On Stability and Convergence of the Population-Dynamics in Differential Evolution

Sambarta Dasgupta [1], Swagatam Das[1], Arijit Biswas [1] and Ajith Abraham[2]

[1] Dept. of Electronics and Telecommunication Engg,
Jadavpur University, Kolkata, India

[2]Center of Excellence for Quantifiable Quality of Service,
Norwegian University of Science and Technology, Norway
ajith.abraham@ieee.org

Abstract —Theoretical analysis of the dynamics of evolutionary algorithms is believed to be very important to understand the search behavior of evolutionary algorithms and to develop more efficient algorithms. In this paper we investigate the dynamics of a canonical Differential Evolution (DE) algorithm with DE/rand/1 type mutation and binomial crossover. Differential Evolution (DE) is well-known as a simple and efficient algorithm for global optimization over continuous spaces. Since its inception in 1995, DE has been finding many important applications in real-world optimization problems from diverse domains of science and engineering. The paper proposes a simple mathematical model of the underlying evolutionary dynamics of a one-dimensional DE-population. The model shows that the fundamental dynamics of each search-agent (parameter vector) in DE employs the gradient-descent type search strategy (although it uses no analytical expression for the gradient itself), with a learning rate parameter that depends on control parameters like scale factor F and crossover rate CR of DE. The stability and convergence-behavior of the proposed dynamics is analyzed in the light of Lyapunov's stability theorems very near to the islolated equilibrium points during the final stages of the search. Empirical studies over simple objective functions are conducted in order to validate the theoretical analysis.

1. Introduction

In Artificial Intelligence (AI), an evolutionary algorithm (EA) is a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. An EA uses some mechanisms inspired by biological evolution: reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the objective function (also known as cost function or fitness function in literature) determines the environment within which the solutions "live". Evolution of the population then takes place after the repeated application of the above operators. Theoretical analysis of evolutionary algorithms has received increasing attention in the recent years [1]. A few examples of interesting topics are, among many others, convergence analysis [2, 3], dynamics of evolution strategies [4], genetic algorithms [5, 6], and analysis of average computational time [7]. However, the dynamics of EAs during optimization and the roles of each genetic operator are still unclear and stand as a significant research problem at its own right. The analysis of dynamics of EAs is very helpful not only to understand working mechanism of EAs [8] but also to improve

performance of EAs and to propose new algorithms [9] because the solution of an optimizer is the result of the dynamics of EAs. Recently the convergence and stability of another state-of-the-art real parameter optimization technique called Particle Swarm Optimization (PSO) [10] has been undertaken by Trelea [11], Poli *et al.* and Kadirkamanathan *et al.* [12]. In [12] the authors have used the Lyapunov stability theorems to judge the stability and convergence of the search-agents (called *particles*) in PSO.

Since its inception in 1995, a good volume of research has been undertaken in order to improve the performance of the DE algorithm over complex and multi-modal fitness landscapes. There exists a plethora of works concerning the empirical study of parameter selection and tuning process in DE [13-18] and its application to optimization problems [19, 20]. Little research has, however, 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. Some significant work in this direction was reported in [21, 22, and 13] by Zaharie, where she theoretically analyzed the influence of the variation operators and their parameters on the expected population variance. Zaharie [21] showed that the expected population variance (after applying mutation and crossover or recombination) of DE is greater than that of the ES algorithm analyzed in [23]. This finding could explain to some extent the excellent performance of DE on certain test functions. The works of Zaharie [21] however did not focus on modeling DE as a dynamical system and analyzing its stability and convergence properties from there. Neither have they accounted for the control parameters that govern the final convergence of all the DE vectors to an isolated optimum. The study undertaken in this paper, attempts to make a humble contribution in these contexts.

In this paper, we provide a simple mathematical model of DE/rand/1/bin scheme (which is the most popular variant of DE family [19]). Each parameter vector is modeled as a search-agent moving under the directives of the DE algorithm, over the fitness landscape in continuous time searching for the optima. The *survival-of-the-fittest* type selection mechanism in DE has been modeled with the unit step function and then approximated using the continuous logistic function in order to apply standard calculus techniques for further analysis. Our model attempts to find out an expected *velocity* of each parameter vector towards the optimum over successive time-steps. It also tries to relate the search mechanism of DE with that of the classical gradient descent search technique [24, 25]. A few earlier attempts to hybridize DE and GA with the gradient descent techniques, can be found in [26, 27]. Our model, however, indicates that the fundamental equation governing the expected velocity of the search agents over a continuous fitness landscape in DE has itself got a striking resemblance with that of the steepest descent search. The term analogous to the learning rate in steepest descent, for DE, becomes a function of control parameters like *F* and *CR*. Our analysis indicates that DE employs some kind of estimation of the gradient (*not* any analytical expression of the gradient itself though) in order to direct its search towards the optima. Based on the proposed model, the stability and convergence of the DE-vectors in a small neighborhood centered on an isolated equilibrium point, has been investigated with the Lyapunov stability theorems [28, 29]. The Lyapunov's theorems are widely used in nonlinear system analysis to determine the necessary conditions for stability of a dynamical system. The theoretical results, presented in this context, show that the crossover rate *CR* mainly governs the time taken by a single search-agent to converge to an arbitrarily small neighborhood around the optimum. Future works may consider some

special tuning mechanisms for *CR* that facilitate quick convergence to an equilibrium (which is usually an optimum during the final stages of search). Simple experimental results have also been provided to support the theoretical claims made in the paper. In the appendix A.2 we provide an equivalent mathematical model for the DE/current-to-rand/1 scheme which uses arithmetic recombination operator so that the trial vectors may remain rotationally invariant [19].

2. The Classical DE Algorithm – an Outline

Like any other evolutionary algorithm, DE starts with a population of *NP D*-dimensional parameter vectors representing the candidate solutions. We shall denote subsequent generations in DE by $G = 0,1..., G_{max}$. Since the parameter 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 as:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, ..., x_{D,i,G}].$$ (1)

For each search-variable of the problem, there may be a user-specified range within which value of the variable should lie for more accurate search results at less computational cost. The initial population (at $G = 0$) should cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds: $\vec{X}_{min} = \{x_{1,min}, x_{2,min}, ..., x_{D,min}\}$ and $\vec{X}_{max} = \{x_{1,max}, x_{2,max}, ..., x_{D,max}\}$. Hence we may initialize the *j*-th component of the *i*-th vector as,

$$x_{j,i,0} = x_{j,min} + rand_{i,j}(0,1) \cdot (x_{j,max} - x_{j,min}),$$ (2)

where $rand_{i,j}(0,1)$ is a uniformly distributed random number lying between 0 and 1 and is instantiated independently for each component of each vector. Following steps are taken next: mutation, crossover, and selection, which are explained in the following subsections.

a) Mutation:

After initialization, DE creates a *donor* vector $\vec{V}_{i,G}$ corresponding to each population member or *target* vector $\vec{X}_{i,G}$ in the current generation through mutation. It is the method of creating this donor vector, which differentiates between the various DE schemes. 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 below:

$$\text{"DE/rand/1": } \vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}).$$ (3)

$$\text{"DE/best/1": } \vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}).$$ (4)

$$\text{"DE/target-to-best/1": } \vec{V}_{i,G} = \vec{X}_{i,G} + F \cdot (\vec{X}_{best,G} - \vec{X}_{i,G}) + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}).$$ (5)

"DE/best/2": $\vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r_1^i,G} - \vec{X}_{r_2^i,G}) + F \cdot (\vec{X}_{r_3^i,G} - \vec{X}_{r_4^i,G})$. (6)

"DE/rand/2": $\vec{V}_{i,G} = \vec{X}_{r_1^i,G} + F \cdot (\vec{X}_{r_2^i,G} - \vec{X}_{r_3^i,G}) + F \cdot (\vec{X}_{r_4^i,G} - \vec{X}_{r_5^i,G})$. (7)

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*], and all are different from the index *i*. These indices are randomly generated once for each donor vector. The scaling factor *F* is a positive control parameter for scaling the difference vectors. $\vec{X}_{best,G}$ is the best individual vector with the best fitness (i.e. lowest objective function value for minimization problem) 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.

b) Crossover:

To increase the potential diversity of the population, a crossover operation comes into play after generating the donor vector through mutation. The DE family of algorithms can use two kinds of crossover schemes - *exponential* and *binomial* [1-3]. The donor vector exchanges its components with the target vector $\vec{X}_{i,G}$ under this operation to form the *trial* vector $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, ..., u_{D,i,G}]$. In exponential crossover, we first choose an integer *n* randomly among the numbers $[1, D]$. This integer acts as a starting point in the target vector, from where the crossover or exchange of components with the donor vector starts. We also choose another integer *L* from the interval $[1, D]$. *L* denotes the number of components; the donor vector actually contributes to the target. After a choice of *n* and *L* the trial vector:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, ..., \langle n-L+1 \rangle_D \\ x_{j,i,G}, & \text{for all other } j \in [1, D] \end{cases}$$ (8)

where the angular brackets $\langle \ \rangle_D$ denote a modulo function with modulus *D*. The integer *L* is drawn from $[1, D]$ according to the following lines of pseudo code.

$L = 0$;
DO
{
    $L=L+1$;
} WHILE $(((rand(0,1) < Cr) \text{ AND } (L < D))$;

Hence in effect, probability $(L \geq \upsilon) = (Cr)^{\upsilon-1}$ for any $\upsilon > 0$. '*Cr*' is called *crossover rate* and it appears as a control parameter of DE just like *F*. For each donor vector, a new set of *n* and *L* must be chosen randomly as shown above.

On the other hand, binomial crossover is performed on each of the *D* variables whenever a randomly picked number between 0 and 1 is less than or equal to 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,

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } (rand_{i,j}(0,1) \leq Cr \text{ or } j = j_{rand}) \\ \\ x_{j,i,G}, & \text{otherwise} \end{cases} \tag{9}$$

where $rand_{i,j}(0,1) \in [0,1]$ is a uniformly distributed random number, which is called a new for each *j*-th component of the *i*-th parameter vector. $j_{rand} \in [1,2,....,D]$ is a randomly chosen index, which ensures that $\vec{U}_{i,G}$ gets at least one component from $\vec{V}_{i,G}$.
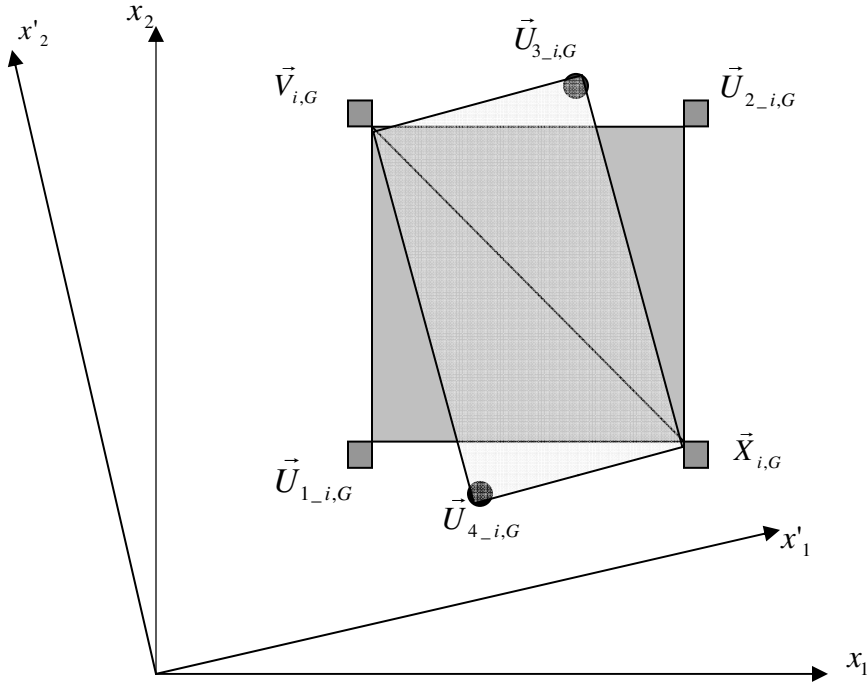
Fig. 1: Change of the trial vectors generated through the crossover operation described in equation (9) due to rotation of the coordinate system

The crossover operation described in equation (9) is basically a discrete recombination [3]. Figure 1 illustrates a two-dimensional example of recombining the parameters of two vectors $\vec{X}_{i,G}$ and $\vec{V}_{i,G}$, according to this crossover operator, where the potential trial vectors are generated at the corners of a rectangle. As can be seen from Figure 1, discrete recombination is a rotationally variant operation. Rotation transforms the coordinates of both vectors and thus changes the shape of rectangle as shown in Figure 1. Consequently, the potential location of the trial vector

moves from the possible set $(\vec{U}_{1\_i,G}, \vec{U}_{2\_i,G})$ to $(\vec{U}_{3\_i,G}, \vec{U}_{4\_i,G})$. To overcome this limitation, a new trial vector generation strategy 'DE/current-to-rand/1' is proposed in [18], which replaces the crossover operator prescribed in equation (9) with the rotationally invariant arithmetic crossover operator to generate the trial vector $\vec{U}_{i,G}$ by linearly combining the target vector $\vec{X}_{i,G}$ and the corresponding donor vector $\vec{V}_{i,G}$ as follows:

$$\vec{U}_{i,G} = \vec{X}_{i,G} + K \cdot (\vec{V}_{i,G} - \vec{X}_{i,G}).$$

Now incorporating equation (3) in (10) we have:

$$\vec{U}_{i,G} = \vec{X}_{i,G} + K \cdot (\vec{X}_{r_1,G} + F \cdot (\vec{X}_{r_2,G} - \vec{X}_{r_3,G}) - \vec{X}_{i,G}).$$

which further simplifies to:

$$\vec{U}_{i,G} = \vec{X}_{i,G} + K \cdot (\vec{X}_{r_1,G} - \vec{X}_{i,G}) + F' \cdot (\vec{X}_{r_2,G} - \vec{X}_{r_3,G}), \tag{10}$$

where K is the combination coefficient, which has been shown [18] to be effective when it is chosen with a uniform random distribution from [0, 1] and $F' = K.F$ is a new constant here.

c) Selection:

To keep the population size constant over subsequent generations, the next step of the algorithm calls for *selection*. This operation determines which one of the target and the trial vector survives to the next generation i.e. at $G = G + 1$. The selection operation may be outlined as:

$$\vec{X}_{i,G+1} = \vec{U}_{i,G}, \qquad \text{if } f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G})$$

$$= \vec{X}_{i,G}, \qquad \text{if } f(\vec{U}_{i,G}) > f(\vec{X}_{i,G}) \tag{11}$$

where $f(\vec{X})$ is the function to be minimized. So if the new trial vector yields a lower value of the objective function, it replaces the corresponding target vector in the next generation; otherwise the target is retained in the population. Hence the population either gets better (w.r.t. the minimization of the objective function) or remains constant, but never deteriorates. The complete pseudo code has been provided below:

<u>Pseudo-code for the DE algorithm family</u>

Step 1. Set the generation number $G = 0$ and randomly initialize a population of NP individuals $P_G = \{\vec{X}_{1,G}, \ldots, \vec{X}_{NP,G}\}$ with $\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \ldots, x_{D,i,G}]$ and each individual uniformly distributed in the range $[\vec{X}_{\min}, \vec{X}_{\max}]$,

where $\vec{X}_{\min} = \{x_{1,\min}, x_{2,\min}, \ldots, x_{D,\min}\}$ and $\vec{X}_{\max} = \{x_{1,\max}, x_{2,\max}, \ldots, x_{D,\max}\}$ with $i = [1,2,\ldots, NP]$.

Step 2. WHILE stopping criterion is not satisfied

DO

FOR $i = 1$ to NP //do for each individual sequentially

Step 2.1 *Mutation Step*

Generate a donor vector $\vec{V}_{i,G} = \{v_{1,i,G},........,v_{D,i,G}\}$ corresponding to the *i*-th target vector $\vec{X}_{i,G}$ via one of the different mutation schemes of DE (equations (3) to (7)).

Step 2.2 *Crossover Step*

Generate a trial vector $\vec{U}_{i,G} = \{u_{1,i,G},........,u_{D,i,G}\}$ for the *i*-th target   vector $\vec{X}_{i,G}$ through binomial crossover (equation (9)) or exponential crossover (equation (8)) or through the arithmetic crossover (equation (10)).

Step 2.3 *Selection Step*

Evaluate the trial vector $\vec{U}_{i,G}$

IF $f(\vec{U}_{i,G}) \leq f(\vec{X}_{i,G})$, THEN $\vec{X}_{i,G+1} = \vec{U}_{i,G}$, $f(\vec{X}_{i,G+1}) = f(\vec{U}_{i,G})$

    IF $f(\vec{U}_{i,G}) < f(\vec{X}_{best,G})$, THEN $\vec{X}_{best,G} = \vec{U}_{i,G}$, $f(\vec{X}_{best,G}) = f(\vec{U}_{i,G})$

    END IF

    END IF

ELSE $\vec{X}_{i,G+1} = \vec{X}_{i,G}$, $f(\vec{X}_{i,G+1}) = f(\vec{X}_{i,G})$

END FOR

Step 2.4 *Increase the Generation Count* $G = G + 1$

END WHILE

3. The Mathematical Model of the Population-Dynamics in DE

3.1 Assumptions

Suppose $f(x) : \Re \to \Re$ be the function of a single variable x and 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 objective function $f(x)$ is assumed to be of class C^2 (please note that a function f is said to be of class C^k if the derivatives $f^1, f^2,..., f^{(k)}$ exist and are continuous [30]). Also let $f(x)$ be Lipschitz continuous [31], that is given any two points x and $y \in \Re$, f satisfies the Lipschitz condition

$|f(x) - f(y)| \le L.|x - y|$ with the value of Lipschitz constant $L \le 1$ in the region of the fitness landscape (i.e. $f(x)$ is actually a contraction mapping), where our analysis applies. Moreover the objective function is unimodal in the region of interest.

Explanation: Goal of our work is to analyze stability of DE population during the final stages of the search. To study stability and convergence, we assume that already due to DE-type mutation and crossover operations, the population have crowded into a small neighborhood surrounding an optimum. The above hypotheses of regularity made on the objective function indicate that the value of the gradient becomes small under such conditions. This facilitates the analysis as will be evident from appendix A.1.

ii) The population of NP trial solutions is limited within a small region i.e. individual trial solutions are located very close to each other. According to [21] and [32], this is usually the case during the later stages of the search, when the parameter vectors concentrate in a compact cluster around the global optimum, and especially when the scaling factor F is set at 0.5. Please note that the justification for this assumption has been provided in appendix A.1.

iii) Dynamics is modeled assuming the vectors as search-agents moving in continuous time.

Explanation: A population member (trial solution) may change its position with time, which in turn may change its objective function value. Computer simulation of the algorithm, however, proceeds through discrete generations. Certain amount of processor time is elapsed between two successive generations. Thus, in the virtual world of simulations, a solution can change its position only at certain discrete time instants. This change of position is ideally instantaneous. In between two successive generations it remains practically immobile. Without losing generality, the time between two successive generations may be defined as unit time for the following derivation. But we have assumed continuous time to model it as a dynamic physical system, where it is not possible to have instantaneous position change. Hence, we assume within two successive generations the position shifts continuously and linearly. In practice, time between two successive generations i.e. computational time of generation is very small. This ensures that the approximation is fairly good.

Figure 2 depicts a favorable portion of a one- dimensional arbitrary objective function for our analysis.
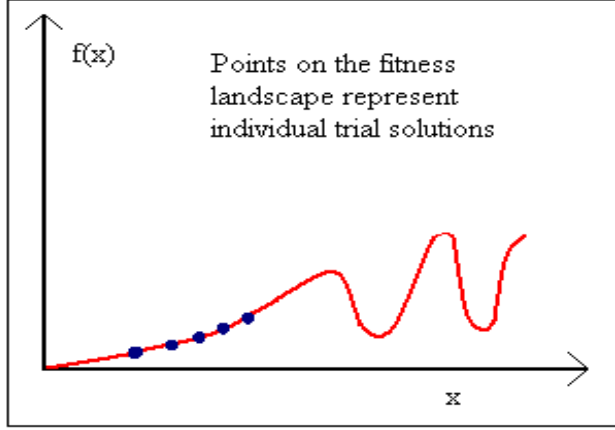
Fig. 2: A DE population dispersed on a one-dimensional arbitrary fitness landscape

3.2 Modeling Different Steps of DE

Let x_m be the m-th individual of the population, where $m = 1,2,..., NP$. During an generation of DE, it undergoes three steps: mutation, crossover, and selection. Each step is modeled individually and finally they are merged to get a generalized expression for the expectation value of the trial vector formed this way. 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:
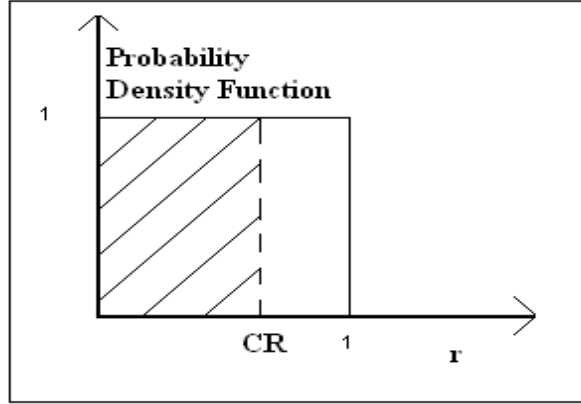
Figure 3: Probability density function of r

Theorem 1: The expected value of a trial vector U_m corresponding to the target vector x_m is given by

$$E(U_m) = (1 - CR)x_m + CRx_{av} \tag{12}$$

and the expected value of U_m^2 is then given by,

$$E(U_m^2) = (1 - CR)x_m^2 + CR(2F^2 + 1)Var(x) + CRx_{av}^2 \tag{13}$$

where x_{av} is the mean of the population i.e. $x_{av} = \dfrac{1}{NP}\displaystyle\sum_{i=1}^{NP} x_i$ and $Var(x)$ is the variance of the target population.

Proof: From Figure 3, probability of the event, $r \leq CR = P(r \leq CR) =$ Area of the shaded region.

$$= 1 \times CR = CR$$

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

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

$$\therefore E(U_m) = P(r > CR)x_m +$$

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

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}.

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

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

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

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

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

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

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

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

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

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

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,

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

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

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

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

and $\because \displaystyle\sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP} x_i x_j = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP} x_j x_k = \sum_{i=1}^{NP}\sum_{j=1}^{NP}\sum_{k=1}^{NP} x_k x_i = NP\left(\sum_{i=1}^{NP} x_i\right)^2$, we have,

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

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

Where, $Var(x) = \dfrac{1}{NP}\displaystyle\sum_{i=1}^{NP} x_i{}^2 - (\dfrac{1}{NP}\sum_{i=1}^{NP} x_i)^2$ and $x_{av} = \dfrac{1}{NP}\displaystyle\sum_{i=1}^{NP} x_i$, and hence the proof.

Remark: Note that if $CR = 0$, $E(U_m) = x_m$ and $E(U_m^2) = x_m^2$, i.e. the expected value of the trial vector remains same as that of the target vector. This is intuitively simple as for $CR = 0$, the trial vector can only inherit from the target but nothing from the donor/mutant vector. Again if $CR = 1$, $E(U_m) = x_{av}$ and $E(U_m^2) = (2F^2+1).Var(x) + x_{av}^2$. Clearly if $0 < CR < 1$, the expected value of the trial vector lies in between x_m and x_{av}.

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:

$$E(\frac{dx_m}{dt}) = -\frac{k}{8}CR\{(2F^2+1)Var(x) + (x_{av} - x_m)^2\}f^{'}(x_m) + \frac{1}{2}CR(x_{av} - x_m) \tag{14}$$

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 Heaviside's unit step function [33] , which is defined as follows:

$$u(p) = 1 \quad if \quad p \geq 0$$
$$= 0 \quad otherwise.$$

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

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

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

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

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

$$\therefore \frac{dx_m}{dt} = u[-f'(x_m)\frac{dx_m}{dt}](U_m - x_m) \tag{15}$$

Now, we have to replace unit step function by logistic function to carry out the analysis. Ideally,
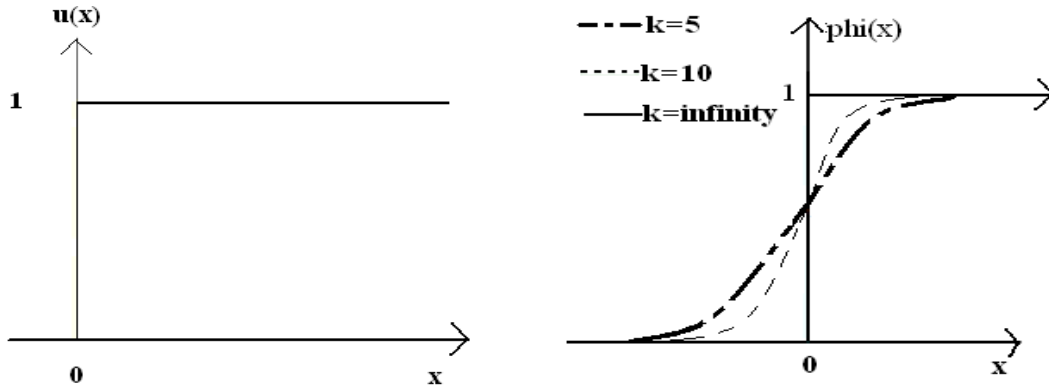
$$u(p) = \underset{k \to \infty}{lt}\frac{1}{1+e^{-kp}}.$$

Let us take a moderate value of k for analysis. $u(p) \approx \dfrac{1}{1+e^{-kp}}$. 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 $(1-\dfrac{kp}{2})^{-1}$ we obtain,

$$u(p) \approx \frac{1}{2} + \frac{k}{4}p \tag{16}$$

Fig. 4: The unit step and the logistic functions.

Now, the population has a small divergence. $\therefore U_m - x_m$ is not very large (Explained in Appendix A1) and as $\left|\dfrac{dx_m}{dt}\right|$ is either 0 or $\left|U_m - x_m\right|$. This ensures $\left|\dfrac{dx_m}{dt}\right|$ 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 $\left| f^{'}(x_m)\dfrac{dx_m}{dt} \right|$ is small. Thus from equations (14) and (15) we get,

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

$$\Rightarrow \frac{dx_m}{dt} = \frac{\dfrac{1}{2}(U_m - x_m)}{1 + \dfrac{k}{4}f^{'}(x_m)(U_m - x_m)} \tag{17}$$

Now, $\left| \dfrac{k}{4}f^{'}(x_m)(U_m - x_m) \right|$ is small $\therefore [1 + \dfrac{k}{4}f^{'}(x_m)(U_m - x_m)]^{-1} \approx 1 - \dfrac{k}{4}f^{'}(x_m)(U_m - x_m)$.

From equation (17) we get,

$$\frac{dx_m}{dt} = -\frac{k}{8}(U_m - x_m)^2 f^{'}(x_m) + \frac{U_m - x_m}{2} \tag{18}$$

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

Let us try to compute its expected value.

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

$$\Rightarrow E(\frac{dx_m}{dt}) = -\frac{k}{8}f^{'}(x_m)[E(U_m^2) + x_m^2 - 2x_m E(U_m)] + \frac{1}{2}[E(U_m) - x_m] \tag{19}$$

Substituting values of $E(U_m), E(U_m^2)$ from equation (12) and (13) to equation (19) 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 4: Let x_{av} denote the centroid (mean of all points) of the current population and $x_{av} = \dfrac{1}{NP}\sum\limits_{m=1}^{NP} x_m$. 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,

$$E(\frac{dx_{av}}{dt}) = -\frac{k}{8}CR(2F^2 + 1)Var(x)f_{av}^{'} - \frac{k}{8}CR(\frac{1}{N}\sum\limits_{m=1}^{NP}\varepsilon_m^2 f^{'}(x_{av} + \varepsilon_m)) \tag{20}$$

Proof: Now, $x_{av} = \dfrac{1}{NP}\sum\limits_{i=1}^{NP} x_i = \dfrac{1}{NP}\sum\limits_{m=1}^{NP} x_m$

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

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

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

Now, $\sum\limits_{m=1}^{NP}(x_{av}-x_m) = 0$ $\therefore E\left(\dfrac{dx_{av}}{dt}\right) = \dfrac{1}{NP}\left(-\dfrac{k}{8}CR.\sum\limits_{m=1}^{NP}\{(2F^2+1)Var(x)+(x_{av}-x_m)^2\}f'(x_m)\right)$

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

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

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

$$\therefore E\left(\frac{dx_{av}}{dt}\right) = -\frac{k}{8}CR(2F^2+1)Var(x)f'_{av} - \frac{k}{8}CR\left(\frac{1}{N}\sum_{m=1}^{NP}\varepsilon_m^2 f'(x_{av}+\varepsilon_m)\right) \tag{21}$$

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

Remark: From theorem 3, we may write,

$$E\left(\frac{dx_m}{dt}\right) = -\alpha_{DE}f'(x_m) + \beta_{DE} \tag{22}$$

Where, $\alpha_{DE} = -\dfrac{k}{8}CR\{(2F^2+1)Var(x)+(x_{av}-x_m)^2\}$ and $\beta_{DE} = \dfrac{1}{2}CR(x_{av}-x_m)$

The classical gradient descent search algorithm is given by the following dynamics (continuous) in single dimension [25]:

$$\frac{d\theta}{dt} = -\alpha.G + \beta \tag{23}$$

where α is the learning rate and β is the momentum.

The resemblance of equations (22) and (23) is not difficult to recognize and it suggests that, the dynamics of actual DE uses some kind of estimation for the gradient of the objective function. In equation (20), $-\alpha_{DE} f'(x_m)$ term on the R.H.S. is responsible for moving along the direction of the negative gradient, whereas β_{DE} represents a component of velocity of a trial solution towards the mean vector (center of mass) of the population.

Evidently very near to an optimum, when $f'(x_m) \to 0$,

$$E(\frac{dx_m}{dt}) \approx \beta_{DE} = \frac{1}{2} CR(x_{av} - x_m) \tag{24}$$

Clearly if the population converges towards the optimum, $(x_{av} - x_m)$ tends to zero and $E(\frac{dx_m}{dt}) \to 0$, thus once reaching the optimum, average velocity of the population members ceases to exist. Thus $Var(x) \to 0$, $x_{av} - x_m \to 0$ and also $\varepsilon_m \to 0$ and from (24) we get $E(\frac{dx_m}{dt}) \to 0$ and $E(\frac{dx_{av}}{dt}) \to 0$.

4. Lyapunov Stability Analysis of the DE-Population

In this section we analyze the stability of the population-dynamics represented by equation (3.16) using the concept of Lyapunov stability theorems [28]. We begin this treatment by explaining some basic concepts and their interpretations from the standard literature on nonlinear control theory [29, 28].

Definition 1

A point $\vec{x} = \vec{x}_e$ is called an *equilibrium state,* if the dynamics of the system is given by

$$\frac{d\vec{x}}{dt} = f(\vec{x}(t))$$

becomes zero at $\vec{x} = \vec{x}_e$ for any *t* i.e. $f(\vec{x}_e(t)) = 0$. The equilibrium state is also called equilibrium (stable) point in *D*-dimensional hyperspace, when the state $\vec{x}_e$ has *D*-components.

Definition 2

A scalar function $V(\vec{x})$ is said to be *positive definite* with respect to the point $\vec{x}_e$ in the region $\|\vec{x} - \vec{x}_e\| \le K$, if $V(\vec{x}) > 0$ at all points of the region except at $\vec{x}_e$ where it is zero.

Definition 3

A scalar function $V(\vec{x})$ is said to be *negative definite* if $-V(\vec{x})$ is positive definite.

Definition 4

A dynamics $\dfrac{d\vec{x}}{dt} = f(\vec{x}(t))$ is *asymptotically stable* at the equilibrium point $\vec{x}_e$, if

a) it is stable in the sense of Lyapunov, i.e., for any neighborhood $S(\varepsilon)$ surrounding $\vec{x}_e$ ($S(\varepsilon)$ contains points $\vec{x}$ for which $\left\| \vec{x} - \vec{x}_e \right\| \leq \varepsilon$) where there is a region $S(\delta)$ ($S(\delta)$ contains points $\vec{x}$ for which $\left\| \vec{x} - \vec{x}_e \right\| \leq \delta$), $\delta < \varepsilon$, such that trajectories of the dynamics starting within $S(\delta)$ do not leave $S(\varepsilon)$ as time $t \to \infty$ and

b) the trajectory starting within $S(\delta)$ converges to the origin as time t approaches infinity.

The sufficient condition for stability of a dynamics can be obtained from the Lyapunov's theorem, presented below.

Lyapunov's stability theorem [28, 34]

Given a scalar function $V(\vec{x})$ and some real number $\varepsilon > 0$, such that for all $\vec{x}$ in the region $\left\| \vec{x} - \vec{x}_e \right\| \leq \varepsilon$ the following conditions hold:

1) $V(\vec{x}_e) = 0$

2) $V(\vec{x}) > 0$ for $\vec{x} \neq \vec{x}_e$, i.e. $V(\vec{x})$ is positive definite.

3) $V(\vec{x})$ has continuous first partial derivatives with respect to all components of $\vec{x}$.

Then the equilibrium state $\vec{x}_e$ of the system $\dfrac{d\vec{x}}{dt} = f(\vec{x}(t))$ is

a) *asymptotically stable* if $\dfrac{dV}{dt} < 0$, i.e. $\dfrac{dV}{dt}$ is negative definite, and

b) *asymptotically stable* in the large if $\dfrac{dV}{dt} < 0$ for $\vec{x} \neq \vec{x}_e$, and in addition, $V(\vec{x}) \to \infty$ as $\left\| \vec{x} - \vec{x}_e \right\| \to \infty$.

Remark: Lyapunov stability analysis is based on the idea that if the total energy in the system continually decreases, then the system will asymptotically reach the zero energy state associated with an equilibrium point of the system. A system is said to be asymptotically stable if all the states approach the equilibrium state with time.

To study stability of DE algorithm we first model it as an autonomous control system. Here each population member x_m is a state variable of the control system. From equation (14) we get,

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

Assuming the population to be concentrated into a small neighborhood around an optimum in a flatter portion of the function, we have $\left| f'(x_m) \right| << 1$. Hence the equation can be written as,

$$E(\frac{dx_m}{dt}) = \frac{1}{2}CR(x_{av} - x_m), \qquad \text{for } i = 1,2,..., NP$$

$$E(\frac{dx_m}{dt}) = \frac{1}{2}CR(\frac{1}{NP}\sum_{j=1}^{NP} x_j - x_m), \qquad \text{for } i = 1,2,..., NP \qquad (25)$$

Actually (25) represents NP number of simultaneous equations. Next, we represent them using matrix notation. From (25) we get,

$$\begin{bmatrix} E(\frac{dx_1}{dt}) \\ E(\frac{dx_2}{dt}) \\ ..... \\ ..... \\ E(\frac{dx_{NP}}{dt}) \end{bmatrix} = \frac{1}{2}CR \begin{bmatrix} \frac{1}{NP}-1 & \frac{1}{NP} & .... & \frac{1}{NP} \\ \frac{1}{NP} & \frac{1}{NP}-1 & .... & \frac{1}{NP} \\ ...... & ...... & ..... & ..... \\ ...... & ...... & ..... & ..... \\ \frac{1}{NP} & \frac{1}{NP} & .... & \frac{1}{NP}-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ .... \\ .... \\ .... \\ .... \\ x_{NP} \end{bmatrix} \qquad (26)$$

The above matrix equation is of the form $\left[E\left(\frac{d\vec{x}}{dt}\right) \right] = A[\vec{x}]$, Where $\vec{x} = [x_1, x_2,..., x_{NP}]^T$ is the set of state variables and

$$A = \frac{1}{2}CR \begin{bmatrix} \frac{1}{NP}-1 & \frac{1}{NP} & .... & \frac{1}{NP} \\ \frac{1}{NP} & \frac{1}{NP}-1 & .... & \frac{1}{NP} \\ ...... & ...... & ..... & ..... \\ ...... & ...... & ..... & ..... \\ \frac{1}{NP} & \frac{1}{NP} & .... & \frac{1}{NP}-1 \end{bmatrix}$$

We know that eigenvalues of system-matrix A are the poles of the system. Eigenvalues are those values of λ for which $\det[\lambda I - A] = 0$ is satisfied, where I is the identity matrix of order NP.

$$\det[\lambda I - A] = 0 \Rightarrow \det \begin{bmatrix} \frac{2}{CR}\lambda - \frac{1}{NP}+1 & -\frac{1}{NP} & .... & -\frac{1}{NP} \\ -\frac{1}{NP} & \frac{2}{CR}\lambda - \frac{1}{NP}+1 & .... & -\frac{1}{NP} \\ ...... & ...... & ..... & ..... \\ ...... & ...... & ..... & ..... \\ -\frac{1}{NP} & -\frac{1}{NP} & .... & \frac{2}{CR}\lambda - \frac{1}{NP}+1 \end{bmatrix} = 0 \qquad (27)$$

After doing simple algebraic operations on the rows of the determinant in LHS of (27) we get,

$$\lambda(\lambda + \frac{CR}{2})^{NP-1} = 0 \qquad (28)$$

Clearly equation (28) is the characteristic equation of matrix A. From (28) we get the system eigenvalues as:

$$\lambda = 0, -\frac{CR}{2}, -\frac{CR}{2}, ..., -\frac{CR}{2}.$$

These values of λ are the system poles. We observe that one of these eigenvalues is zero and the rest are negative. Since one eigenvalue is zero, the system is not asymptotically stable and must have a DC component in the output. In the following section, we investigate whether the system is stable in the sense of Lyapunov.

Theorem 5: The system defined in equations (25) and (26) is stable in the sense of Lyapunov.

Proof: We are assuming the population is located very close to optima. Hence value of the gradient is negligibly small. So equation (25) holds true in such a region.

$$E(\frac{dx_m}{dt}) = \frac{1}{2} CR (\frac{1}{NP} \sum_{j=1}^{NP} x_j - x_m), \qquad \text{for } m = 1,2,..., NP$$

The condition for an equilibrium point is $E(\frac{dx_m}{dt}) = 0$, for $m = 1,2..., NP$ [according to definition 1]. We consider the case where the DE population is confined within a small neighborhood of an isolated optimum and over the entire population value of the gradient is very small. In this case, the preferred equilibrium point should be the optimum itself. This ensures that with time there is no change in values of state variables i.e. positions of the population members after they hit the optimum. Now from equation (25),

$$E(\frac{dx_m}{dt}) = 0$$

$$\Rightarrow \frac{1}{2} CR(\frac{1}{NP} \sum_{j=1}^{NP} x_j - x_m) = 0 \quad , \text{for } i = 1,2..., NP .$$

$$\Rightarrow x_i = \frac{1}{NP} \sum_{j=1}^{NP} x_j = x_{av} , \text{for } i = 1,2.NP .$$

This is possible only if all of the state variables are equal in value i.e. $x_1 = x_2 = ... = x_{NP} = x_e$, where x_e is the equilibrium position. At this point we would like to mention that as the search progresses; the population-members in DE get to the better portions of the search space or remain constant owing to its greedy selection strategy. In case of a smooth, unimodal fitness landscape, the solution vectors generally crowd into a small neighborhood surrounding the optimum. Thus during the later stages of search, the equilibrium point x_e basically is identical to the optimum, once reaching at which point, population members are expected not to change any further

and thus this point should satisfy the condition $x_1 = x_2 = ... = x_{NP}$ as well. This section examines the stability of the solution vectors very near to such an optimum point of the search space. Figure 5 shows a fitness landscape and an equilibrium position at the optimum. Next, we define Lyapunov's Energy function V as,

$$V(x_1, x_2, ...x_{NP}, t) = \sum_{i=1}^{NP}(x_i - x_{av})^2 \tag{29}$$

Clearly $\qquad V = 0$, if $x_1 = x_2 = ... = x_{NP} = x_e$

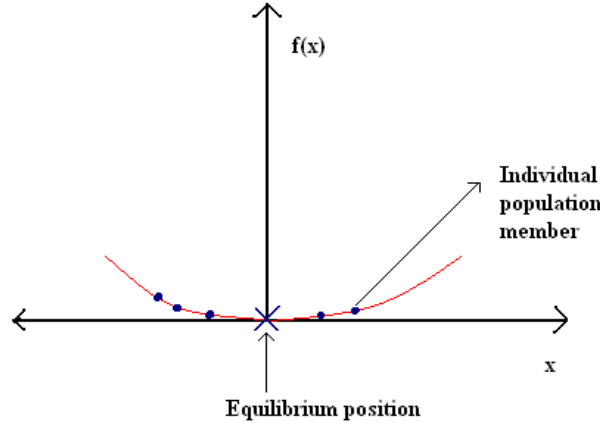$\qquad\qquad\qquad > 0$, otherwise.

Fig 5: State variables along with equilibrium position.

Energy function is always positive except the equilibrium, where it becomes zero. So, energy function is a positive definite with respect to equilibrium [from definition 2]. It is also to be noted that basically $V = NP(Var(x))$. Differentiating equation (29) with respect to time we get,

$$\frac{dV}{dt} = 2\sum_{m=1}^{NP}(x_m - x_{av})(\frac{dx_m}{dt} - \frac{dx_{av}}{dt})$$

$$E\left(\frac{dV}{dt}\right) = 2\sum_{m=1}^{NP}(x_m - x_{av})(E\left(\frac{dx_m}{dt}\right) - E\left(\frac{dx_{av}}{dt}\right)) \tag{30}$$

From (25) we get ,

$$E(\frac{dx_m}{dt}) = \frac{1}{2}CR(x_{av} - x_m) \text{ and } E(\frac{dx_{av}}{dt}) = E(\frac{d}{dt}(\frac{1}{NP}\sum_{i=1}^{NP}x_i)) = \frac{1}{NP}E\left(\sum_{i=1}^{NP}\frac{dx_i}{dt}\right) = 0.$$

Putting expectation values in (30),

$$E\left(\frac{dV}{dt}\right) = -CR\sum_{i=1}^{NP}(x_i - x_{av})^2 \tag{31}$$

From equation (30) it is evident that $E\left(\dfrac{dV}{dt}\right)$ is 0 when $x_1 = x_2 = \ldots\ldots\ldots = x_{NP} = x_e$ and is negative

otherwise. Hence $E\left(\dfrac{dV}{dt}\right)$ is a negative definite with respect to equilibrium point. Here V is positive definite and

$E\left(\dfrac{dV}{dt}\right)$ is negative definite, satisfying Lyapunov's stability theorem. We can infer that the system is stable in the

sense of Lyapunov.

Remark: Clearly V has continuous partial derivative. It can be noted that the population average or the center of

the mass of the system does not change with time (as $E(\dfrac{dx_{av}}{dt}) = E(\dfrac{d}{dt}(\dfrac{1}{NP}\sum\limits_{i=1}^{NP} x_i)) = \dfrac{1}{NP} E\left(\sum\limits_{i=1}^{NP} \dfrac{dx_i}{dt}\right) = 0.$).

Already we have mentioned condition for equilibrium is $x_1 = x_2 = \ldots\ldots\ldots = x_{NP} = x_e$, where x_e the

equilibrium position is. Now, if all population members are equal, then each of them equals to population average,

i.e. $x_1 = x_2 = \ldots\ldots\ldots = x_{NP} = x_{av}$. This leads to the conclusion $x_e \equiv x_{av}$. Initially, population was scattered

within a small region around the optima. So, the average was also very close to the actual optima. Lyapunov's

function in this case is directly proportional to the population variance. With time the initially dispersed populations

gather at the center of mass of system (which almost remains steady in the time interval), and eventually population

variance diminishes to 0. This leads to convergence of system. Average velocity of *m*-th population member

is $\dfrac{1}{2} CR(x_{av} - x_m)$. Average acceleration is $\dfrac{1}{2} CR(\dfrac{dx_{av}}{dt} - \dfrac{dx_m}{dt}) = -\dfrac{1}{2} CR \dfrac{dx_m}{dt}$ (as of mass remains

unchanged). So, acceleration is directly proportional to velocity and the negative sign signifies it acts in opposite

direction. This characteristic of the system dynamics suggests that near the optima the algorithm acts as a

mechanical damper, and average position acts as a stable attracter. The velocity of a population member gradually

attenuates to zero by the damping force.

Theorem 6: An estimate of the system time-constant can be $\dfrac{1}{CR}$

Proof: Using equation (27), equation (29) can be written as

$$\frac{V}{-E\left(\dfrac{dV}{dt}\right)} = \frac{1}{CR}. \tag{32}$$

The term in the denominator of L.H.S of above expression is the expected or average value of time rate of change of

energy function. Let the process be carried out repeatedly for same initial conditions and parameter values and an

average of energy function is calculated for the runs and the average of the energy function be denoted by $\langle V \rangle$.

Time rate of change of the average is also computed and let it be denoted as $\dfrac{d\langle V\rangle}{dt}$. We assume that the runs of the algorithm are independent and probability associated with selecting a population member in any stage of the algorithm does not change with time i.e. the process is time invariant. In that case we may expect from equation (32)

$$\frac{\langle V\rangle}{-\dfrac{d\langle V\rangle}{dt}} = \frac{1}{CR}$$

$$\Rightarrow \langle V\rangle = V_0 \exp(-\frac{t}{1/CR}) \tag{33}$$

where V_0 is the initial value of energy function. We have seen that energy function decreases with time. We may define a time-constant for the system as the time interval in which the energy function reduces to $\dfrac{1}{e}$ part of its initial value. If we denote this time-constant by T, Putting $\langle V\rangle = \dfrac{V_0}{e}$ and $t = T$ in (31), we have time-

constant $= T = \dfrac{1}{CR}$.

5. Experimental Results

In this section we provide the phase plots ($v = \dfrac{dx}{dt}$ 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. The vectors are randomly initialized in the interval (-5, 5). In Figure 6 four phase-trajectories have been shown 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 $E(\dfrac{dx}{dt})$ of individual member of population gradually approaches zero. Experimental results suggest that the D.C component at the output of the system defined in equation (26) is actually zero, leading all the vectors to converge to the equilibrium point, which is identical to the optimum for a uni-modal function.
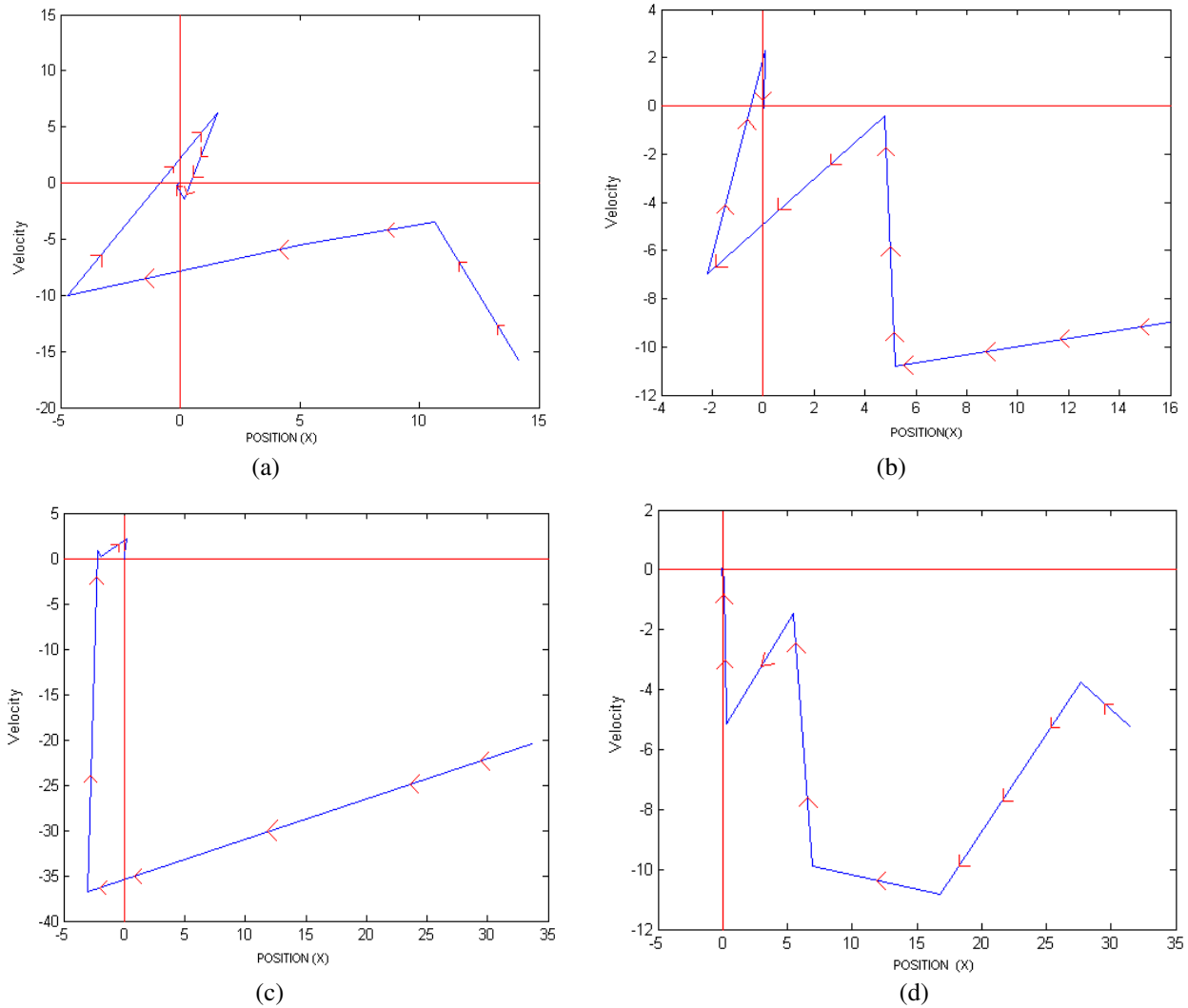
Fig. 6: Phase trajectory of the median order vector (in a population of size $NP = 11$) for 4 independent runs (with different seeds for the random number generator) for $f(x) = x^2$

Similarly, we construct phase trajectories for objective function $f(x) = 1 - e^{-x^2}$. New set of phase trajectories is shown in Figure 7. The vectors are randomly initialized in the interval (-5, 5).

(a)

(b)

(c)
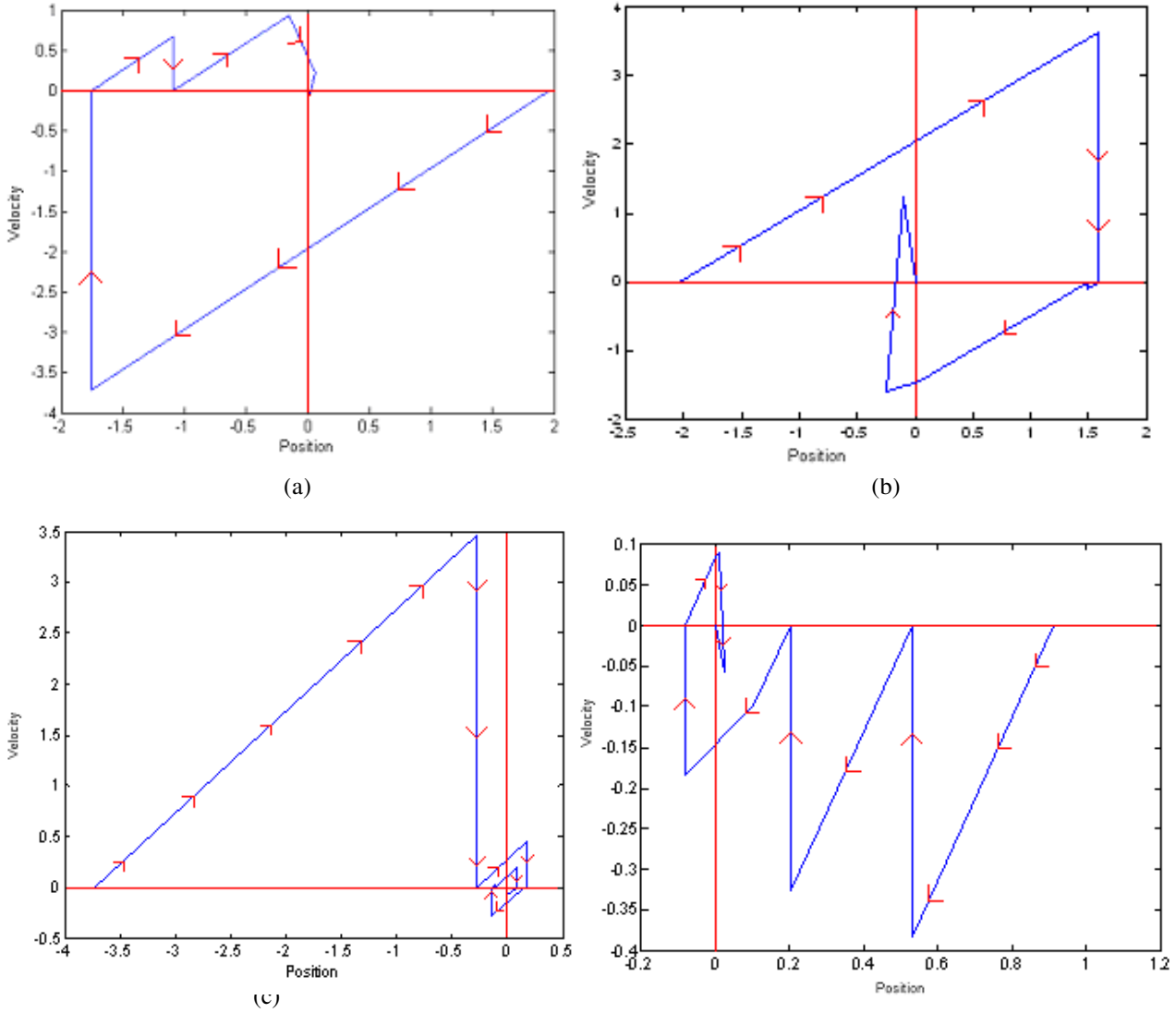
Fig. 7: Phase trajectory of the median order vector (in a population of size $NP = 11$) for 4 independent runs (with different seeds for the random number generator) for $f(x) = 1 - e^{-x^2}$

We have estimated time-constant of Lyapunov energy function in theorem 6. Now, according to equation (33) convergence time is inversely proportionate to crossover probability. In Figure 8 plots of time variations of Lyapunov's energy function is provided for various crossover probabilities (objective function used $f(x) = x^2$). From Figure 8 we observe as crossover probability increases convergence time gradually decreases. This matches with our theoretical finding of theorem 6. From Figure 8 we graphically determine time-constant for the energy function, which is the time in which Lyapunov energy function diminishes to e-th (approx 2.71) fraction of its initial value. In Table 1 below we make a comparison between convergence time measured from Figure 7 and found from equation (33).
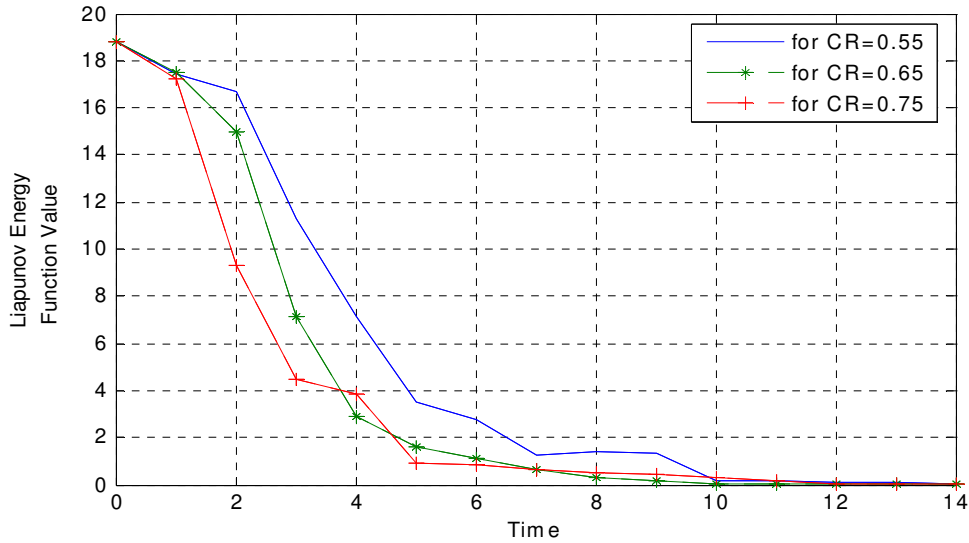
Fig. 8: Convergence characteristics for various values of crossover probability.

Table 1 shows that the theoretically predicted convergence time-constant closely matches its experimentally found counterpart. This confirms the finding of theorem 6.

Table 1: Comparison between calculated and experimental convergence time.

Crossover probability	Convergence time (Expressed in number of generations)	
	Measured graphically	Calculated theoretically
0.55	2.4	2.31
0.65	2.1	1.94
0.75	1.8	1.73

6. 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. We apply simple statistical and calculus-based methods in order to derive a dynamical model of the DE-population that undergoes mutation, binomial crossover and selection. The selection mechanism in DE has been modeled by the well-known unit step function, which was subsequently approximated by continuous logistic function. One important finding of our analysis is the similarity of the fundamental differential equation governing the expected velocity of a search-agent in DE with that of the classical gradient descent search with momentum. This suggests that DE uses a stochastic estimate of the gradient of the objective function (which was assumed to be continuous in our analysis in order to keep the mathematics less rigorous) in order to locate the optima of the function. It is due to the gradient descent type search strategy, that DE converges much faster than algorithms like GA or Particle Swarm Optimization (PSO) over unimodal benchmarks [35]. However, the actual algorithm does not take into account any analytical expression of the true function-gradient and due to the

randomness introduced by mutation and crossover operations into the dynamics of an individual, can escape trapping in local optima in many cases.

Based on the mathematical model derived here, we also analyze the stability of a DE population, very near to an isolated optimum, which acts as the equilibrium point for the dynamics. Application of Lyapunov's stability theorems reveals that the near-equilibrium behavior of a DE population is inherently stable and free from any kind of oscillatory behaviors seen in other optimization algorithms like Bacterial Foraging Optimization (BFO) [36] or PSO [37]. Our analysis reveals that the control parameter CR governs the rate of convergence of a DE population to an optimum. Future research may focus on analyzing the stability of the DE dynamics based on a stochastic Lyapunov energy function approach [38].

Appendix A1

Explanation of assumption ii) in section 3.1: In this work we aim at a stability analysis of a DE population. For that we assume the population is closely spaced and study how the population finally converges. Apart from this reason this assumption serves another purpose too. It allows us to carry out some simplifications to reach equation (12) and to carry out analysis further. After (11) it is assumed that $|U_m - x_m|$ is very small.

U_m can assume only two values x_m and V_m respectively.

Then, $|U_m - x_m| \leq |V_m - x_m|$, equality holds if $U_m = V_m$.

$$|V_m - x_m| = |(X_{r1} - x_m) + F(X_{r2} - X_{r3})| \leq |X_{r1} - X_{r2}| + F|X_{r2} - X_{r3}| \leq R(1+F)$$

Where, R is the range of the population i.e. $R = x_{max} - x_{min}$.

So, $|U_m - x_m| \leq |V_m - x_m| \leq R(1+F) \qquad \Rightarrow |U_m - x_m| \leq R(1+F)$.

Population variance is defined as $Var(x) = \dfrac{1}{NP}\displaystyle\sum_{i=1}^{NP}(x_i - \bar{x})^2$, where NP is population size.

$$Var(x) = \frac{1}{NP}\sum_{i=1}^{NP}(x_i - \bar{x})^2 \geq \frac{1}{NP}\left((x_{max} - \bar{x})^2 + (x_{min} - \bar{x})^2\right)$$

[considering only two specific terms out of NP number of terms]

$$(x_{max} - \bar{x})^2 + (x_{min} - \bar{x})^2 = \frac{1}{2}\left[(x_{max} + x_{min} - 2\bar{x})^2 + (x_{max} - x_{min})^2\right] \quad [\text{as } a^2 + b^2 = \left(\frac{a+b}{2}\right)^2 + \left(\frac{a-b}{2}\right)^2]$$

$$(x_{max} - \bar{x})^2 + (x_{min} - \bar{x})^2 \geq \frac{1}{2}[x_{max} - x_{min}]^2 = \frac{R^2}{2}$$

So, $Var(x) \geq \dfrac{1}{NP}\left((x_{max} - \bar{x})^2 + (x_{min} - \bar{x})^2\right) \geq \dfrac{R^2}{2NP} \qquad \Rightarrow \sqrt{Var(x)} \geq \dfrac{R}{\sqrt{2NP}}$.

Finally $\Rightarrow |U_m - x_m| \le R(1+F) \le \sqrt{2NP(Var(x))}$. Thus if f $Var(x)$ is small (which is the case during the final stages of search on a unimodal fitness landscape for DE) in that case $|U_m - x_m|$ is also very small.

Appendix A.2

In this section we carry out a similar analysis for the DE/current-to-rand/1 scheme illustrated in equation (10). Next we carry out previous analysis for 'DE/current-to-rand/1'. Besides previous assumptions described in section 3.1 we also assume $k_1 = k_2 = ... = k_{NP} = k_{crossover}$. This assumption is made to simplify the analysis. Similar to the derivations done in theorem 1 and 2, we calculate the following expectations.

$$E(U_m - x_m) = k_{crossover}(x_{av} - x_m) \tag{34}$$

$$E(U_m - x_m)^2 = k_{crossover}^2 \langle x^2 \rangle_{av}(1+2F^2) + (k_{crossover}^2 x_m^2 - 2k_{crossover}^2 x_m x_{av} - 2k_{crossover}^2 F^2 x_{av}^2), \tag{35}$$

where, $\langle x^2 \rangle_{av} = \dfrac{1}{NP}\sum_{i=1}^{NP} x_i^2$

Selection step is exactly same in the two versions of algorithms. Theorem 3 also holds for this case. From theorem 3, we obtain expression for $E\left(\dfrac{dx_m}{dt}\right)$, which is as following,

$$E(\frac{dx_m}{dt}) = -\frac{k}{8}f'(x_m)E(U_m - x_m)^2 + \frac{1}{2}E(U_m - x_m) \tag{36}$$

Substituting values from equations (32) and (33) we get,

$$E(\frac{dx_m}{dt}) = \alpha_{new}f'(x_m) + \beta_{new}, \tag{37}$$

where, $\alpha_{new} = -\dfrac{k}{8}k_{crossover}^2 \langle x^2 \rangle_{av}(1+2F^2) + (k_{crossover}^2 x_m^2 - 2k_{crossover}^2 x_m x_{av} - 2k_{crossover}^2 F^2 x_{av}^2$

and $\beta_{new} = \dfrac{k_{crossover}}{2}(x_{av} - x_m)$.

Equation (35) shows that the fundamental dynamics of 'DE/current-to-rand/1' near an optimum also has a resemblance with the classical gradient descent algorithm. We carry out stability tests in a way exactly similar to that of done in section 4. We found that 'DE/current-to-rand/1' is also asymptotically stable, satisfying Liapunov's criterion. In this case convergence time becomes $\dfrac{1}{k_{crossover}}$

References

1. Reeves, C. R. and Rowe, J. E., Genetic Algorithms – Principles and Perspectives: A Guide to GA Theory, Kluwer Academic Publishers, 2003.

2. Rudolph, G., Convergence Analysis of Canonical Genetic Algorithms, IEEE Transactions on Neural Networks, 5(1), pages 96–101, 1994.

3. Th. Baeck., Order Statistics for Convergence Velocity Analysis of Simplified Evolutionary Algorithms. Foundations of Genetic Algorithms, pages 91-102, 1994.

4. Beyer, H.-G., On the Dynamics of EAs without Selection, Proceedings of Foundations of Genetic Algorithms, 5 (FOGA-5), pages 5–26, 1999.

5. Vose, M. D., The Simple Genetic Algorithm: Foundations and Theory, The MIT Press, 1999.

6. Pruegel-Bennett, A., Modeling Genetic Algorithm Dynamics, Theoretical Aspects of Evolutionary Computing, pages 59–85, 2001.

7. He, J. and Yao, X., Towards an analytic framework for analyzing the computation time of evolutionary algorithms, Artificial Intelligence, 145(1-2), pages 59-97, 2003.

8. Okabe, T., Jin, Y. and Sendhoff, B., On the Dynamics of Evolutionary Multi-Objective Optimization, Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2002), pages 247–255, 2002.

9. Okabe, T., Jin, Y., and Sendhoff, B., Evolutionary Multi-Objective Optimization with a Hybrid Representation, Proceedings of Congress on Evolutionary Computation (CEC-2003), pages 2262–2269, 2003.

10. Trelea, I. C., The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters, vol. 85, pp. 317-325, Mar. 2003.

11. Poli, R. and Broomhead, D., Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (London, England, July 07 - 11, 2007). GECCO '07. ACM, New York, NY, 134-141, 2007.

12. Kadirkamanathan, V., Selvarajah, K., .Fleming, P. J., Stability analysis of the particle dynamics in particle swarm optimizer, IEEE Transactions on Evolutionary Computing, vol.10, no.3, pp.245-255, Jun. 2006.

13. 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, June 2003.

14. 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, Sept. 2003.

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

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

17. 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.

18. Das, S., Konar, A., and 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.

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

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

21. Zaharie, D., On the Explorative Power of Differential Evolution, 3rd International Workshop on Symbolic and Numerical Algorithms on Scientific Computing, SYNASC-2001, Timişoara, Romania, October 2 – 4, 2001.

22. Zaharie, D., Critical Values for the Control Parameters of Differential Evolution Algorithms. In: Matouk, Radek and Oera, Pavel (eds.) (2002). Proceedings of MENDEL 2002, 8th International Mendel Conference on Soft Computing, June 5., 2002, Brno, Czech Republic, pp. 62,. Brno University of Technology, Faculty of Mechanical Engineering, Brno (Czech Republic).

23. Beyer, H. –G, On the Explorative Power of ES/EP-like Algorithms, Technical Report, University of Dortmund, 1998.

24. Efimov, A.V., Modulus of continuity, Encyclopaedia of Mathematics, Springer, 2001.

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

26. Magoulas, G.D., Plagianakos, V.P., and Vrahatis, M.N., Hybrid methods using evolutionary algorithms for on-line training Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on Volume 3, Page(s):2218 – 2223, 15-19 July 2001.

27. Ranasinghe, M., Mixing Paradigms: A Genetic Operator that Approximates Gradient Descent, Genetic Algorithms and Genetic Programming at Stanford 2003 (Book of Student Papers from J. R. Koza's Course at Stanford on Genetic Algorithms and Genetic Programming) Stanford University Bookstore

28. Hahn, W., Theory and Application of Liapunov's Direct Method, Prentice-Hall, Englewood Cliffs, N.J., 1963.

29. Haddad, W. M. and Chellaboina, V., Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach, Princeton University Press, 2008.

30. Kirk, W. A. and Sims, B., Handbook of Metric Fixed Point Theory, Kluwer Academic, London, 2001.

31. Fletcher, R., Practical Methods of Optimization, Second ed., John Wiley & Sons, Chichester, 1987.

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

33. Anwal, R. P.: Generalized Functions: Theory and Technique, 2nd ed. Boston, MA: Birkhãuser, 1998.

34. Kuo, B. C., Automatic Control Systems, Prentice-Hall, Englewood Cliffs, NJ, 1987.

35. Vesterstrøm, J. and Thomson, R., A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, in Proc. Sixth Congress on Evolutionary Computation (CEC-2004), IEEE Press, 2004.

36. Dasgupta, S., Das, S., Abraham, A. and Biswas, A., Adaptive Computational Chamotaxis in Bacterial Foraging Optimization – An Analysis, IEEE Transactions on Evolutionary Computation , 2009 (in Press).

37. Clerc, M. and Kennedy, J., The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6 (1), 58–73, 2002.

38. 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.