the convergence. Consequently we have taken C_r =0.5, which is neither too high nor too low. All the algorithms are executed on a PIV PC, using DEV C++, thirty times for each problem. We have compared the algorithms taking two criteria one error and other evaluation so according to these, termination criteria is different in both cases. For error the termination criteria maximum number of function evaluation

Code	Name	Search range
f_{sph}	Sphere	[-100,100]
f_{ack}	Ackley	[-32,32]
f_{sch}	Generalized Schwefel	[-500,500]
f_{sal}	Salomon	[-100,100]
f_{ras}	Rastrigin	[-5,5]
f_{ros}	Rosenbrock	[-100,100]
f_{grw}	Griewank	[-600,600]
f_{pn1}	Generalized Penalized 1	[-50,50]
f_{pn2}	Generalized Penalized 2	[-50,50]
f_{wht}	Whitely	[-100,100]

Table 1: Benchmark problems

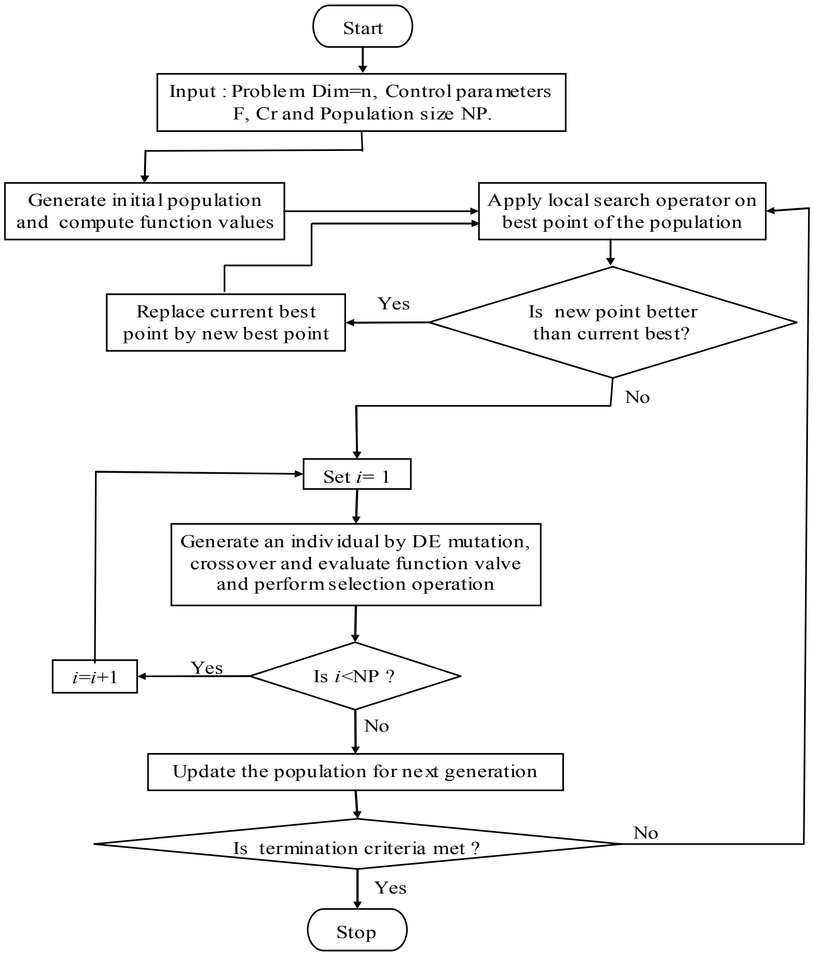

Figure 1: Flow chart of proposed algorithms

(NFE=10,000*Dim). For evaluation termination criterion is $\left|f^* - f_{\min}\right| \le \varepsilon = 10^{-6}$.
Where f^* is global optimum.

Benchmark problems:

The performances of the proposed algorithms are tested on a set of ten benchmark problems taken from literature [Noman and Iba, 2008]. These all are scalable problems, here we have taken dimension thirty. This test set though small provides a suitable platform for testing the efficiency of an optimization algorithm. A brief description of the name of the function, its code and search range are given Table I.

Real life problems:

To further validate the algorithm we have considered three real life problems. These problems are described in the following subsections.

5.1 Water pumping system formulation

A water pumping system [Stoecker , 1971] consists of two parallel pumps drawing water from a lower reservoir and delivering it to another that is 40 m higher, as shown in fig 2. In addition to overcoming the pressure difference due to the elevation, the friction in the pipe is $7.2w^2$ kPa, where w is the combined flow rate in kilograms per second. The pressure-flow-rate characteristics of the pumps are:

$$\text{Pump 1:} \quad \Delta P(kPa) = 810 - 25w_1 - 3.75w_1^2 \tag{4}$$
$$\text{Pump 2:} \quad \Delta P(kPa) = 900 - 65w_2 - 30w_2^2 \tag{5}$$

Where w_1 and w_2 are the flow rates through pump 1 and pump 2, respectively. The system can be represented by four simultaneous equations. The pressure difference due to elevation and friction is:

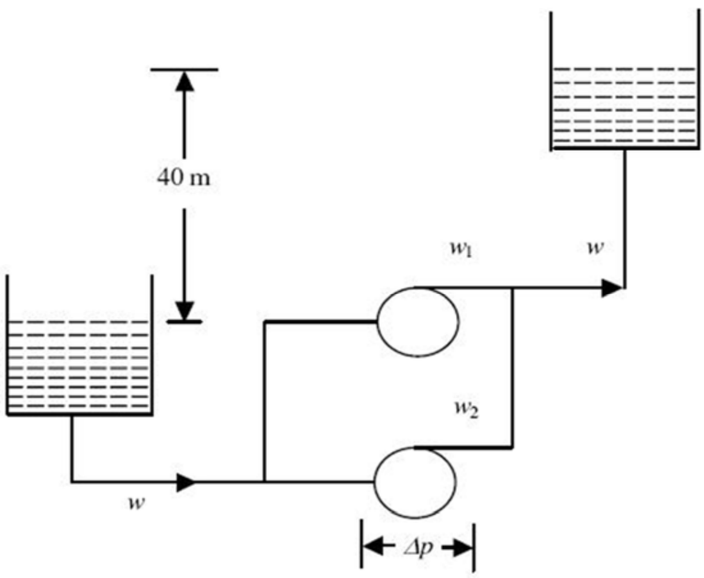

Figure 2: Water Pumping System

$$\Delta P = 7.2w^2 + \frac{(40m)(1000kg/m)(9.807m/s)}{(1000\,Pa/kPa)} \quad (6)$$

Equation (4), (5) and mass balance $w = w_1 + w_2$ (7)

The objective here is to minimize ΔPsubject to the constraints (4), (5), (6), and (7). [Leibman et al., 1986] modified the above problem as given below:

$$\min f = x_3$$

Subject to;

$$x_3 = 250 + 30x_1 - 6x_1^2, x_3 = 300 + 20x_2 - 12x_2^2,$$
$$x_3 = 150 + .5(x_1 + x_2)^2,$$
$$0 \le X \le (9.422, 5.903, 276.42)$$

In general, the equality constraints are difficult to deal with. So there is a need to transform equality constraints into inequality constraints by some means or the other. Typically, they are handled by either of the following two methods, viz., (1) eliminating the parameter and hence reducing the dimensions of the problem (2) an equality constraint is formulated into two inequalities by introducing deviation variables on problem parameter. In the present study, one variable is eliminated while the other two equalities are transformed into inequalities using method 1. Hence, the reformulated problem is as follows:

$$\min f = x_3 = 150 + .5(x_1 + x_2)^2$$

Subject to;

$$g_1 = 6x_1^2 - 30x_1 - 249.9999999 + 150 + .5(x_1 + x_2)^2 \ge 0$$
$$g_2 = 12x_2^2 - 20x_2 - 299.9999999 + 150 + .5(x_1 + x_2)^2 \ge 0$$
$$0 \le X \le (9.422, 5.903)$$

The global optimum obtained is: (x; *f*) = (6.293429, 3.821839; 201.159334).

5.2 Parameter estimation of water quality model

The second problem is a one-dimensional diffusion pollution problem[Wang et al., 2008]. Consider a homogeneous stream region, where it is assumed that the flow field is steady. In this circumstance, the general advection-diffusion-reaction equation of the pollutant is defined by [Loughlin and Bowmer, 1975].

fun	Error			std		
	DE	DETLS	DEILS	DE	DETLS	DEILS
f_{sph}	5.50e-97	2.87e-93	0.00e+00	4.55e-96	2.34e-95	0.00e+00
f_{ack}	3.70e-15	3.69e-15	3.70e-15	8.34e-23	5.54e-24	7.43e-21
f_{sch}	6.82e-04	6.82e-04	6.82e-04	7.43e-07	7.54e-06	5.43e-06
f_{sal}	1.99e-01	9.99e-02	9.99e-02	5.34e-02	6.45e-02	1.35e-02
f_{ras}	8.26e+01	6.26e+01	5.46e+01	1.45e+01	1.43e+01	5.57e+01
f_{ros}	2.45e+01	2.13e+01	2.05e+01	3.45e+02	4.54e+01	4.39e+01
f_{grw}	0.00e+00	0.00e+00	0.00e+00	4.32e-31	4.55e-35	8.55e-34
f_{pn1}	1.35e-19	1.35e-19	1.35e-19	4.32e-26	9.36e-32	5.35e-28
f_{pn2}	1.29e-19	1.29e-19	1.29e-19	2.43e-23	4.56e-22	7.54e-22
f_{wht}	3.63e+02	1.36e+02	3.53e+02	5.59e+01	5.54e+01	4.44e+01

Table 2: Comparison of proposed DEILS and DETLS with basic DE in terms of Mean error and standard deviation for benchmark functions

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial X} = D\frac{\partial^2 C}{\partial X^2} - KC \tag{8}$$

Where $C(X, t)$ (*mass/volume*) is the concentration of the pollutant at downstream distance X and time t; X is distance from upstream boundary condition (*length*); u the cross-sectional mean stream velocity (*length/time*); D is longitudinal dispersion coefficient (*length^2/time*); K denotes the first order reaction rate of the pollutant (*time^{-1}*).

Assuming that the pollutant is injected instantaneously into the stream, we can obtain the equation of the pollutant for the process of diffusion, convection, and absorption, as follows:

$$\begin{cases} \frac{\partial C}{\partial t} + u\frac{\partial C}{\partial X} = D\frac{\partial^2 C}{\partial X^2} - KC \quad (0 < X, 0 < t) \\ C(X,0) = 0 \qquad\qquad\qquad\qquad (0 < X) \\ C(0,t) = C_0\delta(t) \quad C(\infty,t) = 0, \ (0 \le t) \end{cases} \tag{9}$$

Where $C_0 = m/Au$, m the mass of the pollutant, A the mean cross sectional area of the stream. By using Laplace transform, we can easily obtain the analytic solution of (9):

$$C(X,t) = \frac{m}{A\sqrt{4\pi Dt}}\exp\left(-\frac{(X-ut)^2}{4Dt} - Kt\right) \tag{10}$$

According to the measured histories of the pollutant concentration, estimation of parameter vector $\mathbf{Q} = (u, D, K)$ can be formulated as an optimization problem, whose objective function is the total summation of square error (*SSE*). The total summation of square error, for the predicted data compared to measured data, can be cast in the discrete normalized form as

$$SSE = \sum_{i=1}^{n}(C(X_0,t,Q) - C'(X_0,t))^2$$

Where $C'(X_0, t)$ is the measured data of the pollutant concentration at X_0, C (.) the predicted data at X_0, and n the number of data

fun	DE	DETLS	DEILS
f_{sph}	21270	21110	19482
f_{ack}	31950	30976	26839
f_{sch}	300000	300000	300000
f_{sal}	300000	300000	300000
f_{ras}	300000	300000	300000
f_{ros}	300000	300000	300000
f_{grw}	23370	22474	19876
f_{pn1}	20730	20335	16521
f_{pn2}	21780	21885	18445
f_{wht}	300000	300000	300000

Table 3: Mean number of function evaluation to achieve accuracy 10^{-6}.

5.3 Transistor Modelling

The mathematical model of the transistor design [Price, 1983] is given by,

$$\text{Minimize } f(x) = \gamma^2 + \sum_{k=1}^{4}(\alpha_k{}^2 + \beta_k{}^2) \tag{11}$$

Where $\alpha_k = (1 - x_1x_2)x_3\{\exp[x_5(g_{1k} - g_{3k}x_7 \times 10^{-3} - g_{5k}x_8 \times 10^{-3})] - 1\}g_{5k} + g_{4k}x_2$

$\beta_k = (1 - x_1x_2)x_4\{\exp[x_6(g_{1k} - g_{2k} - g_{3k}x_7 \times 10^{-3} + g_{4k}x_9 \times 10^{-3})] - 1\}g_{5k}x_1 + g_{4k}$

$\gamma = x_1x_3 - x_2x_4$

Subject to: $x_i \geq 0$ and the numerical constants g_{ik} are given by the matrix.

$$\begin{bmatrix} 0.485 & 0.752 & 0.869 & 0.982 \\ 0.369 & 1.254 & 0.703 & 1.455 \\ 5.2095 & 10.0677 & 22.9274 & 20.2153 \\ 23.3037 & 101.779 & 111.461 & 191.267 \\ 28.5132 & 111.8467 & 134.3884 & 211.4823 \end{bmatrix}$$

This objective function provides a least-sum-of-squares approach to the solution of a set of nine simultaneous nonlinear equations, which arise in the context of transistor modeling.

6 Numerical results and comparisons

6.1 Comparison between DETLS, DEILS and DE

In this section we compare DETLS and DEILS with DE algorithm. The results in terms of average error and standard deviation are listed in Table II. Average error is defined as the difference between the true global optimum value and the value

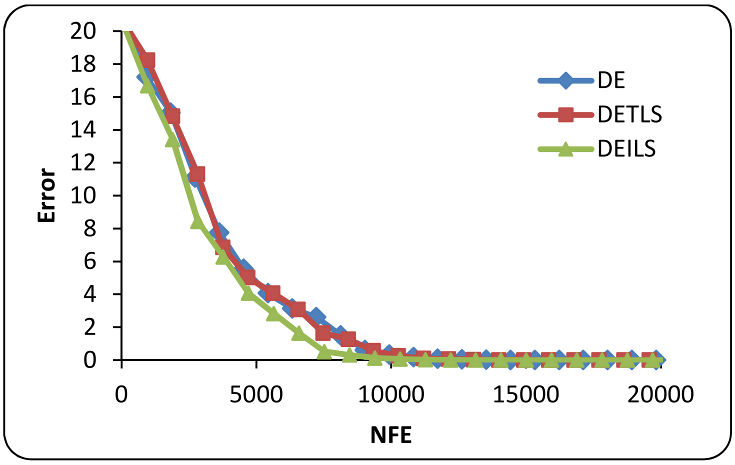

Figure 3(a): Performance curves of Ackley function

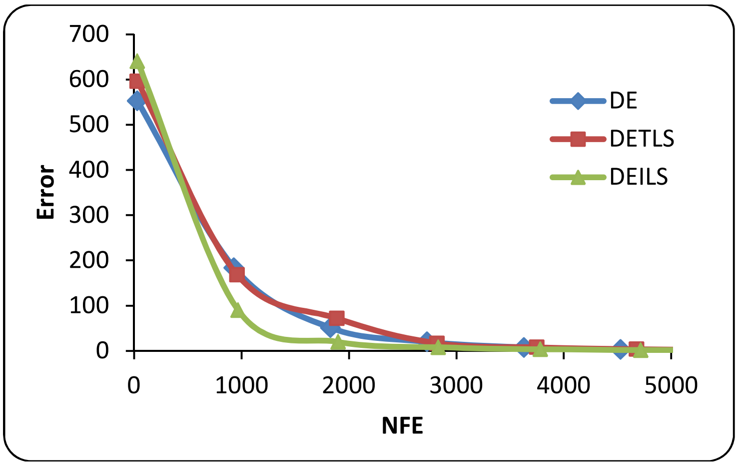

Figure 3(b): Performance curves of Griewank function

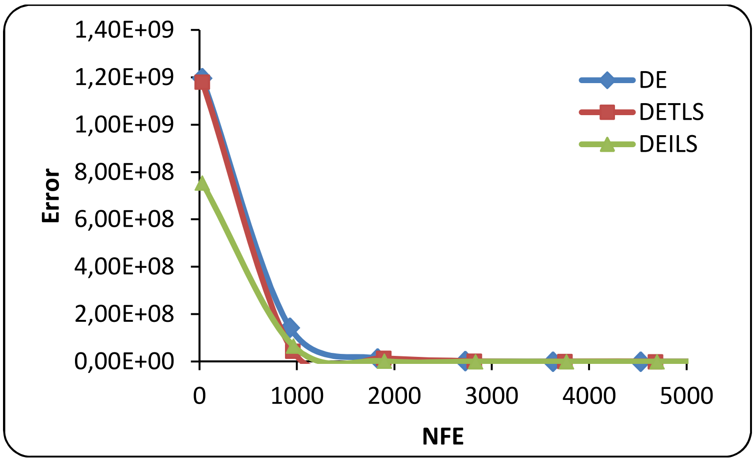

Figure 3(c): Performance curves of PN1 function

obtained by the algorithm. Table III provides number of function evaluations (NFE) obtained to achieve the desired accuracy of error i.e. 10^{-06}. As it is clear from Table II that in term of average error and standard deviation all the algorithms give more or less similar results although in some cases DEILS performs slightly better than other algorithms. The superior performance of the proposed algorithms is more evident from Table III, which gives the average number of functions evaluations. If the algorithm does not achieve the desired accuracy then we have taken maximum number of function evaluation (=3, 00,000). From Table III we can see that DEILS takes less number of function evaluations to achieve the required fitness in comparison to the basic DE and DETLS in five cases and there is a tie in remaining five cases. Second place goes to the DETLS algorithm. Performance curves

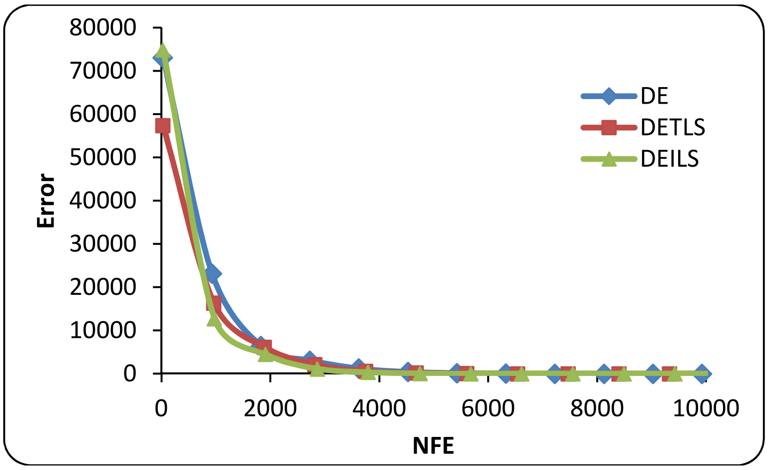

Figure 3(d): Performance curves of sphere function

fun	DETLS	DEILS	Deahc SPX	TDE	MGG +UNDX	G3+PCX	MGG +SPX
f_{sph}	2.87e-93 2.34e-95	0.00e+00 0.00e+00	9.50e-31 3.09e-30	4.53e-99 4.22e-98	1.73e-11 1.94e-11	3.58e-81 1.36e-81	8.75e+00 2.87e+00
f_{ack}	3.69e-15 5.54e-24	3.69e-15 7.43e-21	1.78e-14 1.01e-32	7.25e-13 2.33e-30	8.23e-07 4.64e-07	1.48e+01 4.17e+00	1.68e+00 2.99e-01
f_{sch}	6.81e-04 7.54e-06	6.81e-04 5.43e-06	2.89e+02 5.48e+02	6.81e-01 1.23e-01	4.12e+03 1.72e+03	4.04e+03 1.09e+03	8.70e+03 2.41e+02
f_{sal}	9.99e-02 6.45e-02	9.99e-02 1.34e-02	1.83e-01 3.45e-03	1.99e-01 9.98e-02	1.50e-01 4.95e-02	4.64e+00 4.74e+00	3.82e-01 4.29e-02
f_{ras}	6.25e+01 1.43e+01	5.46e+01 5.56e+01	2.45e+01 2.43e+01	1.29e+01 8.56e+01	1.35e+00 1.03e+00	1.75e+02 3.37e+01	5.78e+00 1.83e+00
f_{ros}	2.12e+01 4.54e+01	2.058e+01 4.38e+01	1.87e+00 5.01e+01	2.23e+00 1.02e-01	2.81e+01 1.23e+01	4.18e+00 9.68e+01	1.38e+03 6.45e+02
f_{grw}	0.00e+00 4.54e-35	0.00e+00 8.54e-34	4.94e-03 4.79e-04	7.39e-03 1.08e-19	2.96e-04 1.48e-03	1.07e-02 1.30e-02	1.09e+00 2.24e-02
f_{pn1}	1.35e-19 9.35e-32	1.35e-19 5.34e-28	3.45e-06 5.53e-05	1.35e-19 1.84e-18	4.93e-02 3.50e-02	4.35e+00 6.94e+00	2.57e-01 6.90e-02
f_{pn2}	1.29e-19 4.55e-22	1.29e-19 7.54e-22	2.39e-30 4.34e-31	1.29e-19 7.34e-21	4.39e-04 2.20e-03	1.50e+01 1.58e+01	2.29e+00 3.72e-01
f_{wht}	1.35e+02 5.54e+01	3.52e+02 4.43e+01	3.05e+02 2.34e+02	1.49e+02 4.34e+02	4.28e+02 3.82e+01	7.90e+02 1.27e+02	3.28e+03 2.77e+03

Table 4: comparison of proposed DETLS and DEILS with other algorithms in terms of mean error and standard deviation (STD) for 10 benchmark function

(convergence graphs) of few selected functions are given in Fig3(a) – Fig3(d).From these illustrations it is evident that the convergence of proposed algorithms is faster than basic DE.

6.2 Comparisons of proposed DETLS and DEILS with other algorithms

The performances of the proposed algorithms are further compared with two other modified versions of DE namely DEahcSPX [Noman and Iba, 2008] and Trigonometric Mutation Differential Evolution, TDE [Fan and Lampinen, 2003]. We executed these algorithms with same parameter settings as given in relevant literature. Also we have compared the proposed algorithms with some state of the art evolutionary algorithms like G3+PCX, MGG+UNDX and MGG+SPX. For these three algorithms we have taken the results given in [Noman and Iba, 2008]. All these algorithms have reportedly given a good performance for a set of benchmark problems. The results obtained are summarized in Tables IV and V. In Table IV, the results are compared in term of average error and standard deviation. From this Table we can see that the proposed algorithms perform better in almost cases in comparison to the other algorithms. From Table V, which gives the number of function evaluations (NFE) we can see that DEILS takes less NFE in most of the functions to achieve the accuracy given in last column. Performance curves (convergence graphs) of few selected functions showing the comparison of the proposed algorithms with TDE and DESPX are illustrated in Fig. 4(a) – 4(d).

fun	DEahcSPX	TDE	DETLS	DEILS
f_{sph}	87013	30750	21110	19482
f_{ack}	129189	44340	30976	26839
f_{sch}	300000	300000	300000	300000
f_{sal}	300000	300000	300000	300000
f_{ras}	300000	300000	300000	300000
f_{ros}	299913	193380	300000	300000
f_{grw}	121579	29880	22474	19876
f_{pn1}	96121	37500	20335	16521
f_{pn2}	85432	39060	21885	18445
f_{wht}	300000	300000	300000	300000

Table 5: Mean number of function evaluation to achieve accuracy 10-6

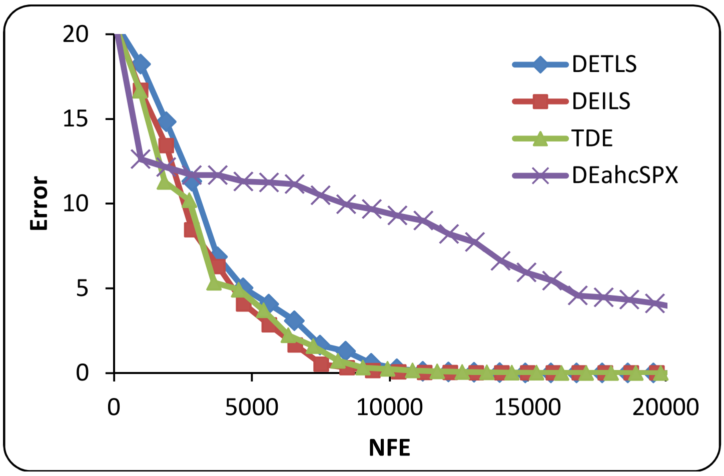

Figure 4(a): Performance curves of Ackley function

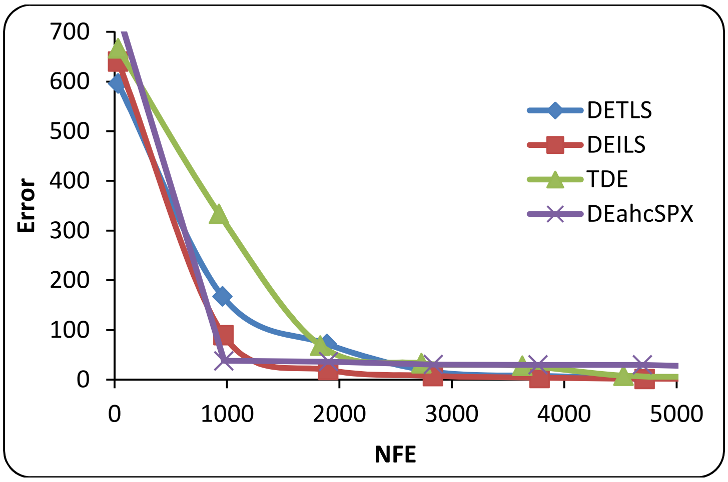

Figure 4(b): Performance curves Griewank function

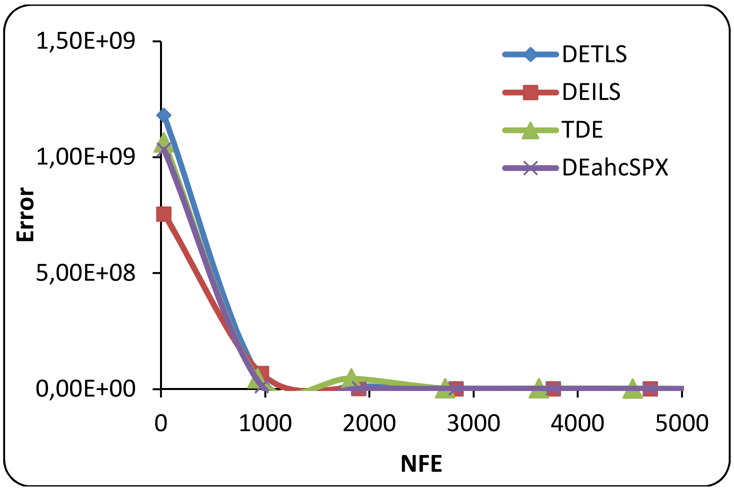

Figure 4(c): Performance curves of PN1 function

6.3 Numerical results of real life problems

The experimental setting for this comparison is same as for benchmark problems except the population size *NP*=50. Method proposed by Deb [Deb, 2000], is used in the present study to handle constraints. Numerical result of water pumping system is given in Table VI. From this table it is clear that all the algorithms give almost same solution but DEILS takes lesser number of functions evaluations to find out the solution in comparison to other algorithms.

Numerical results for parameter estimation of water quality model are summarized in Table VII. For the solution of this problem parameters are taken as X0 = 8km, the injected pollutant m=378.52g, cross sectional area A=14m2. And the exact values

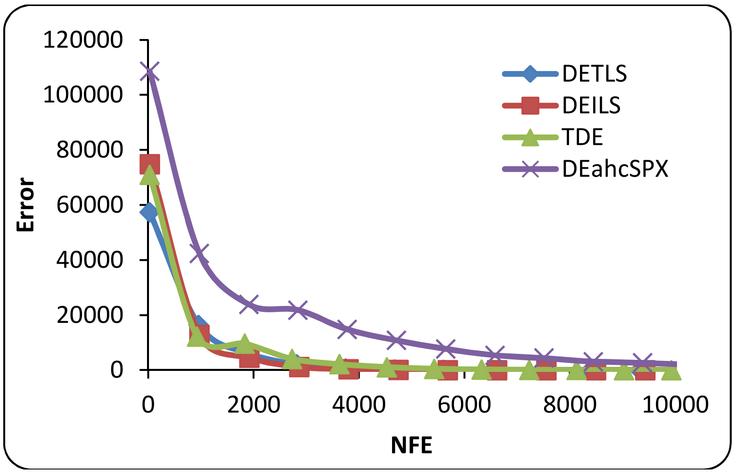

Figure 4(d): Performance curves of sphere function

	DE	DETLS	DEILS
x1	6.29343	6.29343	6.29343
x2	3.82184	3.82184	3.82184
x3	201.15978	201.15934	201.15933
g1	8.78565e-05	6.79518e-05	6.32518e-05
g2	7.89362e-05	2.58852e-05	2.58832e-05
f(X)	201.15978	201.15934	201.15933
NFE	8762	6532	5825

Table 6: Parameters, fitness and nfe values for the water pumping system

	DE	DETLS	DEILS
x1=u	0.530027	0.530003	0.52856
x2=D	22.0001	22.0001	20.36604
x3=K	0.0669998	0.0670001	0.07087
f(X)	3.17559e-11	2.24631e-16	7.40720e-07
NFE	6820	5630	6154

Table 7: Parameters, fitness and nfe values for the water quality model problem

of parameters (u, D, K) were assumed to be fixed at 0.53m/s, 22.0m2/s and 0.065/h respectively. We have taken a sample of 20 points with same interval 0.1158h between 3.2h and 5.4h. From Table VII it is clear that DETLS outperforms all the algorithms. The numerical results of transistor modeling problem is given in Table VIII. Here the proposed algorithms are compared with DE in terms objective function value and NFEs. From this Table we can see the excellent performance of DEILS which is clearly better than all remaining algorithms.

7 Discussions and conclusions

In this paper we proposed two LS schemes; Trigonometric Local Search (TLS) and Interpolated Local Search (ILS). These schemes are embedded in the structure of basic DE and are applied adaptively to explore the neighborhood of the best individual of the population. The corresponding algorithms are termed as DETLS and DEILS. Both the schemes make a judicious use of the exploration and exploitation abilities of the search mechanism and are therefore more likely to avoid false or premature convergence. These schemes are rather greedy in nature as they bias the new trial solution strongly in the direction where the best one of three individuals chosen for the mutation is and can therefore be viewed as good local searches.

The simulation of results showed that the resulting DE variants are quite competent for solving problems of different dimensions in less number of function evaluations without compromising with the quality of solution. We have also compared our results with other variants of DE as well as with some other algorithms (DEahcSPX, TDE, G3+PCX, MGG+UNDX and MGG+SPX). The set of problems considered, though small and limited show the promising nature of proposed algorithms. However, we would like to maintain that the work is still in the preliminary stages and efforts continue for advanced study. We intend to apply it for more complex problems and compare its performance with other versions of DE and with other optimization algorithms.

	DE	DETLS	DEILS
x1	0.901340	0.901337	0.901337
x2	0.891164	0.891043	0.891043
x3	3.87857	3.87943	3.87943
x4	3.94653	3.94663	3.94663
x5	5.32623	5.32509	5.32509
x6	10.6267	10.6171	10.6171
x7	0.0	0.0	0.0
x8	1.08924	1.08832	1.08832
x9	0.705675	0.706734	0.706734
f(X)	0.0937829	0.0643636	0.0643636
NFE	15940	11780	10710

Table 8: Parameters, fitness and nfe values for the transistor modeling problem

References

[Ali et al., 1997] Ali, M.M., Torn, A. and Vitanen. S. (1997), A numerical comparison of some modified controlled random search algorithms. *J. Global Optim.*, 11:341-359.

[Ali et al., 2009]Ali, M. Pant, M and Abraham, A. (2009), Mixed Strategy Embedded Differential Evolution, IEEE Congress on Evolutionary Computation, Norway, pp. 2841-2849.

[Brest et al., 2009]Brest, Janez, Zamuda ales, Boskovic, Boroko, Mirjam Sepesy Maucec and Viljem Zumer, (2009), Dynamic Optimization using Differential Evolution, IEEE Congress on Evolutionary Computation, Norway, pp. 415-421.

[Chakraborty, 2008] Chakraborty U. K. (Ed.) (2008), *Advances in Differential Evolution*, Springer-Verlag, Heidelberg.

[Deb, 2000] Deb, K. (2000), An efficient constraint handling method for genetic algorithm, computer method in applied mechanics and engineering, 186(2/4), pp 311-338.

[Deep and Das, 2008] Deep K.and Das, K.N. (2008), Quadratic approximation based hybrid genetic algorithm for function optimization, Applied Mathematics and Computation, 203, 86-98.

[Epitropakis et al., 2009]Epitropakis, M.G, Plagianakos, V.P. and Vrahatis, M.N. (2009), Evolutionary Adaption of the Differential Evolution Control Parameters, IEEE Congress on Evolutionary Computation, Norway, pp. 1359-1366.

[Faith et al., 2009] Faith Tasgetiren M., Pan Quan-Ke, Suganthan P.N., and Yun Chia Liang, (2009), A Differential Evolution Algorithm with Variable Parameter Search for Real Parameter Continuous Function Optimization, IEEE Congress on Evolutionary Computation, Norway, pp. 1247-1254.

[Fan and Lampinen, 2003] Fan, Hui-Yuan, Lampinen, Jouni (2003), A Trigonometric Mutation Operation to Differential Evolution, *Journal of Global Optimization* 2003, 27:105-129.

[Lai et al., 2009] Lai, JCY Leung FHF and Ling SH. (2009), A new Differential Evolution Algorithm with Wavelet Theory based Mutation operation. IEEE Congress on Evolutionary Computation, Norway, pp. 1116-1122.

[Lampinen and Zelinka, 2000] Lampinen, J. and Zelinka, I. (2000), On stagnation of the differential evolution algorithm, in: Pavel Ošmera, (ed.) Proc. of MENDEL 2000, 6th International Mendel Conference on Soft Computing, pp. 76 – 83.

[Leibman et al., 1986] Leibman, J., Lasdon, L., Schrage, L. and Waren, A. (1986), Modeling and optimization with GINO.*The Scientific Press*, Palo Alto, CA.

[Li et al., 2005] Li, H., Jiao, Y.C. and Wang, Y.P. (2005), Integrating the simplified interpolation into the genetic algorithm for constrained optimization problems, Springer-Verlag Berlin Heidelberg ,247-254.

[Loughlin and Bowmer, 1975] Loughlin, E. M. O., Bowmer, K. H. (1975), Dilution and decay of aquatic herbicides in flowing channels, *J. Hydro*, Vol. 26, pp. 217-235.

[Menchaca-Mendez and Coello Coello, 2009]Menchaca-Mendez Adriana and Coello Coello Carlos A. (2009), A new proposal to hybridize the Nelder Mead Differential Evolution Algorithm for Constrained Optimization, IEEE Congress on Evolutionary Computation, Norway, pp. 2598-2605.

[Mohan and Shanker, 1994] Mohan C. and Shanker, K., (1994), A Controlled Random Search Technique For Global Optimization using Quadratic Approximation, Asia-Pacific Journal of Operational Research, Vol. 11, pp. 93-101.

[Noman and Iba, 2008] Noman, N. and Iba, H.(2008), Accelerating differential evolution using an adaptive local search, IEEE transactions on evolutionary computation 12(1), 107-125.

[Omran et al., 2009] Omran, G.H. Mohammad, and Engelbrecht, Andries P. (2009), Free Search Differential Evolution. IEEE Congress on Evolutionary Computation, Norway, pp. 110-117.

[Pant et al., 2009] Pant, M. Ali, M. and Singh, V.P. (2009), Parent centric differential evolution algorithm for global optimization, Opsearch 46(2), pp 153-168.

[Pant et al., 2009]Pant, M. Thangaraj R., Abraham,A. and Grosan, C. (2009), Differential Evolution with Laplace Mutation Operator, IEEE Congress on Evolutionary Computation, Norway, pp. 2841-2849.

[Price et al., 2005] Price, K. V., Storn, R. M. and Lampinen J. A. (2005)*, Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag.

[Price, 1983] Price, W.L. (1983), global optimization by controlled random search, *Journal of optimization theory and applications*, 40, 3, 333-348.

[Rahnamayan et al., 2007] Rahnamayan, Shahryar Tizhoosh, H.R..Salama, M.M.A. (2007), A novel population initialization method for accelerating evolutionary algorithms, *computer and applied mathematics with application* (53), pp 1605-1614.

[Stoecker , 1971] Stoecker, W. F. (1971), Design of Thermal Systems. 3rd ed.,McGraw-Hill International edition, Singapore, pp 117-121.

[Storn and Price, 1995] Storn.R and Price, K. (1995), Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, *Technical Report TR-95-012, Berkeley, CA*.

[Storn and Price, 1997]Storn,R. and Price, K. V. (1997), Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces, *J. Global Opt.*, vol. 11, no. 4, pp. 341–359.

[Wang et al., 2008] Wang, K., Wang, X. Wang, J. G.Lv, M.Jiang, C.Kang and L.Shen, (2008), particle swarm optimization for calibrating stream water quality model, second international symposium on intelligent information technology application, pp 682-686.

[Zhang et al., 2009] Zhang, Li, Jiao, Yong-Chang, Li, Hong, Zhang, Fu-Shun. (2009), Hybrid Differential Evolution and the Simplified Quadratic Interpolation for Global Optimization. *GEC'09,* June 12–14, Shanghai, China. ACM 978-1-60558-326-6/09/06.