Discrete Optimization

Bare bones differential evolution

Mahamed G.H. Omran a,*, Andries P. Engelbrecht b, Ayed Salman c

a Department of Computer Science, Gulf University for Science and Technology, Kuwait, Kuwait
b Department of Computer Science, University of Pretoria, Pretoria, South Africa
c Computer Engineering Department, Kuwait University, Kuwait, Kuwait

ARTICLE INFO

Article history:
Received 22 August 2007
Accepted 29 February 2008
Available online xxxx

Keywords:
Evolutionary computation
Differential evolution
Particle swarm optimization
Optimization

ABSTRACT

The barebones differential evolution (BBDE) is a new, almost parameter-free optimization algorithm that is a hybrid of the barebones particle swarm optimizer and differential evolution. Differential evolution is used to mutate, for each particle, the attractor associated with that particle, defined as a weighted average of its personal and neighborhood best positions. The performance of the proposed approach is investigated and compared with differential evolution, a Von Neumann particle swarm optimizer and a barebones particle swarm optimizer. The experiments conducted show that the BBDE provides excellent results with the added advantage of little, almost no parameter tuning. Moreover, the performance of the barebones differential evolution using the ring and Von Neumann neighborhood topologies is investigated. Finally, the application of the BBDE to the real-world problem of unsupervised image classification is investigated. Experimental results show that the proposed approach performs very well compared to other state-of-the-art clustering algorithms in all measured criteria.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Particle swarm optimization (PSO) (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) and differential evolution (DE) (Storn and Price, 1995) are two stochastic, population-based optimization methods, which have been applied successfully to a wide range of problems as summarized in Engelbrecht (2005), Kennedy et al. (2001) and Price et al. (2005). One class of variations includes hybrids between PSO and DE, where the advantages of the two approaches are combined. This paper empirically analyzes another PSO–DE hybrid algorithm proposed by the authors (Omran et al., 2007), which combines concepts from the barebones PSO (Kennedy, 2003) and the recombination operator of DE. The resulting algorithm, referred to as the barebones DE (BBDE) eliminates the control parameters of PSO and replaces the static DE control parameters with dynamically changing parameters to produce an almost parameter-free, self-adaptive, optimization algorithm. Furthermore, the performance of the BBDE using the ring and Von Neumann neighborhood topologies is investigated. Finally, a BBDE-based clustering algorithm is proposed and applied to the real-world problem of unsupervised image classification with promising results.

The remainder of the paper is organized as follows: differential evolution is summarized in Section 2. Section 3 provides an overview of PSO and variants used in this paper. Several PSO and DE hybrids are discussed in Section 4. The barebones DE (BBDE) is presented in Section 5. Section 6 presents and discusses the results of the experiments. The problem of unsupervised image classification is investigated in Section 7. Finally, Section 8 concludes the paper.

2. Differential evolution

Differential evolution (DE) is an evolutionary algorithm proposed by Storn and Price (1995). While DE shares similarities with...
other evolutionary algorithms (EA), it differs significantly in the sense that distance and direction information from the current population is used to guide the search process. DE uses the differences between randomly selected vectors (individuals) as the source of random variations for a third vector (individual), referred to as the target vector. Trial solutions are generated by adding weighted difference vectors to the target vector. This process is referred to as the mutation operator where the target vector is mutated. A recombination, or crossover step is then applied to produce an offspring which is only accepted if it improves on the
previously generated offspring. Liu and Lampinen (2002) proposed a fuzzy DE where the control parameters (i.e. \( F = 0.5 \)) provides on average good results. Liu and Lampinen (2002) proposed a fuzzy DE where the control parameters (i.e. \( F = 0.5 \)) provides on average good results.

### Nomenclature

- \( F \): DE scalar factor
- \( p_r \): DE probability of reproduction
- \( N_d \): number of parameters of a single individual
- \( v_i \): current velocity of particle \( i \)
- \( y_i \): personal best position of particle \( i \)
- \( y_i^* \): neighborhood best position of particle \( i \)
- \( y \): position of the global best particle
- \( w \): PSO inertia weight
- \( c_1 \) and \( c_2 \): are the PSO acceleration coefficients
- \( s \): population size
- \( f \): denotes the function being optimized
- \( t \): denotes time or time steps

Although empirical studies have shown that DE convergence is relatively insensitive to control parameter values, performance can be greatly improved if parameter values are optimized. A number of DE strategies have been developed where values for control parameters adapt dynamically. Abbass et al. (2001) proposed an approach where a new value is sampled for the scale factor for each application of the mutation operator. The scale factor is sampled from a Gaussian distribution, \( F \sim N(0,1) \). This approach is also used in Rönkkönen et al. (2005) and Qin and Suganthan (2005). In Qin and Suganthan (2005), the mean of the distribution was changed to 0.5 and the deviation to 0.3 (i.e. \( F \sim N(0.5,0.3) \)), due to the empirical results that suggest that \( F = 0.5 \) provides on average good results. Liu and Lampinen (2002) proposed a fuzzy DE where the control parameters (i.e. \( F \) and \( p_r \)) are adapted using fuzzy logic. Human knowledge and previous experience are used to establish the fuzzy rules and membership functions. A problem with this approach is that it is not very objective, and depends on how good the knowledge of the expert is.

Abbass (2002) proposed the Self-adaptive Pareto DE (SPDE), a self-adaptive approach to DE for multi-objective optimization problems. In SPDE, the parameter \( F \) is generated for each variable from a normal distribution, \( N(0,1) \). Each individual, \( i \), has its own probability of reproduction, \( p_i \). The parameter \( p_i \) is first initialized for each individual in the population from a uniform distribution between 0 and 1. Then, \( p_i \) is adapted as

\[
p_i(t) = p_i(t_0) + N(0, 0.1) \times (p_i(t_0) - p_i(t)),
\]

where \( t_i, t_2, t_3 \sim U(1, \ldots, s) \), and \( t_i = t_2 \neq t_3 \). According to Abbass (2002), the proposed approach performed well compared to other evolutionary multi-objective optimization approaches.

Omran et al. (2005c) proposed a self-adaptive DE strategy which makes use of the approach in Eq. (2) to dynamically adapt the scale factor. The mutation operator changes as follows:

\[
\mu(t) = \begin{cases} 
\nu_j(t) & \text{if } U(0, 1) < p_j \text{ or } j = r \\
\nu_j(t) & \text{otherwise}
\end{cases}
\]

where \( j = 1, \ldots, N_d \) refers to a specific dimension, \( N_d \) is the number of genes (parameters) of a single chromosome, and \( r \sim U(1, \ldots, N_d) \). In the above, \( p_j \) is the probability of reproduction (with \( p_j \in [0,1] \)).

Thus, each offspring is a stochastic linear combination of three randomly chosen individuals when \( U(0, 1) < p_j \); otherwise the offspring inherits directly from the parent. Even when \( p_j = 0 \), at least one of the parameters of the offspring will differ from the parent (forced by the condition \( j = r \)).

Selection: DE evolution implements a very simple selection procedure. The generated offspring, \( \mu(t) \), replaces the parent, \( \chi(t) \), only if the fitness of the offspring is better than that of the parent.

Storn (1996) and Storn and Price (1997) proposed ten different strategies for DE based on the individual being perturbed, the number of individuals used in the mutation process and the type of crossover used. The strategy described above is known as DE/rand/1, meaning that the target vector is randomly selected, and only one difference vector is used. This strategy is considered to be the most widely used and it is the one used in this paper. Other main strategies include DE/best/1, DE/best/2, DE/rand/2 and DE/rand-to-best/1. The notation, DE/x/y, is used where \( x \) represents the individual being perturbed and \( y \) is the number of difference vectors used to perturb \( x \). For a detailed discussion of these strategies, the reader is referred to Storn (1996).

### 3. Particle swarm optimization

Particle swarm optimization (PSO) (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) is a stochastic, population-based optimization algorithm modeled after the simulation of social behavior of bird flocks. In a PSO system, a swarm of individuals (called particles) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its neighborhood (i.e. the experience of neighboring particles). Particle position, \( \chi \), is adjusted using...
\[ x_i(t+1) = x_i(t) + \psi_i(t+1), \]
\[ \text{where the velocity component, } \psi_i \text{, represents the step size. For the basic PSO,} \]
\[ v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2r_{2j}(t)(\bar{y}_j(t) - x_{ij}(t)), \]
\[ \text{where } w \text{ is the inertia weight (Shi and Eberhart, 1998), } c_1 \text{ and } c_2 \text{ are the acceleration coefficients,} \]
\[ r_{1j}, r_{2j} \sim \text{U(0,1),} \]
\[ y_{ij}(t) \text{ is the personal best position of particle } i, \]
\[ \bar{y}_j \text{ is the neighborhood of particle } i. \]
\[ \text{The neighborhood best position } y_{ij} \text{ of particle } i \text{ depends on the neighborhood topology used (Kennedy and Mendes, 2002). If a fully connected topology is used, then } y_{ij} \text{ refers to the best position found by the entire swarm. That is} \]
\[ y_{ij}(t) = \{y_{ij}(t), y_{j1}(t), \ldots, y_{ji}(t)\} = \min \{f(y_{i1}(t)), f(y_{i2}(t)), \ldots, f(y_{in}(t))\}, \]
\[ \text{where } s \text{ is the swarm size.} \]

The resulting algorithm is referred to as the global best (gbest) PSO. For the ring topology, the swarm is divided into overlapping neighborhoods of particles. In this case, \( y_{ij} \) is the best position found by the neighborhood of particle \( i \). The resulting algorithm is referred to as the local best (lbest) PSO. The Von Neumann topology defines neighborhoods by organizing particles in a lattice structure. A number of empirical studies have shown that the Von Neumann topology outperforms other neighborhood topologies (Kennedy and Mendes, 2002; Peer et al., 2003). It is important to note that neighborhoods are determined using particle indices, and are not based on any spatial information.

A large number of PSO variations have been developed, mainly to improve the accuracy of solutions, diversity, and convergence behavior (Engelbrecht, 2005; Kennedy et al., 2001). This section reviews those variations used in this study, from which concepts have been borrowed to develop a new, almost parameter-free PSO algorithm.

van den Bergh and Engelbrecht (2006) and Clerc and Kennedy (2002) proved that each particle converges to a weighted average of its personal best and neighborhood best position, that is,
\[ \lim_{t \to \infty} x_{ij}(t) = \frac{c_1y_{ij} + c_2\bar{y}_j}{c_1 + c_2}. \]
This theoretically derived behavior provides support for the barebones PSO (BB) developed by Kennedy (2003). The BB replaces Eqs. (3) and (4) with
\[ x_{ij}(t+1) = N\left(\frac{y_{ij}(t) + \bar{y}_j(t)}{2}, |y_{ij}(t) - \bar{y}_j(t)|\right). \]

Particle positions are therefore randomly selected from a Gaussian distribution with the mean given as the weighted average of the personal best and global best positions, i.e. the swarm attractor. Note that exploration is facilitated via the deviation, \( y_{ij}(t) - \bar{y}_j(t) \), which approaches zero as \( t \) increases. In the limit, all particles will converge on the attractor point.

Kennedy (2003) also proposed an alternative version of the BB, referred to as BBExp in this paper, where Eqs. (3) and (4) are replaced with
\[ x_{ij}(t+1) = \begin{cases} \left(\frac{N(y_{ij}(t) + \bar{y}_j(t))}{2}, |y_{ij}(t) - \bar{y}_j(t)|\right) & \text{if } U(0,1) > 0.5 \\ y_{ij}(t) & \text{otherwise} \end{cases} \]
Based on the above equation, there is a 50% chance that the \( j \) th dimension of the particle dimension changes to the corresponding personal best position. This version of PSO biases towards exploiting personal best positions.

4. PSO and DE hybrids

Hendtlass (2001) used the DE perturbation approach to adapt particle positions. Particle positions are updated only if their offspring have better fitnesses. That is (assuming a minimization problem),
\[ x_i(t+1) = \begin{cases} x_i(t) + \frac{1}{2} (\bar{y}_j(t) - x_i(t)) & \text{if } f(x_i(t+1)) < f(x_i(t)) \\ x_i(t) & \text{otherwise} \end{cases} \]

The DE reproduction process is applied to the particles in a PSO swarm at specified intervals. At the specified intervals, the PSO swarm serves as the population for a DE algorithm, and the DE is executed for a number of generations. After execution of the DE, the evolved population is then further optimized using PSO. Kannan et al. (2004) applied DE to each particle for a number of iterations, and replaces the particle with the best individual obtained from the DE process.

Zhang and Xie (2003) and Talbi and Batouche (2004) used the DE operator to provide mutations. A trait point \( y_i(t) \) is calculated as follows:
\[ y_{ij}(t) = \begin{cases} y_j(t) + \frac{y_{ij}(t) - y_{ij}(t)}{2} & \text{if } U(0,1) < p_r, \text{ or } j = r \\ y_j(t) & \text{otherwise} \end{cases} \]
where \( j = 1, \ldots, N_d, r \sim \text{U(1, } \ldots, N_d) \) and \( y_{ij}(t) \) and \( y_{j2}(t) \) are randomly chosen personal best positions. Then, \( y_i(t) = y_{ij}(t) \text{ only if } f(y_{ij}(t)) < f(y_j(t)) \) (assuming a minimization problem).

5. Barebones differential evolution

Both PSO and DE have their strengths and weaknesses. One of the problems which both algorithms share is that control parameters need to be optimized for each new problem, as best parameter values are problem dependent. PSO has the advantage that formal proofs exist to show that particles will converge to a single attractor. The barebones PSO utilizes this information by sampling candidate solutions, normally distributed around the formally derived attractor point. Additionally, the barebones PSO has no parameters to be tuned. DE has the advantage of not being biased towards any prior defined distribution for sampling mutational step sizes and its selection operator follows a hill-climbing process. Mutational step sizes are determined as differences between individuals in the current population. Furthermore, very efficient self-adaptive DE strategies have been developed that eliminate the need for optimizing control parameters.

The barebones DE strategy proposed in this section capitalizes on these strengths of both the barebones PSO and self-adaptive DE strategies, to form a new, efficient hybrid optimization algorithm. For the barebones DE, position updates are done as follows:
\[ x_{ij}(t) = \begin{cases} p_{ij}(t) + r_{2j} \times (x_{ij}(t) - x_{ij}(t)) & \text{if } U(0,1) > p_r \\ y_{ij}(t) & \text{otherwise} \end{cases} \]
where from Eq. (6),
\[ p_{ij}(t) = r_{ij}(t)y_{ij}(t) + (1 - r_{ij}(t))y_{ij}(t) \]
with \( i_1, i_2, i_3 \sim \text{U(1, } \ldots, s), \) \( i_1 \neq i_2 \neq i \), \( r_{1j}, r_{2j} \sim \text{U(0,1)} \) and \( p_r \) is the probability of recombination.

Referring to Eq. (10), \( p_r(t) \) represents the particle attractor as a stochastic weighted average of personal best and global best positions, borrowing from the barebones PSO. Referring to Eq. (9), the mutation operator of DE is used to explore around the current attractor, \( p_r(t) \), by adding a difference vector to the attractor. Cross-over is done with a randomly selected personal best, \( y_{ij} \), as these personal bests represent a memory of best solutions found by individuals since the start of the search process. Also note that the
scale factor is a random variable. Using the position update in Eq. (10), for a proportion of \((1 - p_t)\) of the updates, information from a randomly selected personal best, \(y_j\), is used (facilitating exploitation), while for a proportion of \(p_t\) of the updates step sizes are mutations of the attractor point, \(p\) (facilitating exploration). Mutation step sizes are based on the difference vector between randomly selected particles, \(x_i\) and \(x_k\). Using the above, the BBDE achieves a good balance between exploration and exploitation. It should also be noted that the exploitation of personal best positions does not focus on a specific position. The personal best position, \(y_j\), is randomly selected for each application of the position update.

6. Experimental results

This section compares the performance of the barebones DE (BBDE) with that of the two barebones PSO algorithms discussed in Section 3, a PSO using the Von Neumann neighborhood topology, and the DE/rand/1/bin strategy. For the DE, \(F = 0.5\) and \(p_t = 0.9\), as suggested in Storn and Price (1997). For the PSO algorithms, \(w = 0.72\) and \(c_1 = c_2 = 1.49\). These values were used as defaults for all experiments which use static strategies in all the unimodal benchmark functions. The results have been shown to provide very good results (Clerc and Kennedy, 2002; van den Bergh, 2002; van den Bergh and Engelbrecht, 2006).

For all the algorithms used in this section, \(s = 50\). All functions were implemented in 30 dimensions except for the two-dimensional Camel-back function. Unless otherwise specified, these values were used as defaults for all experiments which use static control parameters. The initial population was generated from a uniform distribution in the ranges specified below.

The following functions have been used to compare the performance of the BBDE with that of other methods. These benchmark functions provide a balance of unimodal and multimodal functions.

For each of these functions, the goal is to find the global minimizer, formally defined as

\[
\text{minimize } f(x), \quad x \in \mathbb{R}^n
\]

where \(x^* = \arg\min_{x \in \mathbb{R}^n} f(x)\).

F. Generalized Swefel's Problem 2.26 (Yao et al., 1999), defined as

\[
f(x) = -\sum_{i=1}^{N_x} \left( x_i \sin(\sqrt{|x_i|}) \right),
\]

where \(x^* = (420.9687, \ldots, 420.9687)\) and \(f(x^*) = -12569.5\) for \(-500 \leq x_i \leq 500\).

G. Rastrigin function, defined as

\[
f(x) = \sum_{i=1}^{N_x} \left( x_i^2 - 10 \cos(2\pi x_i) + 10 \right),
\]

where \(x^* = 0\) and \(f(x^*) = 0\) for \(-5.12 \leq x_i \leq 5.12\).

H. Ackley's function, defined as

\[
f(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{N_x} \sum_{i=1}^{N_x} x_i^2} \right) - \exp \left( \frac{1}{N_x} \sum_{i=1}^{N_x} \cos \left( 2\pi x_i \right) \right) + 20 + e,
\]

where \(x^* = 0\) and \(f(x^*) = 0\) for \(-32 \leq x_i \leq 32\).

I. Griewank function, defined as

\[
f(x) = \frac{1}{4000} \sum_{i=1}^{N_x} x_i^2 - \sum_{i=1}^{N_x} \frac{x_i^2}{i} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1,
\]

where \(x^* = (0, 0, \ldots, 0)\) and \(f(x^*) = 0\) for \(-600 \leq x_i \leq 600\).

J. Six-hump Camel-back function, defined as

\[
f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_2^6 + x_1x_2 - 4x_1^2 + 4x_2^2,
\]

where \(x^* = (-0.08983, 0.7126), (-0.08983, 0.7126)\) and \(f(x^*) = -1.0316285\) for \(-5 \leq x_i \leq 5\).

Sphere, Schwefel's Problem 2.22, Rosenbrock and rotated hyper-ellipsoid are unimodal, while the Step function is a discontinuous unimodal function. Schwefel's Problem 2.26, Rastrigin, Ackley and Griewank are difficult multimodal functions where the number of local optima increases exponentially with the problem dimension. The Camel-back function is a low-dimensional function with only a few local optima.

The results reported in this section are averages and standard deviations over 30 simulations. Each simulation was allowed to run for 50,000 evaluations of the objective function. The statistically significant best solutions have been shown in bold (using the z-test with \(z = 0.05\)).

Table 1 summarizes the results obtained by applying the different approaches to the unimodal benchmark functions. The results show that the DE is the best performer followed by the BBDE. Thus, the BBDE performed better than (or at least equal to) the other PSO strategies in all the unimodal functions. Fig. 1 illustrates results for selected functions. For the Sphere function, Fig. 1a shows that the BBDE achieved a faster reduction in fitness than the other algorithms. For the Rotated hyper-ellipsoid function, Fig. 1b shows that the DE reached a good solution faster than the other approaches. The BBDE was slower than the DE but better than the remaining approaches. The barebones PSO algorithms had problems with the Rotated hyper-ellipsoid function, with results significantly worse than that of the other approaches. Fig. 2 illustrates diversity for selected functions. Diversity has been calculated using

\[
\text{diversity} = \frac{1}{S} \sum_{i=1}^{S} \sqrt{\frac{1}{N_x} \sum_{j=1}^{N_x} (x_j(t) - \bar{x}_j(t))^2},
\]

where \(x_j(t)\) is the average of the jth dimension over all individuals, i.e.,

\[
\bar{x}_j(t) = \frac{1}{S} \sum_{i=1}^{S} x_j(t).
\]
For the Sphere function, the BBDE exhibited the fastest reduction in diversity enabling it to converge faster than the other approaches. For the Rotated hyper-ellipsoid function, both the BBDE and DE exhibited the fastest reduction in diversity. The BBExp was the slowest in diversity reduction which might cause its slow convergence.

Table 2 summarizes the results obtained by applying the different approaches to the multimodal functions. The results show that the BBDE performed better than (or at least equal to) the other methods in all the test functions except the Rastrigin function where the BBExp found a better solution. The DE had problems with the Rastrigin function, with results significantly worse than that of the other methods. VN achieved the fastest reduction in fitness when applied to the Rastrigin function, nevertheless, the BBExp reached a better solution. The BB has the lowest diversity which explains its premature convergence behavior. On the other hand, the DE has the highest diversity, hence, it needs more time to converge to a good solution.

For the Griewank function, Figs. 3b and 4b show that the BBDE exhibited the fastest reduction in fitness and diversity but does not prematurely converge as in the case for the BB.

Table 2
Mean and standard deviation (±SD) of the multimodal function optimization results

<table>
<thead>
<tr>
<th>Function</th>
<th>Schwefel problem 2.26</th>
<th>Rastrigin</th>
<th>Ackley</th>
<th>Griewank</th>
<th>Camel-back</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>11052.61419 (1178.89586)</td>
<td>140.59464 (27.27772)</td>
<td>0 (0)</td>
<td>0.005007 (0.00676)</td>
<td>-1.031628 (0)</td>
</tr>
<tr>
<td>VN</td>
<td>8629.400262 (588.165184)</td>
<td>37.470096 (9.435686)</td>
<td>0 (0)</td>
<td>0.004360 (0.005910)</td>
<td>-1.031628 (0)</td>
</tr>
<tr>
<td>BBExp</td>
<td>9091.017809 (561.296234)</td>
<td>87.572345 (23.70764)</td>
<td>2.264577 (5.959689)</td>
<td>0.011234 (0.012018)</td>
<td>-1.031628 (0)</td>
</tr>
<tr>
<td>BBDE</td>
<td>11649.008729 (272.707782)</td>
<td>37.551246 (15.254959)</td>
<td>0 (0)</td>
<td>0.000657 (0.002583)</td>
<td>-1.031628 (0)</td>
</tr>
</tbody>
</table>

Please cite this article in press as: Omran, M.G.H. et al., Bare bones differential evolution, European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.02.035
6.1. Effect of noise on performance

This section investigates the effect of noise on the performance of BBDE. The noisy versions of the benchmark functions are defined as

$$f_{\text{Noisy}}(x) = f(x) + N(0, 1),$$

where $N(0, 1)$ represents the normal distribution with zero mean and standard deviation of one.

Tables 3 and 4 summarize the results obtained for the noisy problems for the unimodal and multimodal functions, respectively. Table 3 shows that for the Sphere, Schwefel Problem 2.22 and Step functions, the BBDE was less prone to noise than the other approaches. Table 4 shows that the BBDE retained its position as the best performer when applied to multimodal functions even in the presence of noise. The only exception is the noisy Rastrigin function where BBExp outperformed the BBDE. Hence, compared to the other tested approaches, the BBDE seems to be less badly affected by noise. This is a significant improvement over the DE which is known to be badly affected by the presence of noise (Krink et al., 2004). There are two expected reasons for this improvement. The first is that the BBDE does not use a deterministic selection mechanism. Deterministic selection is expected to be a main factor for the DE deficiency when applied to noisy problems (Krink et al., 2004). The second reason is that the BBDE uses a random scaling factor that enables it to be more stochastic and, thus, has a better exploration capability than the conventional DE. Being less prone to noise is an important advantage for the BBDE given that optimizing noisy or imprecise functions occurs in many engineering applications.

6.2. Scalability study

When the dimension of the functions increases from 30 to 100, the performance of the DE degraded significantly as shown in Tables 5 and 6. The results show that the BBDE is the best performer followed by the VN. Fig. 5 illustrates results for representative functions. Figs. 5a and d show that the BBDE achieved a faster reduction in fitness than the other algorithms when applied to the Sphere and Griewank functions. For the Rotated hyper-ellipsoid and Rastrigin functions, Figs. 5b and c show that the VN reached better solutions faster than the other methods. In general, Fig. 5 shows that the barebones PSO methods had difficulties with the high-dimensional problems. Fig. 6 illustrates diversity for the selected functions. For the Sphere and Griewank functions (Figs. 6a and d), the BBDE exhibited the fastest reduction in diversity.
Fig. 5. Performance Comparison of the different methods when applied to representative functions (Nd = 100).

<table>
<thead>
<tr>
<th>Table 4</th>
<th>Mean and standard deviation (±SD) of the noisy multimodal function optimization results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Schwefel problem 2.26</td>
</tr>
<tr>
<td>DE</td>
<td>-10925.046322 (1254.046813)</td>
</tr>
<tr>
<td>VN</td>
<td>-8715.290515 (634.156051)</td>
</tr>
<tr>
<td>BB</td>
<td>-1219.050539 (406.954831)</td>
</tr>
<tr>
<td>BBEXP</td>
<td>-10271.531191 (478.895628)</td>
</tr>
<tr>
<td>BBDE</td>
<td>-11584.222894 (287.881147)</td>
</tr>
</tbody>
</table>

Table 5 | Mean and standard deviation (±SD) of the unimodal function optimization results (Nd = 100) |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Schwefel problem 2.22</td>
</tr>
<tr>
<td>DE</td>
<td>0.009534 (0.011473)</td>
</tr>
<tr>
<td>VN</td>
<td>0.000578 (0.000137)</td>
</tr>
<tr>
<td>BB</td>
<td>4.764215 (4.733918)</td>
</tr>
<tr>
<td>BBEXP</td>
<td>0.388195 (0.203691)</td>
</tr>
<tr>
<td>BBDE</td>
<td>0 (0)</td>
</tr>
</tbody>
</table>

Table 6 | Mean and standard deviation (±SD) of the multimodal function optimization results (Nd = 100) |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Schwefel problem 2.26</td>
</tr>
<tr>
<td>DE</td>
<td>-19438.479681 (2294.731006)</td>
</tr>
<tr>
<td>VN</td>
<td>-24706.562508 (1445.236154)</td>
</tr>
<tr>
<td>BB</td>
<td>-28796.484336 (1227.159007)</td>
</tr>
<tr>
<td>BBEXP</td>
<td>-29791.507120 (910.130844)</td>
</tr>
<tr>
<td>BBDE</td>
<td>-34746.152554 (3750.927593)</td>
</tr>
</tbody>
</table>

Please cite this article in press as: Omran, M.G.H. et al., Bare bones differential evolution, European Journal of Operational Research (2008), doi:10.1016/j.ejor.2008.02.035
enabling it to converge faster than the other approaches. For the Rotated hyper-ellipsoid and Rastrigin functions (Figs. 6b and c), the BB exhibited the fastest reduction in diversity resulting in its premature convergence. In general, the BBExp was the slowest in diversity reduction which might cause its slow convergence.

6.3. Influence of \( p_r \) on the performance of BBDE

The performance of the barebones DE has been evaluated for \( p_r \in \{0.1, 0.3, 0.5, 0.7, 0.9\} \). Table 7 summarizes the results. The results suggest that smaller values of \( p_r \) (i.e. \( p_r = 0.1 \)) are preferable for unimodal functions while \( p_r = 0.7 \) yields the best results for multimodal functions. In general, \( p_r = 0.7 \) seems to generate the best results when applied to the benchmark problems. On the other hand, \( p_r = 0.9 \) degraded the performance of the BBDE significantly.

The effect of using different strategies to calculate \( p_r \) is summarized in Table 8. The two strategies investigated were as follows:

A. Parameter-free BBDE (PFBBDE): The same strategy as the BBDE, except that \( p_r = N(0.5, 0.15) \). The rationale behind using a normal distribution \( N(0.5, 0.15) \) for \( p_r \) is that \( N(0.5, 0.15) \) will generate values in the range of \([0.5 - 3 \times 0.15, 0.5 + 3 \times 0.15]\) which covers the \( p_r \) boundary giving more probability to values surrounding 0.5. The reason for preferring values surrounding 0.5 is that \( p_r = 0.5 \) represents a uniform crossover (i.e. there is an equal probability that the new offspring will be chosen from either the mutated attractor point or from a randomly chosen personal best).

B. Dynamically adjustable BBDE (DBBDE): The same strategy as the BBDE, except that \( p_r = t/t_{max} \), where \( t \) is the current iteration and \( t_{max} \) is the maximum number of iterations. Thus, \( p_r \) is dynamically adjusted by increasing it linearly over time. Hence, the initial small value of \( p_r \) favors exploration in the early stages, and exploitation in the later stages.

Table 8 shows that BBDE and PFBBDE performed comparably for all the benchmark functions. In addition, they generally performed

**Table 7**

<table>
<thead>
<tr>
<th>Function</th>
<th>( p_r ) = 0.1</th>
<th>( p_r ) = 0.3</th>
<th>( p_r ) = 0.5</th>
<th>( p_r ) = 0.7</th>
<th>( p_r ) = 0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0.030934 (0.034928)</td>
</tr>
<tr>
<td>Schwefel problem</td>
<td>32.262004 (24.894417)</td>
<td>42.934501 (32.734171)</td>
<td>47.857080 (31.835408)</td>
<td>82.042890 (59.119772)</td>
<td>17527.428158 (35073.731685)</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0.415136 (0.362744)</td>
</tr>
<tr>
<td>Step</td>
<td>20.020990 (16.114680)</td>
<td>31.737502 (15.15135)</td>
<td>56.47487 (38.975253)</td>
<td>323.816583 (233.859497)</td>
<td>28.133333 (64.607426)</td>
</tr>
<tr>
<td>Rotated hyper-ellipsoid</td>
<td>–10682.8391 (510.01006)</td>
<td>–11291.0108 (325.921626)</td>
<td>–11649.08729 (272.707782)</td>
<td>–11792.802318 (224.658980)</td>
<td>1.031628 (0.00000)</td>
</tr>
<tr>
<td>Schwefel problem</td>
<td>2.26</td>
<td>153.779164 (16.391473)</td>
<td>106.832999 (12.946721)</td>
<td>37.551246 (15.254959)</td>
<td>15.245124 (4.110377)</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>2.651095 (1.404765)</td>
</tr>
<tr>
<td>Ackley</td>
<td>0.004763 (0.006335)</td>
<td>0.002545 (0.0006012)</td>
<td>0.000657 (0.002583)</td>
<td>0.000985 (0.003208)</td>
<td>2.651095 (1.404765)</td>
</tr>
<tr>
<td>Griewank</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031380 (0.000564)</td>
</tr>
<tr>
<td>Camel-back</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031628 (0.00000)</td>
<td>–1.031380 (0.000564)</td>
</tr>
</tbody>
</table>

Fig. 6. Performance Comparison of the different methods when applied to representative functions (\( N_d = 100\)). The vertical axis represents the diversity and the horizontal axis represents the number of generations.
better than the DBBDE for all the functions except for the Rastrigin function. Fig. 7 illustrates the results for selected functions. For the Rotated hyper-ellipsoid function, Figs. 7a and b show that the BBDE and the PFBBDE have very similar convergence and diversity characteristics. The DBBDE has a lower diversity and converged to a worse solution. For the Rastrigin function, Figs. 7c and d show that the BBDE and the PFBBDE have very similar (almost identical) convergence and diversity characteristics. On the other hand, the DBBDE exhibited a fast reduction in diversity after 500 iterations and reached a better solution.

6.4. Using neighborhood topologies with BBDE

The effect of neighborhoods on DE has been investigated by Omran et al. (2005c). The results in Omran et al. (2005c) showed that using the ring neighborhood topology (Fig. 8a) with DE (known as DE/lbest/1) generally improves the performance of DE. On the other hand, as already stated, using the Von Neumann topology (Fig. 8b) with PSO generally improves the performance of PSO. The purpose of this subsection is to investigate the performance of the BBDE using the ring and Von Neumann neighborhood topologies.

Table 8
Comparison of different variants of BBDE

<table>
<thead>
<tr>
<th>Function</th>
<th>BBDE</th>
<th>PFBBDE</th>
<th>DBBDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>Schwefel problem 2.22</td>
<td>47.857080 (31.835408)</td>
<td>55.552183 (59.870164)</td>
<td>76.371400 (90.076614)</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>0 (0)</td>
<td>0 (0)</td>
<td>0 (0)</td>
</tr>
<tr>
<td>Schwefel problem 2.26</td>
<td>56.467487 (38.975253)</td>
<td>72.755559 (51.147255)</td>
<td>185.222391 (112.377535)</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>-11649.00873 (272.707782)</td>
<td>-11601.481689 (290.840666)</td>
<td>-11098.220557 (367.719656)</td>
</tr>
<tr>
<td>Ackley</td>
<td>37.551246 (15.2549959)</td>
<td>34.132272 (17.814442)</td>
<td>10.785169 (5.924399)</td>
</tr>
<tr>
<td>Griewank</td>
<td>0 (0)</td>
<td>0.000657 (0.002583)</td>
<td>0.001479 (0.003461)</td>
</tr>
<tr>
<td>Camel-back</td>
<td>-1.031628 (0.000000)</td>
<td>-1.031628 (0.000000)</td>
<td>-1.031628 (0.000000)</td>
</tr>
</tbody>
</table>

The two proposed variants are as follows:
A. VNBDBE: For the VNBDBE, y_i(t) in Eq. (10) is the best position found so far in the ring neighborhood of the i-th individual (i.e. the two adjacent left and right neighbors of i).
B. RBBDE: For the RBBDE, y_i(t) in Eq. (10) is the best position found so far in the Von Neumann neighborhood of the i-th individual (i.e. the four immediately adjacent left, right, up and down neighbors of i).
Unsupervised image classification is the process of identifying groups of similar image primitives. These image primitives can be pixels, regions, line elements and so on, depending on the problem encountered. Many basic image processing techniques such as quantization, segmentation and coarsening can be viewed as different instances of the clustering problem (Puzicha et al., 2000).

Herein the terminology used throughout the rest of the paper is defined. A measure is given to quantify the quality of a clustering algorithm, after which the BBDE-based clustering algorithm is introduced.

Define the following symbols:

- \( N_s \) denotes the number of spectral bands of the image set.
- \( N_p \) denotes the number of image pixels.
- \( N_c \) denotes the number of spectral classes (as provided by the user).
- \( \mathbf{z}_p \) denotes the \( N_s \) components of pixel \( p \).
- \( \mathbf{m}_j \) denotes the centroid (mean) of cluster \( j \).

7.1. Measure of quality

Different measures can be used to express the quality of a clustering algorithm. The most general measure of performance is the quantization error, defined as

\[
J_e = \sum_{k=1}^{K} \left( \frac{\sum_{i \in C_k} d(\mathbf{z}_p, \mathbf{m}_k)}{n_k} \right) / K,
\]

where \( C_k \) is the \( k \)th cluster, and \( n_k \) is the number of pixels in \( C_k \).

7.2. The BBDE-based clustering algorithm

In the context of data clustering, a single individual represents the \( K \) cluster centroids. That is, each individual \( \mathbf{x}_i \) is constructed as \( \mathbf{x}_i = (\mathbf{m}_{1k}, \ldots, \mathbf{m}_{Kk}) \) where \( \mathbf{m}_{jk} \) refers to the \( k \)th cluster centroid vector of the \( j \)th individual. Therefore, a population represents a number of candidate data clusterings. The quality of each individual is measured using

\[
f(\mathbf{x}_i, \mathbf{Z}) = \frac{d_{\text{max}}(\mathbf{Z}, \mathbf{x}_i) + J_e}{d_{\text{min}}(\mathbf{Z}, \mathbf{x}_i)},
\]

where \( \mathbf{Z} \) is a matrix representing the assignment of patterns to the clusters of individual \( i \). Each element \( z_{ik,p} \) indicates if pattern \( p \) belongs to cluster \( C_k \) of individual \( i \) (in which case \( z_{ik,p} \)). Also,

\[
d_{\text{max}}(\mathbf{Z}, \mathbf{x}_i) = \max_{k=1 \ldots K} \left\{ \frac{\sum_{p} d(\mathbf{z}_p, \mathbf{m}_k)}{n_k} \right\}
\]

is the maximum average Euclidean distance of individuals to their associated clusters, and

\[
d_{\text{min}}(\mathbf{x}_i) = \min_{k,k',k'' \neq k} \{ d(\mathbf{m}_k, \mathbf{m}_{k''}) \}
\]

is the minimum Euclidean distance between any pair of clusters. In the above, \( n_{kj} \) is the number of patterns that belong to cluster \( C_{kj} \) of individual \( i \).

The fitness function in Eq. (11) has as objective to simultaneously minimize the intra-cluster distance between patterns and their cluster centroids, as quantified by \( d_{\text{max}}(\mathbf{Z}, \mathbf{x}_i) \) and the quantization error, as quantified by \( J_e \), and to maximize the inter-cluster distance between any pair of clusters, as quantified by \( d_{\text{min}}(\mathbf{x}_i) \).

According to the definition of the fitness function, a small value of \( f(\mathbf{x}_i, \mathbf{Z}) \) suggests compact and well-separated clusters (i.e. good clustering).

The BBDE clustering algorithm is summarized in Fig. 9.

1. Initialize each individual to contain \( K \) randomly selected cluster centroids
2. For \( t = 1 \) to \( t_{\text{max}} \)
   (a) For each individual \( i \)
      i. For each pattern \( z_p \)
         - Calculate \( d(\mathbf{z}_p, \mathbf{m}_{ik}) \) for all clusters \( C_{ik} \)
         - Assign \( z_p \) to \( C_{ik} \) where
           \[
d(\mathbf{z}_p, \mathbf{m}_{ik}) = \min_{k \neq k'} \{ d(\mathbf{z}_p, \mathbf{m}_{ik'}) \}\n\]
      ii. Calculate the fitness, \( f(\mathbf{x}_i, \mathbf{Z}_i) \)
   (b) Find the global best solution \( \hat{\mathbf{y}}(t) \)
   (c) Apply the BBDE as described in Section 5 and update the cluster centroids using Eq. 9.

Fig. 9. The BBDE clustering algorithm.
7.3. Simulation results

The BBDE-based clustering algorithm has been applied to three types of imagery data, namely synthetic, MRI and LANDSAT 5 MSS (79 m GSD) images:

- **Synthetic image**: Fig. 10a shows a 100 × 100 8-bit gray scale image created to specifically show that the BBDE algorithm does not get trapped in the local minimum. The image was created using two types of brushes, one brighter than the other.
- **MRI image**: Fig. 10b shows a 300 × 300 8-bit gray scale image of a human brain, intentionally chosen for its importance in medical image processing.
- **Remotely sensed imagery data**: Fig. 10c shows band 4 of the four-channel multispectral test image set of the Lake Tahoe region in the US. Each channel is comprised of a 300 × 300, 8-bit per pixel (remapped from the original 6 bit) image. The test data are one of the North American Landscape Characterization (NALC) Landsat multispectral scanner data sets obtained from the U.S. Geological Survey (USGS).

The performance of the BBDE is compared to K-means (Forgy, 1965), FCM (Bezdek, 1980), a PSO clustering algorithm (Omran et al., 2005b), a DE clustering algorithm (Omran et al., 2005a), BB and BBExp. The fitness function for PSO, DE, BBExp and BBDE is defined in Eq. (11). For DE, PSO, BB, BBExp and BBDE, 50 individuals were trained for 100 iterations; for the other algorithms 5000 function evaluations were used. For K-means, FCM and PSO, the parameters were set as in Omran (2005). For DE, the parameters were set as in Fig. 10.

<table>
<thead>
<tr>
<th>Image</th>
<th>$J_e$</th>
<th>$d_{max}$</th>
<th>$d_{min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>K-means 20.21225 (0.937836)</td>
<td>28.04049 (2.777938)</td>
<td>78.4975 (7.0628718)</td>
</tr>
<tr>
<td></td>
<td>FCM 20.731920 (0.650023)</td>
<td>28.559214 (2.21067)</td>
<td>82.434116 (4.404686)</td>
</tr>
<tr>
<td></td>
<td>PSO 17.284 (0.09)</td>
<td>22.457 (0.414)</td>
<td>90.06 (0.712)</td>
</tr>
<tr>
<td></td>
<td>DE 17.349039 (0.024415)</td>
<td>22.080808 (0.045002)</td>
<td>89.674503 (0.0717472)</td>
</tr>
<tr>
<td></td>
<td>BB 17.324602 (0.026192)</td>
<td>22.292777 (0.036073)</td>
<td>89.706956 (0.029075)</td>
</tr>
<tr>
<td></td>
<td>BBExp 17.337303 (0.024775)</td>
<td>22.117476 (0.035281)</td>
<td>89.691711 (0.029087)</td>
</tr>
<tr>
<td></td>
<td>BBDE 17.329013 (0.023745)</td>
<td>22.224411 (0.034509)</td>
<td>89.703907 (0.030338)</td>
</tr>
<tr>
<td>MRI</td>
<td>K-means 7.3701 (0.042809)</td>
<td>13.214369 (0.761599)</td>
<td>9.93435 (7.308529)</td>
</tr>
<tr>
<td></td>
<td>FCM 7.205987 (0.166418)</td>
<td>10.851742 (0.960273)</td>
<td>19.517755 (2.014138)</td>
</tr>
<tr>
<td></td>
<td>PSO 7.819 (0.238)</td>
<td>9.197 (0.56)</td>
<td>29.45 (1.481)</td>
</tr>
<tr>
<td></td>
<td>DE 8.489362 (0.518571)</td>
<td>11.183335 (0.620451)</td>
<td>26.561583 (1.339439)</td>
</tr>
<tr>
<td></td>
<td>BB 7.722666 (0.091995)</td>
<td>8.940551 (0.563716)</td>
<td>29.969806 (0.909355)</td>
</tr>
<tr>
<td></td>
<td>BBExp 7.813303 (0.165253)</td>
<td>9.283251 (0.489863)</td>
<td>29.309396 (0.833855)</td>
</tr>
<tr>
<td></td>
<td>BBDE 8.149742 (0.386591)</td>
<td>10.311062 (1.399248)</td>
<td>29.37867 (0.977154)</td>
</tr>
<tr>
<td>Tahoe</td>
<td>K-means 3.280730 (0.095188)</td>
<td>5.234911 (0.312989)</td>
<td>9.402616 (2.823284)</td>
</tr>
<tr>
<td></td>
<td>FCM 3.164670 (0.000004)</td>
<td>4.999294 (0.000009)</td>
<td>10.706077 (0.000015)</td>
</tr>
<tr>
<td></td>
<td>PSO 3.882 (0.274)</td>
<td>5.036 (0.368)</td>
<td>16.410 (1.231)</td>
</tr>
<tr>
<td></td>
<td>DE 4.190698 (0.303445)</td>
<td>5.216843 (0.321845)</td>
<td>16.902606 (1.089620)</td>
</tr>
<tr>
<td></td>
<td>BB 3.980572 (0.102029)</td>
<td>5.131814 (0.246145)</td>
<td>17.037257 (0.861458)</td>
</tr>
<tr>
<td></td>
<td>BBExp 4.015715 (0.102513)</td>
<td>5.201299 (0.18996)</td>
<td>17.31936 (0.579856)</td>
</tr>
<tr>
<td></td>
<td>BBDE 4.007243 (0.117684)</td>
<td>5.130046 (0.155744)</td>
<td>17.102415 (0.514053)</td>
</tr>
</tbody>
</table>
Omran et al. (2005a). The results are summarized in Table 10. These results are averages and standard deviations over 20 simulation runs. Table 10 shows that BBDE generally outperformed K-means and FCM in \(d_{min}\) and \(d_{max}\), while performing comparably with respect to \(J_c\). The DE, PSO, BB and BBExp and BBDE showed similar performance. However, compared to DE and PSO, the BB, BBExp and BBDE almost require no parameter tuning. Thus, these approaches are the best choices to use. The segmented images resulting from the BBDE-based clustering algorithms are shown in Fig. 10.

8. Conclusions

This paper investigated a new population-based algorithm, as a hybrid of the barebones particle swarm optimizer (PSO) and differential evolution (DE). The particle position update is changed to probabilistically base a new position on a randomly selected personal best position, or a mutation of the particle attractor (i.e. weighted average of the personal best and neighborhood best). The BBDE does not make use of the standard PSO parameters (i.e. inertia weight, acceleration coefficients, and velocity clamping), and also removes the DE scale parameter. The only parameter is the probability of recombination. The approach was tested on ten benchmark functions where it generally performed very well compared to the other approaches. This paper investigated the effect of noise on the performance of BBDE and found that the BBDE is less prone to noise than the other approaches. Empirical results show that, in general, the BBDE provided the best results when applied to high-dimensional problems. Moreover, using different neighborhood topologies with BBDE was investigated. Finally, a clustering approach using BBDE was proposed. The BBDE clustering algorithm has as objective to simultaneously minimize the quantization error and intra-cluster distances, and to maximize the inter-cluster distances. The BBDE clustering algorithm was compared against K-means, FCM, DE, PSO, BB and BBDE with encouraging results.

References


