The Population Dynamics of Differential Evolution: A Mathematical Model

Sambarta Dasgupta, Arijit Biswas, Swagatam Das and Ajith Abraham

Abstract— **Differential Evolution (DE) is well known as a simple and efficient algorithm for global optimization over continuous spaces. This article provides a simple mathematical model of the underlying evolutionary dynamics of a one-dimensional DE. The model relates the search process of DE with the classical gradient descent search and also analyzes the convergence behavior of a DE population, very near to optima.**

I. Introduction

Differential Evolution [1, 2] is a simple real coded Genetic Algorithm (GA) [3], which shows strikingly good performance as compared to classical GA or swarm intelligence based meta-heuristic like Particle Swarm Optimization (PSO) [4] over several numerical benchmarks [5]. DE has successfully been applied to divert domains of science and technology such as pattern recognition, signal processing, mechanical engineering design, and chemical process plant optimization [6-11] etc. As in other evolutionary strategies two main processes drive the evolution in DE: firstly, the variation process, which enables exploring the different regions of the search space and then the selection process that ensures the exploitation of previous knowledge about fitness landscape. Although there exists a plethora of works concerning the empirical study of parameter selection and tuning process in DE [12-20] and its application to optimization problems, little research has been undertaken to model the underlined search dynamics of DE, which would enable us to understand how and why DE manages to find the optima of many difficult numerical functions so fast. A significant work in this direction was reported in [21] by Zaharie, where she theoretically analyzed the influence of the variation *operators* and their parameters on the expected population variance.

In this article we provide a simple mathematical model of DE/rand/1/bin scheme (which is the most popular variant of DE family [2]). The model attempts to find out the velocity of each search variable vector towards the optima over successive generations. It also tries to relate the search mechanism in DE with that of the classical gradient descent search technique [22]. Moreover, it explores the behavior of population members near optima.

II. The Differential Evolution Algorithm – An Outline

Like any other evolutionary algorithm, DE also starts with a population of *NP* D-dimensional parameter vectors. We shall represent subsequent generations in DE by discrete time steps like t=0, 1, 2 ...t, t+1 etc. Since the vectors are likely to be changed over different generations we may adopt the following notation for representing the i-th vector of the population at the current generation (i.e. at time t=t) as:

$$X_i(t) = [x_{i,1}(t), x_{i,2}(t), x_{i,3}(t) ..... x_{i,D}(t)] \qquad (1)$$

For each parameter of the problem, there may be a certain range within which value of the parameter should lie for better search results. At the very beginning of a DE run or at t = 0, problem parameters or independent variables are initialized somewhere in their feasible numerical range. So, if the j-th parameter of the given problem has its lower and upper bound as $x_{\min,j}$ and $x_{\max,j}$ respectively, then we may initialize the j-th component of the i-th population members as,

$$x_{i,j}(0) = x_{\min,j} + rand_j(0,1).(x_{\max,j} - x_{\min,j}) \qquad (2)$$

Where $\text{rand}_j(0,1)$ is the j-th instantiation of a uniformly distributed random number lying between 0 and 1. The following steps are taken next:

2.1 Mutation

After initialization, DE creates a *donor* vector $V_{i,G}$ corresponding to each population member or *target* vector $X_{i,G}$ in the current generation through mutation. It is the method of creating this donor vector, which demarcates between the various DE schemes. For example, five most frequently referred mutation strategies implemented in the public-domain DE codes available online at http://www.icsi.berkeley.edu/~storn/code.html are listed as follows:

Sambarta Dasgupta, Arijit Biswas and Swagatam Das are with Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India.

Ajith Abraham is with the Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, Norway (e-mail: ajith.abraham@ieee.org).

978-1-4244-1823-7/08/$25.00©2008 IEEE

"DE/rand/1": $V_i(t) = X_{r_1^i}(t) + F.(X_{r_2^i}(t) - X_{r_3^i}(t))$ (3.1)

"DE/best/1": $V_i(t) = X_{best}(t) + F.(X_{r_1^i}(t) - X_{r_2^i}(t))$ (3.2)

"DE/target-to-best/1":

$V_i(t) = X_i(t) + F.(X_{best}(t) - X_i(t)) + F.(X_{r_1^i}(t) - X_{r_2^i}(t))$

(3.3)

"DE/best/2":

$V_i(t) = X_{best}(t) + F.(X_{r_1^i}(t) - X_{r_2^i}(t)) + F.(X_{r_3^i}(t) - X_{r_4^i}(t))$ (3.4)

"DE/rand/2":

$V_i(t) = X_{r_1^i}(t) + F.(X_{r_2^i}(t) - X_{r_3^i}(t)) + F.(X_{r_4^i}(t) - X_{r_5^i}(t))$ (3.5)

The indices r_1^i, r_2^i, r_3^i, r_4^i and r_5^i are mutually exclusive integers randomly chosen from the range [1, *NP*], which are also different from the index *i*. These indices are randomly generated once for each mutant vector. The scaling factor *F* is a positive control parameter for scaling the difference vectors. $X_{best,G}$ is the best individual vector with the best fitness function value in the population at generation *G*. The general convention used for naming the various mutation strategies is DE/x/y/z, where DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed and y is the number of difference vectors considered for perturbation of x. z stands for the type of crossover being used (exp: exponential; bin: binomial). The following section discusses the crossover step in DE.

2.2 Crossover

Next, to increase the potential diversity of the population a crossover scheme is undertaken. DE family of algorithms can use two kinds of cross over schemes namely *exponential* and *binomial*. The donor vector exchanges its "body parts" i.e. components with the target vector $X_i(t)$ under this scheme to form the *trial* vector $U_i(t)$. We here outline the binomial crossover scheme, which comes into play in our present analysis. In this case the crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the CR value. In this case the number of parameters inherited from the mutant has a (nearly) binomial distribution. The scheme may be outlined as follows:

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t) & \text{If } (rand_i(0,1) \le CR) \text{ or } (j = rn(i)) \\ x_{i,j}(t) & \text{If } (rand_i(0,1) > CR) \text{or } (j \ne rn(i)) \end{cases} \quad (4)$$

where $rand_j$ (0, 1)∈ [0,1] is the j-th evaluation of a uniform random number generator. *rn(i)* $\in [1,2,....,D]$ is a randomly chosen index which ensures that $U_i(t)$ gets at least one component from $V_i(t)$. It is instantiated once for each vector. In this article we have not taken into account the term *rn(i)* so that CR may be exactly equal to the cross-over probability p_{Cr}.

2.3 Selection

In this way for each target vector $X_i(t)$ a trial vector $U_i(t)$ is created. To keep the population size constant over subsequent generations, the next step of the algorithm calls for 'selection' to determine which one of the target and the trial vector will survive in the next generation i.e. at time t = t+1. DE actually involves the Darwinian principle of "Survival of the fittest" in its selection process, which may be outlined as,

$$X_i(t+1) = \begin{cases} U_i(t) & \text{if } f(U_i(t)) \le f(X_i(t)) \\ X_i(t) & \text{if } f(U_i(t)) > f(X_i(t)) \end{cases} \quad (5)$$

where f( ) is the function to be minimized. So if the new trial vector yields a better value of the fitness function, it replaces its parent in the next generation; otherwise the parent is retained in the population. Hence the population either gets better (w.r.t the fitness function) or remains constant but never deteriorates. The DE/rand/1 algorithm is outlined below:

Pseudo-code for DE algorithm

```
Step 1: Set the time-step $t = 0$, and randomly initialize a
        population of NP individuals
        $P = \{X_1(0),......, X_{NP}(0)\}$ with
        $X_i(0) = \{x_{i,1}(0),......, x_{i,D}(0)\}$ and
        $i = [1,2,...., NP]$ uniformly distributed in the
        range $[X_{\min}, X_{\max}]$, where
        $X_{\min} = \{x_{\min,1},......, x_{\min,D}\}$ and
        $X_{\max} = \{x_{\max,1},......, x_{\max,D}\}$
Step 2: WHILE stopping criterion is not satisfied
        DO
        Step 2.1 Mutation Step
            FOR $i = 1$ to NP
              Generate a donor vector
              $V_i(t) = \{v_{i,1}(t),......., v_{i,D}(t)\}$ corresponding
              to the target vector $X_i(t)$ via one of the
              different mutation schemes of DE (equations
              (3)).
            END FOR
        Step 2.2 Crossover Step
            Generate a trial vector $U_i(t)$ for each target
            vector $X_{i,G}$ through binomial crossover
            (equation (4)).
        Step 2.3 Selection Step
            FOR $i = 1$ to NP
              Evaluate the trial vector $U_i(t)$
              IF $f(U_i(t)) \le f(X_i(t))$, THEN
              $X_i(t+1) = U_i(t)$, $f(X_i(t+1)) = f(U_i(t))$
              ELSE $X_i(t+1) = X_i(t)$
                  END IF
            END FOR
```

III. The Mathematical Model of the Population Dynamic in Differential Evolution

In what follows we outline a simple mathematical model of the dynamics of the parameter vectors of a DE population over a one-dimensional fitness landscape .The analysis is based on simple probability theory and differential calculus .It provides an insight into the search process developed through DE type mutation and crossover operators. Although there are a number of different DE mutation strategies, we confine ourselves to the most popular one among them – DE/rand/1/bin scheme for the sake of simplicity of expressions that allow one to visualize how different operators in DE contribute to the dynamics of the population. The analysis can be easily extended to other DE type mutations.

3.1 Assumptions

Suppose $f(x)$, function of single variable x, is to be optimized using the DE Algorithm. Let $\{x_1, x_2, ... x_{NP}\}$ be a set of trial solutions forming the population subjected to DE search where NP denotes the population size. In order to validate our analysis, we make certain assumptions, which are listed below:

i. The population of NP trial solutions is limited within a small region i.e. individual trial solutions are located very close to each other.

ii. A region of fitness landscape, having moderate value of gradient at each point, is prescribed as the domain of the analysis.

iii. Dynamics is modeled assuming continuous time.

Figure1 depicts a favorable portion of a one- dimensional arbitrary objective function for our analysis.

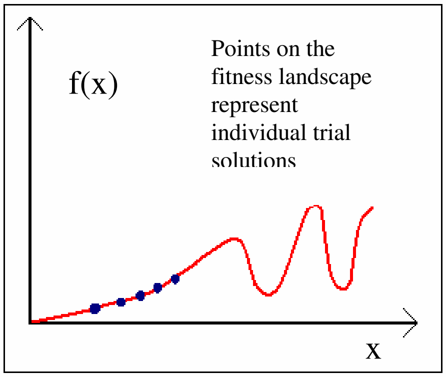


Figure 1: A DE population dispersed on a one-dimensional arbitrary fitness landscape

3.2 Modeling Different Steps of DE

Let x_m be the m-th individual of the population, where $m = 1(1)NP$. It is used as a target vector in a particular DE iteration. During an iteration of DE, it undergoes three steps namely mutation, crossover and selection. Each step is modeled individually and finally they are merged to get a generalized expression. In the following analysis, upper case letter denotes random variables.

Three trial solutions are chosen at random from the population. Let X_{r1}, X_{r2}, X_{r3} be three trial solutions (random variables) picked up randomly from population. Here, we assume trial solutions are drawn with replacement. i. e. each trial solution chosen at a particular draw is returned to the population before next draw. This assumption makes X_{r1}, X_{r2}, X_{r3} independent of each other.

This means $P(X_{ri} = x_l \mid X_{rj} = x_k) = P(X_{ri} = x_l)$

$$\Rightarrow P(X_{ri} = x_l \cap X_{rj} = x_k) = P(X_{ri} = x_l) P(X_{rj} = x_k)$$

Where, $i, j = 1,2,3$ and $k, l = 1(1)NP$ and $i \neq j$

Difference of X_{r2}, X_{r3} is scaled by a factor F and then X_{r1} is added with the scaled difference. Let V_m be the generated donor vector.

$$\therefore V_m = X_{r1} + F(X_{r2} - X_{r3})$$

For the one-dimensional analysis we omit the restriction that at least one component of the trial vector must come from the donor. Hence in this case CR equals the true probability of the event that $U_m = V_m$. Equipped with these assumptions we may assert the following theorems:

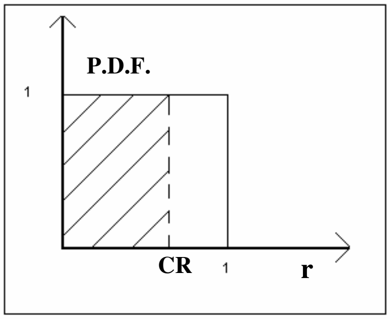


Figure 2. Probability density function of r

***Theorem 1*:** The expected value of U_m^2 can be given by:

$$E(U_m{}^2) = (1 - CR)x_m{}^2 + CR(2F^2 + 1)Var(x) + CRx_{av}{}^2 \qquad (6)$$

Proof: From Figure 2, probability of the event $r \leq CR$

$= P(r \leq CR) =$ Area of the shaded region.

$= 1 \times CR = CR$.

Now, $r \le CR$ and $r > CR$ are mutually exclusive and exhaustive events.

$$\therefore P(r > CR) = 1 - P(r \le CR) = 1 - CR$$

$$\therefore E(U_m) = P(r > CR)x_m + \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}[P\{(r \le CR)\cap((X_{r1}=x_i)\cap(X_{r2}=x_j)\cap(X_{r3}=x_k))\}$$
$$\{x_i + F(x_j - x_k)\}]$$

Now, we have assumed that mutation and crossover are independent of each other i.e. r is independent of X_{r1}, X_{r2}, X_{r3}.

$$P\{(r\le CR)\cap((X_{r1}=x_i)\cap(X_{r2}=x_j)\cap(X_{r3}=x_k))$$
$$= P(r\le CR)P((X_{r1}=x_i)\cap(X_{r2}=x_j)\cap(X_{r3}=x_k))$$

$$\therefore E(U_m) = P(r>CR)x_m + P(r\le CR)$$
$$\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}P\{(X_{r1}=x_i)\cap(X_{r2}=x_j)\cap(X_{r3}=x_k)\}[x_i+F(x_j-x_k)]$$

X_{r1}, X_{r2}, X_{r3} are independent random variables.

Hence,

$$P\{(X_{r1}=x_i)\cap(X_{r2}=x_j)\cap(X_{r3}=x_k)\} =$$
$$P(X_{r1}=x_i)P(X_{r2}=x_j)P(X_{r3}=x_k)$$
$$\therefore E(U_m)=P(r>CR)x_m +$$
$$P(r\le CR)\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}P(X_{r1}=x_i)P(X_{r2}=x_j)P(X_{r3}=x_k)[x_i+F(x_j-x_k)]$$

Now, $P(X_{r1}=x_i)=P(X_{r2}=x_j)=P(X_{r3}=x_k)=\frac{1}{NP}$

$$\therefore E(U_m)=P(r>CR)x_m +$$
$$P(r\le CR)\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}\frac{1}{NP^3}[x_i+F(x_j-x_k)]$$

$$\therefore E(U_m)=(1-CR)x_m +$$
$$CR\frac{1}{NP^3}\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}[x_i+F(x_j-x_k)]$$
$$\therefore E(U_m)=(1-CR)x_m +$$
$$CR\frac{1}{NP^3}[\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_i + F(\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_j - \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_k)]$$

$$\therefore E(U_m)=(1-CR)x_m + CR\frac{1}{NP^3}[\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_i]$$
$$[\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_j = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_k]$$

$$\therefore E(U_m)=(1-CR)x_m+CR\frac{1}{NP}\sum_{i=1}^{NP}x_i$$

$$\therefore E(U_m)=(1-CR)x_m+CRx_{av} \qquad (7)$$

Now, similar to the previous one,

$$\therefore E(U_m^2)=P(r>CR)x_m^2 +$$
$$\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}[P\{(r\le CR)\cap((X_{r1}=x_i)\cap(X_{r2}=x_j)\cap(X_{r3}=x_k))\}$$
$$\{x_i+F(x_j-x_k)\}^2]$$

Proceeding in the same manner,

$$\therefore E(U_m^2)=(1-CR)x_m^2 +$$
$$CR\frac{1}{NP^3}\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}\{x_i+F(x_j-x_k)\}^2$$

$$\Rightarrow E(U_m^2)=(1-CR)x_m^2 +$$
$$CR\frac{1}{NP^3}\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}\{x_i^2+F^2(x_j-x_k)^2+2Fx_i(x_j-x_k)\}$$

$$\Rightarrow E(U_m^2)=(1-CR)x_m^2 +$$
$$CR[(2F^2+1)\frac{1}{NP}\sum_{i=1}^{NP}x_i^2 - 2F^2(\frac{1}{NP}\sum_{i=1}^{NP}x_i)^2]$$

$$\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_ix_j = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_ix_k$$

and $\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_i^2 = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_j^2 = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_k^2$

also $\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_ix_j = NP\sum_{i=1}^{NP}x_i\sum_{j=1}^{NP}x_j = NP\left(\sum_{i=1}^{NP}x_i\right)^2$

and
$$\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_ix_j = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_jx_k =$$
$$\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP}x_kx_i = NP\left(\sum_{i=1}^{NP}x_i\right)^2$$

So,
$$E(U_m^2)=(1-CR)x_m^2 +$$
$$CR(2F^2+1)[\frac{1}{NP}\sum_{i=1}^{NP}x_i^2 - (\frac{1}{NP}\sum_{i=1}^{NP}x_i)^2] + CR(\frac{1}{NP}\sum_{i=1}^{NP}x_i)^2$$

$$\Rightarrow E(U_m^2)=(1-CR)x_m^2 +$$
$$CR(2F^2+1)Var(x)+CRx_{av}^2$$

Where, $Var(x)=\frac{1}{NP}\sum_{i=1}^{NP}x_i^2 - (\frac{1}{NP}\sum_{i=1}^{NP}x_i)^2$ and

$x_{av}=\frac{1}{NP}\sum_{i=1}^{NP}x_i$, and hence the proof.

Theorem 2: If the DE population may be modeled as a continuous-time, dynamic system, then the expectation value of the velocity of an individual point on the fitness landscape may be given as:

$$E(\frac{dx_m}{dt}) = -\frac{k}{8}CR\{(2F^2+1)Var(x) + (x_{av}-x_m)^2\}f^{'}(x_m) + \frac{1}{2}CR(x_{av}-x_m) \qquad (8)$$

Proof: Let us assume that mutation and crossover occur in unit time to give rise to off offspring. In selection x_m is replaced by U_m if the objective function value for $x = U_m$ is less than or equal to that for $x = x_m$.This decision-making is performed using unit step function, which is defined as follows:

$u(p) = 1$ If $p \geq 0$

=0 otherwise.

Now, let at time t position of mth trial solution be x_m and at $t + \Delta t$ it is changed to $x_m + \Delta x_m$

Then, $\frac{\Delta x_m}{\Delta t} = u[\frac{f(x_m) - f(x_m + \Delta x_m)}{\Delta t}](U_m - x_m)$

$$\Rightarrow \frac{\Delta x_m}{\Delta t} = u[\frac{f(x_m) - f(x_m + \Delta x_m)}{\Delta x_m}\frac{\Delta x_m}{\Delta t}](U_m - x_m)$$

$$\Rightarrow \underset{\Delta t \to 0}{Lt}\frac{\Delta x_m}{\Delta t} = \underset{\Delta t \to 0}{Lt}\, u[\frac{f(x_m) - f(x_m + \Delta x_m)}{\Delta x_m}\frac{\Delta x_m}{\Delta t}](U_m - x_m)$$

$$\Rightarrow \underset{\Delta t \to 0}{Lt}\frac{\Delta x_m}{\Delta t} = \underset{\Delta t \to 0}{Lt}\, u[-\frac{f(x_m + \Delta x_m) - f(x_m)}{\Delta x_m}\frac{\Delta x_m}{\Delta t}](U_m - x_m)$$

$$\therefore \frac{dx_m}{dt} = u[-f^{'}(x_m)\frac{dx_m}{dt}](U_m - x_m) \qquad (9)$$

Now, we have to replace unit step function by logistic function to carry out the analysis.

Ideally, $u(p) = \underset{k \to \infty}{lt}\frac{1}{1 + e^{-kp}}$

Let us take a moderate value of k for analysis. Here we commit some error due to approximation.

$$u(p) \approx \frac{1}{1 + e^{-kp}}$$

Now, if p is very small. Then, $e^{-kp} \approx 1 - kp$ [neglecting higher order terms]

$$\therefore u(p) \approx \frac{1}{1 + e^{-kp}} \approx \frac{1}{2 - kp} = \frac{1}{2}(1 - \frac{kp}{2})^{-1}$$

Again assuming p to be very small and neglecting higher order terms in expansion of $(1 - \frac{kp}{2})^{-1}$ we obtain,

$$u(p) \approx \frac{1}{2} + \frac{k}{4}p \qquad (10)$$

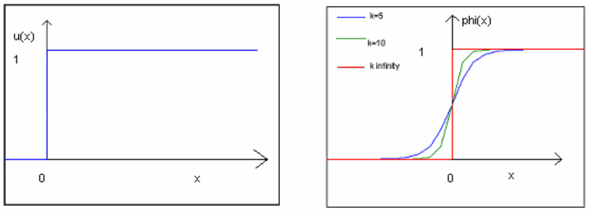


Figure 3: The unit step and the logistic functions. Now, the population has a small divergence.

$\therefore U_m - x_m$ is not very large and hence $\left|\frac{dx_m}{dt}\right|$ is small.

Also we have assumed that fitness landscape has a moderate slope i.e. $\left|f^{'}(x_m)\right|$ is also small, which in turn suggests that $\left|f^{'}(x_m)\frac{dx_m}{dt}\right|$ is small. Thus from (9) and (10) we get,

$$\frac{dx_m}{dt} = [\frac{1}{2} - \frac{k}{4}f^{'}(x_m)\frac{dx_m}{dt}](U_m - x_m)$$

$$\Rightarrow \frac{dx_m}{dt} = \frac{\frac{1}{2}(U_m - x_m)}{1 + \frac{k}{4}f^{'}(x_m)(U_m - x_m)} \qquad (11)$$

Now, $\left|\frac{k}{4}f^{'}(x_m)(U_m - x_m)\right|$ is small

$\therefore [1 + \frac{k}{4}f^{'}(x_m)(U_m - x_m)]^{-1} \approx 1 - \frac{k}{4}f^{'}(x_m)(U_m - x_m)$ from (11) we get,

$$\frac{dx_m}{dt} = -\frac{k}{8}(U_m - x_m)^2 f^{'}(x_m) + \frac{U_m - x_m}{2} \qquad (12)$$

Now U_m is a random variable. $\therefore \frac{dx_m}{dt}$, which is a function of U_m, is also a random variable.

Let us try to compute its expected value.

$$E(\frac{dx_m}{dt}) = -\frac{k}{8}f^{'}(x_m)E(U_m - x_m)^2 + \frac{1}{2}E(U_m - x_m)$$

$$\Rightarrow E(\frac{dx_m}{dt}) = -\frac{k}{8}f^{'}(x_m)[E(U_m{}^2) + x_m{}^2 - 2x_m E(U_m)] + \frac{1}{2}[E(U_m) - x_m] \qquad (13)$$

Substituting values of $E(U_m), E(U_m{}^2)$ from equation (7) and (8) to equation (13) we get,

$$E(\frac{dx_m}{dt}) = -\frac{k}{8}CR\{(2F^2+1)Var(x) + (x_{av} - x_m)^2\}f^{'}(x_m) + \frac{1}{2}CR(x_{av} - x_m)$$

and hence the proof.

Theorem 3: Let x_{av} denote the centroid (mean of all points) of the current population and $x_{av} = \sum_{m=1}^{NP} x_m$. Also let us denote $\varepsilon_m = x_{av} - x_m$ = deviation of individual from average. Then expected velocity of the centroid of the population may be given by,

$$E(\frac{dx_{av}}{dt}) = -\frac{k}{8}CR(2F^2+1)Var(x)f_{av}{}' \\ -\frac{k}{8}CR(\frac{1}{N}\sum_{m=1}^{NP}\varepsilon_m{}^2 f'(x_{av}+\varepsilon_m)) \qquad (14)$$

Proof: Now, $x_{av} = \frac{1}{NP}\sum_{i=1}^{NP} x_i = \frac{1}{NP}\sum_{m=1}^{NP} x_m$

$$\Rightarrow \frac{dx_{av}}{dt} = \frac{d}{dt}(\frac{1}{NP}\sum_{m=1}^{NP} x_m) = \frac{1}{NP}\sum_{m=1}^{NP}\frac{dx_m}{dt}$$

$$\Rightarrow E(\frac{dx_{av}}{dt}) = E(\frac{1}{NP}\sum_{m=1}^{NP}\frac{dx_m}{dt}) = \frac{1}{NP}\sum_{m=1}^{NP}E(\frac{dx_m}{dt})$$

$$E(\frac{dx_{av}}{dt}) = \frac{1}{NP}(-\frac{k}{8}CR\sum_{m=1}^{NP}\{(2F^2+1)Var(x)+(x_{av}-x_m)^2\}f'(x_m) \\ +\frac{1}{2}CR\sum_{m=1}^{NP}(x_{av}-x_m))$$

Now, $\sum_{m=1}^{NP}(x_{av}-x_m) = 0$

$$\therefore E(\frac{dx_{av}}{dt}) \\ = \frac{1}{NP}(-\frac{k}{8}CR\sum_{m=1}^{NP}\{(2F^2+1)Var(x)+(x_{av}-x_m)^2\}f'(x_m)$$

Let us denote $\varepsilon_m = x_{av} - x_m$ = deviation of individual from average.

$$\therefore E(\frac{dx_{av}}{dt}) = -\frac{k}{8}CR(2F^2+1)Var(x)(\frac{1}{NP}\sum_{m=1}^{NP}f'(x_m)) \\ -\frac{k}{8}CR(\frac{1}{NP}\sum_{m=1}^{NP}\varepsilon_m{}^2 f'(x_m))$$

$$\Rightarrow E(\frac{dx_{av}}{dt}) = -\frac{k}{8}CR(2F^2+1)Var(x)\left(\frac{1}{NP}\sum_{m=1}^{NP}f'(x_m)\right) \\ -\frac{k}{8}CR(\frac{1}{NP}\sum_{m=1}^{NP}\varepsilon_m{}^2 f'(x_{av}+\varepsilon_m))$$

$$\therefore E(\frac{dx_{av}}{dt}) = -\frac{k}{8}CR(2F^2+1)Var(x)f_{av}{}' \\ -\frac{k}{8}CR(\frac{1}{N}\sum_{m=1}^{NP}\varepsilon_m{}^2 f'(x_{av}+\varepsilon_m)) \qquad (15)$$

Where, $f'_{av} = \frac{1}{NP}\sum_{m=1}^{NP}f'(x_m)$ = average of the gradients for trial solution points on fitness landscape. This completes the proof.

3.3 Mathematical Analogy between DE and Classical Gradient Descent Search

From (14) we may infer

$$E(\frac{dx_m}{dt}) = -\alpha_{DE}f'(x_m)+\beta_{DE} \qquad (16)$$

where $\alpha_{DE} = -\frac{k}{8}CR\{(2F^2+1)Var(x)+(x_{av}-x_m)^2\}$

and $\beta_{DE} = \frac{1}{2}CR(x_{av}-x_m)$

Expression (16) has similarity with characteristic equation of classical gradient descent algorithm. So, DE may be termed as a modified gradient descent search under certain constraints. Here $-\alpha_{DE}f'(x_m)$ term on the R.H.S. of (16) is responsible for gradient descent, whereas β_{DE} represents a component of velocity of a trial solution towards the mean vector (center of mass) of the population.

Evidently, when $f'(x_m) \to 0$,

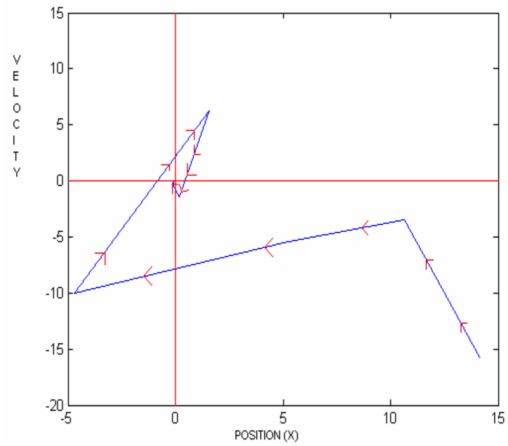
$$E(\frac{dx_m}{dt}) \approx \beta_{DE} = \frac{1}{2}CR(x_{av}-x_m)$$

Now, if the population is not much scattered $(x_{av}-x_m)$ is very small. $\therefore E(\frac{dx_m}{dt}) \to 0$

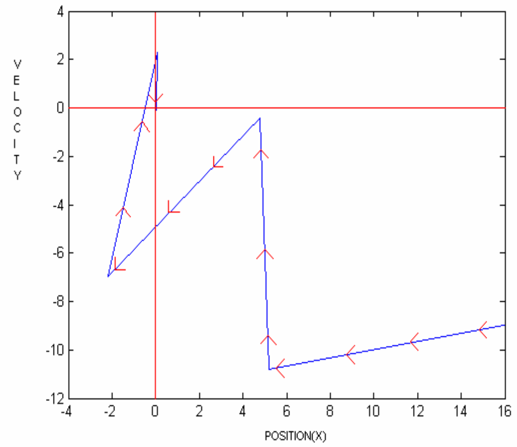
So, it converges towards optima. Under these circumstances, the population members are confined within a small region of the search space and hence $\mathrm{var}(x) \to 0$, $x_{av}-x_m \to 0$ and also $\varepsilon_m \to 0$. Thus, from (16) and (17) we get $E(\frac{dx_m}{dt}) \to 0$ and $E(\frac{dx_{av}}{dt}) \to 0$.

IV. Experiment results

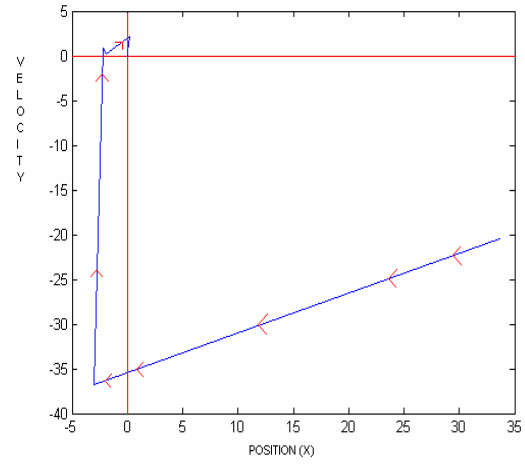
In this section we provide the phase plots ($v = \frac{dx}{dt}$ versus x plots) for DE/rand/1/bin, which supports the theoretical results derived in the earlier section. A population of 11 vectors is taken to optimize the single dimensional sphere function $f(x) = x^2$ using the DE Algorithm. In Figure 4 four phase-trajectories are illustrated for the median vector (when the population is ranked according to the final fitness values of the vectors) over four independent runs (with different initial populations). These phase trajectories verifies our theoretical finding that near an optima, the expected velocity $E(\frac{dx}{dt})$ of individual member of population gradually approaches zero.



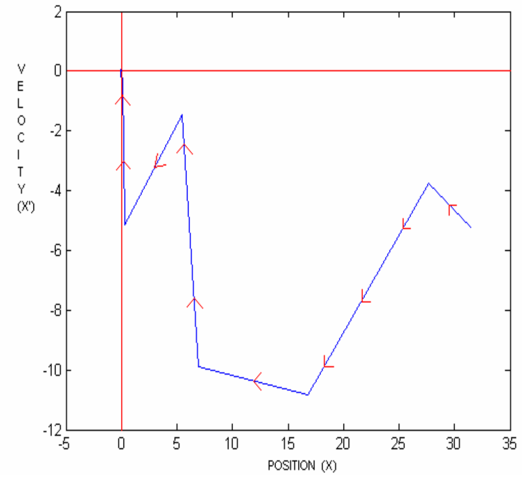
(a)



(b)



(c)



(d)

Figure 4 (a-d): Phase trajectory of the median order vector (in a population of size *NP* = 11) for 4 independent runs (with different seeds for the random number generator)

V. Conclusions

Differential Evolution (DE) has been regarded as a competitive form of Evolutionary Algorithm for function optimization problems in recent years. In this article we provide a simple analysis of the evolutionary dynamics undertaken by each of the population members in DE/rand/1/bin, which appears as one of the most popular and widely used variant of the DE. The distribution of the difference vector under convolution is first considered to explain why DE works and the implications are given for optimization. In addition, DE has been modeled as a modified gradient descent algorithm, which should provide the further insight into the evolution of the DE population.

Future research may focus on analyzing the stability of the DE dynamics based on a stochastic Lyapunov energy function approach [23].

References

[1] Storn, R., Price, K.: Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, *Technical Report TR-95-012, ICSI*, 1995. http.icsi.berkeley.edu/~storn/litera.html

[2] Price, K., Storn, R., and Lampinen, J.: Differential Evolution - A Practical Approach to Global Optimization, Springer, Berlin, 2005.

[3] Holland, J. H.: *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, (1975).

[4] Kennedy, J. and Eberhart, R.: Particle swarm optimization, In Proceedings of IEEE International conference on Neural Networks. 1942-1948, (1995).

[5] Vesterstrøm, J., Thomson, R.: A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems, Proc. Sixth Congress on Evolutionary Computation (CEC-2004). IEEE Press, (2004).

[6] Rogalsky, T., Derksen, R.W., and Kocabiyik, S.: Differential Evolution in Aerodynamic Optimization, In: Proc. of 46th Annual Conf of Canadian Aeronautics and Space Institute (1999) 29-36.

[7] Joshi, R., and Sanderson, A.C.: Minimal Representation Multisensor Fusion Using Differential Evolution, IEEE Trans. Systems, Man, and Cybernetics, Part A, vol 29, no. 1, pp. 63-76, 1999.

[8] Das, S., Konar, A.: Design of Two dimensional IIR Filters with Modern Search Heuristics: A Comparative Study, International Journal of Computational Intelligence and Applications, World Scientific Press, 2008 (in press).

[9] Omran, M., Engelbrecht, A, P., Salman, A.: Differential Evolution Methods for Unsupervised Image Classification, Proc. Seventh Congress on Evolutionary Computation (CEC-2005). IEEE Press.

[10] Das, S., Abraham, A., and Konar, A.: Adaptive Clustering Using Improved Differential Evolution Algorithm, IEEE Transactions on Systems, Man and Cybernetics – Part A, IEEE Press, USA, Volume 38, No.1, pp. 218-237, 2008.

[11] Lampinen, J.: A Bibliography of Differential Evolution Algorithm. Technical Report. Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing, 1999. http://www.lut.fi/~jlampine/debiblio.htm.

[12] Gamperle, R., Muller, S. D. and Koumoutsakos, A.: Parameter Study for Differential Evolution," in WSEAS NNA-FSFS-EC 2002, Interlaken, Switzerland, Feb. 11-15 2002. WSEAS.

[13] J. Ronkkonen, S. Kukkonen, and K. V. Price. Real Parameter Optimization with Differential Evolution, in The 2005 IEEE Congress on Evolutionary Computation (CEC2005), vol. 1, pages 506 – 513. IEEE Press, 2005.

[14] Liu, J., Lampinen, J.: A Fuzzy Adaptive Differential Evolution Algorithm, In Soft computing- A Fusion of Foundations, Methodologies and Applications, v.9 n.6, p.448-462, April 2005.

[15] Zaharie, D.: Control of Population Diversity and Adaptation in Differential Evolution Algorithms, In Matousek, D., Osmera P. (eds.), Proc. of MENDEL 2003, 9th International Conference on Soft Computing, Brno, Czech Republic, pp. 41-46, 2003.

[16] Zaharie D. and Petcu D.: Adaptive Pareto Differential Evolution and its Parallelization, Proc. of 5th International Conference on Parallel Processing and Applied Mathematics, Czestochowa, Poland, 2003.

[17] Abbass, H.: The Self-Adaptive Pareto Differential Evolution Algorithm, In: Proceedings of the 2002 Congress on Evolutionary Computation (2002) 831-836.

[18] Omran, M., Salman, A., Engelbrecht, A, P.: Self-adaptive Differential Evolution, Computational Intelligence And Security, PT 1, Proceedings Lecture Notes In Artificial Intelligence 3801: 192-199 2005.

[19] Brest, J., Greiner, S., Boskovic, B., Mernik, M. and Zumer, V., Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems, IEEE Transactions on Evolutionary Computation, Vol. 10, and Issue: 6, Dec. 2006, pp. 646 – 657.

[20] Das, S., Konar, A., Chakraborty, U.K., Two improved differential evolution schemes for faster global search, ACM-SIGEVO Proceedings of GECCO, Washington D.C., June 2005, pp. 991-998.

[21] Zaharie, D.: Control of Population Diversity and Adaptation in Differential Evolution Algorithms, in Matousek, R. and Osmera, P. (eds.), Proc. of Mendel 2003, 9th International Conference on Soft Computing, Brno, Czech Republic, 2003, pp. 41-46.

[22] Snyman, J. A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing (2005).

[23] Semenov, M. A. and Terkel, D. A.: Analysis of Convergence of an Evolutionary Algorithm with Self-Adaptation Using a Stochastic Lyapunov Function, Evolutionary Computation, 363-379, (2003).