European Journal of Operational Research xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

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.

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). It is, however, the case that the performance of these methods are greatly influenced by their control parameters. Empirical and theoretical studies have shown that the convergence behavior of PSO is strongly dependent on the values of the inertia weight and the acceleration coefficients (Clerc and Kennedy, 2002; van den Bergh, 2002; van den Bergh and Engelbrecht, 2006). Wrong choices of values for these parameters may result in divergent or cyclic particle trajectories. The performance of DE, on the other hand, is influenced mainly by the scale parameter and the probability of recombination. Although recommendations for values of these parameters have been made in the literature (based on empirical studies) (Storn and Price, 1997), these values are not universally applicable. The best values for DE control parameters remain problem dependent, and need to be fine tuned for each problem.

A number of variations of both PSO and DE have been developed in the past decade to improve the performance of these algorithms (Engelbrecht, 2005; Kennedy et al., 2001; 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

* Corresponding author. Tel.: +965 530 7433; fax: +965 2645795.
E-mail addresses: Omran.m@gust.edu.kw (M.G.H. Omran), engel@cs.up.ac.za (A.P. Engelbrecht), ayed@eng.kuniv.edu.kw (A. Salman).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.02.035

Nomenclature

F	DE scalar factor	$\hat{\boldsymbol{y}}$	position of the global best particle
p_r	DE probability of reproduction	w	PSO inertia weight
N_d	number of parameters of a single individual		c_1 and c_2 are the PSO acceleration coefficients
$\boldsymbol{v}_i$	current velocity of particle i	s	population size
$\boldsymbol{y}_i$	personal best position of particle i	f	denotes the function being optimized
$\hat{\boldsymbol{y}}_i$	neighborhood best position of particle i	t	denotes time or time steps

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 fitness of the parent individual.

The basic DE algorithm is described in more detail below with reference to the three evolution operators: mutation, crossover, and selection.

Mutation: For each parent, $x_i(t)$, of generation t, a trial vector, $v_i(t)$, is created by mutating a target vector. The target vector, $\boldsymbol{x}_{i_3}(t)$, is randomly selected, with $i \neq i_3$. Then, two individuals $\boldsymbol{x}_{i_1}(t)$, and $\boldsymbol{x}_{i_2}(t)$ are randomly selected with $i_1 \neq i_2 \neq i_3 \neq i$, and the difference vector, $\boldsymbol{x}_{i_1} - \boldsymbol{x}_{i_2}$, is calculated. The trial vector is then calculated as

$$\boldsymbol{v}_i(t) = \boldsymbol{x}_{i_3}(t) + F(\boldsymbol{x}_{i_1}(t) - \boldsymbol{x}_{i_2}(t)), \tag{1}$$

where the last term represents the mutation step size. In the above, F is a scale factor used to control the amplification of the differential variation. Note that $F \in (0, \infty)$.

Crossover: DE follows a discrete recombination approach where elements from the parent vector, $\boldsymbol{x}_i(t)$, are combined with elements from the trial vector, $\boldsymbol{v}_i(t)$, to produce the offspring, $\boldsymbol{\mu}_i(t)$. Using the binomial crossover,

$$\mu_{ij}(t) = \begin{cases} v_{ij}(t) & \text{if } U(0,1) < p_r \text{ or } j = r \\ x_{ij}(t) & \text{otherwise} \end{cases},$$

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_r is the probability of reproduction (with $p_r \in [0,1]$).

Thus, each offspring is a stochastic linear combination of three randomly chosen individuals when $U(0,1) < p_r$; otherwise the offspring inherits directly from the parent. Even when $p_r = 0$, 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, $\boldsymbol{\mu}_i(t)$, replaces the parent, $\boldsymbol{x}_i(t)$, 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).

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 values of 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

$$p_i(t) = p_{i_1}(t) + N(0,1) \times (p_{i_2}(t) - p_{i_3}(t)), \tag{2}$$

where i_1, i_2, $i_3 \sim U(1,\ldots,s)$, and $i_1 \neq i_2 \neq i_3$. 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:

$$\boldsymbol{v}_i(t) = \boldsymbol{x}_{i_3}(t) + F_i(t)(\boldsymbol{x}_{i_1}(t) - \boldsymbol{x}_{i_2}(t)),$$

where

$$F_i(t) = F_{i_4}(t) + N(0,0.5) \times (F_{i_5}(t) - F_{i_6}(t))$$

with $i_4 \neq i_5 \neq i_6$ and i_4, i_5, $i_6 \sim U(1,\ldots,s)$.

The crossover probability can be sampled from a Gaussian distribution as discussed above, or adapted according to Eq. (2).

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, $\boldsymbol{x}_i$, are adjusted using

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

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \boldsymbol{v}_i(t+1), \tag{3}$$

where the velocity component, $\mathbf{v}_i$, represents the step size. For the basic PSO,

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + c_2 r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t)), \tag{4}$$

where w is the inertia weight (Shi and Eberhart, 1998), c_1 and c_2 are the acceleration coefficients, r_{1j}, $r_{2j} \sim U(0,1)$, $\boldsymbol{y}_i$ is the personal best position of particle i, and $\hat{\boldsymbol{y}}_i$ is the neighborhood best position of particle i.

The neighborhood best position $\hat{\boldsymbol{y}}_i$, of particle i depends on the neighborhood topology used (Kennedy, 1999; Kennedy and Mendes, 2002). If a fully connected topology is used, then $\hat{\boldsymbol{y}}_i$ refers to the best position found by the entire swarm. That is

$$\hat{\boldsymbol{y}}_i(t) \in \{\boldsymbol{y}_0(t), \boldsymbol{y}_1(t), \ldots, \boldsymbol{y}_s(t)\} = \min\{f(\boldsymbol{y}_0(t)), f(\boldsymbol{y}_1(t)), \ldots, f(\boldsymbol{y}_s(t))\}, \tag{5}$$

where s 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, $\hat{\boldsymbol{y}}_i$ 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,

$$\lim_{t \to +\infty} x_{ij}(t) = \frac{c_1 y_{ij} + c_2 \hat{y}_{ij}}{c_1 + c_2}. \tag{6}$$

This theoretically derived behavior provides support for the barebones PSO (BB) developed by Kennedy (2003). The BB replaces Eqs. (3) and (4) with

$$x_{ij}(t+1) = N\left(\frac{y_{ij}(t) + \hat{y}_j(t)}{2}, |y_{ij}(t) - \hat{y}_j(t)|\right). \tag{7}$$

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, $y_{ij}(t) - \hat{y}_j(t)$, 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

$$x_{ij}(t+1) = \begin{cases} N\left(\frac{y_{ij}(t) + \hat{y}_j(t)}{2}, |y_{ij}(t) - \hat{y}_j(t)|\right) & \text{if } U(0,1) > 0.5 \\ y_{ij}(t) & \text{otherwise} \end{cases}. \tag{8}$$

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),

$$\boldsymbol{x}_i(t+1) = \begin{cases} \boldsymbol{x}'_i(t+1) & \text{if } f(\boldsymbol{x}'_i(t+1)) < f(\boldsymbol{x}_i(t)) \\ \boldsymbol{x}_i(t) & \text{otherwise} \end{cases}.$$

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 $\ddot{\boldsymbol{y}}_i(t)$ is calculated as follows:

$$\ddot{y}_{ij}(t) = \begin{cases} \hat{y}_{ij}(t) + \frac{(y_{1j}(t) - y_{2j}(t))}{2} & \text{if } U(0,1) < p_r \text{ or } j = r \\ \hat{y}_{ij}(t) & \text{otherwise} \end{cases},$$

where $j = 1, \ldots, N_d$, $r \sim U(1, \ldots, N_d)$ and $\boldsymbol{y}_1(t)$ and $\boldsymbol{y}_2(t)$ are randomly chosen personal best positions. Then, $\boldsymbol{y}_i(t) = \ddot{\boldsymbol{y}}_i(t)$ only if $f(\ddot{\boldsymbol{y}}_i(t)) < f(\boldsymbol{y}_i(t))$ (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:

$$x_{ij}(t) = \begin{cases} p_{ij}(t) + r_{2j} \times (x_{i_1 j}(t) - x_{i_2 j}(t)) & \text{if } U(0,1) > p_r \\ y_{i_3 j}(t) & \text{otherwise} \end{cases}, \tag{9}$$

where from Eq. (6),

$$p_{ij}(t) = r_{1j}(t) y_{ij}(t) + (1 - r_{1j}(t)) \hat{y}_{ij}(t) \tag{10}$$

with i_1, i_2, $i_3 \sim U(1, \ldots, s)$, $i_1 \neq i_2 \neq i$, r_{1j}, $r_{2j} \sim U(0,1)$ and p_r is the probability of recombination.

Referring to Eq. (10), $\boldsymbol{p}_i(t)$ 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, $\boldsymbol{p}_i(t)$, by adding a difference vector to the attractor. Crossover is done with a randomly selected personal best, $\boldsymbol{y}_{i_3}$, as these personal bests represent a memory of best solutions found by individuals since the start of the search process. Also note that the

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

scale factor is a random variable. Using the position update in Eq. (10), for a proportion of $(1 - p_r)$ of the updates, information from a randomly selected personal best, $\boldsymbol{y}_{i_3}$, is used (facilitating exploitation), while for a proportion of p_r of the updates step sizes are mutations of the attractor point, $\boldsymbol{p}_i$(facilitating exploration). Mutation step sizes are based on the difference vector between randomly selected particles, $\boldsymbol{x}_{i_1}$ and $\boldsymbol{x}_{i_2}$. 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, $\boldsymbol{y}_{i_3}$, 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 $p_r = 0.9$, as suggested in Storn and Price (1997). For the PSO algorithms, $w = 0.72$ and $c_1 = c_2 = 1.49$. These values 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

Given f: $\Re^{N_d} \rightarrow \Re$
find $\boldsymbol{x}^* \in \Re^{N_d}$ such that $f(\boldsymbol{x}^*) \leqslant f(\boldsymbol{x}), \forall \boldsymbol{x} \in \Re^{N_d}$
The following functions were used:
A. *Sphere* function, defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_d} x_i^2,$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.
B. *Schwefel*'s Problem 2.22 (Yao et al., 1999), defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_d} |x_i| + \prod_{i=1}^{N_d} |x_i|,$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-10 \leqslant x_i \leqslant 10$.
C. *Step* function, defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_d} (\lfloor x_i + 0.5 \rfloor)^2,$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.
D. *Rosenbrock* function, defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_d-1} \left(100\left(x_i - x_{i-1}^2\right)^2 + (x_{i-1} - 1)^2\right),$$

where $\boldsymbol{x}^* = (1,1,\ldots,1)$ and $f(\boldsymbol{x}^*) = 0$ for $-30 \leqslant x_i \leqslant 30$.
E. *Rotated hyper-ellipsoid* function, defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_d} \left(\sum_{j=1}^{i} x_j\right)^2,$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-100 \leqslant x_i \leqslant 100$.
F. *Generalized Swefel*'s Problem 2.26 (Yao et al., 1999), defined as

$$f(\boldsymbol{x}) = -\sum_{i=1}^{N_d} \left(x_i \sin\left(\sqrt{|x_i|}\right)\right),$$

where $\boldsymbol{x}^* = (420.9687,\cdots,420.9687)$ and $f(\boldsymbol{x}^*) = -12569.5$ for $-500 \leqslant x_i \leqslant 500$.
G. *Rastrigin* function, defined as

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_d} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right),$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-5.12 \leqslant x_i \leqslant 5.12$.
H. *Ackley*'s function, defined as

$$f(\boldsymbol{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{N_d} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{N_d} \cos(2\pi x_i)\right) + 20 + e,$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-32 \leqslant x_i \leqslant 32$.
I. *Griewank* function, defined as

$$f(\boldsymbol{x}) = \frac{1}{4000}\sum_{i=1}^{N_d} x_i^2 - \prod_{i=1}^{N_d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

where $\boldsymbol{x}^* = \mathbf{0}$ and $f(\boldsymbol{x}^*) = 0$ for $-600 \leqslant x_i \leqslant 600$.
J. *Six-hump Camel-back* function, defined as

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

where $\boldsymbol{x}^* = (-0.08983, 0.7126)$, $(-0.08983, 0.7126)$ and $f(\boldsymbol{x}^*) = -1.0316285$ for $-5 \leqslant x_i \leqslant 5$.

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 $\alpha = 0.05$).

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

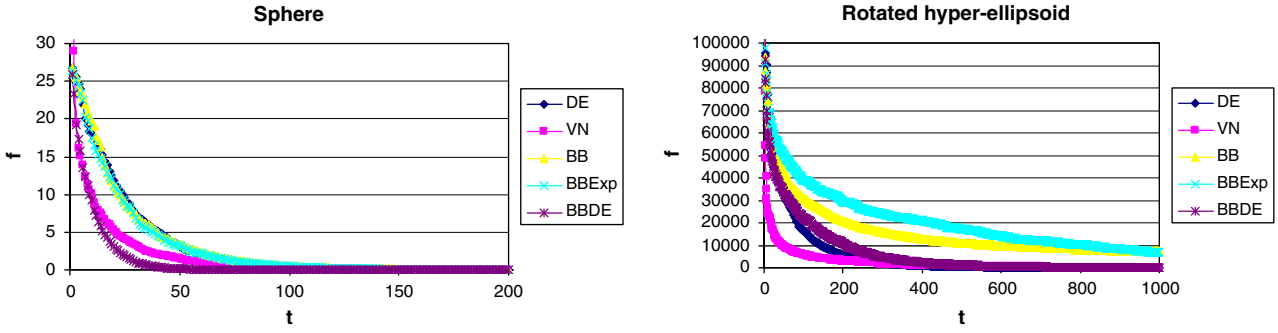
$$\text{diversity} = \frac{1}{s}\sum_{i=1}^{s}\sqrt{\sum_{j=1}^{N_d}\left(x_{ij}(t) - \bar{x}_j(t)\right)^2},$$

where $\bar{x}_j(t)$ is the average of the jth dimension over all individuals, i.e.

$$\bar{x}_j(t) = \frac{1}{s}\sum_{i=1}^{s} x_{ij}(t).$$

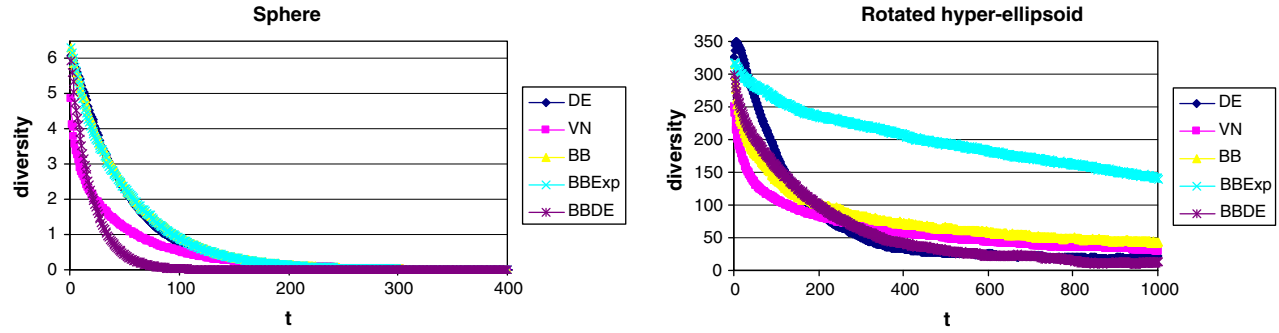
Table 1
Mean and standard deviation (±SD) of the unimodal function optimization results

	Sphere	Schwefel Problem 2.22	Rosenbrock	Step	Rotated hyper-ellipsoid
DE	0 (0)	0 (0)	**32.28566 (19.876539)**	0 (0)	**11.659804 (7.749608)**
VN	0 (0)	0 (0)	45.188135 (28.289829)	0 (0)	91.092175 (41.162374)
BB	0 (0)	4.333333 (8.976342)	15560.221552 (33898.184895)	0 (0)	7228.468118 (5581.910757)
BBExp	0 (0)	0 (0)	77.131243 (55.570480)	0 (0)	6881.687373 (3242.314613)
BBDE	0 (0)	0 (0)	47.857080 (31.835408)	0 (0)	56.467487 (38.975253)



(a) Sphere (zoomed) (b) Rotated hyper-ellipsoid

Fig. 1. Performance Comparison of the different methods when applied to selected unimodal functions.



(a) Sphere (zoomed) (b) Rotated hyper-ellipsoid

Fig. 2. Comparison between the different approaches for selected unimodal functions. The vertical axis represents the diversity and the horizontal axis represents the number of generations.

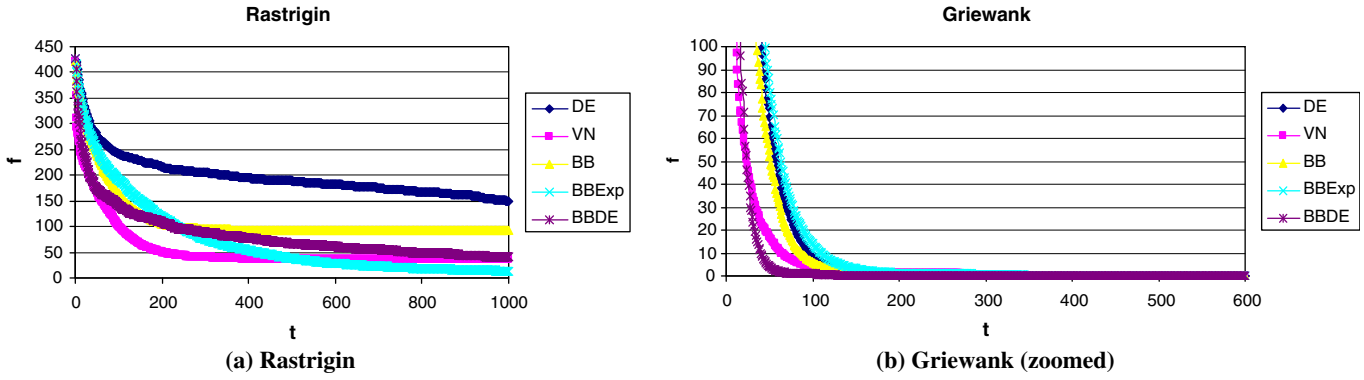
For the Sphere function, the BBDE exhibited the fastest reduction in diversity enabling it to converge faster than the other approaches. For the Rotated hyper-ellipsoid function, both the BBDE and DE exhibited the fastest reduction in diversity. The BBExp was the slowest in diversity reduction which 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. Fig. 3a shows that the VN achieved the fastest reduction in fitness when applied to the Rastrigin function, nevertheless, the BBExp reached a better solution. The DE had problems with the Rastrigin function, with results significantly worse than that of the other methods. Fig. 4a depicts the diversity for the Rastrigin function. The figure shows that 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

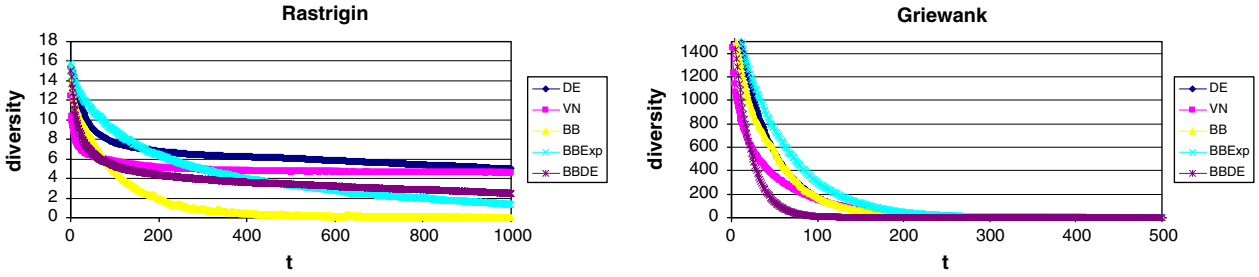
	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
DE	−11052.61419 (1178.89586)	140.59464 (27.27772)	0 (0)	0.005007 (0.00676)	−1.031628 (0)
VN	−8629.400262 (588.165184)	37.470096 (9.435686)	0 (0)	0.004360 (0.006910)	−1.031628 (0)
BB	−9091.017809 (561.296234)	87.572345 (23.790764)	2.264577 (5.959689)	0.011234 (0.012018)	−1.031628 (0)
BBEXP	−10471.819885 (405.232815)	**13.465041 (3.908130)**	0 (0)	**0.000878 (0.002531)**	−1.031628 (0)
BBDE	**−11649.008729 (272.707782)**	37.551246 (15.254959)	0 (0)	**0.000657 (0.002583)**	−1.031628 (0)

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



(a) Rastrigin (b) Griewank (zoomed)

Fig. 3. Performance Comparison of the different methods when applied to selected multimodal functions.



(a) Rastrigin (b) Griewank (zoomed)

Fig. 4. Comparison between the different approaches for selected multimodal functions. The vertical axis represents the diversity and the horizontal axis represents the number of generations.

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

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

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

Table 3
Mean and standard deviation (±SD) of the noisy unimodal function optimization results

	Sphere	Schwefel problem 2.22	Rosenbrock	Step	Rotated hyper-ellipsoid
DE	0.000252 (0.000294)	0.004822 (0.004532)	**37.222951 (24.240263)**	**0.000095 (0.000096)**	**14.700399 (16.041069)**
VN	0.000131 (0.000112)	0.007422 (0.009574)	49.331559 (38.702919)	**0.000070 (0.000109)**	100.543200 (37.625735)
BB	0.003779 (0.005281)	4.525207 (6.629674)	3394.157615 (16396.762517)	0.000330 (0.000418)	7848.947152 (4530.047241)
BBExp	0.000938 (0.001128)	0.135526 (0.292442)	80.799413 (50.289245)	0.000201 (0.000192)	7915.471099 (3410.026066)
BBDE	**0.000057 (0.000052)**	**0.000658 (0.000747)**	71.812221 (58.263287)	**0.000054 (0.000085)**	81.771119 (40.879498)

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

Table 4
Mean and standard deviation (±SD) of the noisy multimodal function optimization results

	Schwefel problem 2.26	Rastrigin	Ackley	Griewank	Camel-back
DE	−10925.046322 (1254.046813)	154.564926 (23.902508)	15.537647 (3.799861)	0.001864 (0.002269)	**−5.153808 (0.204848)**
VN	−8715.290151 (634.156051)	35.742326 (8.464964)	**1.609258 (0.591330)**	0.001466 (0.001263)	−5.458725 (0.218398)
BB	−9129.050539 (406.954831)	94.212615 (28.232227)	16.756511 (0.264990)	0.069609 (0.161420)	**−5.116395 (0.229806)**
BBEXP	−10271.531191 (478.895628)	**18.341674 (4.977051)**	16.700183 v	0.007150 (0.010329)	−5.468067 (0.144862)
BBDE	**−11584.222894 (287.881147)**	37.397204 (18.291796)	**2.417560 (4.705577)**	**0.000548 (0.001134)**	**−5.208338 (0.115516)**

Table 5
Mean and standard deviation (±SD) of the unimodal function optimization results ($N_d = 100$)

	Sphere	Schwefel problem 2.22	Rosenbrock	Step	Rotated hyper-ellipsoid
DE	0.009534 (0.011473)	1.960783 (1.682477)	33984.386973 (59798.047911)	128.100000 (166.231034)	68497.115587 (15790.603308)
VN	0.000578 (0.000137)	1.367757 (1.835326)	1237.939115 (446.834989)	**6.133333 (8.186336)**	**33775.945696 (7098.248666)**
BB	4.764215 (4.733918)	210.756552 (209.083838)	10312425.024713 (24211050.245722)	10700.500000 (8741.615395)	213527.892344 (23035.788775)
BBExp	0.388195 (0.203691)	2333333388.276991 (4301830640.691421)	2024138.254810 (1951057.144980)	711.600000 (414.566712)	316239.880949 (51134.508287)
BBDE	**0 (0)**	**0 (0)**	**312.632070 (195.546311)**	**2.700000 (5.018243)**	212418.632768 (24845.151018)

Table 6
Mean and standard deviation (±SD) of the multimodal function optimization results ($N_d = 100$)

	Schwefel problem 2.26	Rastrigin	Ackley	Griewank
DE	**−19438.479681 (2294.731006)**	716.837974 (100.264563)	3.183406 (0.945352)	1.401776 (0.850903)
VN	−24706.562508 (1445.236154)	**174.575750 (25.457402)**	2.059572 (0.448335)	0.584462 (0.131999)
BB	−26976.484336 (1227.159007)	736.013775 (65.466151)	19.496031 (0.930001)	114.318158 (92.74845)
BBEXP	−29791.567120 (910.138084)	533.652677 (86.672692)	18.327547 (3.991001)	9.870790 (4.233975)
BBDE	−34746.152554 (3750.927593)	616.194754 (38.115845)	**0 (0.000001)**	**0.001640 (0.005296)**

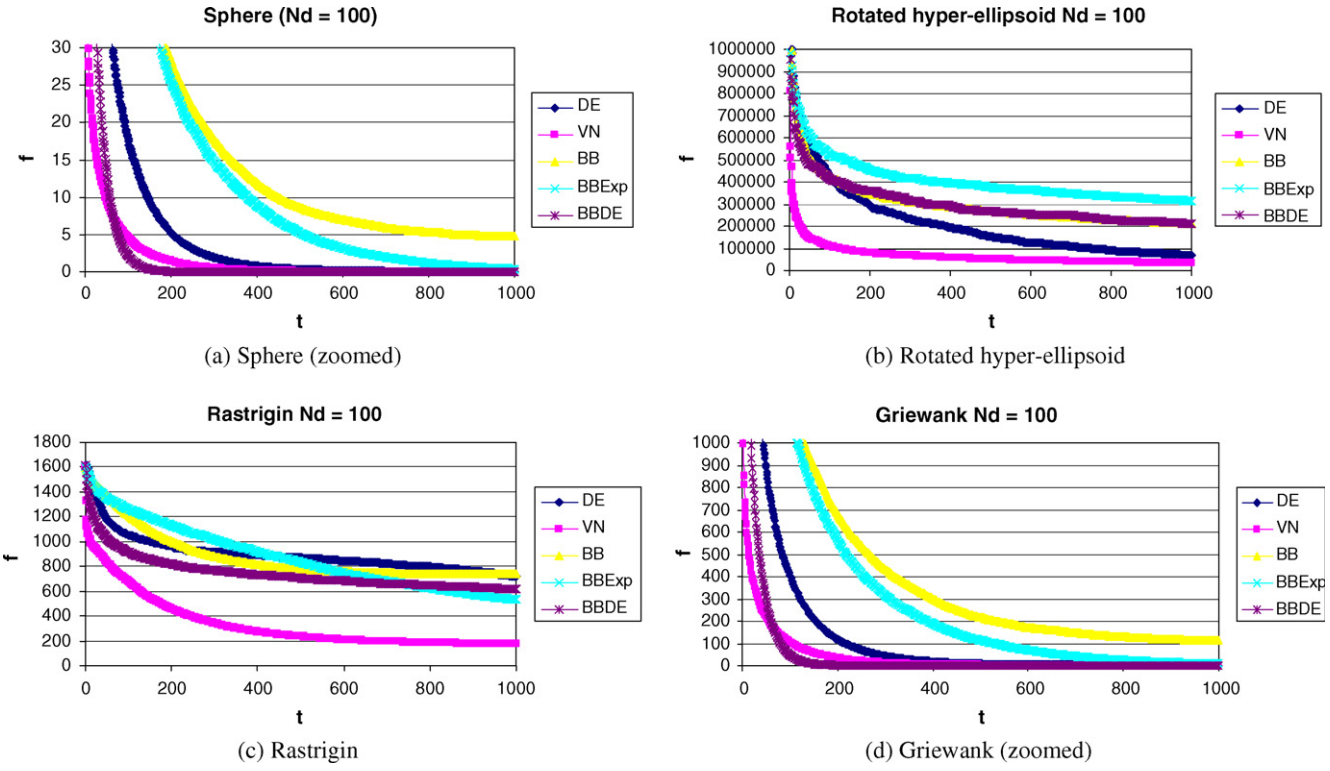


Fig. 5. Performance Comparison of the different methods when applied to representative functions ($N_d = 100$).

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

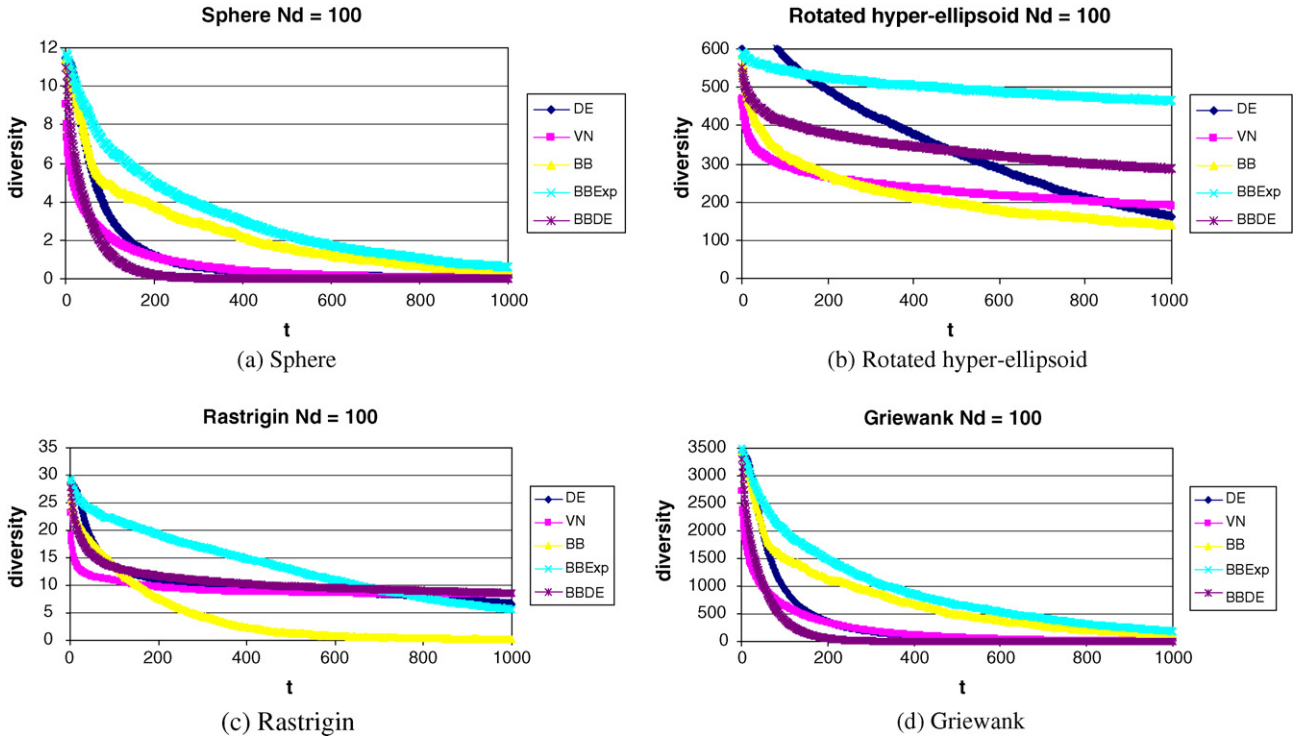


Fig. 6. Performance Comparison of the different methods when applied to representative functions ($N_d = 100$). The vertical axis represents the diversity and the horizontal axis represents the number of generations.

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 covergence.

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 $p_r \sim N(0.5, 0.15)$. 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
Influence of p_r on the performance of BBDE

	$p_r = 0.1$	$p_r = 0.3$	$p_r = 0.5$	$p_r = 0.7$	$p_r = 0.9$
Sphere	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**	0.030934 (0.034928)
Schwefel problem 2.22	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**	0.415136 (0.362744)
Rosenbrock	**32.262004 (24.894417)**	**42.934501 (32.734171)**	47.857080 (31.835408)	82.042890 (59.119772)	17527.428158 (35073.731685)
Step	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**	28.133333 (64.607426)
Rotated hyper-ellipsoid	**20.020990 (16.114680)**	31.737502 (15.151355)	56.467487 (38.975253)	323.816583 (233.859497)	2139.329671 (648.025177)
Schwefel problem 2.26	−10682.8391 (510.011006)	−11291.0108 (325.921626)	−11649.008729 (272.707782)	**−11792.802318 (224.658980)**	−11688.001932 (239.308673)
Rastrigin	155.779164 (16.391473)	106.833299 (12.946721)	37.551246 (15.254959)	**6.473762 (2.990750)**	15.245124 (4.110377)
Ackley	**0 (0)**	**0 (0)**	**0 (0)**	**0 (0)**	2.651095 (1.404765)
Griewank	0.004763 (0.006335)	**0.002545 (0.006012)**	**0.000657 (0.002583)**	**0.000985 (0.003208)**	1.682496 (1.273646)
Camel-back	**−1.031628 (0.000000)**	**−1.031628 (0.000000)**	**−1.031628 (0.000000)**	**−1.031628 (0.000000)**	−1.031380 (0.000564)

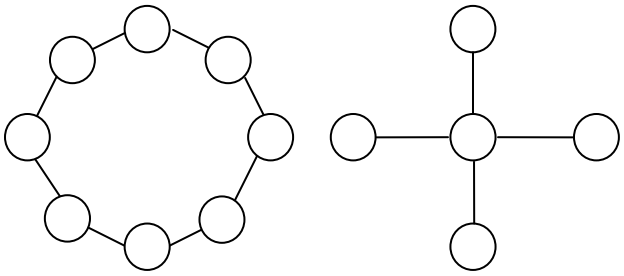
Table 8
Comparison of different variants of BBDE

	BBDE	PFBBDE	DBBDE
Sphere	0 (0)	0 (0)	0 (0)
Schwefel problem 2.22	0 (0)	0 (0)	0 (0)
Rosenbrock	47.857080 (31.835408)	55.552183 (59.870164)	76.371490 (90.076614)
Step	0 (0)	0 (0)	0.066667 (0.253708)
Rotated hyper-ellipsoid	**56.467487 (38.975253)**	**72.755559 (51.147255)**	185.222391 (112.377535)
Schwefel problem 2.26	**−11649.00873 (272.707782)**	**−11601.481689 (290.840666)**	−11098.220557 (367.719656)
Rastrigin	37.551246 (15.254959)	34.132272 (17.814442)	**10.785169 (5.924399)**
Ackley	0 (0)	0 (0)	0 (0)
Griewank	**0.000657 (0.002583)**	**0.001479 (0.003461)**	0.004517 (0.006755)
Camel-back	−1.031628 (0.000000)	−1.031628 (0.000000)	−1.031628 (0.000000)

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.



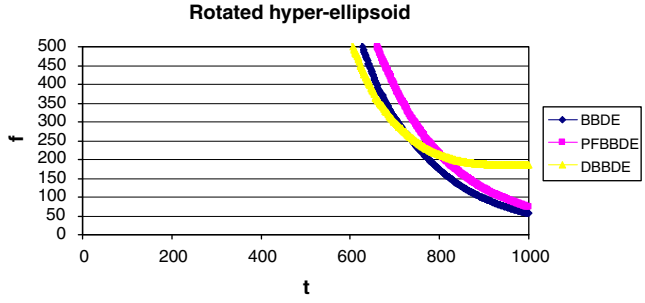
(a) Ring Topology (b) Von Neumann Topology

Fig. 8. A diagrammatic representation of neighborhood topologies.

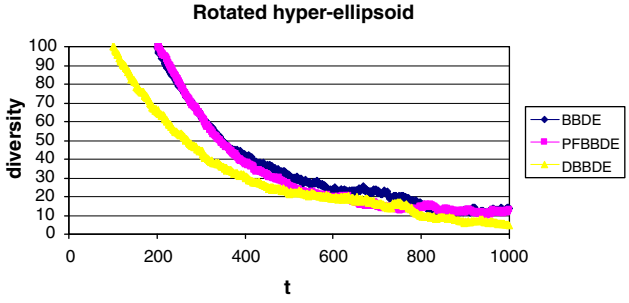
The two proposed variants are as follows:

A. VNBBDE: For the VNBBDE, $\hat{\boldsymbol{y}}_i(t)$ in Eq. (10) is the best position found so far in the ring neighborhood of the ith individual (i.e. the two adjacent left and right neighbors of i).

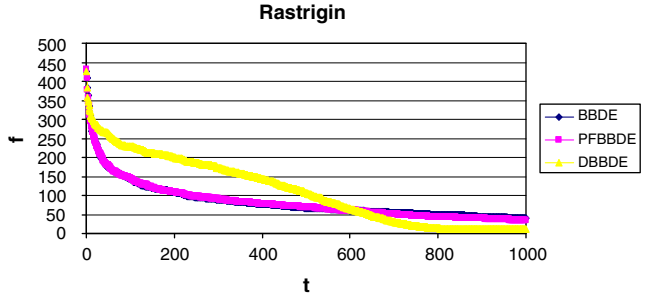
B. RBBDE: For the RBBDE, $\hat{\boldsymbol{y}}_i(t)$ in Eq. (10) is the best position found so far in the Von Neumann neighborhood of the ith individual (i.e. the four immediately adjacent left, right, up and down neighbors of i).



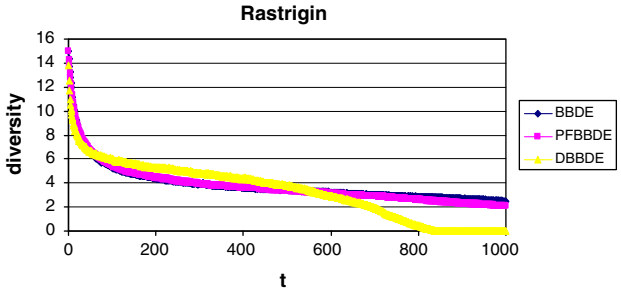
(a) Rotated hyper-ellipsoid (zoomed)



(b) Rotated hyper-ellipsoid (diversity - zoomed)



(c) Rastrigin



(d) Rastrigin (diversity)

Fig. 7. Performance Comparison of the BBDE, PFBBDE and DBBDE when applied to representative functions.

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

Table 9
Comparison of different neighborhood topologies

	BBDE	VNBBDE	RBBDE
Sphere	0 (0)	0 (0)	0 (0)
Schwefel problem 2.22	0 (0)	0 (0)	0 (0)
Rosenbrock	47.857080 (31.835408)	**33.835765 (20.302641)**	53.911486 (29.917795)
Step	0 (0)	0 (0)	0 (0)
Rotated hyper-ellipsoid	**56.467487 (38.975253)**	3737.803843 (1290.674863)	11799.276804 (3129.255101)
Schwefel problem 2.26	−11649.00873 (272.707782)	−12131.922676 (184.745498)	**−12307.231539 (274.314345)**
Rastrigin	**37.551246 (15.254959)**	77.757479 (8.314326)	85.331301 (10.549080)
Ackley	0 (0)	0 (0)	0 (0)
Griewank	0.000657 (0.002583)	0 (0)	0 (0)
Camel-back	−1.031628 (0)	−1.031628 (0)	−1.031628 (0)

Table 9 summarizes the results obtained by applying the different methods to the benchmark problems. The results show that the BBDE outperformed the other strategies when applied to the Rotated hyper-ellipsoid and Rastrigin functions. On the other hand, for the Rosenbrock function, the VNBBDE was the best performer. For the Schwefel Problem 2.26, the RBBDE outperformed the other approaches. The three approaches performed comparably when applied to the remaining functions.

7. Unsupervised image classification

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_b 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).
- $\boldsymbol{z}_p$ denotes the N_b components of pixel p.
- $\boldsymbol{m}_j$ 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

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

where $\boldsymbol{C}_k$ is the kth cluster, and n_k is the number of pixels in $\boldsymbol{C}_k$.

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 $\boldsymbol{x}_i$ is constructed as $\boldsymbol{x}_i = (\boldsymbol{m}_{i,1}, \ldots, \boldsymbol{m}_{i,k}, \ldots, \boldsymbol{m}_{i,K})$ where $\boldsymbol{m}_{i,k}$ refers to the kth cluster centroid vector of the ith individual. Therefore, a population represents a number of candidate data clusterings. The quality of each individual is measured using

$$f(\boldsymbol{x}_i, \boldsymbol{Z}_i) = \frac{\bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i) + J_{e,i}}{d_{\min}(\boldsymbol{Z}_i, \boldsymbol{x}_i)}, \tag{11}$$

where $\boldsymbol{Z}_i$ is a matrix representing the assignment of patterns to the clusters of individual i. Each element $z_{i,k,p}$ indicates if pattern $\boldsymbol{z}_p$ belongs to cluster $\boldsymbol{C}_k$ of individual i (in which case $z_{i,k,p}$). Also,

$$\bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i) = \max_{k=1,\ldots,K}\left\{\sum_{\forall z_{i,k,p}=1} d(\boldsymbol{z}_p, \boldsymbol{m}_{i,k}) \Big/ n_{i,k}\right\}$$

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

$$d_{\min}(\boldsymbol{x}_i) = \min_{\forall k,kk,k\neq kk}\{d(\boldsymbol{m}_{i,k}, \boldsymbol{m}_{i,kk})\}$$

is the minimum Euclidean distance between any pair of clusters. In the above, $n_{i,k}$ is the number of patterns that belong to cluster $\boldsymbol{C}_{i,k}$ 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 $\bar{d}_{\max}(\boldsymbol{Z}_i, \boldsymbol{x}_i)$ and the quantization error, as quantified by J_e, and to maximize the inter-cluster distance between any pair of clusters, as quantified by, $d_{\min}(\boldsymbol{x}_i)$.

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

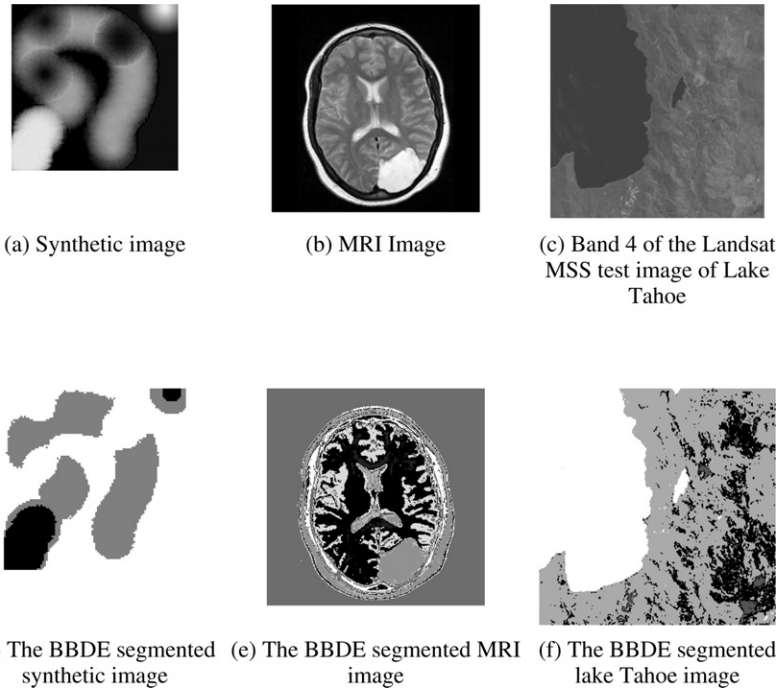
The BBDE clustering algorithm is summarized in Fig. 9.

1. Initialize each individual to contain K randomly selected cluster centroids
2. For $t = 1$ to $t_{\max}$
(a) For each individual i
i. For each pattern z_p
- Calculate $d(z_p, \boldsymbol{m}_{i,k})$ for all clusters $\boldsymbol{C}_{i,k}$
- Assign z_p to $\boldsymbol{C}_{i,k}$ where

$$d(z_p, \boldsymbol{m}_{i,k}) = \min_{\forall k=1,\ldots,K}\{d(z_p, \boldsymbol{m}_{i,k})\}$$

ii. Calculate the fitness, $f(\boldsymbol{x}_i, \boldsymbol{Z}_i)$
(b) Find the global best solution $\hat{\boldsymbol{y}}(t)$
(c) Apply the BBDE as described in Section 5 and update the cluster centroids using Eq. 9.

Fig. 9. The BBDE clustering algorithm.



(a) Synthetic image (b) MRI Image (c) Band 4 of the Landsat MSS test image of Lake Tahoe

(d) The BBDE segmented synthetic image (e) The BBDE segmented MRI image (f) The BBDE segmented lake Tahoe image

Fig. 10. The original images and the segmented images resulting from the BBDE clustering algorithms.

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, BB, 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 iterations were used (i.e. all algorithms have performed 5000 function evaluations). For K-means, FCM and PSO, the parameters were set as in Omran (2005). For DE, the parameters were set as in

Table 10
Comparison between K-means, FCM, PSO, DE, BB, BBEXP and BBDE

Image		J_e	$\bar{d}_{max}$	d_{min}
Synthetic	K-means	20.21225 (0.937836)	28.04049 (2.7779388)	78.4975 (7.0628718)
	FCM	20.731920 (0.650023)	28.559214 (2.221067)	82.434116 (4.404686)
	PSO	17.284 (0.09)	22.457 (0.414)	90.06 (0.712)
	DE	17.349039 (0.024415)	22.208008 (0.045002)	89.674503 (0.071472)
	BB	17.324602 (0.026192)	22.229277 (0.036073)	89.706956 (0.029875)
	BBExp	17.337303 (0.024775)	22.211746 (0.035281)	89.691711 (0.029087)
	BBDE	17.329013 (0.023745)	22.224411 (0.034509)	89.703907 (0.030338)
MRI	K-means	7.3703 (0.042809)	13.214369 (0.761599)	9.93435 (7.308529)
	FCM	7.205987 (0.166418)	10.851742 (0.960273)	19.517755 (2.014138)
	PSO	7.839 (0.238)	9.197 (0.56)	29.45 (1.481)
	DE	8.489362 (0.518571)	11.193335 (0.620451)	26.561583 (1.339439)
	BB	7.722666 (0.091995)	8.940551 (0.563716)	29.969806 (0.969355)
	BBExp	7.813303 (0.165253)	9.283251 (0.489863)	29.309536 (0.833855)
	BBDE	8.149742 (0.386591)	10.311062 (1.399248)	29.37867 (0.977154)
Tahoe	K-means	3.280730 (0.095188)	5.234911 (0.312988)	9.402616 (2.823284)
	FCM	3.164670 (0.000004)	4.999294 (0.000009)	10.970607 (0.000015)
	PSO	3.882 (0.274)	5.036 (0.368)	16.410 (1.231)
	DE	4.190698 (0.302445)	5.216843 (0.321865)	16.906206 (1.089620)
	BB	3.980572 (0.192029)	5.131814 (0.266169)	17.037257 (0.861458)
	BBExp	4.051751 (0.120513)	5.210299 (0.18998)	17.31936 (0.579856)
	BBDE	4.007243 (0.117684)	5.130046 (0.155744)	17.102415 (0.514053)

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

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 $\bar{d}_{\max}$, while performing comparably with respect to J_e. 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

Abbass, H., 2002. The self-adaptive Pareto differential evolution algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 831–836.

Abbass, H., Sarker, R., Newton, C., 2001. PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, pp. 971–978.

Bezdek, J., 1980. A convergence theorem for the fuzzy ISODATA clustering algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 2, 1–8.

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

Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micromachine and Human Science, pp. 39–43.

Engelbrecht, A., 2005. Fundamentals of Computational Swarm Intelligence. Wiley & Sons.

Forgy, E., 1965. Cluster analysis of multivariate data: Efficiency versus interpretability of classification. Biometrics 21, 768–769.

Hendtlass, T., 2001. A combined swarm differential evolution algorithm for optimization problems. In: Proceedings of the Fourteenth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Lecture Notes in Computer Science, vol. 2070. Springer-Verlag, pp. 11–18.

Kannan, S., Slochanal, S., Subbaraj, P., Padhy, N., 2004. Application of particle swarm optimization technique and its variants to generation expansion planning. Electric Power Systems Research 70 (3), 203–210.

Kennedy, J., 1999. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 3, pp. 1931–1938.

Kennedy, J., 2003. Bare bones particle swarms. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 80–87.

Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proceedings of the IEEE International Joint Conference on Neural Networks. IEEE Press, pp. 1942–1948.

Kennedy, J., Mendes, R., 2002. Population structure and particle performance. In: Proceedings of the IEEE Congress on Evolutionary Computation. IEEE Press, pp. 1671–1676.

Kennedy, J., Eberhart, R.C., Shi, Y., 2001. Swarm Intelligence. Morgan Kaufmann.

Krink, T., Filipic, B., Fogel, G., 2004. Noisy optimization problems – A particular challenge for differential evolution. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 332–339.

Liu, J., Lampinen, J., 2002. A fuzzy adaptive differential revolution algorithm. In: Proceedings of the IEEE International Region 10 Conference, pp. 606–611.

Omran, M., 2005. Particle swarm optimization methods for pattern recognition and image processing. PhD Thesis, University of Pretoria.

Omran, M., Engelbrecht, A., Salman, A., 2005a. Differential evolution methods for unsupervised image classification. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, pp. 966–973.

Omran, M., Engelbrecht, A., Salman, A., 2005b. Particle swarm optimization method for image clustering. International Journal of Pattern Recognition and Artificial Intelligence. 19 (3), 297–322.

Omran, M., Salman, A., Engelbrecht, A., 2005c. Self-adaptive differential evolution. Lecture Notes in Artificial Intelligence 3801, 192–199.

Omran, M., Engelbrecht, A., Salman, A., 2007. Differential evolution based on particle swarm optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp. 112–119.

Peer, E., van den Bergh, F., Engelbrecht, A., 2003. Using neighborhoods with the guaranteed convergence PSO. In: Proceedings of the IEEE Swarm Intelligence Symposium. IEEE Press, pp. 235–242.

Price, K., Storn, R., Lampinen, J., 2005. Differential Evolution: A Practical Approach to Global Optimization. Springer.

Puzicha, J., Hofmann, T., Buhmann, J.M., 2000. Histogram clustering for unsupervised image segmentation. In: IEEE Proceedings of the Computer Vision and Pattern Recognition, vol. 2, pp. 602–608.

Qin, A., Suganthan, P., 2005. Self-adaptive differential evolution algorithm for numerical optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1785–1791.

Rönkkönen, J., Kukkonen, S., Price, K.V., 2005. Real-parameter optimization with differential evolution. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 506–513.

Shi, Y., Eberhart, R., 1998. A modified particle swarm optimizer. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 69–73.

Storn, R., 1996. On the usage of differential evolution for function optimization. In: Proceedings of the Biennieal Conference of the North American Fuzzy Information Processing Society, pp. 519–523.

Storn, R., Price, K., 1995. Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute.

Storn, R., Price, K., 1997. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 11 (4), 359–431.

Talbi, H., Batouche, M., 2004. Hybrid particle swarm with differential evolution for multimodal image registration. In: Proceedings of the IEEE international conference on industrial technology, vol. 3, pp. 1567–1573.

van den Bergh, F., 2002. An Analysis of Particle Swarm Optimizers Department of Computer Science. University of Pretoria, Pretoria, South Africa.

van den Bergh, F., Engelbrecht, A., 2006. A study of particle swarm optimization particle trajectories. Information Sciences 176 (8), 937–971.

Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3 (2), 82–102.

Zhang, W.-J., Xie, X.-F., 2003. DEPSO: Hybrid particle swarm with differential evolution operator. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, pp. 3816–3821.

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