A Differential Covariance matrix Adaptation Evolutionary Algorithm for Global Optimization

Saurav Ghosh[1], Subhrajit Roy[1], Sk. Minhazul Islam[1], Swagatam Das[1], and P. N. Suganthan[2]
[1]Dept. of Electronics and Telecommunication Engg. , Jadavpur University, Kolkata 700 032, India.
[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore
E-mail: saurav_online@yahoo.in, roy.subhrajit20@gmail.com, skminha.isl@gmail.com, swagatamdas19@yahoo.co.in, epnsugan@ntu.edu.sg

Abstract— **Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is arguably one of the most powerful stochastic real-parameter optimization algorithms in current use for non-linear non-convex functions with parameter linkages. Differential Evolution (DE) is again a very powerful but simple evolutionary algorithm for real parameter optimization. In this article we propose a simple but very efficient hybrid evolutionary algorithm named Differential Covariance Matrix Adaptation Evolutionary Algorithm (DCMA-EA), where it creates new population members by using controlled share of its target and the population mean, the scaled difference from current population and the step-size generated through the Covariance Matrix Adaptation. It also incorporates the selection and crossover strategies of DE. The proposed hybrid algorithm has more pronounced explorative and exploitative behaviors than its two ancestors (CMA-ES and DE). We compare DCMA-EA with original CMA-ES, some of the most known DE-variants: SaDE and JADE, and a PSO-based state-of-the-art real optimizer: DMS-PSO (Dynamic Multi Swarm Particle Swarm optimization) and DE/Rand/1/Bin over a test-suite of 20 shifted, rotated, and compositional numerical benchmarks.**

I. INTRODUCTION

In this paper we propose a simple yet very powerful hybrid EA that synergistically combines the features of two global optimizers: Differential Evolution (DE)[1-4] and Covariance Matrix Adaptation Evolution strategy (CMA-ES)[5]. CMA-ES has emerged as a very competitive real-parameter optimizer for continuous search spaces. Compared to other evolutionary algorithms, an important property of CMA-ES is its invariance against the linear transformations in the search space. In other words, it exhibit the same performances for a given objective functions $f : x \in \Re^n \mapsto f(x) \in \Re$ where $n \in N$ or for the same function where a linear transformation is applied, *i.e.* $f_R : x \in \Re^n \mapsto f(Rx) \in \Re$ where R denotes the linear transformation. In practice this transformation is learned by the CMA. In DE community, the individual trial solutions (which constitute a population) are called *parameter vectors* or *genomes*. Unlike traditional EAs, DE perturbs the current-generation vectors with the scaled differences of two randomly selected and mutually distinct population vectors. For over past few years, researchers have been trying to improve the performance of DE by devising new kinds of mutation schemes and parameter adaptation strategies for the algorithm. A thorough account of research on and with DE could be found in [4] and the references therein. Recently Mallipeddi *et al.* [6] proposed an ensemble of mutation strategies and control parameters with the DE so that the algorithm itself may employ different sets of control parameter values and mutation strategies at different stages of the optimization process as needed. In our proposed hybrid scheme, new solution is generated by taking a portion of the current generation mean and its target. It also takes the help of a scaled difference from two vectors of the current generation. Finally it also adds the step size for the current generation guided by the Covariance matrix Adaptation. We also have introduced the selection and crossover strategy of the DE to have its own advantage in a evolutionary algorithm. The proposed Scheme takes the advantage of the CMA-ES of overcoming badly scaled or high non-separable functions into premature convergence and also utilizes the DE with the introduction of crossover which ensures component injection and selection ensuring of absence of any inferior vectors in the next generations.

We choose a test-suite comprising of 20 numerical benchmarks. In order to illustrate the effectiveness of the incorporation of DE-based operators in CMA-ES, we compare its performance with those of the original DE and CMA-ES. We compare DCMA-EA with SaDE [7] and JADE [8]. Finally we also compare DCMA-EA with also DMS-PSO [9] and DE/rand/1/Bin (canonical version of DE). Our study indicates that DCMA-EA is superior to or comparable with some of the most competitive real-parameter optimizers.

II. CLASSICAL CMA-ES

The CMA-ES is a stochastic method for real parameter optimization of non-linear, non-convex functions. The algorithm includes adaptation of covariance matrix which is basically an alternative method of traditional Quasi-Newton method for optimization based on gradients. Here are some basic tools for realizing the CMA-ES algorithm.

A. Eigen decomposition of a positive definite matrix

A symmetric positive definite matrix, $C \in R^{n \times n}$, is defined such a way that for all $x \in R^n \setminus \{0\}$ holds $x^T C x > 0$. The eigendecomposition of C obeys

$$C = BD^2B^T, \quad (1)$$

978-1-61284-072-7/11/$26.00 ©2011 IEEE

$D^2 = DD = [diag(d_1,....d_n)]^2 = diag(d_1^2,.....d_n^2)$ is a diagonal matrix with eigenvalues of C as diagonal elements. Here, B is an orthogonal matrix; $B^T B = BB^T = I$ .Columns of B forms an orthonormal basis of eigenvectors.

$$C^{-1} = \left(BD^2B^T\right)^{-1} = B^{T^{-1}}D^{-2}B^{-1} = BD^{-2}B^T$$
$$= B\,diag\left(\frac{1}{d_1^2},.....,\frac{1}{d_n^2}\right)B^T, \quad (2)$$

From equation (2) we naturally define

$$C^{\frac{1}{2}} = BDB^T \quad (3)$$

and therefore

$$C^{-\frac{1}{2}} = BD^{-1}B^T = B\,diag\left(\frac{1}{d_1},.....,\frac{1}{d_n}\right)B^T \quad (4)$$

B. Introduction to multivariate normal distribution

A multivariate normal distribution $N(m,C)$, has a uni-modal, bell-shaped density, where the modal value corresponds to the distribution mean m .The distribution $N(m,C)$ is uniquely determined by its mean $m \in R^n$ and it's symmetric and positive definite covariance matrix $C \in R^{n\times n}$.

The normal distribution $N(m,C)$ can be written in different ways

$$N(m,C) \sim \; m + N(m,C) \;\sim\; m + C^{\frac{1}{2}}N(0,I)$$
$$\sim \; m + BD\underbrace{B^T N(0,I)}_{\sim N(0,I)} \;\sim\; m + BD\underbrace{N(0,I)}_{\sim N(0,D^2)} \quad (5)$$

where "~" denotes equality in distribution.

C. Sampling

The new population vectors in the subsequent generations are generated in the CMA evolution strategy by means of sampling which takes the help of the following equation.

$$x_k^{(g+1)} \sim m^{(g)} + \sigma^{(g)}N\left(0,C^{(g)}\right) \quad (6)$$

$N\left(0,C^{(g)}\right)$ is a multivariate normal distribution with zero mean with covariance matrix $C^{(g)}$. $m^{(g)} \in R^n$ and $\sigma^{(g)} \in R_+$ is the mean and overall standard deviation (step-size) at generation *g* respectively.

D. Recombination

The new mean $m^{(g+1)}$ of the all current population vectors is the weighted average of μ selected vectors from the samples $x_1^{(g+1)},....., x_\lambda^{(g+1)}$:

$$m^{(g+1)} = \sum_{i=0}^{\mu} w_i x_{i:\lambda}^{(g+1)} \quad (7)$$

$$\text{Here, } \sum_{i=1}^{\mu} w_i = 1, \quad w_1 \geq w_2 \geq ..... \geq w_\mu > 0 \quad (8)$$

Here, $\mu \leq \lambda$ (parent population size) is the number of selected points from the whole population. $w_{i=1,2,...\mu} \in R_+$ are positive weight coefficients for recombination. $x_{i:\lambda}^{(g+1)}$ is the *i*-th best individual from $x_1^{(g+1)},....., x_\lambda^{(g+1)}$ from eqn (8). Where, $f\left(x_{1:\lambda}^{(g+1)}\right) \leq f\left(x_{2:\lambda}^{(g+1)}\right) \leq f\left(x_{3:\lambda}^{(g+1)}\right) \leq ..... \leq f\left(x_{\lambda:\lambda}^{(g+1)}\right)$

Here, a new term related to μ is introduced whose measure

$$\mu_{eff} = \left(\frac{\|w\|_1}{\|w\|_2}\right)^2 = \frac{\|w\|_1^{\,2}}{\|w\|_2^{\,2}} = \frac{1}{\|w\|_2^2} = \left(\sum_{i=1}^{\mu} {w_i}^2\right)^{-1} \quad (9)$$

is termed as variance effective selection mass.

E. Covariance Matrix Adaptation

In this section the adaptation of covariance matrix has been derived which is the essence of the CMA-ES algorithm.

1) Estimating the covariance without any update

At first, the original covariance matrix $C^{(g)}$ can be estimated using the sample population from equation (6), $x_1^{(g+1)},....., x_\lambda^{(g+1)}$ via the empirical covariance matrix

$$C_{emp}^{(g+1)} = \frac{1}{\lambda - 1}\sum_{i=1}^{\lambda}\left(x_i^{(g+1)} - \frac{1}{\lambda}\sum_{j=1}^{\lambda}x_j^{(g+1)}\right)\left(x_i^{(g+1)} - \frac{1}{\lambda}\sum_{j=1}^{\lambda}x_j^{(g+1)}\right)^T$$

(10)

The empirical covariance matrix $C_{emp}^{(g+1)}$ is an estimator of $C^{(g)}$ .Considering a slight different approach to get an estimator for $C^{(g)}$

$$C_\lambda^{(g+1)} = \frac{1}{\lambda}\sum_{i=1}^{\lambda}\left(x_i^{(g+1)} - m^{(g)}\right)\left(x_i^{(g+1)} - m^{(g)}\right)^T \quad (11)$$

The later equation is also another unbiased estimator of $C^{(g)}$. To estimate a better co-variance matrix equation (11) is modified and the same weighted selection mechanism of equation (7) is used.

$$C_\mu^{(g+1)} = \sum_{i=1}^{\mu}\left(x_{i:\lambda}^{(g+1)} - m^{(g)}\right)\left(x_{i:\lambda}^{(g+1)} - m^{(g)}\right)^T \quad (12)$$

The matrix $C_\mu^{(g+1)}$ is an estimator for $C^{(g)}$. Sampling from $C_\mu^{(g+1)}$ tends to reproduce selected, *i.e.* successful steps giving a justification for a better co-variance matrix.

2) Rank-μ-update

To achieve fast search (opposite to more robust and more global search) the population size and μ_{eff} should be quite small ($\mu_{eff} \leq 1 + \ln n$).To use information from previous generation additionally, after a sufficient number of

generations the mean of the estimated co-variance matrices from all generations,

$$C^{(g+1)} = \frac{1}{g+1}\sum_{i=0}^{g}\frac{1}{\sigma^{(i)^2}}C_{\mu}^{(i+1)} \qquad (13)$$

becomes a reliable estimator for the selected steps. Choosing $C^{(0)} = I$ to be unity matrix and a learning rate $0 \le c_{\mu} \le 1$, then $C^{(g+1)}$ is given by

$$\begin{aligned} C^{(g+1)} &= \left(1-c_{\mu}\right)C^{(g)} + c_{\mu}\frac{1}{\sigma^{(g)^2}}C_{\mu}^{(g+1)} \\ &= \left(1-c_{\mu}\right)C^{(g)} + c_{\mu}\sum_{i=1}^{\mu} w_i y_{i:\lambda}^{(g+1)} y_{i:\lambda}^{(g+1)^T} \end{aligned} \qquad (14)$$

Here, $c_{\mu} \le 1$ is the learning rate for updating the co-variance matrix and $c_{\mu} \approx \min(1, {\mu_{eff}}/{n^2})$ is the reasonable choice and $y_{i:\lambda}^{(g+1)} = {(x_{i:\lambda}^{(g+1)} - m^{(g)})}/{\sigma^{(g)}}$.

This co-variance matrix update is called Rank-μ-update, because the sum of outer products in equation (14) is of rank $\min(\mu, n)$ (with probability 1). A first order approximation for a good choice of c_{μ} is that $c_{\mu} = {\mu_{eff}}/{n^2}$.

3) Rank-one-update

In this section a so-called evolution path is finally used for a rank-one update of the covariance matrix. At first a specific method has been considered to produce n-dimensional normal distribution with zero mean. Let the vectors $y_1, \ldots\ldots, y_{g0} \in R^n, g0 \ge n$, span R^n and let $N(0,1)$ denote independent $(0,1)$-normally distributed random numbers, then

$$N(0,1)y_1 + \ldots\ldots\ldots + N(0,1)y_{g0} \quad \sim \quad N\left(0, \sum_{i=1}^{g0} y_i y_i^T\right) \qquad (15)$$

is a normally distributed random vector with zero mean and co-variance matrix $\sum_{i=1}^{g0} y_i y_i^T$. The singular distribution $N(0,1)y_i \sim N(0, y_i y_i^T)$ generates the vector y_i with maximum likelihood considering all normal distributions with zero mean. Let the sum in equation (14) consists of a single summand only (*e.g.* $\mu = 1$), and let $y_{g+1} = \frac{x_{1:\lambda}^{(g+1)} - m(g)}{\sigma^{(g)}}$. Then the rank-one-update for the covariance matrix is given by

$$C^{(g+1)} = \left(1-c_1\right)C^{(g)} + c_1 y_{g+1} y_{g+1}^T \qquad (16)$$

4) Cumulation: Utilizing the evolution path

To construct an evolution path the step size σ is constructed by

$$\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} = \frac{m^{(g)} - m^{(g-1)}}{\sigma^{(g-1)}} = \frac{m^{(g-1)} - m^{(g-2)}}{\sigma^{(g-2)}} \qquad (17)$$

To construct the evolution path $p_c \in R^n$ it starts with $p_c^{(0)} = 0$.

$$p_c^{(g+1)} = \left(1-c_c\right)p_c^{(g)} + \sqrt{c_c\left(2-c_c\right)\mu_{eff}}\,\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \qquad (18)$$

Here, $p_c \in R^n$ is the evolution path at generation g. $c_c \le 1$. So the rank-one-update of the covariance matrix $C^{(g)}$ via the evolution path $p_c^{(g+1)}$ is given by

$$C^{(g+1)} = \left(1-c_1\right)C^{(g)} + c_1 p_c^{(g+1)} p_c^{(g+1)^T} \qquad (19)$$

As a last step equation (13) and (16) has been combined.

5) Combining rank-μ-update and Cumulation

The final covariance matrix adaptation update of the covariance matrix combines equation (13) and (14)

$$C^{(g+1)} = \left(1-c_1-c_{\mu}\right)C^{(g)} + c_1\underbrace{p_c^{(g+1)}p_c^{(g+1)^T}}_{rank-one\ update} + c_{\mu}\underbrace{\sum_{i=1}^{\mu} w_i y_{i:\lambda}^{(g+1)}\left(y_{i:\lambda}^{(g+1)}\right)^T}_{rank-\mu\ update} \qquad (20)$$

Here, $c_1 \approx {2}/{n^2}$ .and $c_{\mu} \approx \min\left({\mu_{eff}}/{n^2}, 1-c_1\right)$

$y_{i:\lambda}^{(g+1)} = {(x_{i:\lambda}^{(g+1)} - m^{(g)})}/{\sigma^{(g)}}$

F. Step-size control

To control the step size $\sigma^{(g)}$ the evolution path is utilized. To construct the evolution path, p_c from (21) depends on its direction. Initialized with $p_{\sigma}^{(0)} = 0$ is given by

$$p_{\sigma}^{(g+1)} = \left(1-c_{\sigma}\right)p_{\sigma}^{(g)} + \sqrt{c_{\sigma}\left(2-c_{\sigma}\right)\mu_{eff}}\,C^{(g)^{-\frac{1}{2}}}\,\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}} \qquad (21)$$

Here $p_{\sigma}^{(g)}$ is the conjugate evolution path at generation *g*. The update of $\sigma^{(g)}$ is given by the following equation.

$$\ln\sigma^{(g+1)} = \ln\sigma^{(g)} + \frac{c_{\sigma}}{d_{\sigma}E\left\|N(0,I)\right\|}\left(\left\|p_{\sigma}^{(g+1)}\right\| - E\left\|N(0,I)\right\|\right) \qquad (22)$$

Here, $d_{\sigma} \approx 1$ is damping parameter which scales the change magnitude of $\ln\sigma^{(g)}$.

$$E\left\|N(0,I)\right\| = \sqrt{2}\,\Gamma\left(\frac{n+1}{2}\right)\Big/\,\Gamma\left(\frac{n}{2}\right) \approx \sqrt{n} + \mathrm{O}(1/n) \qquad (23)$$

It is the expectation of the Euclidean norm of a $N(0,I)$ distributed random vector. After some rigorous modification it becomes

$$\sigma^{(g+1)} = \sigma^{(g)} + \exp\left(\frac{c_{\sigma}}{d_{\sigma}}\left(\frac{\left\|p_{\sigma}^{(g+1)}\right\|}{E\left\|N(0,I)\right\|} - 1\right)\right) \qquad (24)$$

The length of the evolution path is an intuitive and empirically well validated goodness measure for the overall step-length.

III. CLASSICAL DE

DE is a simple real-coded evolutionary algorithm. In this section we describe the basic operations of DE and introduce necessary notations and terminologies which facilitate the explanation of our adaptive DE algorithm later.

A. Initialization of the Parameter Vectors

DE searches for a global optimum point in a *D*-dimensional continuous hyperspace. It begins with a randomly initiated population of *Np* (population number) *D* dimensional real-valued parameter vectors. Each vector, also known as *genome/chromosome*, forms a candidate solution to the multi-dimensional optimization problem. We shall denote subsequent generations in DE by $g = 0,1,...., g_{\max}$ .Here we adopt the following notation for representing the *i*-th vector of the population at the current generation:

$$x_i^{(g)} = [\, x_{1,i}^{(g)}, x_{2,i}^{(g)}, x_{3,i}^{(g)}, ..., \ x_{D,i}^{(g)}]. \qquad (25)$$

For each parameter of the problem, there may be a certain range within which the value of the parameter should lie for better search results. 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:

$x_{\min} = \{x_{1,\min}, x_{2,\min}, ..., x_{D,\min}\}$ and

$x_{\max} = \{x_{1,\max}, x_{2,\max}, ..., x_{D,\max}\}$ .Hence we may initialize the *j*-th component of the *i*-th vector as:

$$x_{j,i}^{(0)} = x_{j,\min} + rand_{i,j}(0,1) \cdot (x_{j,\max} - x_{j,\min}) \qquad (26)$$

where rand is a uniformly distributed random number lying between 0 and 1 (actually $0 \le rand_{i,j}(0,1) < 1$) and is instantiated independently for each component of the *i*-th vector.

B. Mutation with Difference Vectors

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 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:

"DE/rand/1": $v_i^{(g)} = x_i^{(g)} + F \cdot (x_{r_1^i}^{(g)} - x_{r_2^i}^{(g)}).$ (27)

"DE/best/1": $v_i^{(g)} = x_{best}^{(g)} + F \cdot (x_{r_1^i}^{(g)} - x_{r_2^i}^{(g)}).$ (28)

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

$$v_i^{(g)} = x_i^{(g)} + F \cdot (x_{best}^{(g)} - x_i^{(g)}) + F \cdot (x_{r_1^i}^{(g)} - x_{r_2^i}^{(g)}). \qquad (29)$$

"DE/best/2":

$$v_i^{(g)} = x_{best}^{(g)} + F \cdot (x_{r_1^i}^{(g)} - x_{r_2^i}^{(g)}) + F \cdot (x_{r_3^i}^{(g)} - x_{r_4^i}^{(g)}). \qquad (30)$$

"DE/rand/2": $v_i^{(g)} = x_{r_1^i}^{(g)} + F \cdot (x_{r_2^i}^{(g)} - x_{r_3^i}^{(g)}) + F \cdot (x_{r_4^i}^{(g)} - x_{r_5^i}^{(g)}).$ (31)

The indices $r_1^i, r_2^i, r_3^i, r_4^i,$ and r_5^i are distinct integers uniformly chosen from the set {1, 2,. . . *Np*} with *i* being the vector concerned in the population. These indices are randomly generated once for each donor 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 (i.e. lowest objective function value for minimization problem) in the population at generation g.

C. Crossover

To enhance the potential diversity of the population, a crossover operation comes into play after generating the donor vector through mutation. The donor vector exchanges its components with the target vector $x_i^{(g)}$ under this operation to form the *trial* vector $u_i^{(g)} = [u_{1,i}^{(g)}, u_{2,i}^{(g)}, u_{3,i}^{(g)}, ..., u_{D,i}^{(g)}]$ .The scheme may be outlined as:

$$u_{j,i}^{(g)} = \begin{cases} v_{j,i}^{(g)}, & \text{if } (rand_{i,j}(0,1) \le Cr \text{ or } j = j_{rand} \\ x_{j,i}^{(g)}, & \text{otherwise,} \end{cases} \qquad (32)$$

where, $rand_{i,j}(0,1)$ is a uniformly distributed random number, 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 $u_i^{(g)}$ gets at least one component from $v_i^{(g)}$. These *F*, *Cr* and population number are the control parameters of the basic DE.

D. Selection

The next step of the algorithm calls for *selection* to determine whether the target or the trial vector survives to the next generation i.e. at $g = g + 1$. The selection operation is described as:

$$x_i^{(g+1)} = \begin{cases} = u_i^{(g)}, & if \quad f(u_i^{(g)}) \le f(x_i^{(g)}) \\ = x_i^{(g)}, & if \quad f(u_i^{(g)}) > f(x_i^{(g)}) \end{cases} \qquad (33)$$

where $f(x)$ is the objective function to be minimized.

IV. THE PROPOSED ALGORITHM DCMA-EA

In our proposed hybridization scheme DCMA-EA the differential mutation, binomial crossover and selection operations of DE have been incorporated into the structure of a CMA-ES algorithm in order to achieve a better performance than both of the ancestors.

A. Introduction of mutation

The traditional CMA-ES algorithm which basically bears the same analogy of the gradient based Quasi-newton method uses the adaptation of the covariance matrix to identify the function landscape which is a convex-quadratic one through the concept of Hessian matrix. What it does is that it sets the covariance matrix of the search distribution to the inverse Hessian matrix which is equivalent to rescaling the ellipsoid projection of the multivariate normal distribution into a spherical one. For a particular function landscape if it becomes qualified to convert it into a spherical projection by correspondingly adapting the eigen-decomposed matrix (B, D) then the algorithm converges fully on the global optima of a given problem. But the main hiccups arrives when more complex functions comes into account like noisy and hybrid composition functions then it fails to detect its landscape so as to adapt the eigen-decomposed matrices for its multivariate normal distribution to detect its function structure and the algorithm becomes liable to be trapped in a local optima. That's the main reason of the traditional CMA-ES failing horribly in the noisy and hybrid functions. In order to remove this major demerit of the CMA-ES a new differential perturbation scheme has been introduced in this proposed algorithm. In the CMA-ES algorithm the new population vector is created by this following equation.

$$x_i^{(g)} = m + \sigma \cdot N(0, C) = m + \sigma \cdot B \cdot D \cdot randn(\dim)^T \quad (34)$$

Here, $randn(\dim)$ is a collection of random numbers taken from a normal distribution of zero mean and standard deviation 1, and having elements equal to the dimension of the function in hand. The determination of m and the evolution of σ has been discussed earlier in section II.

To overcome the limitations of the CMA-ES algorithm discussed above, a DE mutation factor is incorporated in order to enhance the explorative power of the original CMA-ES algorithm. In DE the base vectors gets mutated (secondary parents) with scaled population-derived difference vectors. As generations pass, these differences tend to adapt to the natural scaling of the problem. For example, if the population becomes compact in one variable but remains widely dispersed in another, the difference vectors sampled from it will be small in the former variable, yet large in the latter. This automatic adaptation significantly improves the convergence of the algorithm. The explorative power of DE is also greater as compared to ES and accounts for its better performance significantly. These facts motivated us to use DE-type mutation in CMA-ES. In DCMA-EA the mutated vectors are generated as shown in the following equation below.

$$v_i^{(g)} = m \cdot (1-P) + P \cdot x_i^{(g)} + F \cdot \left(x_{r_1^i}^{(g)} - x_{r_2^i}^{(g)} \right) + \sigma \cdot B \cdot D \cdot randn(\dim)^T \quad (35)$$

Where $x_{r_1^i}^{(g)}$ and $x_{r_2^i}^{(g)}$ are vectors chosen randomly from the current population. m is mean of the current population vectors. P is a typical control parameter which controls the participation of the mean vector of the current population and target vectors. F is the scale factor of differential evolution. σ is standard deviation of current population vectors. The symbols B and D have been discussed earlier.

B. Randomized Scale Factor

The scale factor F is an important control parameter that scales the difference vectors in differential evolution. If the value of F is kept a constant value the diversification of the system is lost as all the particles are perturbed by the same difference vector component. To overcome this drawback the scale factor F_i for each target vector is generated according to a uniformly distributed random number between 0.5 and 1 in each generation as shown below:

$$F_i = 0.5 + 0.5 \cdot rand(0,1) \quad (36)$$

where $rand(0,1)$ is a uniformly distributed random number within the range [0,1]. This permits random variations in the scaling of the difference vector allowing the individuals of the system to sample diverse zones of the search space enhancing the explorative power during the early stages of the search. At the later stages when majority of the population agents point towards the interior of the region in which the suspected global optima lies, due to the stochastically scaled difference vector the donor generated has a moderate chance of jumping to a better location in the multimodal search space lowering the probability to get trapped in a local optima.

C. Significance of the control parameter P

The control parameter in DCMA-EA has been dynamically increased from 0.5 to 1 with generation as shown in the equation below:

$$P = 0.5 \cdot \left(1 + \frac{iter}{iter\_\max} \right) \quad (37)$$

The main reason behind this control parameter is that to set a trade-off between the mean and the target vector in the mutation strategy. At every stage of the algorithm performance it requires a proper trade-off between explorative and exploitative power. Using only the mean vector in mutation does not help the cause because all the vectors then gets perturbed about only the same mean vectors which not only reduces the explorative power of the algorithm in first stage but also hampers the performance in the later stage due to the resultant reduced population diversity where the vectors need to avoid the traps in local optima of multimodal landscape. But when a proper percentage of the mean vector and the current target vectors are used there are some benefits to be had. Incorporation of the target vector helps the generated offspring to keep track of parent vectors to suitably set the evolution path and it also wipes out the stagnancy of the population because the each vector gets mutated by its original parent vector, not only the mean vector. So, this variation of the control parameter helps the vectors at later stages of the algorithm performance by eliminating the possibilities of the vectors to get stagnated in local optima as it compels the target vector to take part more in mutation at the later generations.

D. Introduction to Crossover

To avoid premature convergence at local optima, there is a need to enhance the potential diversity of the population. To circumvent this defect we have implemented the normal DE crossover mechanism after generating the donor vectors through mutation. The basic objective is component injection, i.e. the information contained in the parents is incorporated into the mutated vectors to generate the corresponding trial vectors or offspring. The DE family of algorithms can use two kinds of crossover methods - exponential (or two-point modulo) and binomial (or uniform). In this article we focus on the widely used binomial crossover that is performed on each of the D variables whenever a randomly generated number between 0 and 1 is less than or equal to the Cr value. In this case, the number of parameters inherited from the donor has a (nearly) binomial distribution. The entire scheme is discussed above in section III.

E. Crossover probability determination

At every generation, the crossover probability Cr_i for each mutated vector is independently generated in accordance with a normal distribution of mean Cr_m and standard deviation 0.1 as shown below:

$$Cr_i = Gaussian(Cr_m, 0.1) \qquad (38)$$

Now a value of Cr generated from the distribution described above may be greater than 1. Those Cr values are again regenerated. The value of Cr_m is set to be a slightly higher value (0.6) as it favours non-separable functions which exist largely in real world optimization field. The normal distribution is perfect for the crossover probability values as it has a short tail property. This property do not permit vary large values of Cr greater than 1 which ultimately would have to be truncated to unity. A unity value of Cr means that no component of the parent vector is inherited into the offspring which hinders the basic mechanism of DE.

F. Selection

In EAs exploitation is done via selection that promotes better individuals to the subsequent generation. One of the major demerits of the CMA-ES algorithm is that it is devoid of any selection mechanism and thus the exploitative power of the algorithm is lost. The basic selection procedure is the traditional competitive selection in which individuals are ranked according to their fitness. Next, fittest particles are transferred to the next iteration and those with lower fitness are discarded. But the major shortcomings of this selection are that it ignores the inferior solutions. But in many optimization problems it is seen that the less fit particles may provide useful information about the promising progress direction.

The alternative choice to competitive selection is the one-to-one greedy selection procedure of DE which overcomes this defect by giving preference to the particles with inferior fitness. In this process the fitness of the parents are compared with those of the offspring and the fitter individuals are passed on to the consequent generations. The entire selection operation is outlined in section III. Therefore, if the new trial vector/offspring yields an equal or lower value of the objective function, it replaces the corresponding target(parent) vector in the next generation; otherwise the target is retained in the population. Hence, the population either gets better (with respect to the minimization of the objective function) or remains the same in fitness status, but never deteriorates improving the convergence performance as well as the exploitative power of the proposed hybrid algorithm.

G. Parameter settings in DCMA-EA

In the original CMA-ES algorithm the population size is dependent on problem dimensionality. But in DCMA-EA the population size is kept constant for the DE mutation and crossover to perform successfully on a large no. of vectors. The population size has been kept fixed at 50 for all the problem dimensions considered here. The remaining parameters are initialized as follows:

$$Cr_m = 0.6,\ pop\_size = 50,\ \mu = \frac{pop\_size}{2}$$

$$w_i = \frac{w_i'}{\sum_{j=1}^{\mu} w_j'} \quad , \quad w_i' = \ln(\mu + 0.5) - \ln i \quad \forall\, i(1)\mu$$

$$\mu_{eff} = \frac{\left(\sum_{i=1}^{\mu} w_i\right)^2}{\sum_{i=1}^{\mu} w_i^{\,2}},\ c_\sigma = \frac{\mu_{eff} + 2}{\dim + \mu_{eff} + 5},\ c_c = \frac{4 + \frac{\mu_{eff}}{\dim}}{\dim + 4 + \frac{2 \cdot \mu_{eff}}{\dim}}$$

$$c_1 = \frac{2}{(\dim + 1.3)^2 + \mu_{eff}}$$

$$c_\mu = \min\left(1 - c_1, 2 \cdot \frac{\mu_{eff} - 2 + \frac{1}{\mu_{eff}}}{(\dim + 2)^2 + \mu_{eff}}\right)$$

Here, the symbol pop_size represents the population size and dim symbolizes the problem dimension.

V. Experimental Setup and Results

In order to experimentally illustrate the effectiveness of the proposed hybrid algorithm, we choose a test bench comprising of twenty numerical benchmarks [10, 11] of different characteristics such as multimodality, multiplicative noise in fitness, shift in dimensionality, function rotation as well as composition of functions. The functions are stated below:

1) f_1: Sphere function [10].
2) f_2: Schwefel's Problem 2.22 [10].

3) f_3: Schwefel's Problem 1.2 [10].
4) f_4: Schwefel's Problem 2.21 [10].
5) f_5: Rosenbrock function [10].
6) f_6: Step function [10].
7) f_7: Shifted Schwefel's Problem 1.2 with noise in fitness [11].
8) f_8: Shifted Rotated Griewank's function without bounds [11].
9) f_9: Shifted Rotated Rastrigin function [11].
10) f_{10}: Ackley function [10].
11) f_{11}: Griewank's function [10].
12) f_{12}: Penalized Function 1 [10].
13) f_{13}: Penalized Function 2 [10].
14) f_{14}: Hybrid Composition Function 1 [11].
15) f_{15}: Rotated Hybrid Composition Function 1 [11].
16) f_{16}: Rotated Hybrid Composition Function 1 with noise in fitness [11].
17) f_{17}: Rotated Hybrid Composition Function 2 [11].
18) f_{18}: Rotated Hybrid Composition Function 2 with a Narrow basin for the Global Optimum [11].
19) f_{19}: Rotated Hybrid Composition Function 2 with the Global Optimum on the Bounds [11].
20) f_{20}: Rotated Hybrid Composition Function 4 without Bounds [11].

In order to validate the benefits of incorporating DE-based operators in CMA-ES, we compare its performance with the ancestor algorithms CMA-ES and DE/rand/1/bin. It has also been compared with the recently renowned DE variants SaDE, JADE and a very powerful and real parameter optimizer of current interest: DMS-PSO. For the contestant algorithms we choose the best suited parametric set-up for fair comparison chosen with guidelines from their respective literatures. Table I provides the mean and standard deviation of the best-of-run errors for 50 independent runs of each of the six algorithms on twenty numerical benchmarks for 30 dimensions. A nonparametric statistical test called Wilcoxon's rank sum test for independent samples [12, 13] is conducted at the 5% significance level in order to judge whether the results obtained with the best performing algorithm differ from the final results of rest of the competitors in a statistically significant way. Results of this test are summarized in Table II. In this table, + means result obtained with DCMA-EA is statistically significantly better than the corresponding algorithm, = means difference of the mean errors obtained with DCMA-EA and the corresponding algorithm is not statistically significant and – means results obtained with DCMA-EA is statistically worse than the concerned algorithm.

VI. Conclusion

Hybridization, in the context of evolutionary computing, primarily refers to the process of combining the best features of two or more algorithms together, to form a new EA that is expected to outperform its ancestors over application-specific or general benchmark problems. This proposition of hybridization (DCMA-EA) shows promising results as compared to the both DE and CMA-ES. Here, only the results of 30 dimensions are included due to space limitations. Our simulation studies have indicated that DCMA-EA has been able to achieve better results than both of its ancestors in 50 and 100 dimensions too showing its superiority when applied to higher dimensional problems. DCMA-EA can be adopted in future for tackling multi-objective, dynamic and niching optimization problems.

References

1. R. Storn and K. V. Price, "Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces", *Technical Report TR-95-012, ICSI*, http://http.icsi.berkeley.edu/~storn/litera.html, 1995.
2. R. Storn and K. Price, "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, 11(4) 341–359, 1997.
3. K. Price, R. Storn, and J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, Springer, Berlin, 2005.
4. S. Das and P. N. Suganthan, "Differential Evolution – a survey of the state-of-the-art", *IEEE Transactions on Evolutionary Computation*, (Accepted, 2010).
5. N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies", *Evolutionary Computation*, 9(2) pp. 159–195, 2001.
6. R. Mallipeddi, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies" *Applied Soft Computing,* DOI: 10.1016/j.asoc.2010.04.024.
7. A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization", *IEEE Transactions on Evolutionary Computation,* Vol. 13, Issue 2,, pp. 398-417, April, 2009.
8. J. Zhang, and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive", *IEEE Transactions on Evolutionary Computation*, Vol. 13, Issue 5, Page(s): 945-958, Oct. 2009.
9. S. Z. Zhao, J. J. Liang, P. N. Suganthan and M. F. Tasgetiren, Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search for Large Scale Global Optimization, CEC 2008.
10. X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, 3(2),82-102, July 1999.
11. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization", Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report#2005005, IIT Kanpur, India.
12. F. Wilcoxon, "Individual comparisons by ranking methods", *Biometrics*, 1, pp. 80-83, 1945.
13. S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 special session on real parameter optimization", *Journal of Heuristics*, Vol. 15, Issue 6, pp. 617-644, Dec. 2009.

TABLE I: EXPERIMENTAL RESULTS OF 30-DIMENSIONAL PROBLEMS $F_1 - F_{20}$, AVERAGED OVER 50 INDEPENDENT RUNS FOR 3E+05 FES

Functions	DE/rand/1/bin Mean(Std Dev)	SaDE Mean(Std Dev)	JADE Mean(Std Dev)	DMS-PSO Mean(Std Dev)	CMA-ES Mean(Std Dev)	DCMA-EA Mean(Std Dev)
f_1	9.8154e-30 (5.6276e-29)	4.5263e-36 (6.7834e-36)	1.3287e-72 (9.2543e-72)	1.3872e-62 (2.7583e-62)	4.346e-233 (8.7456e-233)	**3.0221e-293** **(9.8752e-294)**
f_2	1.6243e-18 (2.3453e-18)	1.9675e-23 (1.0547e-23)	3.9845e-32 (2.7345e-31)	9.3734e-30 (6.7324e-30)	2.993e-95 (1.324e-95)	**7.4940e-135** **(6.3575e-135)**
f_3	6.6243e-10 (8.4326e-10)	9.0145e-20 (5.4321e-19)	6.0198e-65 (1.9432e-84)	2.5643e-23 (3.9564e-22)	4.813e-140 (2.364e-140)	**1.5406e-165** **(8.0152e-166)**
f_4	1.0134e+00 (1.123e+00)	7.4123e-07 (1.823e-06)	4.3123e-45 (1.2344e-44)	4.4234e-18 (9.342e-18)	1.143e-73 (2.435e-73)	**1.0751e-98** **(6.9439e-99)**
f_5	2.134e+00 (1.523e+00)	2.1243e+01 (7.8254e+00)	3.2976e-01 (1.1254e+00)	4.784e-01 (1.322e+00)	4.702e-01 (1.234e+00)	**4.9834e-51** **(2.6743e-51)**
f_6	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**
f_7	5.04e-01 (8.58e-01)	5.8160e-06 (1.4479e-05)	5.1159e-07 (4.0194e-07)	2.5487e+03 (3.0638e+02)	1.3840e+04 (6.4478e+03)	**6.3702e-26** **(7.1254e-26)**
f_8	9.65e-001 (9.14e-002)	8.2727e-03 (1.1445e-02)	6.9598e-03 (4.4845e-03)	6.9990e-03 (4.5371e-03)	6.8843e-03 (2.8731e-02)	**1.1102e-16** **(1.1845e-16)**
f_9	6.16e+01 (4.56e+01)	3.5758e+01 (6.0809e+00)	3.0313e+01 (8.3551e+00)	3.7410e+01 (5.2883e+00)	4.8723e+01 (9.8124e+00)	**2.3890e+01** **(9.4523e+00)**
f_{10}	9.7234e-13 (5.0232e-13)	4.3523e-14 (2.6435e-14)	**4.4409e-15** **(0.00e+00)**	4.6534e-10 (6.654e-10)	**4.4409e-15** **(0.00e+00)**	**4.4409e-15** **(0.00e+00)**
f_{11}	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**	2.0143e-04 (1.4034e-03)	1.1078e-05 (1.634e-05)	**0.00e+00** **(0.00e+00)**	**0.00e+00** **(0.00e+00)**
f_{12}	1.1243e-19 (1.043e-19)	1.2453e-26 (2.034e-26)	1.600e-32 (5.500e-48)	1.9234e-18 (3.9876e-18)	1.600e-32 (5.500e-48)	**1.5705e-32** **(0.00e+00)**
f_{13}	7.5085e-20 (4.8452e-20)	1.7435e-28 (2.4345e-28)	1.400e-32 (1.100e-47)	2.9456e-20 (4.836e-20)	1.400e-32 (1.100e-47)	**1.3498e-32** **(0.00e+00)**
f_{14}	4.8443e+002 **(2.1412e+01)**	3.2775e+02 (9.6450e+01)	3.3284e+02 (9.0503e+01)	3.4400e+02 (5.0662e+01)	7.2856e+02 (3.9823e+01)	**3.12e+02** (2.6786e+01)
f_{15}	2.8293e+002 (1.1875e+001)	1.3795e+02 (1.7023e+01)	**7.4383e+01** **(3.9432e+01)**	1.1950e+02 (1.2068e+02)	2.2543e+02 (1.9832e+01)	9.2307e+01 (3.8872e+01)
f_{16}	3.0965e+002 (1.5871e+01)	1.5092e+03 (9.3631e+02)	1.0619e+02 (3.5763e+01)	1.1950e+02 (1.2068e+02)	2.2543e+02 (1.9832e+01)	**9.9328e+01** **(3.9757e+01)**
f_{17}	9.1344e+002 (8.43e-001)	9.5447e+02 (3.4387e+01)	8.9579e+02 (1.6523e-01)	9.1053e+02 (1.5761e+00)	9.0441e+02 (2.8822e-01)	**8.6726e+02** **(1.5209e-01)**
f_{18}	9.1331e+002 (1.21e+000)	8.4583e+02 (6.2155e+01)	8.7644e+02 (1.5419e-01)	9.1060e+02 (1.3383e+00)	9.0431e+02 (2.7115e-01)	**8.4325e+02** **(1.5340e-01)**
f_{19}	9.1332e+002 (1.16e+000)	2.0402e+03 (8.7683e+02)	9.1697e+02 (1.7231e-01)	9.0189e+02 (3.0719e+01)	9.0406e+02 (2.4837e-01)	**8.1626e+02** **(2.6786e-01)**
f_{20}	7.86e+02 (2.17e+01)	5.001e+02 (5.683e-02)	2.1123e+02 (4.2936e+00)	9.889e+02 (3.015e+01)	5.000e+02 (1.311e-04)	**2.1023e+02** **(3.6271e+00)**

TABLE II: WILCOXON RANK-SUM TEST RESULTS OF 30-DIMENSIONAL PROBLEMS $F_1 - F_{20}$

Functio	DE/rand/1/bin	SaDE	JADE	DMS-PSO	CMA-ES
f_1	+	+	+	+	+
f_2	+	+	+	+	+
f_3	+	+	+	+	+
f_4	+	+	+	+	+
f_5	+	+	+	+	+
f_6	=	=	=	=	=
f_7	+	+	+	+	+
f_8	+	+	+	+	+
f_9	+	+	+	+	+
f_{10}	=	=	=	+	=
f_{11}	=	=	+	+	=
f_{12}	+	+	=	+	=
f_{13}	+	+	=	+	=
f_{14}	+	+	+	+	+
f_{15}	+	+	-	+	+
f_{16}	+	+	=	+	+
f_{17}	+	+	+	+	+
f_{18}	+	=	=	+	+
f_{19}	+	+	+	+	+
f_{20}	+	+	=	+	+