Soft Comput (2013) 17:1893–1910
DOI 10.1007/s00500-013-1028-4

METHODOLOGIES AND APPLICATION

A novel memetic algorithm based on invasive weed optimization and differential evolution for constrained optimization

Xinye Cai · Zhenzhou Hu · Zhun Fan

Published online: 22 March 2013

Abstract This paper presents a novel memetic algorithm, named as IWO_DE, to tackle constrained numerical and engineering optimization problems. In the proposed method, invasive weed optimization (IWO), which possesses the characteristics of adaptation required in memetic algorithm, is firstly considered as a local refinement procedure to adaptively exploit local regions around solutions with high fitness. On the other hand, differential evolution (DE) is introduced as the global search model to explore more promising global area. To accommodate the hybrid method with the task of constrained optimization, an adaptive weighted sum fitness assignment and polynomial distribution are adopted for the reproduction and the local dispersal process of IWO, respectively. The efficiency and effectiveness of the proposed approach are tested on 13 well-known benchmark test functions. Besides, our proposed IWO_DE is applied to four well-known engineering optimization problems. Experimental results suggest that IWO_DE can successfully achieve optimal results and is very competitive compared with other state-of-art algorithms.

Communicated by Y.-S. Ong.

X. Cai (✉) · Z. Hu
College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics,
Jiangsu, China
e-mail: xinye@nuaa.edu.cn

Z. Hu
e-mail: zzhu0612@gmail.com

Z. Fan
Department of Mechatronic Engineering, School of Engineering,
Shantou University, Guangdong, China
e-mail: zfan@stu.edu.cn

Keywords Memetic algorithm · Invasive weed optimization · Differential evolution · Constrained optimization · Multi-objective optimization

1 Introduction

Most real world problems are usually subject to various types of constraints and how to tackle these constrained optimization problems (COPs) has been extensively raised concerns. For COPs, constraint satisfaction is a matter of great account. In most cases, the huge search space but very narrow feasible space and the optimal solutions that lie on constraint boundaries all contribute to great difficulties when tackling COPs. COPs, without loss of generality, can be defined as follows. minimize $f(\mathbf{x})$, subject to

$$\begin{cases} g_j(\mathbf{x}) \leq 0, & j = 1, \ldots, p \\ h_j(\mathbf{x}) = 0, & j = p+1, \ldots, m \end{cases} \quad (1)$$

where $\mathbf{x}$ is the vector of the solutions ($\mathbf{x} = (x_1, x_2, \ldots, x_n)$) and $\mathbf{x} \in \Omega \subseteq \Psi, \Omega$ is the set of feasible solutions that satisfy p inequality constraints and (m-p) equality constraints and Ψ is an n-dimension rectangular space confined by the lower boundary and upper boundary of $\mathbf{x}$ as follows.

$$l_k \leq x_k \leq u_k, \qquad 1 \leq k \leq n \quad (2)$$

where l_k and u_k are the lower boundary and upper boundary for a decision variable x_k, respectively. Usually, equality constraints are transformed into inequality form as follows.

$$|h_j(\mathbf{x})| - \epsilon \leq 0, \qquad j = p+1, \ldots, m \quad (3)$$

where ϵ is an allowed positive tolerance value.

Over recent years, there have been reports on the performance improvements of using hybrid algorithms in context of constraint optimization (Barkat Ullah et al. 2009; Liu et al. 2007; Singh et al. 2010). This type of hybrid approaches is usually categorized as memetic algorithms (MAs) (Moscato 1989; Neri and Cotta 2012; Ong et al. 2010), which is the combination between a population-based global search and a heuristic local search. With the local refinement procedures, the efficiency of a population-based global search algorithm could be enhanced considerably. Usually, the hybridization of the global and local search is considered as a balance between exploration and exploitation in the evolutionary process.

Besides remarkable success of MAs in a wide range of application domains, emerging field of adaptive MAs has also attracted great amount of attention. In a recent survey (Chen et al. 2011), Chen et al. summarized the adaption of MAs into several core designs issues, including the frequency of refinements, selection of individual subset to undergo refinement, intensity of refinement, and choice of procedures to conduct refinement. For instance, Nguyen et al. (2007) investigated the impact of refinement frequency, selection of individual subset and intensity of refinement on MAs through empirical experiments. Ong and Keane (2004) proposed a type of Meta-Lamarckian learning in which refinement procedures are cooperative and competitive according to adaptive strategies. Krasnogor and Gustafson (2004) presented a self-generating mechanism by which various local search mechanisms to be used in memetic algorithm is conducted adaptively.

On the contrary, adaptation issues of MAs in the context of constrained optimization has attracted far less attention, though it plays an even more important role in many difficult constrained optimization problems due to the fact that such problems usually have huge search space but very narrow feasible space. A proper adaptive local search can avoid the waste of computational resources in the undesirable infeasible region and thus make the algorithm more efficient. Motivated by this, IWO is firstly considered as a local refinement procedure in this paper due to its intriguing characteristics of adaptation as follows. In IWO, (1) only individuals satisfying a certain fitness degree are permitted to reproduce offspring, and (2) the number of offsprings each individual reproduces is determined by the fitness value adaptively. With these two characteristics, IWO is able to control the refinement frequency, selection of individual subset and intensity adaptively in different stages of evolution. To the best of our knowledge, there is barely any research that considers IWO as a separate local search model though it owns these attractive characteristics of adaptation. Under these circumstances, the investigation of IWO as a local search model in the context of constrained optimization becomes very meaningful.

To complement with IWO as the local search, DE is used as the population-based global search model. DE is an emerging optimization algorithm that has been extensively investigated and surveyed in (Das and Suganthan 2011; Mezura-Montes et al. 2010). By adding the weighted difference between two randomly selected individuals to a third one, DE possesses a simple yet effective and efficient global optimization ability (Price et al. 2005). Therefore, DE is employed as global search models of MAs in literatures (Gong et al. 2010; Wang and Cai 2012b). In the context of constrained optimization, it is indispensable for exploring the search space effectively, especially when the feasible region is much small. With this consideration and previous researches on DE, we use DE as the global search model to locate the feasible region promptly. Besides, when finding feasible solutions, a suitable local search model to exploit the neighborhood areas nearby feasible solutions becomes necessary.

Unlike unconstrained optimization problems, both constraints and optimizing function of a solution need to be considered in COPs. Especially at the beginning of the search process, population may contain both feasible and infeasible solutions. One simple way to address this is to prioritize feasible solutions. However, on the other hand, some good infeasible solutions, especially solutions close to the constraint boundaries, may carry useful information which could steer the population toward feasible regions. Thus, an adaptive fitness assignment to compromise between feasibility and optimizing function in order to attain satisfactory results is another important issue in our work. This issue is much closely related to the efficiency of the local search, as the adaptation mechanism of IWO tends to allocate solutions with higher assigned fitness more computational resources.

With the analysis above, the main contributions of this paper are as follows.

– A novel constrained optimization memetic algorithm IWO_DE, which first considers IWO as a local refinement procedure, is proposed to tackle constrained optimization problems. To accommodate IWO within the constrained optimization, an adaptive weighted sum fitness assignment and polynomial distribution are adopted for reproduction and local dispersal process of IWO, respectively.
– The effectiveness and efficiency of IWO as the local refinement procedure is studied and the intrinsic mechanism of IWO is preliminarily investigated through experiments.
– The proposed hybrid algorithm is tested and compared with other state-of-art algorithms on 13 well known benchmark test functions (Liang et al. 2006) and four engineering optimization problems.

The remainder of this paper is organized as follows. Since the focus of this paper is on dealing with constrained optimization problems under the concept of memetic algorithm, Sect. 2 reviews related works on constraint-handling techniques and MAs. Subsequently, a brief overview of IWO as well as DE is presented in Sect. 3. Section 4 presents the proposed IWO_DE in detail. The experimental results are reported in Sect. 5. Further discussion of IWO as the local refinement procedure is given in Sect. 6. Finally, Sect. 7 draws the conclusion.

2 Related works

2.1 Review on constraint-handling techniques

Recent years a great deal of efforts has already been made on constraint-handling techniques and an elaborate survey of various constraint-handling techniques is referred to in (Coello 2002; Mezura-Montes and Coello 2011). In general, constraint-handling techniques can be classified into (1) penalty functions; (2) special representations and operators; (3) repair algorithms; (4) separate objective and constraints; and (5) hybrid methods.

Penalty functions are the most simple and common approaches for solving COPs so far. In static penalty function methods, such as (Homaifar et al. 1994), the penalty factors constant during the entire evolutionary process. Dynamic penalty function (Joines and Houck 1994), however, computes the penalty factors based on the generation of the evolutionary process. In adaptive penalty function methods such as (Farmani and Wright 2003) and (Woldesenbet et al. 2009), evolutionary information is fed back to determine the amount of penalty added to the infeasible individuals for the preferable selection of better infeasible individuals.

In addition to penalty function approaches, other constraints handling approaches have been proposed. Runarsson and Yao (2000) proposed a stochastic ranking (SR) method to alleviate the shortcomings associated with penalty factors. In SR, a probability parameter p_f is introduced as the comparison criterion to determine the rank of each individual. Takahama and Sakai (2006) proposed the ε constrained method which coverts a constrained optimization problem into an unconstrained one through defining an order relation with the ε level comparison. In this method, the authors defined the ε level comparison as an new order relation that is relevant to the objective function value and the constraint violation.

More recently, using Pareto dominance under multi-objective optimization concept to solve COPs has become increasingly popular. For COPs, constraints are regarded as one or more objective, and then COPs can be redefined as a multi-objective unconstrained optimization problems.

Zhou et al. (2003) used the Pareto strength approach (Zitzler et al. 2001) to rank individuals for the better selection of individuals. The rank between Individuals is determined in the following rules: (1) the higher Pareto strength value of the individual is preferable; (2) if strength value is equal, the one with lower sum amount of constraint violation is better.

Venkatraman and Yen (2005) presented a generic framework consisted of the following two phases. In the first phase, the goal is to find at least one feasible solution and the comparison of individuals only depends on the sum amount of constraint violation. In the second phase, COPs is considered as a bi-objective unconstrained optimization problems and both objectives (the original objective and the sum amount of constraint violation) are optimized and ranked by non-dominated sorting which is proposed in (Deb et al. 2002).

Wang and Cai (2012b) proposed a dynamic hybrid framework which comprises two main component: global search model and local search model. In this framework, the selection mechanism of these two models is carried out under pareto-dominance concept. The proposed framework has the advantage of implementing global and local search dynamically according to the feasibility proportion in the current population.

In this paper, we also use the concept of multi-objective optimization to address COPs. Under this circumstance, a single-objective constrained optimization problem can be redefined as follows:

minimize

$$F(\mathbf{x}) = (f(\mathbf{x}), G(\mathbf{x})) \tag{4}$$

where $G(\mathbf{x}) = \sum_{j=1}^{m} G_j(\mathbf{x})$ denotes the total amount of constraint violation of a variable $\mathbf{x}$ and $G_j(\mathbf{x})$ is the amount of constraint violation of solution $\mathbf{x}$ on the j-th constraint, calculated as follow.

$$G_j(\mathbf{x}) = \begin{cases} max(0, g_j(\mathbf{x})), & j = 1, \ldots, p \\ max(0, |h_j(\mathbf{x})| - \epsilon), & j = p+1, \ldots, m \end{cases} \tag{5}$$

Thus, a single objective constrained optimization problem is converted into a bi-objective unconstrained one. The first objective is the original objective function $f(\mathbf{x})$, and the second is the total amount of constraint violation $G(\mathbf{x})$.

2.2 Review on MAs

In memetic algorithm(MAs), two or more methods were incorporated for the purpose of use of their advantages to cope with optimization problems. MAs can be considered under the name of hybrid algorithms and be regarded as the combination of population-based global search and local refinement procedures.

Various forms of MAs have already been reported among a wide variety of optimization problems. Barkat Ullah et al. (2009) proposed a new agent based on memetic algorithm for dealing with COPs. Four types of local search techniques are adaptively selected through learning. Liu et al. (2007) proposed a co-evolutionary differential algorithm under the concept of MAs for COPs, in which two population are built and evolved cooperatively by independent differential algorithm. Singh et al. (2010) presented a memetic algorithms in which the strength of evolutionary algorithm and a local search strategy were incorporated to tackle COPs. Gong et al. (2010) presented a hybrid algorithm based on differential evolution and biogeography-based optimization (BBO) for global numerical optimization problems. Kelner et al. (2008) presented a new coupling optimization approach where a local search strategy based on the interior point method was integrated into genetic algorithm. More recently, Wang et al. (2012) proposed a memetic particle swarm optimization for tackling multi-modal optimization problems. In his work, two different local search techniques are used in a cooperative way. Wang et al. (2009) proposed an adaptive hill climbing strategy. The greedy crossover-based hill climbing and steepest mutation-based hill climbing were incorporated and used as the local search procedure within the framework of MAs for solving dynamic optimization problems. Tang et al. (2007) presented a diversity-based adaptive local search strategy based on parameterized Gaussian distribution. The local search strategy is integrated into the framework of the parallel memetic algorithm to address large scale combinatorial optimization problems. Molina et al. (2010) proposed a intense continuous local search in the framework of MAs.

In the above reviews, MAs exhibit very promising performance in various optimization problems. Therefore, a similar hybrid idea is presented in this paper.

3 Overview of invasive weed optimization and differential evolution

3.1 Invasive weed optimization

IWO is first presented by Mehrabian and lucas (2006) to solve numerical optimization problems. IWO simulates the nature principles and behaviors of weedy invasion and colonization in the shifting and turbulent environment. Later, Kundu et al. (2011) proposed a variant of IWO that extends the original IWO to handle multi-objective optimization problems. Generally speaking, there are four steps of IWO.

1. *Initialize a population*: solutions are initialized and dispersed in the given n dimensional search space uniformly and randomly.
2. *Reproduction*: each individual of the population is permitted to reproduce seeds according to its own fitness, the colony's lowest and highest fitness, in this situation, the fitness of each individual is normalized and the number of seeds each individual reproduces depends on a given minimum and maximum and increases linearly.
3. *Spatial dispersal*: offspring are randomly distributed over the n dimensional search space by normally distributed random numbers with mean equal to zero; but varying variance. Through this, a group of offspring are produced around their parent individual and thus weed colony is formed to enhance the search ability. Furthermore, standard deviation (sd) of the normally distributed random function will be reduced from a predefined initial value, sd_{max}, to a final value, sd_{min}, over every generation. the value of sd for a given generation is computed as follows.

$$sd = \frac{(sd_{max} - sd_{min}) * (iter_{max} - iter)^m}{iter_{max}^m} + sd_{min} \quad (6)$$

   where $iter_{max}$ is the maximum number of generations, $iter$ is the current number of generation and m is the nonlinear modulation index.
4. *Competitive exclusion*: with the growth and reproduction of weeds, after passing several generation, the number of weeds in a colony will reach its maximum. Therefore, an essential exclusion mechanism is needed among weeds. The exclusion mechanism is applied to eliminate weeds with low fitness and select good weeds that reproduce more than undesirable ones. Subsequently, the selected ones will be preserved into the next generation and then the steps I-IV are repeated until termination criterion is reached.

3.2 Differential evolution

DE, proposed by Storn and Price (1995), is a powerful search algorithm in the optimal problems, and uses the vector differences of individuals for perturbing the population members.

Initially, DE comprises a population of N and every individual is an n-dimensional vectors $\mathbf{x}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. These vectors are randomly generated in the search space and in the process of evolution, individuals will be tackled by the operations of mutation, crossover and selection.

Mutation operation: in this operation, with the different mutant strategies, the generated way of a mutant vector $\mathbf{v}_i$

is different and the weighted vector difference of individuals is various. The popular mutant strategies are summarized (Das and Suganthan 2011) as follows.

- DE/rand/1: $\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$
- DE/rand/2: $\mathbf{v}_i = \mathbf{x}_{r_1} + F(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) + F(\mathbf{x}_{r_4} - \mathbf{x}_{r_5})$
- DE/best/1: $\mathbf{v}_i = \mathbf{x}_{best} + F(\mathbf{x}_{r_1} - \mathbf{x}_{r_2})$
- DE/best/2: $\mathbf{v}_i = \mathbf{x}_{best} + F(\mathbf{x}_{r_1} - \mathbf{x}_{r_2}) + F(\mathbf{x}_{r_3} - \mathbf{x}_{r_4})$
- DE/current to best/1: $\mathbf{v}_i = \mathbf{x}_i + F(\mathbf{x}_{best} - \mathbf{x}_i) + F(\mathbf{x}_{r_1} - \mathbf{x}_{r_2})$

where the subscript r_1, r_2, r_3, r_4, r_5, which are all different from the index i, are not equal to each other and selected uniformly and randomly from the range [1, N]. $\mathbf{x}_{best}$ is the best individual of the current population, F is a scaling factor that measure the scale of the difference of vectors.

Crossover operation: with the mutant vector $\mathbf{v}_i$ and the target vector $\mathbf{x}_i$, the trial vector $\mathbf{u}_i$ is generated by binomial crossover as follows.

$$\mathbf{u}_{i,j} = \begin{cases} \mathbf{v}_{i,j}, & \text{if } rand_j \le C_r or\, j = j_{rand} \\ \mathbf{x}_{i,j}, & \text{otherwise} \end{cases} \tag{7}$$

where $i = 1, 2, \ldots, N, j = 1, 2, \ldots, n, j_{rand}$ is a selected integer randomly from $[1, n]$ which ensures $\mathbf{u}_i$ inherits at least one component from the mutant vector $\mathbf{v}_i$, $rand_j$ is a uniform random number between 0 and 1. C_r is the crossover probability parameter and its value is within [0, 1].

Selection operation: to keep better individual into next generation, the generated trial vector $\mathbf{u}_i$ is compared with the target vector $\mathbf{x}_i$. The selection operation is described as follows.

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G}, & \text{if } f(\mathbf{u}_{i,G}) \le f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G}, & \text{otherwise} \end{cases} \tag{8}$$

4 Proposed approach

This section presents IWO_DE in details. In this method, the refinement procedures of IWO is incorporated before the population-based search algorithm DE. In addition, single-objective COPs are transformed into bi-objective ones, that is, the first objective is the original objective and the second the sum amount of constraint violation, as described by the formula (4). We minimize these two objectives simultaneously and the concept of pareto dominace is adopted as a measurement criteria to decide which individual to survive in the next generation.

The definition of Pareto dominance is as follows.

Definition (*Pareto Dominance*) a vector $\mathbf{x}_i^1$ is said to Pareto dominance another vector $\mathbf{x}_i^2$ (denoted by $\mathbf{x}_i^1 \prec \mathbf{x}_i^2$), if and only if $\forall i \in \{1, 2, \ldots, n\}, \mathbf{x}_i^1 \le \mathbf{x}_i^2 \wedge \exists i \in \{1, 2, \ldots, n\}, \mathbf{x}_i^1 < \mathbf{x}_i^2$

The following sections explain the components of IWO_DE one by one.

4.1 The local refinement procedure—IWO

From the overview of IWO, we can obtain the idea that every weed will reproduce seeds in a distributional manner around parent weeds. If seeds are distributed close to their parent weeds to a certain extent, the behavior can lead to a local search ability in the weed evolution precess and therefore result in exploiting the local areas effectively.

In this section, we explain the general steps of IWO, as described in Sect. 3.1. However, to accommodate IWO in the context of COPs, an adaptive weighted sum fitness assignment mechanism is adopted to determine the amount of the reproduction for each weed in the *Reproduction* step, and the polynomial distribution function is employed in the *Spatial dispersal* step. Furthermore, we use non-dominated sorting (Deb et al. 2002) in the *competitive exclusive* step.

4.1.1 Adaptive weighted sum fitness assignment

In IWO, the number of seeds each weed generates reflects the ability of reproduction of each weed and better fitness indicates more offspring.

Generally, One simple way to address COPs in local search is to prefer feasible solutions first. However, under this mechanism, much computational resource is allocated to the undesirable feasible solutions with worse objective function values, and the infeasible solutions that lie near the constraint boundaries are barely allocated any computational resource during the local search process, although they are more likely to lead the local search to the optimal solutions.

For the sake of balance between feasibility and objective function for local search of IWO, an adaptive fitness assignment mechanism is adopted to determine the reproduction ability of each weed. Under this adaptation mechanism, we weigh the single objective and the amount of constraint violations adaptively through a weight factor ω, which is the percentage of feasible solutions. The descriptive form of the adaptive fitness assignment formula is following.

$$fitness(\mathbf{x}_i) = \sqrt{\omega f'(\mathbf{x}_i)^2 + (1 - \omega) G'(\mathbf{x}_i)^2} \tag{9}$$

and

$$\omega = \frac{\text{the number of feasible individuals}}{\text{the population size}} \tag{10}$$

and $f'(\mathbf{x})$ and $G'(\mathbf{x})$ are the normalization results of the objective function $f(\mathbf{x})$ and the sum amount of constraint violation $G(\mathbf{x})$ respectively, described as follows.

$$\begin{cases} f'(\mathbf{x}_i) &= \frac{f(\mathbf{x}_i) - \min \mathrm{f}(\mathbf{x})}{\max \mathrm{f}(\mathbf{x}) - \min \mathrm{f}(\mathbf{x})} \\ G'(\mathbf{x}_i) &= \frac{G(\mathbf{x}_i) - \min \mathrm{G}(\mathbf{x})}{\max \mathrm{G}(\mathbf{x}) - \min \mathrm{G}(\mathbf{x})} \end{cases} \tag{11}$$

With the above analysis, in this paper, the number of seeds reproduced by a weed is formulated as follows.

$$seed_{num} = floor(S_{max} - (S_{max} - S_{min})f_i) \tag{12}$$

and

$$f_i = \frac{fitness(\mathbf{x}_i) - \min fitness(\mathbf{x})}{\max fitness(\mathbf{x}) - \min fitness(\mathbf{x})} \tag{13}$$

where S_{max} denotes the permissible maximum number of seed and S_{min} the permissible minimum number of seed. Besides, f_i is the normalized fitness function and the better f_i (without loss of generality, to a minimization problem) of one weed is, the more number of seeds it reproduces.

4.1.2 Polynomial distribution function

Need to point out that in the optimal process of the original IWO (Mehrabian and Lucas 2006), normal distribution function is used as spatial dispersal operator to preserve the ability of exploration and exploitation, but it is very hard for various forms of problems to find out an appropriate consistent initial value of standard deviation and then how to effectively and efficiently decide the step-size of standard deviation of normal distribution function at every generation is problem-dependent. With these considerations, polynomial distribution function is adopted as the spatial dispersal operator.

The polynomial distribution is used as mutation operator in (Deb and Goyal 1996), but we use the polynomial probability distribution to generate whole distribution values on the decision variable space to reproduce offspring around the parent individuals under the IWO framework. The polynomial distribution has its mean at the current variable and its variance depending on a parameter n. This parameter will provide the degree that how far the distributed variables are apart from the current variable.

Besides, The polynomial distribution depends on a perturbance factor δ for calculate distributed values, which is defined as follows.

$$P(\delta) = 0.5(n+1)(1-|\delta|)^n, \delta \in (-1, 1) \tag{14}$$

First, to get distributed values, a random number $u \in (0, 1)$ is needed to be generated. Thereafter, the perturbance factor δ corresponding to u is calculated using the above formula.

$$\bar{\delta} = \begin{cases} (2u)^{1/(n+1)} - 1, & \text{if } u < 0.5 \\ 1 - [2(1-u)]^{1/(n+1)}, & \text{if } u \geq 0.5 \end{cases} \tag{15}$$

Finally, the distributed variable is calculated as follows.

$$c = p + \bar{\delta}\Delta_{max} \tag{16}$$

where c is the distributed variable and Δ_{max} is the maximum perturbance value from the parent variable p.

If the parameter n is larger enough, the distributed variable can be very close to its parent variable. Therefore, polynomial probability distribution function is embedded in the framework of IWO to achieve the local search ability for refining better solutions.

4.1.3 Exclusion mechanism of IWO

Under the framework of IWO, the worse individuals will be eliminated when the population size reaches the maximum. In this paper, nondominated sorting (Deb et al. 2002) is adopted to rank each individual for eliminating the worse individuals. In non-dominated sorting algorithm, each individual is assigned to a non-dominated front it belongs to. Accordingly, the exclusion mechanism of IWO is described as follows.

1. if individuals lie on different non-dominated front, then the individuals lying on lower non-dominated front win;
2. if individuals have the same non-dominated front, then the ones with smaller constraint violation win.

Through the above sorting, better individuals both in terms of the objective value and the amount of constraint violation are obtained and kept for the next generation.

As described in (Deb 2000), diversity is an important aspect in a population and there are several popular ways to maintain diversity, such an niching methods (Deb and Goldberg 1989) and usage of mutation (Goldberg 1989). In this paper, we use polynomial mutation (Deb and Goyal 1996) to maintain the diversity of population.

The pseudocode of the local refinement procedures of IWO is presented in *Algorithm* 1.

Algorithm 1 The local refinement procedure — IWO

```
step 1: input the parent population P;
step 2: R=pd(P);
/* pd is the polynomial distribution function that is acted as the spatial dispersal function
of IWO */
step 3: R_m=pm(R);
/* pm is the polynomial mutation function to maintain the diversity of IWO*/
step 4: P_R=P ∪ R_m;
step 5: If the size of P_R ≥ P_max Then
/* P_max denotes the permissible maximum of population*/
step 6: P=select(P_R)
/* use the selection mechanism, described in Section 4.1.3, to pick out the elitist individuals
to be used by DE */;
step 7: end If
```

4.2 The global search model—DE

After the refinement procedures of IWO, the elitist individuals are picked out. Therefore, the operations of DE are applied among the selected elitist individuals to explore more promising space for the aim of finding out more promising solutions to be refined by IWO. In this paper, we simply adopt the classical version of DE—“DE/rand/1/bin” that has been described in Sect. 3.2.

The operations of mutation and crossover are the same as that are described in Section 3.2 and only the operation of selection is modified to meet needs, described as follows.

1. if the generated trial vector is feasible, then it is compared with all the feasible solutions of population and if the generated trial vector is better than all the feasible solution, then the worse feasible solution is replaced with the generated trial vector;
2. if the generated trial vector is infeasible, then it is compared with the all infeasible solutions of population with the concept of Pareto dominance and if the generated trial vector dominates every infeasible solution, then the all infeasible solutions are ranked with the mechanism as described as Sect. 4.1.3 and the individual with the last rank is selected to be compared with the generated trial vector, if the amount of constraint violation of the generated trial vector is lower than that of the selected individual, then the selected individual is replaced with the generated trial vector.

From the above operations, as for the non-dominated trial vectors, the one with lower amount of constraint violation is preferable.

The pseudocode of the global search model—DE is presented in *Algorithm* 2.

Algorithm 2 The global search model — DE

step 1: input the population P selected by IWO;
step 2: **for** i=1:NP **do**
step 3: randomly select different subscript r_1, r_2, r_3 within [1, NP]-i;
step 4: v_i=$mutation(P_{r_1,r_2,r_3},\ F)$;
/* use the strategy of "DE/rand/1" to generate the mutant vector */
step 5: u_i=$crossover(v_i,\ P_i,\ C_r)$;
/* use the binomial crossover to generate the trial vector */
step 6: decide whether u_i wins or not according to the selection operation, as described in the Section 4.2;
step 7: **end for**

As explained above, for each component of IWO-DE, the attracting characteristics of IWO and DE is merged. That is, for IWO, the characteristics refers to the point that each individual fully exploits the useful information around it and for DE, the excellent characteristics refers to making full use of information provided by IWO to find out more valuable individuals which are to be refined, in return, by IWO.

The whole framework of IWO_DE is presented in *Algorithm* 3.

Algorithm 3

objective function values over 25 runs have been listed under the given parameter settings.

As shown in Table 3, the best solution obtained is very approximate to the known optimal value and the global optimal solution has been found consistently on test functions except for g02 and g13. For g02, the best solution can't be found consistently but the near-optimal solutions have been obtained in 16 runs out of 25 runs and the mean result of the obtained objective function value is also very approximate to the known global optimal value. Similarly, IWO_DE can't find the global optimal solution in all 25 runs for g13 since IWO_DE falls the local optimum in 1 run out of 25 runs.

Furthermore, it can be observed from the standard deviation for the test functions in Table 3 that IWO_DE is stable and robust for solving these problems. Finally, for all the test functions, IWO_DE is capable of obtaining feasible solutions in all test runs.

5.3 Performance analysis of IWO_DE

Several performance evaluation criteria suggested in (Liang et al. 2006) have been introduced to further demonstrate the performance of IWO_DE on test functions.

5.3.1 Efficiency analysis of IWO_DE

Tables 4 and 5 present the function error value ($f(x) - f(x^*)$) ($\mathbf{x}$ is the obtained best feasible solution and $\mathbf{x}^*$ is the best known solution) after different FEs for each run, and with the suggestion in (Liang et al. 2006), Tables 4 and 5 record error value after 5×10^3, 5×10^4, 5×10^5 FEs. In Tables 4 and 5, the statistical measures, such as the best, median, mean, worst and standard deviation of the error values, are listed. Besides, the number of violated constraint that cannot satisfy the feasible condition is computed and presented in the parenthesis near by the best, median and worst of the error values. Similarly, in Tables 4 and 5, the symbol c denotes the number of violated constraint for the median solution obtained by IWO_DE and it has three values. The sequence of the c indicates the number of constraint violations whose amount is more than 0.1, 0.01 and 0.0001 respectively. The symbol $\bar{v}$ denotes the mean value of the constraint violations for the median solution obtained by IWO_DE. Note that for convenience, the error value is recorded as zero when less than 1E−10.

Furthermore, Table 6 has recorded performance evaluation criteria value of the obtained experimental results, such as feasible rate, success rate, success performance and the number of FEs needed to find a solution satisfying the given condition: $f(\mathbf{x}) - f(\mathbf{x}^*) \leq 0.0001$. Feasible rate is the percentage of feasible runs out of total runs, and feasible run is referred to a run during which one feasible solution is found under the maximum FEs. Similarly, success rate is the percentage of successful runs out of total runs, but successful run is referred to a run during which the algorithm obtains a feasible solution $\mathbf{x}$ satisfying the given condition: $f(\mathbf{x}) - f(\mathbf{x}^*) \leq 0.0001$.

As presented in (Liang et al. 2006), success performance is defined as follows:

$$\text{success performance} = \frac{\text{mean}(\#\text{ of FEs for successful runs}) \times (\#\text{ of total runs})}{(\#\text{ of successful runs})} \tag{17}$$

It is shown from Tables 4 and 5 that except for g05 and g13, the feasible solutions of the remaining test functions are found by IWO_DE before 5×10^3 FEs. Furthermore, IWO_DE is able to achieve the known optimal solution before 5×10^4 FEs in 25 runs for test functions except for g02, g10 and g13. For g13, IWO_DE is able to obtain the optimal solution in 24 out of 25 runs and 16 out of the total 25 runs for g02 before 5×10^4 FEs. Finally, the proposed algorithm is able to find the optimal solution for g10 before the allowed maximum FEs. Therefore, the results show that IWO_DE is able to converge to the optimal solution efficiently within the acceptable FEs.

From Table 6, we can see that IWO_DE can achieve 100% feasible rate for all of the test functions, which means IWO_DE can find feasible solutions in all 25 runs consistently. Besides, IWO_DE obtains 100 % success rate for 11 out of 13 test functions, which means IWO_DE have the ability of finding the satisfactory solutions in all runs under the given accuracy level, but for g02 and g13, IWO_DE can't obtain solutions of the given condition consistently within the total runs. Furthermore, we can observe through further analysis from Table 6 that

Table 1 Summary of 13 benchmark test functions

f	n	Type	ρ (%)	LI	LE	NI	NE	a
g01	13	Quadratic	0.0111	9	0	0	0	6
g02	20	Nonlinear	99.9971	0	2	0	0	1
g03	10	Polynomial	0.0000	0	0	0	1	1
g04	5	Quadratic	52.1230	0	6	0	0	2
g05	4	Cubic	0.0000	2	0	0	3	3
g06	2	Cubic	0.0066	0	2	0	0	2
g07	10	Quadratic	0.0003	3	5	0	0	6
g08	2	Nonlinear	0.8560	0	2	0	0	0
g09	7	Polynomial	0.5121	0	4	0	0	2
g10	8	Linear	0.0010	3	3	0	0	6
g11	2	Quadratic	0.0000	0	0	0	1	1
g12	3	Quadratic	4.7713	0	1	0	0	0
g13	5	Nonlinear	0.0000	0	0	0	3	3

Table 2 Parameter values of IWO_DE

Symbol	Description	Value
F	Scaling factor	0.7
C_r	Crossover probability parameter	Between 0.9 and 1
P_{init}	Initial number of population	20
P_{max}	Maximum number of population	60
S_{min}	Minimum number of seed	0
S_{max}	Maximum number of seed	2
pd	Polynomial distribution index	100
pm	Polynomial mutation index	1
M_p	Mutation probability	$1/\eta$

η Denotes the number of decision variables

IWO_DE needs only at most 2×10^5 to find the satisfactory solutions under the given condition.

5.3.2 *Convergence analysis of IWO_DE*

In order to visualize the convergence in the search process, the convergence plots of every test functions have been illustrated in Figs. 1, 2, 3, 4. The convergence plots depict the average convergence rate of each test function within the allowed maximum FEs over total runs. Two axes of the figures demonstrate $log10(f(\mathbf{x}) - f(\mathbf{x}^*))$ vs FEs where $\mathbf{x}$ is the best obtained solution corresponding to a certain number of FEs. Note that the solutions which satisfy $f(\mathbf{x}) - f(\mathbf{x}^*) \leq 0$ have not been plotted, since zero or negative number for $log10$ is incalculable.

It can be observed from Figs. 1, 2, 3, 4 that the convergence rate of most test functions is very fast and this observation is consistent with the observed success performance and function error value of each test functions in Tables 4, 5 and 6. Hence IWO_DE is very efficient in tackling the test functions.

5.4 Comparisons with other state-of-the-art algorithms

IWO_DE is compared against six high-performance algorithms under three performance evaluation criteria: feasible rate, success rate and success performance. These selected state-of-the-art algorithms are PSO (Zielinski and Laur 2006), εDE (Takahama and Sakai 2006), GDE (Kukkonen and Lampinen 2006), MDE (Mezura-Montes et al. 2006), jDE-2 (Brest et al. 2006) and PCX (Sinha et al. 2006). The comparative results have been shown in Tables 7, 8 and 9.

In Table 7, the average of feasible rate and success rate have been listed respectively. With respect to the average of feasible rate, IWO_DE has very similar performance to that of PSO, εDE, MDE, jDE-2 and PCX but IWO_DE achieves better performance than GDE. In terms of the average of success rate, as shown in Table 8, the performance of IWO_DE is worse than εDE and PCX, but superior to PSO, GDE, MDE and jDE-2.

As analysed in Wang and Cai (2012b), success performance can be used to reflect the efficiency of a method to a certain extent under the given condition. In Table 9, we have listed the success performance of these six algorithms and have computed the sum success performance to be used for measuring the efficiency of these algorithms. In Table 9, *NA* denotes that the satisfactory solution can't be obtained under the given condition. From Table 9, IWO_DE exerts higher performance when compared against these six algorithms with the total of success performance. Specifically, for g02, g03, g11 and g13, the success performance of IWO_DE is better than all of the compared algorithms. In the remaining test functions (g01, g04, g05, g06, g07, g08,

Table 3 Experimental results of IWO_DE on the 13 well-known benchmark test functions

f	Optimum	Best	Median	Mean	Worst	SD
g01	−15.0000000000	−15.0000000000	−15.0000000000	−15.0000000000	−15.0000000000	0
g02	−0.8036191042	−0.8036191040	−0.8036191006	−0.7989322351	−0.7783222501	7.1e−03
g03	−1.0005001000	−1.00050010001	−1.00050010001	−1.00050010001	−1.00050010001	3.5e−16
g04	−30665.53867178332	−30665.53867178332	−30665.53867178332	−30665.53867178332	−30665.53867178332	3.7e−12
g05	5126.4967140071	5126.4967140071	5126.4967140071	5126.4967140071	5126.4967140071	2.0e−12
g06	−6961.8138755802	−6.961813875580167	−6.961813875580167	−6.961813875580167	−6.961813875580167	0
g07	24.30620906818	24.30620906818	24.30620906818	24.30620906818	24.30620906818	9.9e−15
g08	−0.0958250414	−0.0958250414	−0.0958250414	−0.0958250414	−0.0958250414	1.1e−15
g09	680.6300573744	680.6300573744	680.6300573744	680.6300573744	680.6300573744	4.1e−13
g10	7049.2480205287	7049.2480205287	7049.2480205287	7049.2480205287	7049.2480205287	2.7e−12
g11	0.7499000000	0.7499000000	0.7499000000	0.7499000000	0.7499000000	1.1e−16
g12	−1.0000000000	−1.0000000000	−1.0000000000	−1.0000000000	−1.0000000000	0
g13	0.0539415140	0.053941514042	0.053941514042	0.069335957792	0.438802607794	7.7e−02

Table 4 Function error values obtained when 5×10^3 FEs, 5×10^4 FEs and 5×10^5 FEs for test functions g01–g07

FEs	g01	g02	g03	g04	g05	g06	g07
5×10^3							
Best	0.9988(0)	0.0748(0)	0.0077(0)	1.6001(0)	5.1262E+03(3)	0.1949(0)	4.3707(0)
Median	2.2461(0)	0.1273(0)	0.0854(0)	6.6813(0)	5.1253E+03(3)	1.5841(0)	11.2875(0)
Worst	3.4988(0)	0.2465(0)	0.4132(0)	20.4654(0)	5.1239E+03(3)	5.0059(0)	21.0487(0)
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	1, 3, 3	0, 0, 0	0, 0, 0
v	0	0	0	0	0.04850	0	0
Mean	2.2161	0.1440	0.1066	7.7189	5.1252E+03	1.9623	11.5714
SD	0.5886	0.0484	0.1047	4.3668	0.6391	1.4562	4.2724
5×10^4							
Best	5.2143E−05(0)	3.2367E−06(0)	1.5584E−09(0)	0(0)	0(0)	0(0)	8.9631E−04(0)
Median	1.9878E−04(0)	1.2462E−05(0)	2.1669E−08(0)	0(0)	0(0)	0(0)	3.1670E−03(0)
Worst	4.5413E−04(0)	2.5310E−02(0)	5.9553E−07(0)	0(0)	0(0)	0(0)	2.9905E−02(0)
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0	0	0	0	0	0	0
Mean	1.9614E−04	6.1079E−03	6.7135E−08	0	0	0	7.6068E−03
SD	1.1329E−04	7.8231E−03	1.2535E−07	0	0	0	8.7596E−03
5×10^5							
Best	0(0)	1.8157E−10(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Median	0(0)	3.5592E−09(0)	0(0)	0(0)	0(0)	0(0)	0(0)
Worst	0(0)	2.5297E−02(0)	0(0)	0(0)	0(0)	0(0)	0(0)
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0	0	0	0	0	0	0
Mean	0	4.6869E−03	0	0	0	0	0
SD	0	7.1443E−03	0	0	0	0	0

Table 5 Function error values obtained when 5×10^3 FEs, 5×10^4 FEs and 5×10^5 FEs for test functions g08–g13

FEs	g08	g09	g10	g11	g12	g13
5×10^3						
Best	7.2256E−09(0)	0.9583(0)	1.4065E+03(0)	1.7317E−09(0)	4.3718E−11(0)	7.7577E−06
Median	5.0106E−07(0)	3.1668(0)	4.1108E+03(0)	2.5274E−08(0)	7.2385E−10(0)	3.4213E−05
Worst	7.2304E−05(0)	9.1730(0)	1.7241E+04(0)	4.1938E−04(0)	2.5238E−08(0)	0.3871
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0	0	0	0	0	0
Mean	5.7752E−06	3.9780	4.9869E+03	1.7391E−05(0)	2.1437E−09	0.0156
SD	1.5532E−05	2.1832	3.8765E+03	8.3797E−05(0)	5.0228E−09	0.0774
5×10^4						
Best	0(0)	3.9379E−09(0)	1.3802(0)	0(0)	0(0)	4.5647E−11
Median	0(0)	5.0507E−08(0)	6.0090(0)	0(0)	0(0)	6.5623E−11
Worst	0(0)	4.9097E−07(0)	100.9495(0)	0(0)	0(0)	0.3849
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	
v	0	0	0	0	0	
Mean	0	7.2777E−08	12.5363	0	0	0.0154
SD	0	9.5074E−08	20.1177	0	0	0.0770
5×10^5						
Best	0(0)	0(0)	0(0)	0(0)	0(0)	4.1898E−11
Median	0(0)	0(0)	0(0)	0(0)	0(0)	4.1898E−11
Worst	0(0)	0(0)	0(0)	0(0)	0(0)	0.3849
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0	0	0	0	0	0
Mean	0	0	0	0	0	0.0154
SD	0	0	0	0	0	0.0770

Table 6 Number of FEs to achieve the given accuracy level, feasible rate, success rate and success performance

f	Best	Median	Mean	Worst	SD	Feasible rate (%)	Success rate (%)	Success performance
g01	44,855	54,129	53,634	63,309	4,885.7	100	100	53,634
g02	24,220	31,545	42,683	162,074	35,041.3	100	64	66,692
g03	12,391	16,044	16,484	23,891	2,883.0	100	100	16,484
g04	19,233	22,605	22,537	25,955	1,787.6	100	100	22,537
g05	20,356	25,178	25,025	29,983	2,040.2	100	100	25,025
g06	9,438	10,804	10,770	11,967	615.0	100	100	10,770
g07	77,381	91,345	93,403	134,319	12,424.9	100	100	93,403
g08	1,838	2,838	2,990	4,367	661.4	100	100	2,990
g09	19,218	23,721	23,990	27,058	1,953.2	100	100	23,990
g10	158,667	180,886	182,112	215,739	15,068.3	100	100	182,112
g11	1,419	2,014	1,976	2,538	325.4	100	100	1,976
g12	919	1,422	1,402	2,076	310.6	100	100	1,402
g13	12,633	16,353	17,114	35,830	4,787.7	100	96	17,827

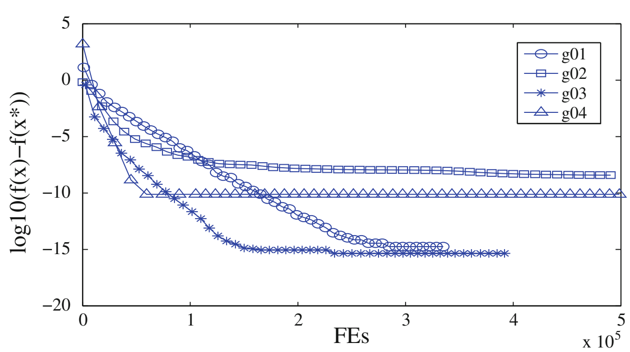

Fig. 1 Convergence graph for g01, g02, and g03

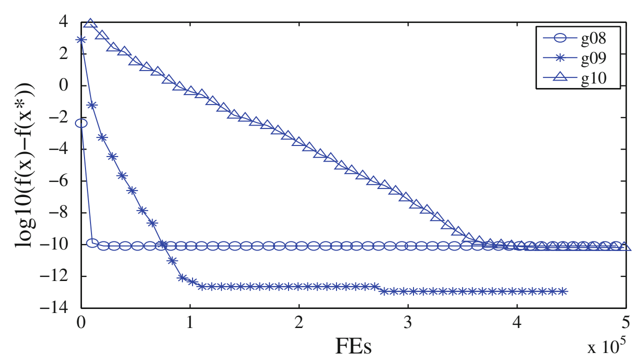

Fig. 3 convergence graph for g07, g08, g09 and g10

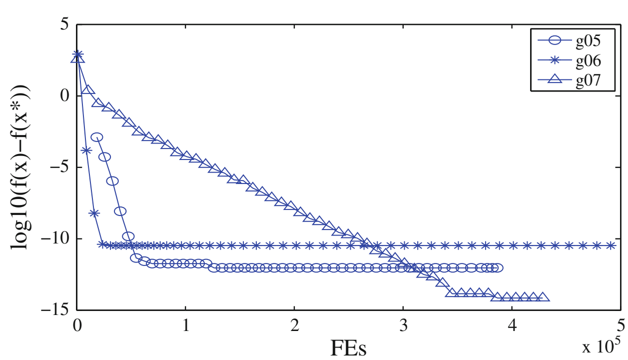

Fig. 2 Convergence graph for g04, g05, and g06

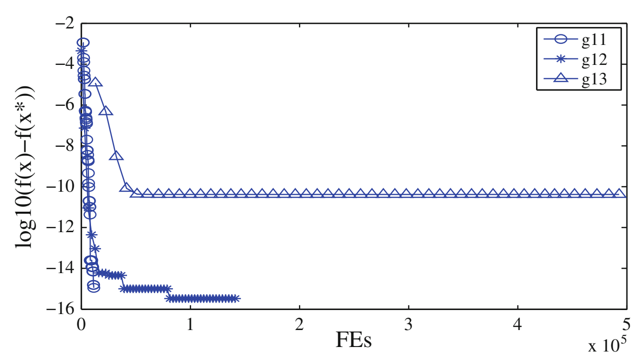

Fig. 4 convergence graph for g11, g12, and g13

g09, g10 and g12), the success performance of IWO_DE is better than more than half of the compared algorithms. Furthermore, as for PSO and jDE-2, there are several functions, i.e. g02, g03 and g13 for PSO, and g03 and g13 for jDE-2, which can't be solved to get satisfactory solutions, but IWO_DE is able to solve all of the test functions within allowed maximum FEs.

5.5 Engineering optimization problems

In this section, we apply IWO_DE to solving four engineering optimization problems to further evaluate its performance. The detailed descriptions of these four problems can be found in (Aguirre et al. 2007; Cagnina et al. 2008) and in this paper, these problems are denoted as follows.

Table 7 Comparison among algorithms in terms of feasible rate

f	Feasible rate						
	PSO (%)	εDE (%)	GDE (%)	MDE (%)	jDE-2 (%)	PCX (%)	IWO_DE (%)
g01	100	100	100	100	100	100	100
g02	100	100	100	100	100	100	100
g03	100	100	96	100	100	100	100
g04	100	100	100	100	100	100	100
g05	100	100	96	100	100	100	100
g06	100	100	100	100	100	100	100
g07	100	100	100	100	100	100	100
g08	100	100	100	100	100	100	100
g09	100	100	100	100	100	100	100
g10	100	100	100	100	100	100	100
g11	100	100	100	100	100	100	100
g12	100	100	100	100	100	100	100
g13	100	100	88	100	100	100	100
mean	100	100	98.46	100	100	100	**100**

Table 8 Comparison among algorithms in terms of success rate

f	Success rate						
	PSO (%)	εDE (%)	GDE (%)	MDE (%)	jDE-2 (%)	PCX (%)	IWO_DE (%)
g01	72	100	100	100	100	100	100
g02	0	100	72	16	92	64	64
g03	0	100	4	100	0	100	100
g04	100	100	100	100	100	100	100
g05	24	100	92	100	68	100	100
g06	100	100	100	100	100	100	100
g07	72	100	100	100	100	100	100
g08	100	100	100	100	100	100	100
g09	100	100	100	100	100	100	100
g10	8	100	100	100	100	100	100
g11	100	100	100	100	96	100	100
g12	100	100	100	100	100	100	100
g13	0	100	40	100	0	100	96
Mean	59.69	**100**	85.23	93.54	81.23	97.23	96.92

- Welded beam design problem.
- Pressure vessel design problem.
- Tension/compression Spring design problem.
- Speed reducer design problem.

For convenience, these engineering problems are abbreviated as WBP, PVP, T/CSP and SRP respectively.

5.5.1 *Experimental results of engineering problems*

We perform IWO_DE in 150,000 FEs for WBP, T/CSP and SRP and 40000 FEs for PVP with 30 runs. The parameters have been set as the same values as that have been used in benchmark test functions and listed in Table 2. The experimental results have been presented in Table 10.

From the statistic measures listed in Table 10, we can conclude that IWO_DE has successfully solved all the engineering optimization problems efficiently.

5.5.2 *Convergence analysis of IWO_DE for engineering problems*

Figures 5, 6, 7, 8 illustrate the convergence situation of IWO_DE when solving WBP, PVP, T/CSP and SRP, respectively.

From these convergence plots, it can be observed that IWO_DE rapidly converges to the best currently known objective function values in all of the four engineering problems, which makes us to believe that IWO_DE is able to tackle real-world engineering problems effectively and efficiently.

5.5.3 *Comparison of algorithms on engineering optimization problems*

In Table 11, the experimental results of IWO_DE are compared with other state-of-art algorithms, taken from literatures (Aguirre et al. 2007; Cagnina et al. 2008; He and Wang 2007; Mezura-Montes and Coello 2005). For convenience, we regard these state-of-art approaches as COPSO (Aguirre et al. 2007), SiC-PSO (Cagnina et al. 2008), Mezura (Mezura-Montes and Coello 2005) and HPSO (He and Wang 2007). It is important to mention that, in Table 11, the experimental results are regained through rerunning IWO_DE with 30 runs within 24,000 FEs at maximum.

Table 9 Comparison among algorithms in terms of success performance

f	Success performance						
	PSO	εDE	GDE	MDE	jDE-2	PCX	IWO_DE
g01	101,825	59,308	40,519	75,373	50,386	55,204	53,634
g02	*NA*	149,825	149,561	96,222	145,899	127,900	**66,692**
g03	*NA*	89,407	3,577,150	44,988	*NA*	34,937	**16,484**
g04	37,802	26,216	15,281	41,562	40,728	30,989	22,537
g05	1528,433	97,431	193,503	21,306	446,839	94,765	25,025
g06	37,946	7,381	6,503	5,202	29,488	33,821	10,770
g07	562,717	74,303	123,996	194,202	127,744	117,121	93,403
g08	3,656	1,139	1,469	918	3,236	2,826	2,990
g09	103,677	23,121	30,230	16,152	54,919	46,527	23,990
g10	6,094,056	105,234	82,604	16,4160	146,150	89,028	182,112
g11	33,073	16,420	8,460	3,000	53,928	38,688	**1,976**
g12	6,906	4,124	3,149	1,308	6,356	8,960	1,402
g13	*NA*	34,738	840,766	21,732	*NA*	53,735	**17,827**
Sum	8,510,091+3*NA*	688,647	5,073,191	686,125	1,105,673+2*NA*	734,501	**518,842**

Bold values indicate the performance of the proposed algorithms, where it is the best compared with that of other approaches on the corresponding benchmark functions

Table 10 Experimental results of IWO_DE on engineering optimization problems

f	Best	Median	Mean	Worst	SD	FEs
WBP	1.724852308597365	1.724852308597365	1.724852308597364	1.724852308597365	1.1e−15	150,000
PVP	6059.714335048436	6059.714335048436	6059.714335048435	6059.714335048436	9.3e−13	40,000
T/CRP	0.012665232788319	0.012665232788319	0.012665232788319	0.012665232788319	1.2e−17	150,000
SRP	2.994471066146820	2.994471066146820	2.994471066146822	2.994471066146820	1.9e−12	150,000

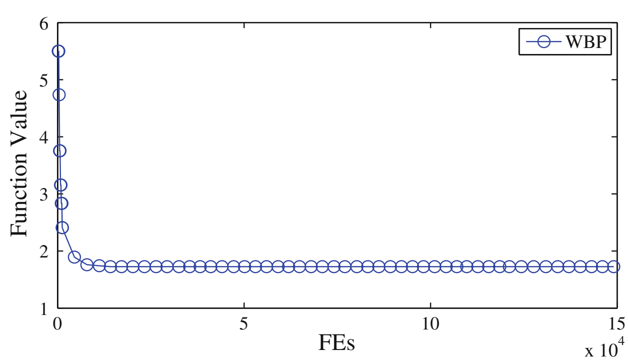

Fig. 5 Convergence *curve* of the objective function value for WBP

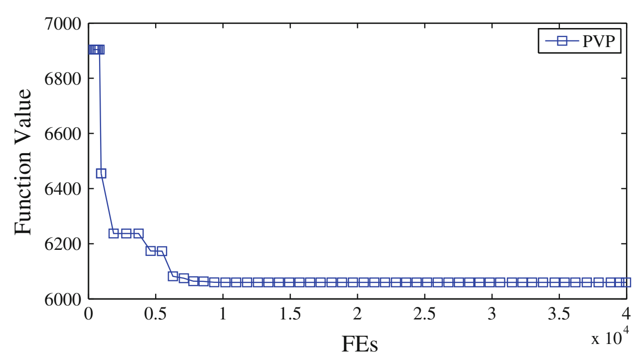

Fig. 6 Convergence *curve* of the objective function value for PVP

In Table 11, It is obvious to see that IOW_DE reports better performance on mean and standard deviation, compared with COPSO, Sic-PSO, Mezura and HPSO for PVP, T/CSP and SRP. For WBP, IWO_DE has slightly worse performance on the best result than COSPO, Sic-PSO, Mezura and HPSO, but the FEs of IWO_DE is less than that of COPSO, Mezura and HPSO. Furthermore, IWO_DE provides better performance on mean results than Sic-PSO, Mezura and HPSO for WBP.

It is worth to note that with the increase of FEs, the performance of IWO_DE on engineering optimization problems remains improving (see Table 10). Another interesting finding is that, IWO_DE obtains an even better solution for SRP than what's been reported so far. This new optimal solution for SRP found by the proposed IWO_DE is $\mathbf{x} = (3.5, 0.7, 17, 7.3, 7.715319911478246, 3.350214666096448, 5.286654464980222)$ with the corresponding $f(\mathbf{x}) = 2994.471066146820$.

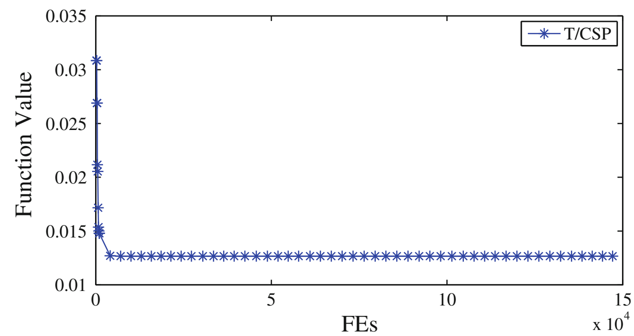

Fig. 7 Convergence *curve* of the objective function value for T/CSP

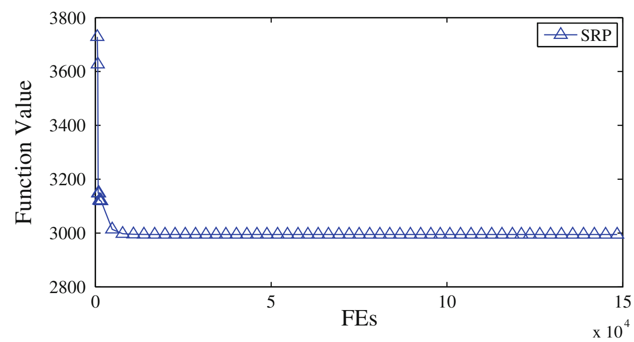

Fig. 8 Convergence *curve* of the objective function value for SRP

6 Discussion

This paper presents IWO_DE as a hybrid model in which IWO is acted as the local refinement procedure to exploit the regions around elite individuals and DE is used as the global search algorithm to explore more promising regions. Thus, this section is dedicated to the discussion on the intrinsic mechanism of IWO_DE.

6.1 DE as the global search model

In context of COPs, how to explore feasible solution promptly is of great importance, especially when the feasible search space is extremely small. DE possesses a simple yet effective and efficient global optimization ability (Price et al. 2005) and there have been reports on successful application of DE for global optimization problems, referred to in Gong et al. (2010), Qin et al. (2009), Wang and Cai (2012a) etc. Among them, DE was reported to be considered as the global search model and local search model simultaneously in tackling COPs in Wang and Cai (2012b). Hence we adopt DE as global search model in our proposed method based on the observations of previous empirical study.

Table 11 Comparison among algorithms on engineering problems

f	COPSO	Sic-PSO	Mezura	HPSO	IWO_DE
WBP					
Best	1.724852	1.724852	1.724852	1.724852	1.724865
Mean	1.724881	2.0574	1.777692	1.749040	1.725046
SD	1.27E−05	2.15E−01	8.8E−2	4.0E−02	1.7E−04
FEs	30,000	24,000	30,000	81,000	24,000
PVP					
Best	6059.714335	6059.714335	6059.7143	6059.7143	6059.71433505
Mean	6071.013366	6092.0498	6379.938037	6099.9323	**6059.71433522**
SD	15.10	12.17	2.1E+2	86.20	**7.4E−07**
FEs	30,000	24,000	30,000	81,000	24,000
T/CRP					
Best	0.012665	0.012665	0.012689	0.0126652	0.012665233
Mean	0.012666	0.0131	0.013165	0.0127072	**0.012665244**
SD	1.28E−06	4.1E−04	3.9E−4	1.58E−05	**3.6E−08**
FEs	30,000	24,000	30,000	81,000	24,000
SRP					
Best	2996.372448	2996.348165	2996.348094[a]	–	**2994.473177**
Mean	2996.408525	2996.3482	2996.348094[a]	–	**2994.483853**
SD	2.87E−02	0	0	–	**8.3E−03**
FEs	30,000	24,000	30,000	–	24,000

Bold values indicate the performance of the proposed algorithms, where it is the best compared with that of other approaches on the corresponding benchmark functions

– Denotes not available

[a] Denotes infeasible solution

Furthermore, as previously presented in Sect. 5.3.1, IWO_DE can obtain the satisfactory solutions before 2×10^5 FEs under the given condition. Therefore, in order to check the efficiency of DE on the ability of global search, we compare IWO_DE with IWO_non-DE in which the component of DE isn't considered and only run IWO_DE and IWO_non-DE under the maximum 2×10^5 FEs and the other experimental conditions are the same in Sect. 5.1. Note that, for clarity, we only summarize the experimental results in Table 12 which cause significant difference on the compared methods.

It is obvious from Table 12 that IWO_non-DE can't get the satisfactory solutions before the given FEs, and for g05,there are only 5 feasible runs out of total 25 runs. Besides, from the Table 12, it is also demonstrated that DE indeed search more promising region and provides more competitive solutions. Thus in terms of experiments, DE does play an important role of achieving good performance for the proposed IWO_DE.

6.2 Effectiveness of the local search ability of IWO

Similarly, in the context of COPs, when finding feasible solutions, how to exploit the feasible solutions effectively is also important. This paper firstly adopts IWO as the local refinement procedures and in order to demonstrate the local search ability of IWO, we compare IWO_DE with itself only when no refinement procedures of IWO is incorporated. For the convenience, IWO_DE without local search is denoted as non-IWO_DE here. In non-IWO_DE, the number of population is set to equal to P_{max} described in Table 2 and the remaining experimental conditions are the same as that is described in Sect. 5.1 so that we can have a fair comparison. Note that, for clarity, we only summarize

Table 12 Comparison of IWO_DE and IWO_non-DE

Function Optimum	g01 –15.000	g05 5126.496714	g07 24.306209	g09 680.630057	g10 7049.24802	g13 0.0539415
Best						
IWO_DE	–15.0000	5126.496714	24.306209	680.630057	7092.81990	0.0539415
IWO_non-DE	–14.9983	5128.066747	24.4555	680.6411	7163.36559	0.0539449
Mean						
IWO_DE	–15.0000	5126.496714	24.306209	680.630057	7049.24802	0.0693360
IWO_non-DE	–14.9956	5205.599381	25.0084	680.6761	8720.83632	0.0994170
Worst						
IWO_DE	–15.000000	5126.496714	24.306209	680.630057	7049.24802	0.4388026
IWO_non-DE	–14.9900	5334.249436	25.8530	680.7462	10721.00698	0.7060011
Infeasible run						
IWO_DE	0	0	0	0	0	0
IWO_non-DE	0	20	0	0	0	0

Table 13 comparison of IWO_DE and non-IWO_DE

Function Optimal	g01 –15.000	g02 –0.8036191	g03 –1.0005001	g05 5126.496714	g07 24.306209	g09 680.630057	g10 7049.24802	g13 0.0539415
Best								
IWO_DE	–15.000000	–0.8036191	–1.0005001	5126.496714	24.306209	680.630057	7049.24802	0.0539415
Non-IWO_DE	–14.945468	–0.7296201	–0.9784715	5126.496714	24.839557	680.967584	7163.36559	0.0539449
Mean								
IWO_DE	–15.000000	–0.7989322	–1.0005001	5126.496714	24.306209	680.630057	7049.24802	0.0847304
Non-IWO_DE	–13.100752	–0.6070030	–0.7273365	5139.645461	28.180487	690.079299	7902.33239	0.1799978
Worst								
IWO_DE	–15.000000	–0.7783223	–1.0005001	5126.496714	24.306209	680.630057	7049.24802	0.4388026
Non-IWO_DE	10.294968	–0.4431408	–0.2704784	5405.657932	34.289398	714.263329	9391.89257	0.8748623
Infeasible run								
IWO_DE	0	0	0	0	0	0	0	0
Non-IWO_DE	0	0	0	3	0	0	0	6

the experimental results in Table 13 which cause significant difference on the compared methods.

Based on the results from Table 13, non-IWO_DE has obtained much worse performance on all test functions when compared with IWO_DE. Although the results of non-IWO_DE for g05 is very close to that of IWO_DE, the number of infeasible runs of non-IWO_DE for g05 is 3 out of 25 runs and as well for g13, the infeasible runs of non-IWO_DE is 6 out of 25 runs. Furthermore, we can notice that the best results obtained by non-IWO_DE of each test function are very close to that of IWO_DE, and thus we can speculate that if non-IWO_DE exploits solutions sufficiently around the best solutions, it also can obtain the same results as IWO_DE. Therefore, this situation validates that the local search ability of IWO plays an important role on refining good solutions within local areas in the process of evolution.

6.3 Contribution of the local refinement procedure of IWO

The search ability of IWO has been demonstrated in above section, but the internal mechanisms of IWO in which how the colonization and invasion of IWO is operated should be observed. In this section, we try to plot figures to show how weeds reproduce seeds according to ones own fitness for survival and further to illustrate the process of colonization and invasion of weeds.

We describe four states of reproduction of IWO in the evolution process for test functions g01, g02, g07 and g10 to show the ability of colonization and invasion of weeds in Fig. 9 and there are one state corresponding to test function g01, g02, g07 and g10 respectively. For Fig. 9a, the parent weed generates two offspring seeds where one is better in terms of the amount of constraint violation but worse with respect to objective function value and another is better in terms of objective function value and worse with respect to the amount of constraint violation. For Fig. 9b, the parent weed reproduces two offspring seeds where two seeds are both better in terms of the amount of constraint violation and both worse with respect to the objective function value. For Fig. 9c, the parent weed generates two offspring seeds where two seeds are both worse in terms of the amount of constraint violation and both better with respect to the objective function value. For Fig. 9d, the parent weed reproduces two offspring seeds where two seeds are both better in terms of the amount of constraint violation and the objective function value.

After explaining the four representative states in Fig. 9a–d, we can come to the conclusion that IWO indeed plays a role in the local search ability and with the characteristics of colonization and invasion, weeds refine better seeds in the neighborhood of them in terms of either or both of the amount of constraint violation and the objective function value and then the better seed will replace parent weed to capture the suitable space for growth and reproduction. Therefore, the local refinement procedures of IWO

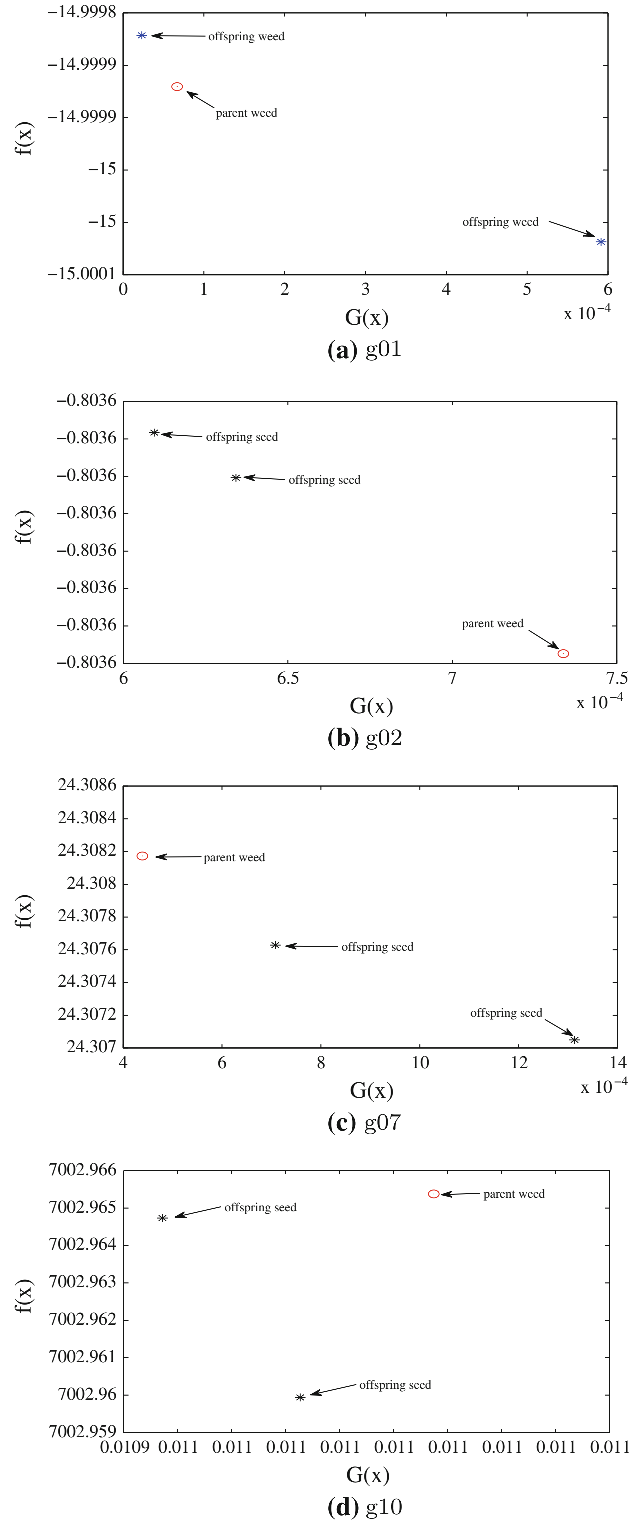

Fig. 9 The state of reproduction of IWO in the evolution process

plays an important role on the performance of hybrid algorithm IWO_DE.

7 Conclusions

This paper presents a novel memetic algorithm, which combines the intriguing characteristics of Invasive Weed Optimization with Differential evolution to deal with COPs. The proposed method regards IWO as the refinement procedures to adaptively exploit the promising local region during the optimization process. Through experimental analysis, we can make the conclusion that IWO does play an indispensable role in local exploitation. On the other hand, the use of DE as the global search model aims to finding more promising solutions among elitist solution refined by IWO. The efficiency and effectiveness of IWO_DE is demonstrated by experiments on the well-known benchmark test functions and engineering optimization problems. Experimental results show that the efficiency of IWO_DE is prominent for achieving the known optimal solution.

Future works include the further investigation on the intrinsic mechanism of IWO so that improved variant of IWO with more powerful local search ability can be proposed.

Acknowledgments The research reported herein was supported in part by the Fundamental Research Funds for the Central Universities of China (NO. NS2012074), Research Fund for the Doctoral Program of Higher Education of China (NO. 20123218120041) and National Natural Science Foundation of China (No. 61175073).

References

Aguirre AH, Zavala AM, Diharce EV, Rionda SB (2007) COPSO: constrained optimization via pso algorithm. Technical report, Center for Research in Mathematics (CIMAT)

Barkat Ullah ASSM, Sarker R, Cornforth D, Lokan C (2009) AMA: a new approach for solving constrained real-valued optimization problems. Soft Comput 13(8–9):741–762

Brest J, Zumer V, Maucec MS (2006) Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In: IEEE congress on evolutionary computation. Vancouver, BC, Canada pp 215–222

Cagnina LC, Esquivel SC, Coello CAC (2008) Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica 32(3):319–326

Chen X, Ong YS, Lim MH, Tan KC (2011) A multi-facet survey on memetic computation. IEEE Trans Evol Comput 15(5):591–607

Coello CAC (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Meth Appl Mech Eng 191(11–12): 1245–1287

Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31

Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):311–338

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

Deb K, Goyal M (1996) A combined genetic adaptive search (geneas) for engineering design. Comput Sci Inf 26(4):30–45

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist nondominated sorting genetic algorithm for multiobjective optimization: NSGA II. IEEE Trans Evol Comput 6:182–197

Farmani R, Wright J (2003) Self-adaptive fitness formulation for constrained optimization. IEEE Trans Evol Comput 7:445–455

Goldberg DE (1989) Genetic algorithms in search optimization and machine learning. Addison-Wesley, Reading

Gong W, Cai Z, Ling CX (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. Soft Comput 15(4):645–665

He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186(2):1407–1422

Homaifar A, Lai SHY, Qi X (1994) Constrained optimization via genetic algorithms. Simulation 62(4):242–254

Joines J, Houck C (1994) On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems wiht GAs. In: Proceedings of congress on evolutionary computation, pp 579–584

Kelner V, Capitanescu F, Lonard O, Wehenkel L (2008) A hybrid optimization technique coupling an evolutionary and a local search algorithm. J Comput Appl Math 215(2):448–456

Krasnogor N, Gustafson S (2004) A study on the use of self-generation in memetic algorithms. Nat Comput 3:53–76

Kukkonen S, Lampinen J (2006) Constrained real-parameter optimization with generalized differential evolution. In; IEEE congress on evolutionary computation. Vancouver, BC, Canada, pp 207–214

Kundu D, Suresh K, Ghosh S, Das S, Panigrahi BK, Das S (2011) Multi-objective optimization with artificial weed colonies. Inf Sci 181(12):2441–2454

Liang JJ, Runarsson TP, Mezura-Montes E, Clerc M, Suganthan PN, Coello CAC, Deb K (2006) Problem definitions and evaluation criteria for the CEC 2006. Nanyang Technol. Univ., Singapore, Tech. Rep.

Liu B, Ma H, Zhang X, Zhou Y (2007) A memetic co-evolutionary differential evolution algorithm for constrained optimization. In: IEEE congress on evolutionary computation, pp 2996-3002

Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inf 1:355–366

Mezura-Montes E, Coello C (2005) Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Proceedings of the 4th Mexican international conference on artificial intelligence (MICAI). Lecture notes on artificial intelligence (LNAI) 3789:652–662

Mezura-Montes E, Coello CAC (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 1(4):173–194

Mezura-Montes E, Velazquez-Reyes J, Coello CAC (2006) Modified differential evolution for constrained optimization. In: IEEE congress on evolutionary computation, Vancouver, BC, Canada pp 25–32

Mezura-Montes E, Miranda-Varela ME, del Carmen Gmez-Ramn R (2010) Differential evolution in constrained numerical optimization: an empirical study. Inf Sci 180(22):4223–4262

Molina D, Lozano M, Garca-Martnez C, Herrera F (2010) Memetic algorithms for continuous optimisation based on local search chains. Evol Comput 18(1):27–63

Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: toward memetic algorithms. Tech.

Rep. Caltech Concur-rent Computation Program, California Instit. Technol., Pasadena, CA, Tech. Rep. 826

Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: a literature review. Swarm Evol Comput 2:1–14

Nguyen QH, Ong YS, Lim MH, Krasnogor N (2007) A study on the design issues of memetic algorithm. In: IEEE congress on evolutionary computation, pp 2390–2397

Ong YS, Keane AJ (2004) Meta-Lamarckian learning in memetic algorithms. IEEE Trans Evol Comput 8(2):99–110

Ong YS, Lim MH, Chen X (2010) Memetic computation: past, present and future. IEEE Comput Intell Mag 5(2):24–31

Price K, Storn R, Lampinen J (2005) Differential evolution: a practical approach to global optimization. Springer, Berlin

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 4(3):284–294

Sinha A, Srinivasan A, Deb K (2006) A population-based parent centric procedure for constrained real parameter optimization. In: IEEE congress on evolutionary computation, Vancouver, BC, Canada, pp 239–245

Singh H, Ray T, Smith W (2010) Performance of infeasibility empowered memetic algorithm for CEC2010 constrained optimization problems. In: IEEE congress on evolutionary computation, pp 1–8

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

Mallipeddi R, Suganthan PN (2010) Problem Definitions and Evaluation Criteria for the CEC 2010 Competition on Constrained Real Parameter Optimization. Nanyang Technological University, Singapore, Technical Report

Takahama T, Sakai S (2006) Constrained optimization by the epsilon constrained differential evolution with gradient-based mutation and feasible elites. In: IEEE congress on evolutionary computation. Vancouver, BC, Canada, pp 308–315

Tang J, Lim M, Ong YS (2007) Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems. Soft Comput 11(9):873–888

Venkatraman S, Yen GG (2005) A generic framework for constrained optimization using genetic algorithms. IEEE Trans Evol Comput 9(4):424–435

Wang Y, Cai Z (2012a) Combining multiobjective optimization with differential evolution to solve constrained optimization problems. IEEE Trans Evol Comput 16(1):117–134

Wang Y, Cai Z (2012b) A dynamic hybrid framework for constrained evolutionary optimization. IEEE Trans Syst Man Cybern Part B Cybern 42(1):203–217

Wang H, Wang D, Yang S (2009) A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. Soft Comput 13(8-9):763–780

Wang H, Moon I, Yang S, Wang D (2012) A memetic particle swarm optimization algorithm for multimodal optimization problems. Inf Sci 197:38–52

Woldesenbet YG, Yen GG, Tessema BG (2009) Constraint handling in multiobjective evolutionary optimization. IEEE Trans Evol Comput 13(3):514–525

Zhou Y, Li Y, He J, Kang L (2003) Multiobjective and MGG evolutionary algorithm for constrained optimization. In: Proceedings of IEEE congress on evolutionary computation, pp 1–5

Zielinski K, Laur R (2006) Constrained single-objective optimization using particle swarm optimization. In: IEEE congress on evolutionary computation, Vancouver, BC, Canada, pp 443–450

Zitzler E, Laumannns M, Thiele L (2001) SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Proceedings of evolutionary methods Des optimization control application industrial problems (EUROGEN), pp 95–100