Engineering Optimization, 2013
Vol. 45, No. 4, 459–481, http://dx.doi.org/10.1080/0305215X.2012.678494

A simulated weed colony system with subregional differential evolution for multimodal optimization

Subhrajit Roy[a], Sk. Minhazul Islam[a], Swagatam Das[b]*, Saurav Ghosh[a] and Athanasios V. Vasilakos[c]

[a]*Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India;* [b]*Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India;* [c]*Department of Computer and Telecommunication Engineering, University of Western Macedonia, Greece*

(*Received 9 September 2011; final version received 22 February 2012*)

This article proposes a two-stage hybrid multimodal optimizer based on invasive weed optimization (IWO) and differential evolution (DE) algorithms for locating and preserving multiple optima of a real-parameter functional landscape in a single run. Both IWO and DE have been modified from their original forms to meet the demands of the multimodal problems used in this work. A *p*-best crossover operation is introduced in the subregional DEs to improve their exploitative behaviour. The performance of the proposed algorithm is compared with a number of state-of-the-art multimodal optimization algorithms over a benchmark suite comprising 21 basic multimodal problems and seven composite multimodal problems. Experimental results suggest that the proposed technique is able to provide better and more consistent performance over the existing well-known multimodal algorithms for the majority of test problems without incurring any serious computational burden.

Keywords: weed colony systems; multimodal optimization; niching; differential evolution; particle swarm optimization

1. Introduction

A multimodal optimization problem amounts to finding multiple optimal solutions and not just one single optimum, as is done in a typical optimization study. If a point-by-point classical optimization approach is used for this task, the approach must be applied several times, each time hoping to find a different optimal solution. Owing to their population-based approach, evolutionary algorithms (EAs) (Eiben and Smith 2003) provide a natural advantage over classical optimization techniques when applied to multimodal optimization problems. They maintain a population of possible solutions, which are processed at every generation, and if multiple solutions can be preserved over all these generations, then at termination of the algorithm they can offer multiple good solutions instead of only the best solution. Note that this is against the natural tendency of EAs, which will always converge to the best solution or a suboptimal solution (in a rugged, not so well-posed function). *Niching* (Mahfoud 1995, Singh and Deb 2006) is a generic term which refers to the techniques of finding and preserving multiple stable *niches*, or favourable parts of

*Corresponding author. Email: swagatam.das@ieee.org

the solution space possibly around multiple solutions, so as to prevent convergence to a single solution. Research on solving multimodal problems with EAs dates back to the landmark work of Goldberg and Richardson in 1987, where they neatly showed how a niche-preserving technique can be introduced in a standard genetic algorithm (GA) and multiple optimal solutions can be obtained. Since that study, many researchers have suggested methodologies for introducing niche-preserving techniques so that, for each optimum solution, a niche is formed in the population of an EA. Currently, the most popular niching techniques used in conjunction with the evolutionary computation community include crowding (Thomsen 2004), fitness sharing (Goldberg and Richardson 1987), restricted tournament selection (Harik 1995) and speciation (Petrowski 1996). Most existing niching methods, however, have difficulties that need to be overcome before they can be applied successfully to real-world multimodal problems. Some identified issues include difficulties in prespecifying some niching parameters, difficulties in maintaining discovered solutions in a run, extra computational overheads and poor scalability when dimensionality is high. The current research on evolutionary multimodal optimization aims to circumvent these problems by devising more and more efficient optimizers.

In this article a simple yet very powerful hybrid multimodal optimizer is proposed that synergistically combines the features of two global optimizers: invasive weed optimization (IWO) (Mehrabian and Lucas 2006) and differential evolution (DE) (Storn and Price 1995, Price *et al.* 2005, Das and Suganthan 2011) for multimodal optimization. IWO is a novel ecologically inspired algorithm that mimics the process of weed colonization and distribution. DE has emerged as a very competitive optimizer for continuous search spaces, exhibiting remarkable performances in several competitions held under the IEEE Congress on Evolutionary Computation (CEC) (*e.g.* Suganthan *et al.* 2005, Tang *et al.* 2007).

Multimodal optimization requires a sufficient amount of exploration of the population agents in hyperspace so that all the local and global attractors can be successfully and quickly detected. However, an efficient multimodal optimization algorithm should exhibit not only a good exploration tendency but also good exploitative power, especially during the later stages of the search, because ultimately it has to ensure accurately a distributed convergence to different optima in the landscape. To preserve such characteristics, a two-stage optimization process is presented that involves an invasive weed optimizer followed by the subregional modified differential evolution (IWO-δ-DE) algorithm. The reason for employing the IWO in a multimodal optimization process is its high explorative power over the bound-constrained search space (Chakraborty *et al.* 2009). Each weed initialized in the search space produces seeds around it in a controlled hyperspace, thus creating a virtual subpopulation of seeds. Hence, after a sufficient number of generations the colony is basically spread in subregions surrounding promising local and global optima. This is the stage where the DE search is invoked in each of these subregions. After IWO coarsely detects the basins of attraction, the whole population is grouped together by means of a subregional detection factor (δ) that can be estimated from the search range and dimensionality of the problem at hand. Then, a modified and more exploitative DE takes over with an efficient subregional search technique in each of the groups to facilitate convergence towards the actual optimal points.

The newly proposed IWO-δ-DE has the capability to converge more accurately to the local and global peaks, starting from a uniform initialization in the search space, in comparison to a number of state-of-the-art algorithms for multimodal optimization. This fact has been demonstrated through comparison with nine other multimodal optimizers over 20 standard benchmarks. The present algorithm involves only one single extra parameter: the subregion detection factor δ. An empirical formula for calculating this parameter depending on the search range and dimensionality specified for the problem at hand is presented. Finally, the efficiency of the proposed algorithm on a practical multimodal optimization problem that involves determination of the structure of a dielectric composite is presented.

2. Evolutionary multimodal optimization: related works

When a single-objective optimization problem has more than one optimal solution, it can be considered as a multimodal optimization problem. The objective of locating different optima in a single run makes it more complicated than single global optimization. *Niching methods*, devised for extending EAs to multimodal optimization, address this issue by maintaining the diversity of certain properties within the population, and in this way they allow parallel convergence into multiple good solutions in multimodal domains. Some of the prominent niching techniques are briefly described below.

- *Crowding and restricted tournament selection*: In 1975, De Jong introduced the crowding method, which tries to maintain population diversity by allowing competition for limited resources among similar individuals in the population. Hence, effectively the competition takes place within each niche. In general, the similarity is measured using Euclidean distance between individuals. The algorithm compares an offspring with some randomly sampled individuals from the current population. The most similar individual will be replaced if the offspring is a superior solution. A parameter *CF*, the crowding factor, is used to control the size of the sample. *CF* is generally set to 2 or 3.
- *Sharing methods*: One of the most well-known methods for creating subpopulations of like individuals is fitness sharing (Goldberg and Richardson 1987). It is based on the concept that a point in a search space has limited resources that need to be shared by any individuals that occupy similar search space behaviours or genetic representations. Sharing in EAs is implemented by scaling the fitness of an individual based on the number of 'similar' individuals present in the population.
- *Clearing*: Unlike fitness sharing, clearing (Petrowski 1996) determines the dominant individuals of the subpopulations and removes the remaining population members from the mating pool. The algorithm first sorts the population in descending order according to the fitness values. Then, it picks one individual at a time from the top and removes all the individuals with worse fitness than the selected one within the specified clearing radius. This step will be repeated until all the individuals in the population are either selected or removed.
- *Speciation*: The concept of speciation (Li *et al.* 2002) depends on a radius parameter r_s, which measures Euclidean distance from the centre of a species to its boundary. The centre of a species is called the species seed. Each of the species is built around the dominating species' seed. All individuals falling within the radius from the species seed are identified as the same species. In this way, the whole population is classified into different groups according to their similarity.

Apart from the above, several other niching methods have also been developed over the years, including derating (Beasley 1993), parallelization (Bessaou *et al.* 2000) and clustering (Yin and Germay 1993). A complete survey on the niching techniques adopted for real-parameter multimodal optimization problems can be found in Das *et al.* (2011).

The concept of niching was primarily incorporated into GAs for tackling multimodal optimization problems. Some of the very prominent GA variants with niching can be found in works such as Mahfoud (1995), Deb and Goldberg (1989), Yao *et al.* (2010) and Deb (1989). Extension of the evolution strategies (ESs) to solve multimodal problems has been reported in works such as Im *et al.* (2004) and Shir and Bäck (2005). Shir *et al.* (2010) applied a new concept of adaptive individual niche radius in conjunction with the covariance matrix adaptation evolution strategy (CMA-ES). The idea is that each individual updates a niche radius every generation along with its adaptive strategy parameters. Two new modifications are implemented. The first technique exploits the cumulative step-size adaptation (CSA) in the CMA-ES mechanism, and couples the individual niche radius to it, and the second one introduces the Mahalanobis distance metric into the niching mechanism.

Researchers also attempted to modify another very popular swarm intelligence algorithm, particle swarm optimization (PSO) (Kennedy and Eberhart 1995), for achieving efficient solutions to multimodal problems. Parsopoulos and Vrahitis (2001, 2004) proposed a method where a potentially good solution is isolated once it is found, and then the fitness landscape is 'stretched' to keep other particles away from this area of the search space (Parsopoulos and Vrahatis 2001). Brits and van den Bergh (2002) extended Parsopoulos and Vrahitis's model by proposing NichePSO, where multiple subswarms are produced from a main swarm population to locate multiple optimal solutions in the search space. Subswarms can merge together, or absorb particles from the main swarm. Recently, Li (2010) used a simple *lbest* PSO employing the typical ring topology to ensure stable niching behaviours. Communication topology, *i.e.* ring topology, has been utilized to control the speed of convergence for the PSO population. The ring topology is advantageous for locating multiple optima primarily because ideally each individual is expected to search thoroughly in its local neighbourhood before propagating the information throughout the population.

Thomsen (2004) integrated the fitness sharing concept with DE to form the sharing DE. Thomsen (2004) also proposed extending DE with a crowding scheme (crowding differential evolution or CDE) to allow it to tackle multimodal optimization problems. CDE, with a crowding factor equal to the population size, has outperformed the sharing DE on standard benchmarks. In CDE, when an offspring is generated, its fitness is only compared with the most similar (in terms of low Euclidean distance) individual in the current population. The offspring will replace this individual if it has a better fitness value. Zaharie (2004) proposed a multi-resolution multi-population differential evolution (MMDE) that divides the population into c equally sized subpopulations. The search space is initially divided into c non-overlapping subdomains, for which the subpopulations are initialized. While the DE algorithm is iteration independent for each subpopulation, the subpopulations are not restricted to the subdomains used in the initialization. Some other prominent approaches involving DE-based niching algorithms were reported by Hendershot (2004) and Rönkkönen and Lampinen (2007). In 2012, Qu *et al.* integrated the concept of Euclidean distance-based neighbourhood in DE for multimodal optimization. The algorithm proposed in this article is conceptually motivated by their approach, although it has its own distinctive features. To the authors' knowledge, IWO has not been applied to solve the multimodal optimization problems to date.

3. A brief account of the ancestors: original IWO and DE

In this section a brief account of the original versions of the IWO and DE algorithms is presented.

3.1. *The invasive weed optimization algorithm*

IWO is a population-based metaheuristic algorithm that mimics the colonizing behaviour of weeds. There are four steps in the algorithm:

1. *Initialization*: A certain number (say m) of weeds is randomly spread over the entire D-dimensional search space. This initial population of each generation will be termed as $\boldsymbol{X} = \{\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_m\}$.
2. *Reproduction*: Each member of the population $\boldsymbol{X}$ is allowed to produce seeds within a specified region centred at its own position. The number of seeds produced by $\vec{X}_i$, $i \in \{1, 2, \ldots, m\}$ depends on its relative fitness in the population with respect to the best and worst fitness.

The number of seeds produced by any weed varies linearly from *min_seed* to *max_seed*, with *min_seed* for the worst member and *max_seed* for the best member in the population.

3. *Spatial distribution*: The generated seeds are randomly distributed over the d-dimensional search space by normally distributed random numbers with zero mean and variance σ^2. However, the standard deviation σ is made to decrease over the generations so that the algorithm gradually moves from exploration to exploitation with increasing generations. If $\sigma_{\max}$ and $\sigma_{\min}$ are the maximum and minimum standard deviation, then the standard deviation in particular generation (or iteration) is given by:

$$\sigma_t = \sigma_{\min} + \left(\frac{\sigma_{\min} - t}{\sigma_{\max}}\right)^{n\_m\_i} \cdot (\sigma_{\max} - \sigma_{\min}), \tag{1}$$

where n_m_i represents the nonlinear modulation index, t is the current iteration number, and $t_{\max}$ is the maximum number of iterations allowed. This step ensures that the probability of dropping a seed in a distant area decreases nonlinearly with iterations, which results in grouping fitter plants and elimination of inappropriate plants.

4. *Competitive exclusion*: Initially, the plants in a colony will reproduce rapidly and all the produced weeds will be included in the colony, until the number of plants reaches a maximum value of *pop_max*. However, it is expected that by this time the fitter plants will have reproduced more than undesirable plants. From then on, only the fittest plants, among the existing ones and the reproduced ones, are taken into the colony, and Steps 1–4 are repeated until the maximum number of iterations [or function evaluations (FEs)] has been reached. So, in every generation the population size must be less than or equal to *pop_max*. This method is known as competitive exclusion and is the selection procedure of IWO.

3.2. *The differential evolution algorithm*

DE starts with a population of *NP* D-dimensional parameter vectors representing the candidate solutions. The subsequent generations in DE are denoted by $t = 0, 1, \ldots, t_{\max}$. For each decision variable of the problem, there may be a user-specified range within which value of the variable should lie for more accurate search results at less computational cost. The initial population (at iteration $t = 0$) should cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum bounds:

$$\vec{X}_{\min} = \{x_{1,\min}, x_{2,\min}, \ldots, x_{D,\min}\} \quad \text{and} \quad \vec{X}_{\max} = \{x_{1,\max}, x_{2,\max}, \ldots, x_{D,\max}\}.$$

(a) *Mutation*: After initialization, DE creates a *donor* vector corresponding to each population member or *target* vectorin the current generation through mutation. The three most frequently used mutation strategies implemented in the public-domain DE codes available online at http://www.icsi.berkeley.edu/∼storn/code.html are:

$$\text{'DE/rand/1'}: \vec{Y}_{i,t} = \vec{X}_{\alpha_1} + F \cdot (\vec{X}_{\alpha_2} - \vec{X}_{\alpha_3}). \tag{2a}$$

$$\text{'DE/best/1'}: \vec{Y}_{i,t} = \vec{X}_{best,t} + F \cdot (\vec{X}_{\alpha_1^i,t} - \vec{X}_{\alpha_2^i,t}). \tag{2b}$$

$$\text{'DE/target-to-best/1'}: \vec{Y}_{i,t} = \vec{X}_{i,t} + F \cdot (\vec{X}_{best,t} - \vec{X}_{i,t}) + F \cdot (\vec{X}_{\alpha_1^i,t} - \vec{X}_{\alpha_2^i,t}) \tag{2c}$$

The indices $\alpha_1^i, \alpha_2^i, \alpha_3^i, \alpha_4^i$ and α_5^i are mutually exclusive integers randomly chosen from the range $[1, NP]$, and all are different from the base index i. These indices are randomly generated once for each donor vector. The scaling factor F is a positive control parameter for scaling the difference vectors. $\vec{X}_{best,t}$ is the best individual vector with the best fitness (*i.e.* lowest objective function value for minimization problem) in the population at iteration t.

(b) *Crossover*: The DE family of algorithms can use two kinds of crossover method: *exponential* (or two-point modulo) and *binomial* (or uniform) (Das and Suganthan 2010). In this article the binomial crossover is used and it is performed on each of the D variables whenever a randomly generated number between 0 and 1 is less than or equal to the Cr value. In this case, the number of parameters inherited from the donor has a (nearly) binomial distribution. The scheme may be outlined as:

$$z_{j,i,t} = \begin{cases} y_{j,i,t}, & \text{if } (rand_{i,j}[0,1] \leq Cr \text{ or } j = j_{rand}) \\ x_{j,i,t}, & \text{otherwise} \end{cases} \tag{3}$$

where, as before, $rand_{i,j}[0, 1)$ is a uniformly distributed random number, which is called anew for each jth component of the ith parameter vector. $j_{rand} \in [1, 2, \ldots, D]$ is a randomly chosen index, which ensures that $\vec{U}_{i,t}$ receives at least one component from $\vec{V}_{i,t}$.

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

$$\begin{aligned} \vec{X}_{i,t+1} &= \vec{Z}_{i,t}, \text{ if } f(\vec{Z}_{i,t}) \leq f(\vec{X}_{i,t}) \\ &= \vec{X}_{i,t}, \text{ if } f(\vec{Z}_{i,t}) > f(\vec{X}_{i,t}) \end{aligned} \tag{4}$$

where $f(\vec{X})$ is the function to be minimized. So if the new trial vector yields a lower value of the objective function, it replaces the corresponding target vector in the next generation; otherwise the target is retained in the population.

4. The proposed IWO-δ-DE algorithm

Multimodal optimization necessitates efficient exploration of the search space at earlier stages of search and at the same time multiple localized convergences during the later part. With the modifications of the seeding and selection processes of the classical IWO and a new exploitative crossover scheme and parameter adaptation schemes for DE, the proposed algorithm achieves balanced explorative and exploitative behaviours on the multimodal search space. In this section the innovative aspects of the two stage optimizer are highlighted and then a pseudo-code of the whole algorithm is presented.

4.1. *Seeding and selection in IWO*

In IWO, the generated seeds are randomly distributed over the search space by normally distributed random numbers with mean equal to zero mean but a varying standard deviation. Since multimodal optimization initially requires the detection of multiple regions surrounding the global and local peaks, IWO appears to be very suitable for this as the plants reproduce the seeds by a very small perturbation around them. This means that seeds will be randomly distributed within a small neighbourhood surrounding the parent plant. This is done by keeping the standard deviation for production of the seeds for a particular plant very low. Consequently, generated seeds have a very low probability of converging towards a single optimal point. This fact has been demonstrated by showing the evolution (with selection disabled) of a two-dimensional toy weed-population,

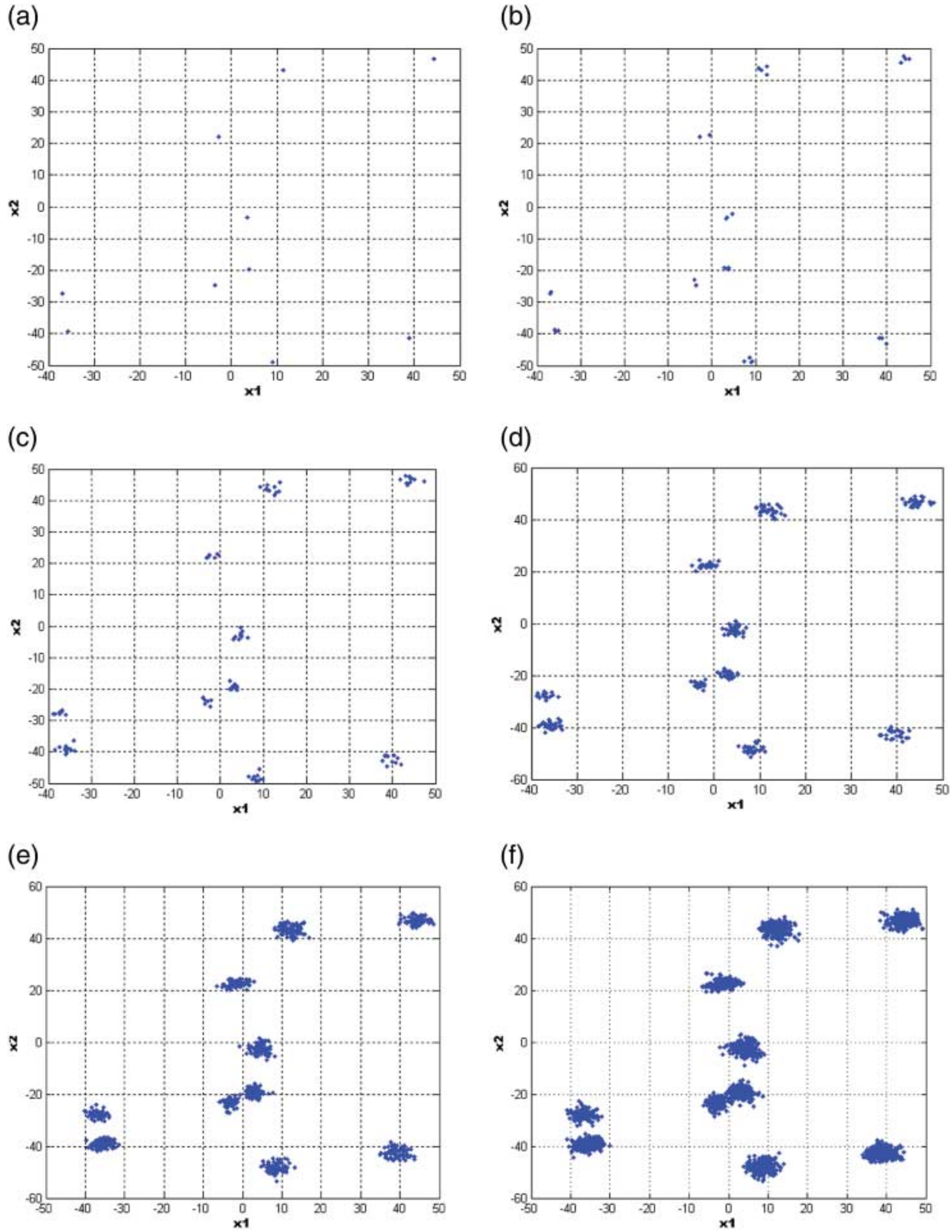


Figure 1. Evolution of a two-dimensional invasive weed optimization (IWO) population with $s = 1$: (a) at 0th generation; (b) at 1st generation; (c) at 3rd generation; (d) at 4th generation; (e) at 5th generation; (f) at 6th generation.

starting from an initial population size of 10, the maximum and minimum number of seeds generated by a weed remaining 3 and 0, respectively, and the standard deviation of each plant being fixed at $\sigma = 1.0$ in Figure 1.

The formation of distributed neighbourhoods around each plant can be clearly seen in Figure 1(f). Size, overlapping and distances among such neighbourhoods can be controlled by varying σ, maximum and minimum number of seeds, *etc*.

As far as the selection process of IWO is concerned, usually it is observed that after passing some iterations, the number of plants in a colony will reach its maximum limit and there is a mechanism for eliminating the plants with poor fitness in the next generation. The traditional competitive selection mechanism works in the following way: when the maximum number of weeds in a colony is reached, the seeds which had been produced by the plants are ranked together with their parents (as a colony of weeds). Next, weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. A major shortcoming of this kind of selection is that the resulting colony may not be able to maintain all the optima until the end of the run because a suspected optimum may be lost in the search space at the end if plants heading towards another optimum have a better fitness value. Subsequently, this kind of selection, although useful when a single global optimum is to be found, is not advantageous for multimodal optimization. Thus, the selection scheme is slightly modified in IWO-δ-DE. At first, for each plant in the colony, the best seed (based on fitness) produced by it has been found out. Then, the nearest plant (based on Euclidean distance) from the best seed has been sorted out from the colony. The next colony would incorporate only the fitter member of a particular best seed and the corresponding nearest plant in the current colony.

4.2. *Grouping*

The two-stage optimization procedure starts with IWO, which runs until 80% of the total budget of FEs. After the IWO phase is over, the final subregions are formed by grouping the solutions with a threshold of mutual Euclidean distance among the weeds and the threshold is called the subregional detection factor (δ). Owing to the relatively poor exploitative behaviour of IWO, the evolved plants cannot fully converge to the actual optimal points. After the IWO phase is over, for the remaining 20% of FEs, solutions from the whole colony are partitioned into disjoint groups and each of these groups is subjected to an exploitative variant of DE. The proposed IWO-δ-DE algorithm introduces a new parameter, δ. A formula for δ is provided in Section 5.3.

The population grouping proceeds in the following way. Starting with the index-wise first weed, the Euclidean distance of this individual is calculated with the rest of the weeds. The weeds having Euclidean distance less than or equal to δ are included in the same group as that of the first member. Now, for the formation of the second group a switch occurs to the second weed. If this weed is already included in the first group, then it is discarded and a switch to the index-wise next weed occurs because the objective is to form non-overlapping groups, *i.e.* a particular weed cannot be present in more than one group. Now, if this second particle is not present in the first group, the same procedure is repeated, *i.e.* the distance of the second particle, then the rest of the particles, avoiding those which are members of the first group, are calculated and the second group is formed with those particles whose distance is $\leq \delta$ along with the second particle. In this way, the grouping procedure will continue until all the population members are considered, forming the maximum possible number of groups.

4.3. *The subregional differential evolution algorithm*

A modified DE with p-best crossover scheme is applied separately to the subregions formed by the grouping technique. DE helps in a detailed search inside the subregions and finally a distributed convergence to multiple optimal solutions.

The mutation in DE is chosen as DE/rand/1/bin of (2a) and the difference vector is formed by choosing population members from the same subregion. Since the number of FEs is limited, to enhance the convergence speed and to promote exploitation, a new p-best crossover scheme is introduced in the subregional DE. At first the group is sorted based on fitness. For each donor, a vector is randomly selected from the p top-ranked individuals in the group population and then binomial crossover is performed between the donor vector and the randomly selected p-best

Table 1. Pseudo-code of the proposed algorithm.

Begin
Initialize the population uniformly within the search bounds
While $t \leq t_{max\_}IWO$
$\sigma_t = \left(\frac{t_{\max\_IWO-t}}{t_{\max}}\right)^{POW} \cdot (\sigma_{\max} - \sigma_{\min}) + \sigma_{\min}$
For $i =$ *to no_plants*
For $se = 1$ *to no_seeds*
For $j = 1$:*dim*
$y_{se,j} = x_{i,j} + randn \cdot \sigma_t$
If $y_{se,j} \geq UB$
$y_{se,j} = UB$
End*If*
If $y_{se,j} \leq LB$
$y_{se,j} = LB$
End*If*
End*for*
End*for*
End*for*
Starting of selection
For $i = 1$ *to no_plants*
$Q =$ *best of the seeds*
$R =$ *nearest of the Q in the plants*
If $f(Q) > f(R)$
R is replaced by Q
End*If*
End*for*
End of selection
$t = t + 1$
End*while*
Grouping of the plants by subregional detection factor calculated as per Equation (7).
For *group* $= 1$ *to no_groups*
Calculate FES and iterations allotted for the group based on the number of elements in that group as in eqn. (6).
Set $F = 0.2$ and $Cr = 0.9$.
Identify the farthest two vectors $VEC1$ and $VEC2$
For $j = 1$ *to* D
$UB_j = MAX(VEC1_j, VEC2_j)$
$LB_j = MIN(VEC1_j, VEC2_j)$
End*for*
For $t = 1$ *to group_iter*
For $i = 1$ to *group_elements*
Form the ith donor vector using Equation (2b) as
For $j = 1$ *to* D
If $y_{i,j} > UB_j$
$y_{i,j} = UB_j$
elseif $y_{i,j} < LB_j$
$y_{i,j} = LB_j$
END*if*
END*for*
End*for*
Calculate p stated in Equation (5)
Sort the group population and randomly select a vector from first p best vectors of the population.
For $i = 1$ to *group_elements*
Generate $j_{rand} = ceil(rand(1, D))$
For $j = 1$ to D
If $j = j_{rand}$ or $\text{rand}(0, 1) \leq \text{Cr}$
$z_{j,i,t} = v_{j,i,t}$
Else
$z_{j,i,t} = x_{j,p-best,t}$

(*Continued*)

Table 1. Continued

End*if*
END*for*
If $f(\vec{Z}_{i,t}) \leq f(\vec{X}_{i,t})$
$\vec{X}_{i,t+1} = \vec{Z}_{i,t}$
Else
$\vec{X}_{i,t+1} = \vec{X}_{i,t}$
End*if*
END*for*
END*for*
Endfor

vector to generate the corresponding trial vector or the offspring. So, the information contained in the better vectors of the current group population is incorporated into the offspring by means of component injection. Thus, the offspring achieve a much higher probability of advancing to the following generation. The estimation of p has been determined by the following equation:

$$p = ceil\left[\left(\frac{group\_size}{2}\right) \cdot \left(1 - \frac{iter - 1}{group\_iter}\right)\right], \tag{5}$$

where *group_size* represents the number of elements in a particular group and *group_iter* is the number of iterations allotted to each DE population that searches a particular subregion, and is calculated as:

$$group\_iter = \frac{FEs\ alloted\ to\ DE}{Total\ population\ size} \tag{6}$$

Note that the different DE populations evolve for the same number of generations, but for different numbers of FEs. The reduction routine of p aims to preserve the components from better individuals and intensifies the search towards the best individuals as time progresses. A complete pseudo-code of the algorithm is provided in Table 1.

5. Experiments and results

5.1. *Numerical benchmarks*

To evaluate the performance of the IWO-δ-DE algorithm, some challenging test functions have been used of various characteristics, such as irregular landscape, symmetrical or equal distribution of optima, unevenly spaced optima, multiple global optima in the presence of multiple local optima, and dimension scalability with function rotation. The collection of the test functions is mainly composed of 21 multimodal general functions (Li 2010) and nine composition functions (Qu and Suganthan 2010). A brief description of the functions is given in Table 2b. Keane's bump problem (f_{21}) is an engineering design problem where the constraints are handled by penalizing the unfeasible individuals. Note that for the composite functions $CF1$–$CF9$, taken from Qu and Suganthan (2010), the dimensionality is 10 and each variable is bounded in $[-5, 5]$. The number of global peaks is six for functions $CF2$–$CF8$ and eight for each of the functions $CF1$ and $CF9$.

5.2. *Comparison of algorithms*

The performance of IWO-δ-DE is compared with the following standard multimodal evolutionary algorithms:

- Crowding DE (CDE) (Thomsen 2004)

Table 2a. Benchmark functions.

Name	Dim	Test function	Range	No. of global peaks
f_1: Two-peak trap	1	$f_1(x) \begin{cases} \frac{160}{15}(15-x) & \text{for } 0 \le x < 15 \\ \frac{200}{5}(x-15) & \text{for } 15 \le x \le 20 \end{cases}$	$0 \le x \le 20$	1
f_2: Central two-peak trap	1	$f_2(x) = \begin{cases} \frac{160}{10}x & \text{for } 0 \le x < 10 \\ \frac{160}{5}(15-x) & \text{for } 10 \le x < 15 \\ \frac{200}{5}(x-15) & \text{for } 15 \le x \le 20 \end{cases}$	$0 \le x \le 20$	1
f_3: Five-uneven-peak trap	1	$f_3(x) = \begin{cases} 80(2.5-x) & \text{for } 0 \le x < 2.5 \\ 64(x-2.5) & \text{for } 2.5 \le x < 5 \\ 64(7.5-x) & \text{for } 5 \le x < 7.5 \\ 28(x-7.5) & \text{for } 7.5 \le x < 12.5 \\ 28(17.5-x) & \text{for } 12.5 \le x < 17.5 \\ 32(x-17.5) & \text{for } 17.5 \le x < 22.5 \\ 32(27.5-x) & \text{for } 22.5 \le x < 27.5 \\ 80(x-27.5) & \text{for } 27.5 \le x \le 30 \end{cases}$	$0 \le x \le 30$	2
f_4: Equal maxima	1	$f_4(x) = \sin^6(5\pi x)$	$0 \le x \le 1$	5
f_5: Decreasing maxima	1	$f_5(x) = \exp\left[-2\log(2) \cdot \left(\frac{x-0.1}{0.8}\right)^2\right] \cdot \sin^6(5\pi x)$	$0 \le x \le 1$	1
f_6: Uneven maxima	1	$f_6(x) = \sin^6(5\pi(x^{\frac{3}{4}} - 0.05))$	$0 \le x \le 1$	5
f_7: Uneven decreasing maxima	1	$f_5(x) = \exp\left[-2\log(2) \cdot \left(\frac{x-0.08}{0.854}\right)^2\right] \cdot \sin^6(5\pi(x^{\frac{3}{4}} - 0.05))$	$0 \le x \le 1$	1
f_8: Himmelblau's function	2	$f_8(\vec{x}) = 200 - x(x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2$	$-4 \le x_1, x_2 \le 4$	4
f_9: Six-hump camel back	2	$f_9(\vec{X}) = -4\left[\left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2\right]$	$-1.9 \le x_1 \le 1.9$ $-1.1 \le x_2 \le 1.1$	2

- Speciation-based DE (SDE) (Li 2005)
- Fitness–Euclidean distance ratio PSO (FER-PSO) (Li 2007)
- Speciation-based PSO (SPSO) (Li 2004)
- r2pso (Li 2004): an PSO with a ring topology; each member interacts with only its immediate member to its right
- r3pso (Li 2010): an PSO with a ring topology; each member interacts with its immediate member on its left and right
- r2psolhc (Li 2010): the same as r2pso, but with no overlapping neighbourhoods, hence acting as multiple local hill climbers, more suitable for finding global as well as local optima
- r3psolhc (Li 2010): the same as r3pso, but with no overlapping neighbourhoods; basically, multiple PSOs search in parallel, like local hill climbers. This variant is more appropriate if the goal of optimization is to find global as well as local optima
- CMA-ES with self-adaptive niche radius (S-CMA) (Shir 2010).

Table 2b. Benchmark functions.

Name	Dim	Test function	Range	No. of global peaks
f_{10}: Shekel's foxholes	2	$f_{10}(\vec{X}) = 500 - \dfrac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1+i+(x_1-a(i))^6+(x_2-b(i))^6}}$ where $a(i) = 16(i \bmod 5 - 2)$, $b(i) = 16(\lfloor (i/5) \rfloor - 2)$	$-65.536 \le x_1, x_2 \le 65.535$	1
f_{11}: 1D inverted Vincent function	1	$f(\vec{X}) \frac{1}{D} \sum_{i=1}^{D} \sin(10 \cdot \log(x_i))$, where n is the dimensionality of the problem	$0.25 \le x_i \le 10$	6
f_{12}: 2D inverted Vincent function	2			36
f_{13}: 3D inverted Vincent function	3			216
f_{14}: Waves	2	$f(\vec{X}) = (0.3x_1)^3 - (x_2^2 - 4.5x_2^2)x_1x_2 - 4.7\cos(3x_1 - x_2^2(2 + x_1)\sin(2.5\pi x_1))$	$-0.9 \le x_1 \le 1.2$ $-1.2 \le x_2 \le 1.2$	1
f_{15}: Shifted Rastrigin	10	$f(\vec{X}) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10) + f\_bias$	$-5 \le x_i \le 5$	1
f_{16}: Branin RCOS	2	$f(\vec{X}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right) + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	$-5 \le x_1 \le 10$ $0 \le x_2 \le 15$	3
f_{17}: Michalewicz function	2	$f(\vec{X}) = \sin(x_1)\sin^{20}\left(\frac{x_1^2}{\pi}\right) + \sin(x_2)\sin^{20}\left(\frac{2x_2^2}{\pi}\right)$	$0 \le x_i \le \pi$	1
f_{18}: Ursem F1 in Ursem (1999)	2	$f(\vec{X}) = \sin(2x_1 - 0.5\pi) + 3\cos(x_2) + 0.5x_1$	$-2.5 \le x_1 \le 3$ $-2 \le x_2 \le 2$	1
f_{19}: Ursem F3 in Ursem (1999)	2	$f(\vec{X}) = \sin(2.2\pi x_1 + 0.5\pi) \cdot \frac{2 - \lvert x_2 \rvert}{2} \cdot \frac{3 - \lvert x_1 \rvert}{2} + \sin(0.5\pi x_2^2 + 0.5\pi) \cdot \frac{2 - \lvert x_2 \rvert}{2} \cdot \frac{2 - \lvert x_1 \rvert}{2}$	$-2.5 \le x_1 \le 3$ $-2 \le x_2 \le 2$	1
f_{20}: Ursem F4 in Ursem (1999)	2	$f(\vec{X}) = 3\sin(0.5\pi x_1 + 0.5\pi) \cdot \frac{2 - \sqrt{x_1^2 + x_2^2}}{4}$	$-2 \le x_i \le 2$	1
f_{21}: Keane's bump problem	20	$f(x) = \dfrac{\left\lvert \sum_{i=1}^{D} \cos^4(x_i) - 2 \cdot \prod_{i=1}^{D} \cos^2(x_i) \right\rvert}{\sqrt{\sum_{i=1}^{D} i \cdot x_i^2}}$	$0 \le x_i \le 10$ $\prod_{i=1}^{D} x_i > 0.75, \sum_{i=1}^{D} x_i < \frac{15 \cdot D}{2}$	1

5.3. *Parametric set-up*

In the actual IWO algorithm each plant produces seeds depending on the highest and lowest fitness of itself as well as the colony. But in the IWO part of IWO-δ-DE this concept is modified in the sense that each plant in the colony is allowed to produce a fixed q number of spatially distributed seeds. The value of q is chosen as 5 for all cases as it gives good results on all the cases tested here. The remaining parameters used in IWO for all the test functions are:

- *pow*: the value of *pow* is set to be 9.
- $sd_{\max}$: is set to be $\frac{\sqrt{\sum_{i=1}^{D}(U_i - L_i)^2}}{10 \cdot \sqrt{D}}$.

- $sd_{\min}$ is set to be $\frac{\sqrt{\sum_{i=1}^{D}(U_i - L_i)^2}}{200 \cdot \sqrt{D}}$.

For the grouping part of the algorithm, empirical studies indicate that parameter δ can be set optimally according to the formula:

$$\delta = \frac{\frac{\sqrt{\sum_{i=1}^{D}(U_i - L_i)^2}}{\sqrt{2500D}} + \frac{\sqrt{\sum_{i=1}^{D}(U_i - L_i)^2}}{\sqrt{100D}}}{2} \tag{7}$$

where U_i and L_i denote the upper and lower bound for the ith dimension respectively. The value of δ should be neither too high nor too low. If the value of δ is too low, then a large number of groups will be formed and the number of search agents in each group will be reduced, as the population size is constant. So it will be difficult for IWO-δ-DE to locate the optima because DE will require a sufficient number of individuals in the basin of attraction of any optima for complete convergence. Again, if δ is too high then overlapping of groups may take place, which may lead to wastage of FEs and missing of some optima because each group is intended to search for a single optimum. For high δ it may occur that each group is heading towards more than one optimum, which is not desirable. Considering these factors, Formula (7) is chosen, from which an optimum value of δ can be obtained. It should be noted that the formula is not dependent on the number of global optima that exist within the fitness landscape. For the contestant algorithms, the best suited parametric set-up is employed, chosen using guidelines from the respective literature.

5.4. *Population size and maximum number of evaluations*

Different population sizes and maximum number of FEs were allotted for different functions depending upon the complications of the functions tested. Table 3 shows the population size and the maximum number of FEs allotted for different benchmark functions throughout the experiments undertaken in this work.

5.5. *Performance measures*

To compare the performance of different multimodal algorithms, a level of accuracy (typically $0 < \varepsilon < 1$) indicating how close the computed solutions are to the known global peaks, needs to be specified. If the difference between a computed solution and a known global optimum is below ε, then the peak is considered to have been found. The performance of all multimodal algorithms is measured in terms of the following four criteria:

- *Success rate*: The percentage of runs in which all global peaks are successfully found.
- *Average number of optima found* (Gan and Warwick 2000).
- *Peak accuracy*: The peak accuracy measure (Thomsen 2004) is calculated as follows: for each optimum to be found, the closest individual x in the population is taken and the absolute difference in fitness values is computed after the maximum number of function evaluations has

Table 3. Population size and function evaluations (FEs) for test functions.

Function number	Population size	Maximum number of FEs
f_1–f_{10}, f_{14}, f_{16}–f_{20}	50	10,000
f_{11}	100	20,000
f_{12}	500	200,000
f_{13}	1000	400,000
f_{14}, f_{15}, $CF1$–$CF9$	600	300,000

elapsed. Then, all these differences are summed and divided by the number of global optima to be found. The peak accuracy calculation is shown below:

$$peak\ accuracy = \sum_{i=1}^{\#peaks} \frac{|f(peak_i) - f(\vec{X})|}{\#peaks} \tag{8}$$

Here, the measure is used for global peaks only.

- *Distance accuracy*: In some multimodal functions some of the global/local peaks are located very close to each other or may have identical peak heights. In such cases the previous metric may produce better results even if the whole population is situated in the basin of attraction of the same peak. As a precaution, the distance accuracy that refers to the dissimilarity in the genotypic space between each peak and its closest individual is also calculated in the same manner as in (8), with the only change that the difference between fitness values is substituted by the Euclidean distance between the two individuals. The distance accuracy metric successfully overcomes the limitations of peak accuracy measurement, which may cause errors owing to coincidental close evaluations.

In order to determine the statistical significance of the advantage of IWO-δ-DE over other algorithms, Wilcoxon's rank sum test (Wilcoxon 1945, Derrac et al. 2011) is applied on the average number of peaks found at the 5% significance level. If the *p*-values obtained with the rank sum test are less than 0.05 (5% significance level), this is strong evidence against the null hypothesis, indicating that the better final results achieved by the best algorithm in each case are statistically significant and have not occurred by chance.

All performances are calculated and averaged over 50 independent runs. All the algorithms are implemented in MATLAB 7.5 and executed using a Pentium core 2 duo machine with 2 GB RAM and 2.23 GHz speed.

5.6. *Results*

This section presents the experimental results. All the algorithms are run until the maximum number of FEs is exhausted. To save space, only the results of the average number of peaks found and the distance accuracy obtained with all the compared algorithms are provided in Tables 4 and 5. The second and third columns of Table 4 indicate the level of accuracy and niche radius (for SDE and SPSO) used in the experiments. Values of niche radius values are chosen as recommended by the authors. It should be noted that the level of accuracy (ε) is taken to be small so that it becomes challenging for the algorithms to locate the peaks.

The statistical test results on the average number of peaks found are shown in Table 6. The numerical values $1, 0, -1$ represent other methods being statistically inferior to, equal to or superior to IWO-δ-DE. By combining the results of Tables 4, 5 and 6, it can be concluded that IWO-δ-DE consistently produces superior results to the other algorithms in a statistically meaningful way.

For an optimization problem with multiple global as well as local optima, there is a need to locate all global optima and some local optima that are also considered as satisfactory solutions. Therefore, a good niching algorithm should have the property of locating both global and local optima. In order to test the ability of locating local optima, Table 7 provides the number of global and local peaks as well as the results for five selected benchmark functions. It should be noted that for f_{10}, a population size of 500 is used with the maximum number of FEs equal to 100,000 for all the algorithms. For the remaining functions the population size, maximum number of FEs and values of the detection factor (δ) used are the same as mentioned in Sections 5.3 and 5.4.

To visualize the performance of IWO-δ-DE, the distribution of solutions over the functional landscapes for the one-dimensional function f_6 and two-dimensional functions f_8 and f_{10} is shown.

Table 4. Average number of peaks found for the test functions (best entries are marked in bold).

Function	ε	r	IWO-δ-DE	S-CMA	CDE	SDE	FERPSO	SPSO	r2pso	r3pso	r2psolhc	r3psolhc
f_1	0.05	0.5	**1**	**1**	**1**	**1**	0.72	0.48	0.76	0.84	0.56	0.6
f_2	0.05	0.5	**1**	**1**	**1**	**1**	**1**	0.44	0.88	0.96	0.44	0.56
f_3	0.05	0.5	**2**	1.92	**2**	1.96	0.8	0.24	0.48	0.6	0.48	0.6
f_4	0.000001	0.01	**5**	4.92	3.84	4.72	4.84	4.88	4.92	4.88	**5**	4.92
f_5	0.000001	0.01	**1**	1	0.72	**1**	**1**	**1**	**1**	**1**	**1**	**1**
f_6	0.000001	0.01	**5**	4.88	3.96	4.6	**5**	4.92	4.88	4.72	4.92	4.88
f_7	0.000001	0.01	**1**	**1**	0.6	**1**	**1**	**1**	**1**	**1**	**1**	**1**
f_8	0.0005	0.5	**3.84**	3.72	0.32	3.72	3.68	0.84	2.92	2.76	3	3.12
f_9	0.000001	0.5	1.64	1.6	0.04	**2**	1.96	0.08	1.44	1.56	1.56	1.48
f_{10}	0.00001	0.5	**1**	0.88	0.52	0.32	**1**	0.56	0.88	0.76	0.72	0.6
f_{11}	0.0001	0.2	**6**	5.56	5.56	4.88	5.36	5.6	5.52	5.16	5.36	5.28
f_{12}	0.001	0.2	32.6	24.6	**33.8**	22.8	23.6	25.72	21.8	22.2	22.52	23.12
f_{13}	0.001	0.2	104.6	75.6	**152**	50.6	68.6	70.12	40.6	45.4	42.2	43.32
f_{14}	0.1	0.5	**0.88**	0.60	0.80	0.64	0.62	0.52	0.48	0.50	0.46	0.54
f_{15}	0.1	1	**0.20**	0.04	0.12	0.16	0.08	0	0	0	0	0
f_{16}	0.1	0.5	**2.88**	2.56	2.76	2.72	2.64	2.48	2.52	2.44	2.36	2.40
f_{17}	0.1	0.5	**0.92**	0.70	0.84	0.82	0.76	0.64	0.68	0.52	0.54	0.58
f_{18}	0.1	0.5	**0.90**	0.72	0.86	0.78	0.80	0.68	0.62	0.58	0.52	0.56
f_{19}	0.1	0.5	**0.84**	0.66	0.82	0.74	0.72	0.62	0.56	0.54	0.48	0.50
f_{20}	0.1	0.5	**0.90**	0.68	0.84	0.78	0.74	0.64	0.58	0.52	0.42	0.48
f_{21}	0.1	1	**0.64**	0.62	0.60	0.52	0.50	0.04	0	0	0	0
$CF1$	0.5	1	**2**	1.08	0	1.8	1.08	0	0	0	0	0
$CF2$	0.5	1	**2**	1.52	1.2	1.2	**2**	0	0	0	0	0
$CF3$	0.5	1	**3.6**	0.52	0.72	1.52	2.52	0	0	0	0	0
$CF4$	0.5	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$CF5$	0.5	1	**2**	1.2	1.12	1.32	**2**	0	0	0	0	0
$CF6$	0.5	1	**1.56**	1.12	0	1.4	1.2	0	0	0	0	0
$CF7$	0.5	1	**1.8**	1.2	0	**1.8**	1.52	0	0	0	0	0
$CF8$	0.5	1	**1.3**	0.72	0	1.1	1.1	0	0	0	0	0
$CF9$	0.5	1	**1.8**	1.08	0	1.3	0	0	0	0	0	0

Plots for the rest of the functions appear more or less similar and are not given, to save space. Function f_6 has five global peaks that are unevenly spaced; this feature makes f_6 a challenging function. Figure 2 shows a simulation run of IWO-δ-DE on f_6 using only 50 plants.

Function f_8 has four global peaks. Since two out of the four global peaks are very close to each other, it is difficult to find all four peaks. Figure 3 shows a simulation run of the IWO-δ-DE on f_8 using only 50 plants.

The results shown above clearly indicate that IWO-δ-DE has outperformed all the compared state-of-the-art niching algorithms in terms of all the performance measures considered. The superiority of the algorithm lies in the fact that it can satisfactorily locate local peaks other than the global peaks, which should be the target of a multimodal optimization algorithm. It has also performed efficiently when the dimensionality is high ($D = 10$). For these high-dimensional composite functions, the traditional PSO variants have completely failed to locate any of the global optima. Tables 4 and 5 indicate that the proposed algorithm exhibits a good exploitative behaviour that facilitates convergence towards different global and local peaks.

5.7. *Sensitivity of the performance of IWO-δ-DE to the variation in sharing of total budget of FEs for IWO and DE*

For multimodal optimization, the algorithm must have a sufficient amount of exploration to successfully detect the optimal zones, and it must also have a good factor of exploitation to accurately detect the optima. That is why, in this article, IWO has been used for the exploration aspect, shown mathematically in Section 5, whereas DE has been used for the proper exploitation

Table 5. Distance accuracy of test functions.

Function	r	IWO-δ-DE	S-CMA	CDE	SDE	FERPSO	SPSO	r2pso	r3pso	r2psolhc	r3psolhc
f_1	0.5	**1.47e−18**	5.47e−08	8.67e−06	3.46e−07	4.65e−01	6.87e−01	8.76e−01	5.93e−01	3.25e−01	4.98e−01
f_2	0.5	**4.73e−17**	4.73e−07	3.45e−06	8.23e−07	3.45e−03	1.43e−01	8.75e−02	9.86e−02	1.98e−01	9.34e−02
f_3	0.5	**8.27e−16**	3.54e−04	7.46e−04	9.78e−05	9.67e−02	9.98e−01	1.83e−01	2.49e−01	9.87e−01	2.04e−01
f_4	0.01	**6.18e−18**	3.64e−06	5.46e−03	8.54e−05	2.46e−06	2.34e−06	8.67e−07	3.45e−06	6.45e−08	6.57e−07
f_5	0.01	**4.19e−18**	4.87e−07	7.37e−05	3.65e−08	7.36e−08	9.68e−08	9.67e−07	2.43e−06	5.67e−06	9.67e−07
f_6	0.01	**8.22e−08**	5.65e−06	4.75e−04	6.57e−06	1.23e−07	5.34e−07	5.76e−05	6.54e−05	8.79e−05	6.87e−05
f_7	0.01	**9.82e−07**	1.82e−06	1.94e−04	6.45e−06	5.86e−05	7.86e−05	1.23e−04	3.54e−04	2.64e−04	7.89e−04
f_8	0.5	**1.65e−02**	5.64e−02	8.67e−01	7.03e−02	2.54e−02	6.78e−01	9.87e−02	1.54e−01	9.12e−02	8.99e−02
f_9	0.5	2.83e−03	3.98e−04	6.45e−03	**3.92e−06**	5.76e−05	4.32e−02	9.78e−03	1.23e−02	8.79e−03	1.14e−02
f_{10}	0.5	**3.54e−06**	9.89e−03	4.65e−02	3.78e−01	9.78e−05	8.67e−03	3.24e−03	1.24e−03	6.57e−03	2.79e−03
f_{11}	0.2	**7.71e−06**	9.78e−02	3.54e−03	2.45e−03	2.43e−03	7.03e−03	8.79e−02	9.78e−02	7.56e−02	8.04e−02
f_{12}	0.2	1.72e−01	5.67e−02	**5.46e−03**	1.03e−02	7.98e−02	9.87e−02	1.09e−01	1.87e−01	2.76e−01	3.08e−01
f_{13}	0.2	5.18e−01	8.67e−01	**7.68e−02**	9.99e−01	8.56e−01	1.05	1.43	2.01	1.62	1.22
f_{14}	0.5	**4.73e−02**	1.34e−01	7.93e−02	8.75e−02	9.73e−02	1.43e−01	3.26e−01	3.87e−01	4.53e−01	3.59e−01
f_{15}	1	**1.45**	3.32	1.71	1.68	2.57	5.06	6.08	6.78	7.65	6.97
f_{16}	0.5	**7.18e−02**	5.45e−01	1.53e−01	2.46e−01	4.56e−01	8.67e−01	7.53e−01	8.45e−01	1.04	9.89e−01
f_{17}	0.5	**5.62e−03**	8.74e−02	1.04e−02	5.43e−02	7.84e−02	6.45e−02	9.65e−02	1.12e−01	1.08e−01	1.54e−01
f_{18}	0.5	**4.27e−03**	1.56e−02	5.87e−03	9.36e−03	1.23e−02	1.45e−02	7.16e−01	9.35e−01	7.26e−01	8.73e−01
f_{19}	0.5	**2.62e−02**	3.98e−01	6.85e−02	9.83e−02	1.13e−01	2.34e−01	8.64e−01	6.57e−01	7.35e−01	6.89e−01
f_{20}	0.5	**7.16e−02**	8.57e−01	1.93e−01	6.42e−01	7.69e−01	8.35e−01	1.08e+00	1.42e+00	1.47e+00	1.53e+00
f_{21}	1	**4.02**	7.53	4.16	6.49	7.45	8.73	10.93	12.43	11.84	12.97
$CF1$	1	**9.88**	25.46	34.65	21.43	22.83	55.47	53.45	59.83	56.74	52.35
$CF2$	1	**12.24**	35.46	29.87	25.46	29.84	61.32	62.95	59.73	60.04	61.19
$CF3$	1	**1.52**	19.78	35.64	32.96	36.45	66.46	64.07	63.85	67.02	69.46
$CF4$	1	**9.27**	27.98	41.23	36.94	40.02	64.35	65.08	67.91	63.40	65.90
$CF5$	1	**11.88**	32.64	38.97	33.72	39.08	69.52	68.03	65.47	63.07	68.79
$CF6$	1	**12.12**	37.65	46.54	37.65	41.14	68.64	69.96	67.34	68.72	66.01
$CF7$	1	**14.37**	48.67	43.27	42.54	46.56	66.73	68.06	67.82	69.01	65.47
$CF8$	1	**19.86**	41.23	49.85	46.57	45.06	69.63	65.42	66.79	68.71	69.05
$CF9$	1	**16.57**	45.34	47.58	48.46	49.47	71.43	69.73	66.03	68.59	70.41

Table 6. Summary of Wilcoxon's rank sum test results on the average number of peaks found for test functions.

Function	S-CMA	CDE	SDE	FERPSO	SPSO	r2pso	r3pso	r2psolhc	r3psolhc
f_1	0	0	0	1	1	1	1	1	1
f_2	0	0	0	0	1	1	1	1	1
f_3	1	0	0	1	1	1	1	1	1
f_4	1	1	1	1	1	0	0	0	0
f_5	0	1	0	0	0	0	0	0	0
f_6	1	1	1	0	1	0	1	0	0
f_7	0	1	0	0	0	0	0	0	0
f_8	1	1	1	1	1	1	1	1	1
f_9	0	1	−1	−1	1	0	0	0	0
f_{10}	1	1	1	0	1	1	1	1	1
f_{11}	1	1	1	1	1	1	1	1	1
f_{12}	1	−1	1	1	1	1	1	1	1
f_{13}	1	−1	1	1	1	1	1	1	1
f_{14}	1	0	1	1	1	1	1	1	1
f_{15}	1	1	0	1	1	1	1	1	1
f_{16}	1	1	1	1	1	1	1	1	1
f_{17}	1	1	1	1	1	1	1	1	1
f_{18}	1	0	1	1	1	1	1	1	1
f_{19}	1	1	1	1	1	1	1	1	1
f_{20}	1	1	1	1	1	1	1	1	1
f_{21}	0	1	1	1	1	1	1	1	1
*CF*1	1	1	0	1	1	1	1	1	1
*CF*2	1	1	1	0	1	1	1	1	1
*CF*3	1	1	1	1	1	1	1	1	1
*CF*4	0	0	0	0	0	0	0	0	0
*CF*5	1	1	1	0	1	1	1	1	1
*CF*6	1	1	0	0	1	1	1	1	1
*CF*7	1	1	0	0	1	1	1	1	1
*CF*8	1	1	0	0	1	1	1	1	1
*CF*9	1	1	1	1	1	1	1	1	1

Table 7. Success rates for locating both local and global optima.

Function	Peaks	ε	r	IWO-δ-DE	S-CMA	CDE	SDE	FER PSO	SPSO	r2pso	r3pso	r2psolhc	r3psolhc
f_1	2	0.05	0.5	**100**	**100**	**100**	84	64	44	72	56	48	52
f_2	2	0.05	0.5	**100**	**100**	**100**	68	88	72	56	32	52	76
f_3	5	0.05	0.5	**100**	48	44	4	0	0	0	0	8	0
f_5	5	0.000001	0.01	**100**	64	48	0	0	**100**	0	0	64	4
f_{10}	25	0.000001	0.5	**100**	60	0	0	0	92	60	52	84	76
f_{14}	10	0.1	0.5	**92**	80	76	56	32	48	56	52	36	32

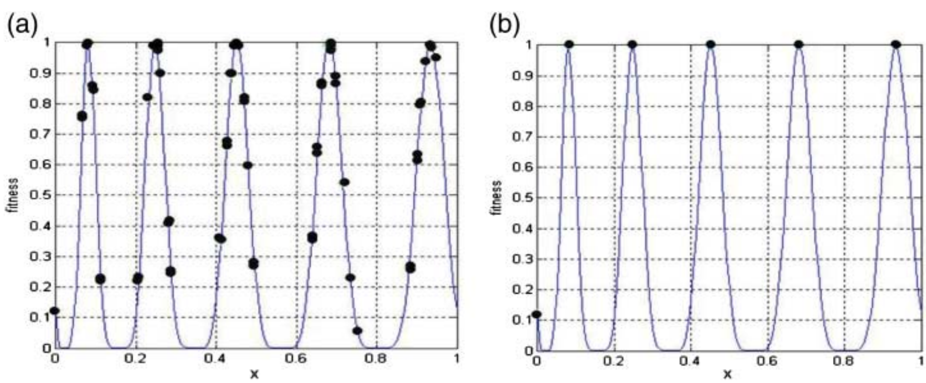
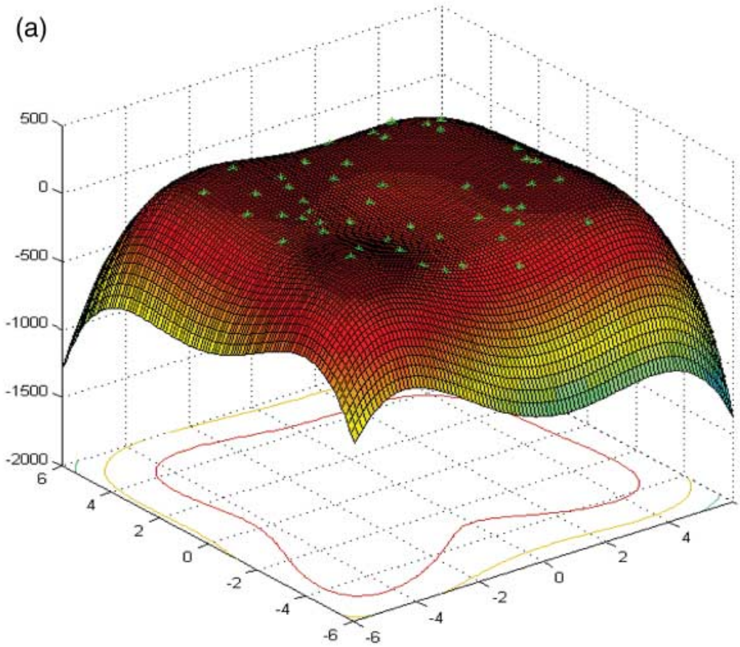


Figure 2. Distribution of individuals (shown as black dots) in the search space during the evolution of the process for f_6: (a) at 5000 function evaluations (FEs); (b) at 10,000 FEs.



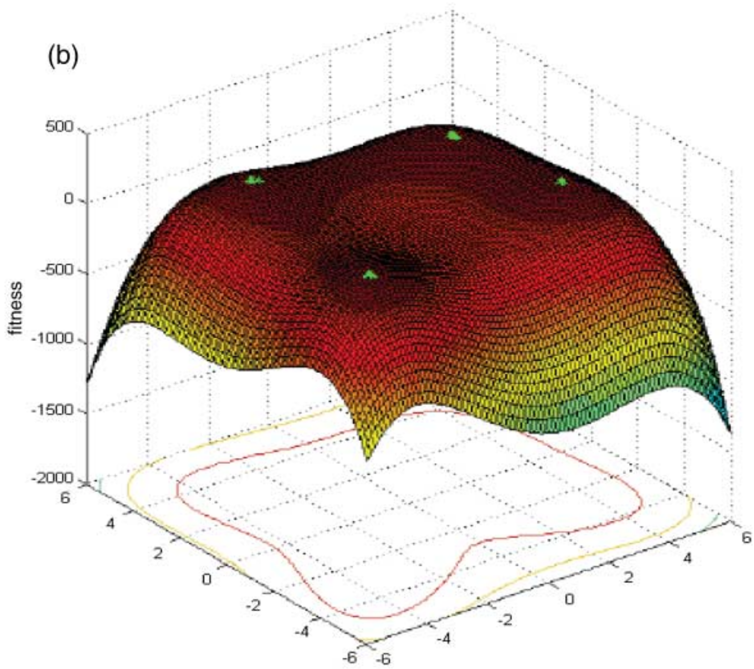


Figure 3. Distribution of individuals (shown as white points) in the search space during the evolution of the process for f_8: (a) initialization; (b) after 10,000 function evaluations.

to detect the optima accurately. Now, the crucial thing is the sharing of the maximum number of function evaluations that should be allotted to both algorithms. The experiments indicate that allocation of 80% to IWO and 20% to DE of the total budget of function evaluations serves as an efficient combination, at least for the benchmarks tested. Table 8 shows the variation in the

Table 8. Variation in the average number of peaks found.

Function	50% IWO–50% DE	60% IWO–40% DE	70% IWO–30% DE	80% IWO–20% DE	90% IWO–10% DE
f_3	**2**	**2**	**2**	**2**	**2**
f_4	4.92	**5**	**5**	**5**	**5**
f_6	4.92	**5**	**5**	**5**	**5**
f_8	3.62	3.74	3.80	**3.84**	3.80
f_9	1.48	1.56	1.56	**1.64**	1.56
f_{11}	5.36	5.52	5.56	**6**	5.56
f_{12}	26.72	28.68	29.32	**32.6**	29.48
f_{13}	96.8	99.6	100.8	**104.6**	100.8
f_{16}	2.68	2.76	2.8	**2.88**	2.84
CF3	3.2	3.32	3.4	**3.6**	3.56

performance of the proposed algorithm with variation in the shares of FEs allocated to IWO and DE.

The performance of IWO-δ-DE is sensitive to the sharing of maximum FEs for IWO and DE. The concept follows that IWO-δ-DE requires huge exploration in the initial stages to detect the global as well as local optimal zones. Although the explorative power of IWO is high, it requires a sufficient amount of FEs to cover all of the optimal zones. So, the majority percentage of maximum FEs should be allotted to IWO because in multimodal optimization the primary objective is to locate the optimal zones. In IWO-δ-DE, the basic function of DE is to provide adequate exploitation at the later stages so that the optimal peaks can be located perfectly. From the above results it is clear that the optimal combination of 80% of maximum FEs allotted to IWO and 20% FEs allotted to DE is best as far as the performance of IWO-δ-DE is concerned. If the percentage of FEs allotted to IWO is decreased below 80%, the multimodal capability of IWO-δ-DE deteriorates as IWO fails to detect the optimal zones. Now, if the combination of 90% IWO–10% DE is chosen, the performance graph of IWO-δ-DE, as per the peak accuracy and distance accuracy metric, falls slightly as DE cannot completely use its potential in a detailed search of small basins owing to the lack of FEs.

6. Application to dielectric composite multimodal optimization problem

Many hybrid evolutionary algorithms have previously been applied to solve real-world optimization problems (Liao 2010). Here, one such problem is considered. The objective of this problem is to determine the structure of a dielectric composite so that its macroscopic effective permittivity ε_{eff} in the direction of the field applied is equal to ε_{eff} (Figure 4).

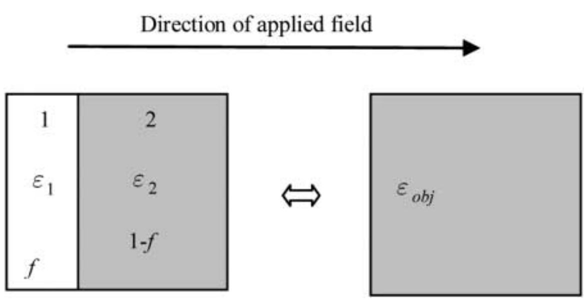


Figure 4. Homogenization of a laminated dielectric structure.

The difficulty in this problem is evident since an analytical expression of the effective permittivity exists. This expression can be obtained by considering the association in series of two dielectrics and is given below (Alami and El Imrani 2008):

$$\varepsilon_{eff} = \frac{\varepsilon_1 \varepsilon_2}{\varepsilon_2 f + \varepsilon_1 (1 - f)}, \tag{9}$$

where ε_1 and ε_2 indicate the permittivity of dielectrics 1 and 2, respectively, and f represents the concentration of material 1. To produce a composite of effective permittivity macroscopic $\varepsilon_{obj} = 1.5$ from a range of material 1 of permittivity varying between 10 and 30, the permittivity of material 2 being imposed ($\varepsilon_2 = 1$) and the concentration of material 1 can vary between 0.1 and 0.9. The problem of optimization to be solved can be expressed in the following way according to the two parameters and f:

$$\begin{cases} \min f_{obj}(f, \varepsilon_1) = |\varepsilon_{eff} - 1.5| \\ 0.1 \leq f \leq 0.9 \\ 10 \leq \varepsilon_1 \leq 30 \end{cases} \tag{10}$$

The solution of Equation (10) is obvious since according to Equation (9), a set of solution couples can be deduced such as $\varepsilon_{eff} = \varepsilon_{obj} = 1.5$ and consequently $f_{obj} = 0$. Figure 5 illustrates a representation of the objective function. The optimal solutions are distributed along a line $\varepsilon_{obj} = 1.5$ located at the intersection of two surfaces of opposite gradient.

The goal of any multimodal algorithm should be to identify the whole solutions. The proposed algorithm is applied to this problem with a population size of 100 and $\delta = 0.5$. The maximum number of function evaluations used is set to 10,000. Figure 6 shows the simulation run of IWO-δ-DE on the above problem.

Note that this problem does not include discrete peaks in the search space; rather, the minima are composed about a line $f_{obj} = 0$. The proposed IWO-δ-DE can easily detect the minimal points

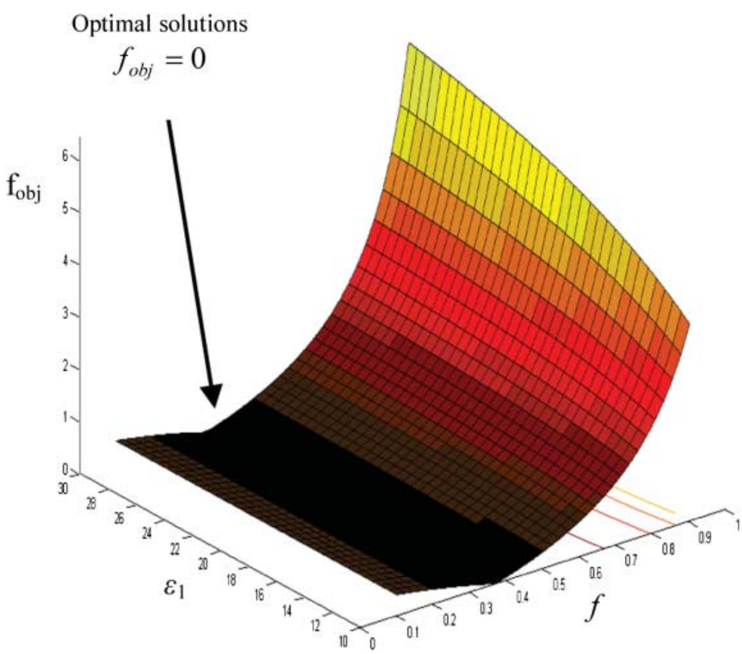


Figure 5. Representation of the objective function.

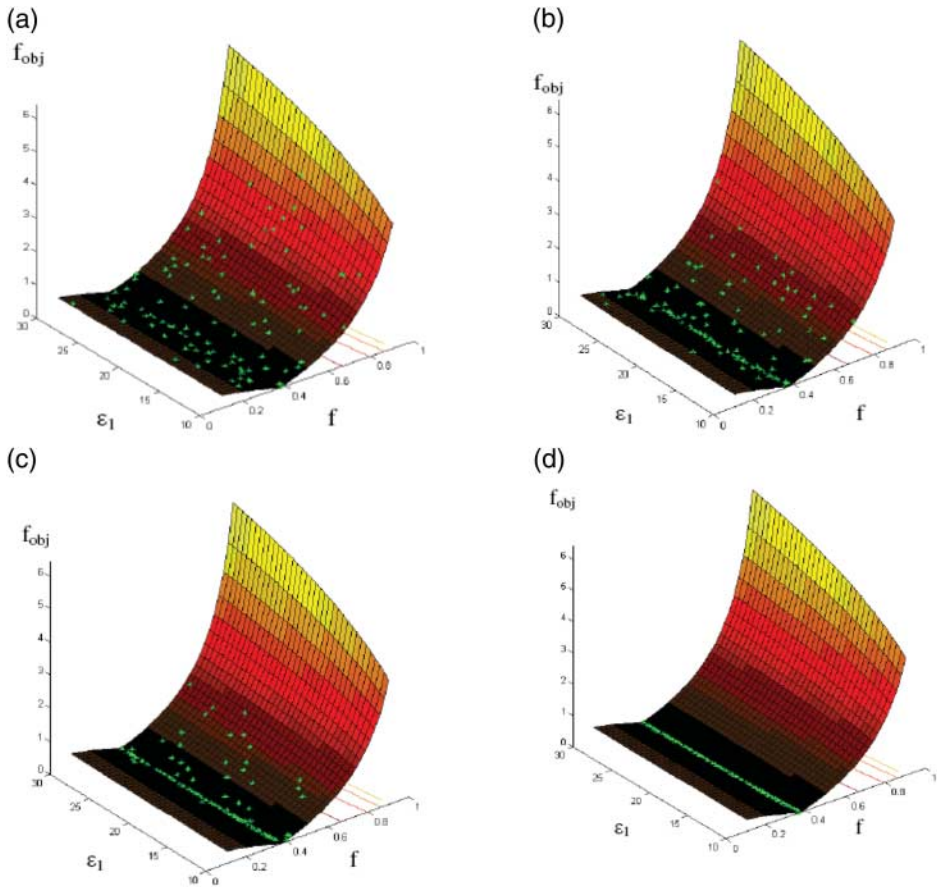


Figure 6. Distribution of individuals in the search space during the evolution of the process: (a) initialization; (b) at 3000 function evaluations (FEs); (c) at 7500 FEs; (d) at 10,000 FEs.

within a mere 10,000 function evaluations, *i.e.* individuals are well distributed along the optimal zone. In this way, the problem considered here, to determine the structure of a dielectric composite so that its macroscopic effective permittivity ε_{eff} in the direction of the field applied is equal to ε_{obj}, has been perfectly solved.

7. Conclusions

In this article a multimodal evolutionary optimization technique that integrates concepts of two powerful modern optimizers, IWO and DE, is proposed. The novel IWO-δ-DE was tested for the optimization of 20 benchmark functions (including seven composite functions) and a real-world multimodal problem concerning the structure of a dielectric composite. To justify its development, results were directly compared with nine state-of-the-art evolutionary multimodal optimizers based on performance metrics such as success rate, average number of peaks found and peak accuracy. The results were further validated using Wilcoxon's rank sum test. The results of the experimental studies suggest that IWO-δ-DE can provide a statistically superior and more consistent performance than the state-of-the-art multimodal optimization algorithms on the tested problems, within the same budget of FEs.

A first step to extend the current work may be to adapt the standard deviation parameter of each seed in the IWO population based on the fitness properties of the weed. This would surely open a path leading to substantially improved performance. Secondly, for a complete and general multimodal optimizer, it would be interesting to study and further modify the proposed approach for application to problems of a dynamic nature, where the landscape changes over time. A very important future research issue would be to develop a tool for estimating the number of local/global optima within the search range from the properties of the subpopulations created by IWO. This knowledge can help in adapting the control parameters for the subregional search with DE more efficiently.

References

Alami, J. and El Imrani, A., 2008. Dielectric composite multimodal optimization using a multipopulation cultural algorithm. *Intelligent Data Analysis*, 12, 359–378.

Beasley, D., Bull, D.R., and Martin, R.R., 1993. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1 (2), 101–125.

Bessaou, M., Petrowski, A., and Siarry, P., 2000. Island model cooperating with speciation for multimodal optimization. *In*: *Proceedings of the 6th international conference on parallel problem solving from nature (PPSN VI)*, Paris. Berlin: Springer, 16–20.

Brits, A.E.R. and van den Bergh, F., 2002. A niching particle swarm optimizer. *In*: *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning (SEAL 2002)*, Singapore, 692–696.

Chakraborty, P., *et al.*, 2009. On population variance and explorative power of the invasive weed optimization algorithm. *In*: *Proceedings of the world congress on nature and biologically inspired computing* (*NaBIC* 2009), Coimbatore, India. Piscataway, NJ: IEEE Press, 227–232.

Das, S. and Suganthan P.N., 2011. Differential evolution—a survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15 (1), 4–31.

Das, S., *et al.*, 2011. Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1 (2), 71–88.

De Jong, K.A., 1975. *An analysis of the behavior of a class of genetic adaptive systems*. Doctoral dissertation, University of Michigan, 1975.

Deb, K., 1989. *Genetic algorithms in multimodal function optimization, the clearinghouse for genetic algorithms*. MS thesis and Report 89002, University of Alabama, Tuscaloosa.

Deb, K. and Goldberg, D.E., 1989. An investigation of niche and species formation in genetic function optimization. *In*: *Proceedings of the third international conference on genetic algorithms*, 42–50.

Derrac, J., *et al.*, 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1 (1), 3–18.

Eiben, A.E. and Smith, J.E., 2003. *Introduction to evolutionary computing*. Berlin: Springer.

Gan, J. and Warwick, K., 2000. A variable radius niching technique for speciation in genetic algorithms. *In*: *Proceedings of the genetic and evolutionary computation conference (GECCO 2000)*, San Francisco, 96–103.

Goldberg, D.E. and Richardson, J., 1987. Genetic algorithms with sharing for multimodal function optimization. *In*: *Proceedings of the second international conference on genetic algorithms*, 41–49.

Harik, G.R., 1995. Finding multimodal solutions using restricted tournament selection. *In*: *Proceedings of the 6th international conference on genetic algorithms*, San Francisco, 24–31.

Hendershot, Z., 2004. A differential evolution algorithm for automatically discovering multiple global optima in multidimensional discontinuous spaces. *In*: *Proceedings of the fifteenth Midwest artificial intelligence and cognitive sciences conference*, Chicago, 16–18, 92–97.

Im, C., *et al.*, 2004. A novel algorithm for multimodal function optimization based on evolution strategy. *IEEE Transactions on Magnetics*, 40 (2).

Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization. *In*: *Proceedings of the IEEE international conference on neural networks (ICNN)*, 4, 1942–1948.

Li, X., 2004. Adaptively choosing neighborhood bests using species in a particle swarm optimizer for multimodal function optimization. *In*: *Proceedings of the conference on genetic and evolutionary computation 2004*, LNCS 3102, 105–116.

Li, X., 2005. Efficient differential evolution using speciation for multimodal function optimization. *In*: *Proceedings of the conference on genetic and evolutionary computation*, Washington DC, 873–880.

Li, X., 2007. Multimodal function optimization based on fitness–Euclidean distance ratio. *In*: *Proceedings of the conference on genetic and evolutionary computation 2007*, 78–85.

Li, X., 2010. Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, 14 (1).

Li, J.-P., *et al.*, 2002. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation*, 10 (3), 207–234.

Liao, T.W., 2010. Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, 10, 1188–1199.
Mahfoud, S., 1995. *Niching methods for genetic algorithms*. Thesis (PhD), University of Illinois, Urbana.
Mehrabian, A.R. and Lucas, C., 2006. A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1, 355–366.
Parsopoulos, K. and Vrahatis, M., 2001. Modification of the particle swarm optimizer for locating all the global minima. In: *Artificial neural networks and genetic algorithms*. Vienna: Springer, 324–327.
Parsopoulos, K. and Vrahatis, M., 2004. On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8 (3), 211–224.
Petrowski, A., 1996. A clearing procedure as a niching method for genetic algorithms. *In*: *Proceedings of the 3rd IEEE congress on evolutionary computation*, New York, 798–803.
Price, K., Storn, R., and Lampinen, J., 2005. *Differential evolution—a practical approach to global optimization*. Berlin: Springer.
Qu, B.Y. and Suganthan, P.N., 2010. Novel multimodal problems and differential evolution with ensemble of restricted tournament selection. *In*: *Proceedings of the IEEE congress on evolutionary computation*, Barcelona.
Qu, B.Y., Suganthan, P.N., and Liang, J.J., 2012. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, in press.
Rönkkönen, J. and Lampinen, J., 2007. On determining multiple global optima by differential evolution. *In*: *Evolutionary and deterministic methods for design, optimization and control*, *Proceedings of Eurogen* 2007, Jyyvaskyla, Finland, 146–151.
Shir, O.M. and Bäck, T., 2005. Niching in evolution strategies. *In*: *Proceedings of the 2005 conference on genetic and evolutionary computation*, New York.
Shir, O.M., Emmerich, M. and Bäck, T., 2010. Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. *Evolutionary Computation*, 18 (1).
Singh, G. and Deb, K., 2006. Comparison of multimodal optimization algorithms based on evolutionary algorithms. *In*: *Proceedings of the conference on genetic and evolutionary computation*, Seattle, WA. New York: ACM Press, 1305–1312.
Storn, R. and Price, K.V., 1995. *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces.* ICSI. Technical Report TR-95-012.
Suganthan, P.N., *et al.*, 2005. *Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization.* Technical Report, Nanyang Technological University, Singapore, May 2005 and KanGAL Report #2005005, IIT Kanpur, India.
Tang, K., *et al.*, 2007. *Benchmark functions for the CEC'2008 special session and competition on large scale global optimization.* Nature Inspired Computation and Applications Laboratory, USTC, China. Technical Report
Thomsen, R., 2004. Multimodal optimization using crowding-based differential evolution. *In*: *Proceedings of the IEEE 2004 congress on evolutionary computation*, 1382–1389.
Ursem, R.K., 1999. Multinational evolutionary algorithms. *In*: *Proceedings of the congress on evolutionary computation (CEC)*, 3. 1633–1640.
Wilcoxon, F., 1945. Individual comparisons by ranking methods. *Biometrics*, 1, 80–83.
Yao, J., Kharma, N., and Grogono, P., 2010. Bi-objective multipopulation genetic algorithm for multimodal function optimization. *IEEE Transactions on Evolutionary Computation*, 14 (1), 80–102.
Yin, X. and Germay, N., 1993. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. *In*: *Proceedings of the international conference on artificial neural networks and genetic algorithm*, 450–457.
Zaharie, D., 2004. A multipopulation differential evolution algorithm for multimodal optimization. *In*: *Proceedings of the 10th MENDEL international conference on soft computing*, Brno, Czech Republic, 17–22.