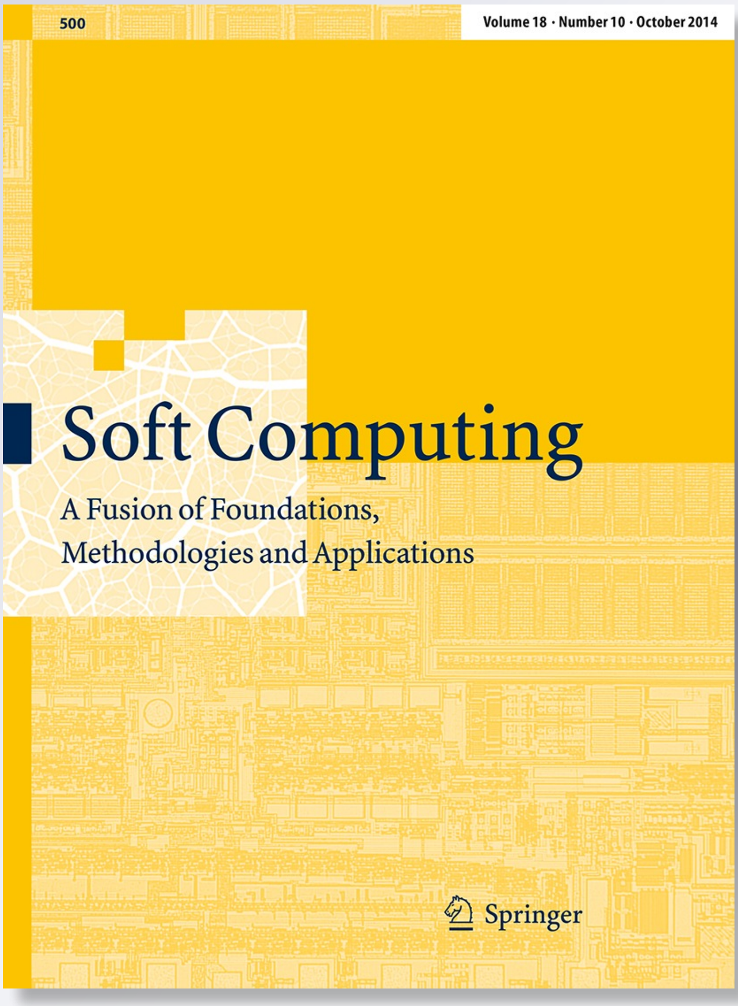
An improved memetic algorithm using ring neighborhood topology for constrained optimization

Zhenzhou Hu, Xinye Cai & Zhun Fan

Soft Computing
A Fusion of Foundations,
Methodologies and Applications

ISSN 1432-7643
Volume 18
Number 10

Soft Comput (2014) 18:2023-2041
DOI 10.1007/s00500-013-1183-7



Springer

Soft Comput (2014) 18:2023–2041
DOI 10.1007/s00500-013-1183-7

METHODOLOGIES AND APPLICATION

An improved memetic algorithm using ring neighborhood topology for constrained optimization

Zhenzhou Hu · Xinye Cai · Zhun Fan

Published online: 29 November 2013

Abstract This paper proposes an improved memetic algorithm relying on ring neighborhood topology for constrained optimization problems based on our previous work in Cai et al. (Soft Comput (in press), 2013). The main motivation of using ring neighborhood topology is to provide a good balance between effective exploration and efficient exploitation, which is a very important design issue for memetic algorithms. More specifically, a novel variant of invasive weed optimization (IWO) as the local refinement procedure is proposed in this paper. The proposed IWO variant adopts a neighborhood-based dispersal operator to achieve more fine-grained local search through the estimation of neighborhood fitness information relying on the ring neighborhood topology. Furthermore, a modified version of differential evolution (DE), known as "DE/current-to-best/1", is integrated into the improved memetic algorithm with the aim of providing a more effective exploration. Performance of the improved memetic algorithm has been comprehensively tested on 13 well-known benchmark test functions and four engineering constrained optimization problems. The experimental results show that the improved memetic algorithm obtains greater competitiveness when compared with the original memetic approach Cai et al. in (Soft Comput (in press), 2013) and other state-of-the-art algorithms. The effectiveness of the modification of each component in the proposed approach is also discussed in the paper.

Keywords Constrained optimization · Invasive weed optimization · Differential evolution · Ring neighborhood topology · Multi-objective optimization

Communicated by Z. Zhu.

Z. Hu · X. Cai (✉)
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Jiangsu, China
e-mail: xinye@nuaa.edu.cn; zzhu0612@gmail.com

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

1 Introduction

Many real-world optimization problems involve various types of constraints. Usually how to locate the optimal solutions with an accepted computational cost, while satisfying the constraints simultaneously, should be carefully taken into consideration. Such problems are even more difficult when they have huge search region but narrow feasible region. In general, constrained optimization problems (COPs), can be defined as follows.

minimize $f(\mathbf{x})$, subject to

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

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

$$l_k \leq x_k \leq u_k, \quad 1 \leq k \leq n \tag{2}$$

where l_k and u_k are the lower boundary and upper boundary for a decision variable x_k, respectively.

Generally, equality constraints are transformed into inequality form as follows.

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

where ϵ is an allowed positive tolerance value. Note that the maximization problems $f(\mathbf{x})$ can be converted into the equivalent minimization problems by adding the negative sign before $f(\mathbf{x})$.

Over the last decades, Memetic algorithms (MAs) have attracted a great amount of attention for tackling optimization problems due to their interesting characteristics through combining a population-based global search with one or more heuristic local refinement procedures. Therefore, MAs have been successfully applied to optimization problems, such as global optimization (Gong et al. 2011), multi-objective optimization (Ishibuchi et al. 2003), combinatorial optimization (Tang et al. 2007) and multi-modal optimization (Wang et al. 2012), etc. More recently, researchers started to tackle COPs in the frameworks of MAs. For example, Kelner et al. (2008) incorporated the local search strategy based on the interior point method into genetic algorithm to solve COPs. Nema et al. (2011) proposed a hybrid cooperative algorithm where particle swarm optimization and gradient search were integrated to balance exploration and exploitation for solving engineering constrained optimization optimization. Ullah et al. (2009), proposed a new agent based on memetic algorithm with four types of local search techniques adaptively selected in the evolution process. Sun et al. (2013) presented an intelligent multi-restart memetic framework for box constrained global optimisation. Handoko et al. (2010) proposed a novel feasibility structure modeling technique to effectively determine the choice of solutions for local refinements, by utilizing information gathered to model the feasibility structure of COPs in the framework of MAs. Wang and Cai (2012b), proposed a dynamic hybrid framework, which is able to implement global and local search dynamically according to the feasibility proportion. More comprehensive surveys of MAs can be found in Moscato (1989), Neri and Cotta (2012), Ong et al. (2010).

Besides the remarkable success of MAs in a wide range of application domains, adaptive forms of MAs have also attracted the increasing attention over the recent years. As surveyed in Chen et al. (2011), several core design issues need to be considered in the adaption of MAs, including the frequency of refinements, selection of individual subset to undergo refinement, intensity of refinement, and choice of procedures to conduct refinement. For instance, empirical experiments were conducted to investigate the impact of refinement frequency, selection of individual subset and intensity of refinement on MAs in Nguyen et al. (2007). A self-generating mechanism to adaptively provide various local search mechanisms used in MAs is presented in Krasnogor and Gustafson (2004). 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 COPs 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. Therefore, in Cai et al. (2013), we proposed a memetic algorithm (IWO_DE) that adopted invasive weed optimization (IWO) as the local search engine and deferential evolution (DE) as the global engine to tackle COPs. IWO is able to control the refinement frequency, selection of individual subset and intensity adaptively in different stages of evolution because it has two interesting characteristics as follows: (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.

Based on our previous work, we propose an improved version of IWO_DE approach (Cai et al. 2013) replying on the ring neighborhood topology as the population structure in this paper. The ring neighborhood topology had been investigated in Kennedy (1999); Kennedy et al. (2002) and found to be able to influence the search tendency of exploration and exploitation. In addition, it has been successfully applied to improve the performance of differential evolution for optimizing problems in Das et al. (2009). The proposed approach in this paper, IWO_DE with ring neighborhood topology, mainly focus on improving the previous IWO_DE approach (Cai et al. 2013) in the following two directions.

1. In IWO, offspring each weed generates are dispersed around their parent in the form of certain distribution(such as normal or polynomial distribution). The standard deviation of the distribution actually determines the dispersal degree of solutions. In the original IWO (Mehrabian and Lucas 2006) and our previous work (Cai et al. 2013), the dimensions of all solutions share the same dispersal degree, which causes a very coarse-grained local refinement. In order to further improve the local refinement ability of IWO, we propose a neighborhood-based dispersal operator through the estimation of neighborhood fitness information replying on the ring neighborhood topology. More specifically, the dispersal degree for different dimensions of each solution is determined by the fitness of this solution's neighborhood in the ring neighborhood topology. Additionally, diversity mutation (Wang et al. 2009) is integrated into the basic framework of IWO in order to maintain a diverse local search for IWO as the local refinement procedure.
2. The original global search algorithm, known as "DE/current-to-best/1" (Das and Suganthan 2011), may cause rapid convergence in the search process (Das et al. 2009). Thus in this paper, a modified version of "DE/current-to-best/1" using the ring neighborhood topology is also adopted to improve the global search ability of the original version in the hope of achieving an effective exploration.

In summary, the framework of our previous work Cai et al. (2013) had proposed the IWO_DE approach,which combined adaptive characteristics of IWO and the global search ability of DE to handle COPs. Based on it, an improved memetic algorithm replying on the ring neighborhood topology is proposed to solve COPs in this paper. For convenience, the improved memetic algorithm is denoted as IWO_DE/Ring.

The main contributions of this paper can be summarized as follows.

1. A novel variant of IWO is proposed relying on the ring neighborhood topology. More specifically, a neighborhood-based dispersal operator is employed firstly to determine the degree of local refinement for different dimensions of each solution through the estimation of the solution's neighborhood fitness.
2. For the global search procedure, the "DE/current-to-best/1", which has been modified to better cater to the characteristic of exploration, is integrated into the improved memetic algorithm with the ring neighborhood topology.
3. The performance of the proposed IWO_DE/Ring approach, both in terms of speed of convergence and optimality, has been tested on 13 well-known benchmark functions and four engineering COPs. The experimental results show the proposed IWO_DE/Ring is very competitive compared with the original IWO_DE (Cai et al. 2013), as well as some other state-of-the-art algorithms.

The rest of this paper is organized as follows. Since the proposed memetic algorithm is applied for COPs, Sect. 2 reviews works on constraint-handling techniques and the descriptions of IWO and DE are briefly reviewed in Sect. 3. In Sect. 4, the ring neighborhood topology adopted in this paper is defined and introduced. Section 5 elaborates the improved memetic algorithm, IWO_DE/Ring, in detail. The experimental results of IWO_DE/Ring on benchmark functions and engineering optimization problems are presented in Sect. 6. Section 7 further discusses and analyzes the performance of IWO_DE/Ring. Finally, Sect. 8 concludes this paper.

2 Related works on constraint-handling techniques

Unlike unconstrained optimization problems, both the objective function to be optimized and constraint satisfaction should be considered when solving COPs. Up to now, different constraint handling techniques have been incorporated with genetic algorithm, differential evolution and particle swarm optimization etc. to tackle COPs (Wang et al. 2009; Tasgetiren and Suganthan 2006; He and Wang 2007). More details of constraint handling techniques have been surveyed in Coello (2002), Mezura-Montes and Coello (2011). Generally, these techniques can be categorized into several classes, which are (1) techniques of penalty functions; (2) techniques of special representations and operators; (3) techniques of multi-objective optimization and (4) techniques of hybrid methods. We will introduce them one by one as follows.

Techniques of penalty functions employing penalty functions is a simple and common approach to solve COPs. The main principal behind is to transform COPs into unconstrained ones through adding a penalty factor to the fitness value of infeasible solutions. However, the disadvantage of using penalty function methods is that the value of penalty factors is usually set up through "trial and error". Penalty factors ususally need to be carefully tuned. Because under- and over-penalization factors can influence the optimal results considerably and usually these penalty factors are problems depended. For instance, Coello (2000) proposed a self-adaptive penalty approach based on the concept of co-evolution under the genetic algorithm framework. The method created two populations that cooperate with each other in such a way that one population evolves penalty factors to be used by another population which focuses on the objective function values. Woldesenbet et al. (2009) also proposed a self-adaptive penalty approach using evolutionary algorithm. In this method, the percentage of feasible solutions plays a significant role in determining the degree of penalty added to infeasible solutions. This method introduced a modified objective function values composed of two components: distance measure and adaptive penalty. More recently, Lin (2013) proposed a novel penalty genetic algorithm based on the rough set theory, which is able to provide an self-adaptive penalty adjustment in the evolution process.

Techniques of special representations and operators except for adopting penalty function approaches, other special representations and operators have been proposed. For example, Runarsson and Yao (2000) proposed a stochastic ranking (SR) method to tackle COPs. SR used a probability parameter p_f as the comparison criterion among individuals, namely (1) if individuals are both feasible, the one with better fitness is selected; or (2) if a uniformly random number within 0 and 1 is less than p_f, the one with better fitness is selected; otherwise, (3) the one which has the small amount of constraint violation is preferable. Besides, SR adopts a dubble-sort-like procedure to achieve the above process. Later, Takahama and Sakai (2006) proposed a ε constrained method in which COPs were transformed into unconstrained ones by defining an order relation under the ε level comparison and ε was controlled by an exponential function and for any ε, its value is greater than zero. The order relation is relevant to the objective function value and the constraint violation. Specifically, assume $\phi_1(x)$ and $\phi_2(x)$ to be the sum amount of constraint violation corresponding to individuals x_1 and

x_2, respectively: (1) if $\phi_1(x),\ \phi_2(x)\ \geq \varepsilon$, then the individual which has better objective function value is preferable; (2) if $\phi_1(x) = \phi_2(x)$, then also the one which has better objective function value is preferable; and otherwise, (3) the one which has small violation of constraint will be selected.

Techniques of multiobjective optimization adopting multiobjective optimization techniques is another way to tackle COPs. Through multiobjective optimization techniques, constraints can be considered as one or more objectives, and thus COPs can be converted into multi-objective unconstrained optimization problems. Wang and Cai (2012a) proposed an algorithm, in which combined the multiobjective optimization and differential evolution to solve the COPs, to overcome the shortcoming of the method (Wang et al. 2007) which also adopted multiobjective optimization for the comparison of individuals. Similarly, Wang and Cai (2008) propose an adaptive model to solve COPs. In this method, the adaptive model can tackle the COPs adaptively in different phase and in the phase that only had infeasible solutions, the multiobjective optimization technique was employed for the comparison of infeasible solutions. Coello and Mezura Montes (2002) proposed a genetic algorithm in which dominance-based tournament selection is used to determine which the infeasible solutions were selected. Venkatraman and Yen (2005) proposed a generic framework which comprised two phases. In the first phase, the goal preferred to find at least one feasible solution and the comparison among individuals only depended on the sum amount of constraint violation. In the second phase, COPs were converted into a bi-objective unconstrained optimization problems and then both objectives (the original objective and the sum amount of constraint violation) were optimized and ranked by the non-dominated sorting which was proposed in Deb et al. (2002).

In this paper, we employ the multi-objective optimization technique to solve COPs. The objective to be optimized and the constraint satisfaction are transformed to the two objectives of a bi-objective optimization problem, which is redefined as follows.

minimize

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

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

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

Based on the above redefinition, this paper considers COPs as a biobjective optimization problem, that is, one objective is the original objective function $f(\mathbf{x})$ and the other is the total amount of constraint violation $G(\mathbf{x})$.

Unlike single-objective optimization, multi-objective optimization usually resorts to the concept of Pareto optimality. Since the multi-objective method is used to handle COPs in this paper, several basic concepts, such as Pareto optimality, in the context of multi-objective optimization need to be introduced as follows.

1. *Pareto dominance*—a vector $\mathbf{x}_i^1$ is said to be Pareto dominance another vector $\mathbf{x}_i^2$ (denoted by $\mathbf{x}_i^1 \prec \mathbf{x}_i^2$), if and only if

$$\forall i \in \{1, 2, \ldots, n\}, \mathbf{x}_i^1 \leq \mathbf{x}_i^2 \wedge \exists i \in \{1, 2, \ldots, n\}, \mathbf{x}_i^1 < \mathbf{x}_i^2$$

2. *Pareto optimality*—a vector $\mathbf{x}_i^1$ is said to be Pareto optimality if and only if

$$\neg \mathbf{x}_i^2, \mathbf{f}(\mathbf{x}_i^2) = (f_1^2(\mathbf{x}), f_2^2(\mathbf{x})) \prec \mathbf{f}(\mathbf{x}_i^1) = (f_1^1(\mathbf{x}), f_2^1(\mathbf{x}))$$

3. *Pareto optimal set*—The Pareto optimal set **PS** is defined as

$$\mathbf{PS} = \{\mathbf{x}_i^1 | \neg \mathbf{x}_i^2, \mathbf{f}(\mathbf{x}_i^2) \prec \mathbf{f}(\mathbf{x}_i^1)\}$$

4. *Pareto optimal front*—The Pareto optimal front **PF** is defined as

$$\mathbf{PF} = \{\mathbf{f}(\mathbf{x}_i) | \mathbf{x}_i \in \mathbf{PS}\}$$

In addition, the vectors in **PS** are called as non-dominated vectors and the schematic diagram of Pareto optimal front when solving COPs under multiobjective optimization technique is plotted in Fig. 1. In Fig. 1, $f(x)$ is the objective function value to be optimized and $G(x)$ is the total amount of constraint violation. The feasible optimal solution is mapped on the intersection between the line of Pareto optimal front and feasible solutions.

Techniques of hybrid methods hybrid methods have received considerable focus over recent years. For hybrid methods, the intriguing characteristics of two or more methods are merged to cope with COPs. Kelner et al. (2008)

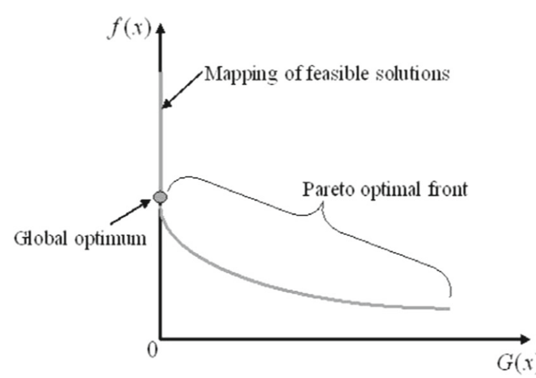


Fig. 1 Pareto optimal front when solving COPs with multiobjective optimization technique

proposed a hybrid optimization technique. In this method, a local search strategy based on the interior point method was hybridized into a genetic algorithm to solve COPs. Wang et al. (2009) proposed a hybrid algorithm in which an adaptive constraint-handling technique was incorporated into evolutionary algorithm to handle COPs. Similarly, Nema et al. (2011) proposed a hybrid cooperative algorithm where particle swarm optimization and gradient search were integrated to balance exploration and exploitation for solving engineering constrained optimization optimization. Accordingly, MAs can be classified into the name of hybrid algorithms and considered as the hybridization between population-based global search and local refinement procedures. 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 in the evolution process. Sun et al. (2013) presented an intelligent multi-restart memetic framework for box constrained global optimisation, in which an estimation of distribution algorithm (EDA) combined with a derivative free local optimizer was developed. Based on the proposed framework, an adaptive multivariate model was proposed with the end to sample offspring. Besides, A derivative-free local optimization algorithm was employed to refine the current best solutions. Handoko et al. (2010) proposed a novel feasibility structure modeling technique to effectively determine the choice of solutions for local refinements, by utilizing information gathered to model the feasibility structure of COPs in the framework of MAs. Wang et al. (2012b) proposed a dynamic hybrid framework, which is able to implement global and local search dynamically according to the feasibility proportion.

Since our proposed approach is an improved version under the framework of IWO_DE, the following section is dedicated to the review of IWO and DE, respectively.

3 Review of IWO and DE

3.1 Invasive weed optimization

IWO proposed by Mehrabian and lucas (2006) is a novel derivative-free and metaheuristic algorithm that mimics the ecological behavior of weeds colonization and dispersion. Subsequently, Kundu et al. (2011) proposed a IWO variant that extends the original IWO to tackle multi-objective optimization problems and Roy et al. (2013) combined IWO with localized group search optimizers to solve multimodal optimization problems.

Generally, there are four steps in IWO.

1. *Initialize a population* initialize and disperse solutions within the given n dimensional search space uniformly and randomly.
2. *Reproduction* each individual of the population reproduces seeds depending on its own fitness, the population's lowest and highest fitness. Under 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* the seeds are randomly dispersed, around each weed, over the n dimensional search space by normally distributed random numbers with mean equal to zero; but varying variance. Under this operation, seeds are dispersed around their parent individual and thus the colony of weeds is formed to enhance the search efficiency. Furthermore, standard deviation (sd) of the normal distribution varies from a predefined initial value, sd_{max}, to a predefined 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 maximal number of generations, $iter$ is the current number of generation and m is the nonlinear modulation index.
4. *Competitive exclusion* with passing several generation and the growth and reproduction of weeds, the number of individuals in a colony will reach the allowed maximum. Therefore, an essential exclusion mechanism is needed to eliminate undesire ones. The exclusion mechanism is adopted to eliminate weeds with low fitness and selects good weeds. Subsequently, the selected ones will be preserved into the next generation and then the steps 1–4 are repeated until satisfactory condition is reached.

3.2 Differential evolution

It is known that DE is a simple and powerful stochastic real-parameter global optimization algorithm (Price et al. 2005) and since its occurrence in 1995 (Storn and Price 1995), DE has drawn much attention on many researchers due to its excellent and efficient performance, which results in a number of improved variants of the original version DE (Das and Suganthan 2011). Furthermore, an empirical study on the COPs by using DE is presented in Mezura-Montes et al. (2010).

Generally, DE comprises N individuals and every individual is an n-dimensional vectors $\mathbf{x}_i = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ that are randomly generated in the search space. Subsequently, the operations of mutation, crossover and selection are executed in the process of evolution.

1. *Mutation operation* with different mutant strategies, the form when generating a mutant vector $\mathbf{v}_i$ is various. The known 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 selected uniformly and randomly within [1, N], are not equal to each other and all different from the index i. $\mathbf{x}_{best}$ is the best individual of the entire current population, F is a scaling factor that measure the scale of the difference between vectors.
2. *Crossover operation* after obtaining the mutant vector $\mathbf{v}_i$, the trial vector $\mathbf{u}_i$ is generated by binomial crossover as follows.

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

   where $\mathbf{x}_i$ is the target vector and $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, n$, 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].
3. *Selection operation* the generated trial vector $\mathbf{u}_i$ is compared with the target vector $\mathbf{x}_i$ to determine whether preserved or not. The selection operation is described as follows.

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

4 Ring neighborhood topology

It has been shown that the topology structure of population could influence the search tendency of exploration and exploitation (Kennedy 1999; Kennedy et al. 2002). The ring neighborhood topology, as one of the most common topology structures, has been successfully applied to handle optimization problems (Das et al. 2009; Li 2010; Omran et al. 2006). Therefore, this paper adopts the ring neighborhood topology as the population structure to determine the neighborhood of individuals, and to further improve IWO_DE approach in Cai et al. (2013). The ring neighborhood topology can be described as follows.

Suppose the population $P = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ where $\mathbf{x}_i$ $(i = 1, 2, \ldots, N)$ is the individuals of population. Conveniently, we organize the individuals of population to be the ring neighborhood topology with respect to their indices as presented in Das et al. (2009) and the ring neighborhood topology used in this paper has been exhibited in Fig. 2. Simply, each individual's neighborhood can be its immediate member on its left and right in the ring neighborhood as shown in Fig. 2. Specifically, the individuals $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ are the immediate neighbors of $\mathbf{x}_i$.

Besides, if the neighborhood radius of the individual $\mathbf{x}_i$ is set to 2, the neighborhood of $\mathbf{x}_i$ are $\mathbf{x}_{i-2}$, $\mathbf{x}_{i-1}$, $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+2}$ and then the neighborhood of $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are overlapped as shown in Fig. 3. Specifically, as for the modified

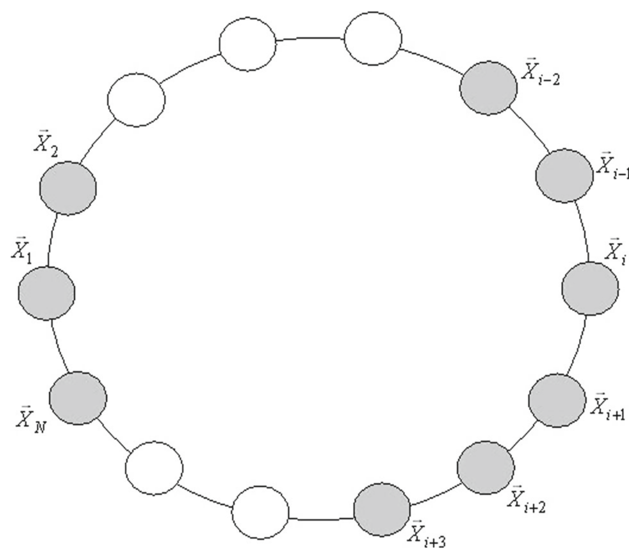


Fig. 2 Ring neighborhood topology

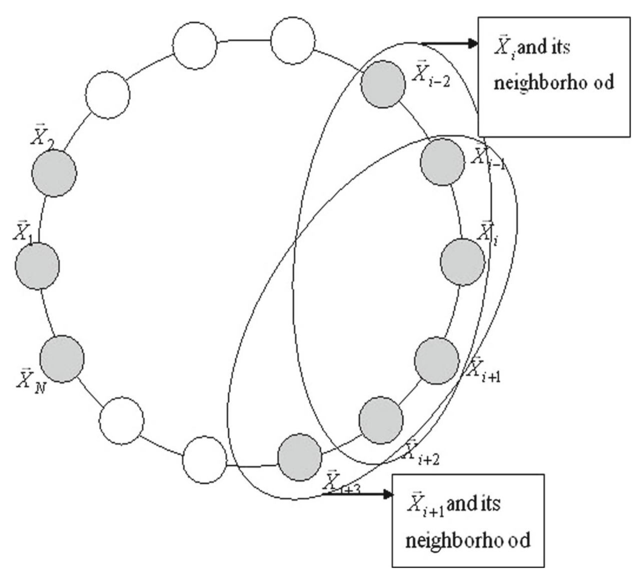


Fig. 3 Ring neighborhood topology with overlapping of neighborhood

version of "DE/current-to-best/1" in this paper, the "best" represents the best individual among $\mathbf{x}_i$ and its neighborhood $\mathbf{x}_{i-2}$, $\mathbf{x}_{i-1}$, $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+2}$. Meanwhile, as for the novel IWO variant in this paper, the neighborhood fitness information of each individual will contribute to the degree of local refinement, that is, with respect to $\mathbf{x}_i$, the fitness information of $\mathbf{x}_{i-2}$, $\mathbf{x}_{i-1}$, $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+2}$ will be employed to determine the local refinement degree of $\mathbf{x}_i$. Note that in this paper the ring neighborhood topology is predefined before the search process and organized on the set of the indices of individuals in the population.

5 The proposed algorithm: IWO_DE/Ring

In this section, we elaborate the improved memetic algorithm in details.

5.1 The novel variant of IWO

5.1.1 An adaptive weighted sum fitness assignment

In IWO, higher fitness of a weed indicates more offspring to produce for it. The number of offspring usually reflects the ability of reproduction for each weed.

When addressing COPs, the feasible solutions should be preferred in the search process. However, by this way great amount of computational resources may also be wasted on the undesirable feasible solutions with very bad objective function values. On the other hand, infeasible solutions with smaller degree of constraint violation and good objective function value should also be preferred, as they are very likely to guide the local search towards the optimal solutions.

Therefore, in order to balance between the feasibility and objective function for the local refinement of IWO, this paper keeps to use the adaptive fitness assignment mechanism presented in Cai et al. (2013) to balance between the feasibility and objective function and determine the number of offspring each weed generates.

The form of adaptive weighted sum fitness assignment Cai et al. (2013) is described as follows.

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

and

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

where weight factor ω is the percentage of feasible solutions. $f'(\mathbf{x})$ 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, as presented below.

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

Therefore, the number of seeds reproduced by a weed is defined below.

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

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} and S_{min} denote the permissible maximal and minimal number of seed respectively. In addition, f_i is the normalized fitness function and the better f_i (without loss of generality, in terms of a minimization problem) of one weed is, the more number of seeds it generates.

5.1.2 A neighborhood-based dispersal operator

In the original IWO (Mehrabian and Lucas 2006), offspring each weed (solution) generates are dispersed around their parent in the form of normal distribution. The standard deviation of the distribution is usually considered as a dispersal parameter that is able to control the dispersal degree of each weed's offspring. Similarly in our previous work (Cai et al. 2013), offspring each weed generates are dispersed around their parent in the form of polynomial distribution. However, both original IWO and our previous work adopt one single dispersal parameter, which means all solutions have the same dispersal degree in all dimensions at a certain iteration. In other words, the original dispersal process in IWO leads to a very coarse-grained local search around the weed and it did not considered the actual local landscape around the weed to adaptively allocate different dispersal degree for different dimensions of different solutions. Thus a new dispersal operator with more powerful adaptive ability comes to the necessity, especially for COPs, when various variables usually have different boundary constraints.

The proposed neighborhood-based dispersal operator is achieved by the estimation of the neighborhood information around each weed (parent solution) to determine the dispersal degree. The "neighborhood" is defined under the ring neighborhood topology detailed in Sect. 4. The formula of the neighborhood-based dispersal operator is given as follows.

$$\mathbf{sd}_i = \mathbf{abs}\left(\left(\frac{\sum_{j=1}^{n_i}(\mathbf{x}_j - \mathbf{x}_i) * f_j}{\sum_{j=1}^{n_i} f_j}\right) / n_i\right) \tag{14}$$

where $\mathbf{abs}(.)$ denotes operation of computing absolute value, $\mathbf{x}_i$ is the i-th weed and $\mathbf{x}_j$ is the j-th neighborhood of $\mathbf{x}_i$. Besides, n_i is the number of neighborhood $\mathbf{x}_i$ has and f_j computed by the formula (13) is fitness of the neighborhood $\mathbf{x}_j$.

It can be seen from the formula (14) that the standard deviation vector $\mathbf{sd}_i$ (dispersal degree for a weed (parent solution) i) is actually determined by the fitness distribution around this very parent solution i, which can be computed as the distances (in the variable space) between the parent solution i and its neighborhood weighted by the neighborhood fitness. In this way, all the neighborhood information of one parent solution has been incorporated into the calculation of the standard deviation. Unlike the formula (6) in Sect. 3.1, where a preset initial value sd_{max} and a final value sd_{min} for standard deviation are required as two parameters, the dispersal degree is adaptively generated for different dimensions of each solution by making use of the neighborhood information, as shown in formula (14). Thus no preset parameters are needed.

5.1.3 Exclusive mechanism of IWO

After the above operation, offspring are reproduced and dispersed in the population. With generations continuing, exclusive mechanism should be employed when the population size reaches the permissible maximum. This paper adopted nondominated sorting (Deb et al. 2002), which had been employed as well in Cai et al. (2013), as the exclusive operator for eliminating the undesired individuals. Using non-dominated sorting algorithm, each individual is allocated to a non-dominated front. Hence, the exclusion mechanism of IWO is presented as follows.

1. if individuals belong to different non-dominated front, then the individuals with lower non-dominated front are better;
2. if individuals hold the same non-dominated front, then the ones having smaller constraint violation are selected.

With the exclusive mechanism, individuals which are competitive both in terms of the objective value and the amount of constraint violation are selected and to be preserved into the next generation.

Except for the above modification of original IWO, diversity mutation proposed in Wang et al. (2009) is employed and integrated into the framework of IWO for the sake of maintaining the diversity of population in IWO.

Diversity mutation the purpose of diversity mutation presented in Wang et al. (2009) is to facilitate a high diversity in the population and the descriptive form of diversity mutation is following.

$$\mathbf{x}_{i,j} = \begin{cases} \mathbf{l}_{i,j} + \beta(\mathbf{u}_{i,j} - \mathbf{l}_{i,j}), & \text{if } j = j_{rand} \\ \mathbf{x}_{i,j}, & \text{otherwise} \end{cases} \tag{15}$$

where $j = 1, 2, \ldots, n$ and a integer j_{rand} is randomly generated between 1 and n with the probability $1/n$. In addition, $\mathbf{l}_i$ and $\mathbf{u}_i$ are real-valued vectors and β is uniformly and randomly generated within [0, 1]. In this paper, $\mathbf{l}_i$ and $\mathbf{u}_i$ are the boundary vectors of $\mathbf{x}_i$.

Algorithm description of the IWO variant is presented in *Algorithm* 1.

Algorithm 1 Procedures of the IWO variant

step 1: input *N* parent weeds, denoted as *P*;
step 2: compute the standard deviation of each weed sd_i, $i = 1, 2, \ldots, N$ by the formula (14);
step 3: *R=ND*(*P*,sd_i);
/* *ND* is the normal distribution function that is acted as the spatial dispersal function of IWO */
step 4: *R_m=DM*(*R*);
/* *DM* is the diversity mutation operation to maintain the diversity of IWO*/
step 5: $P\_R = P \cup R\_m$;
step 6: **If** the size of $P\_R \geq P_{max}$ **Then**
/* P_{max} denotes the permissible maximum of population*/
step 7: *P=Select*(*P_R*)
/* execute exclusive mechanism presented in Section 5.1.3 to select the better individuals*/;
step 8: **end If**

5.2 The modified version of "DE/current-to-best/1"

From the review of DE in Sect. 3.2, there are several popular DE variants. In this paper, we incorporate the DE variant known as "DE/current-to-best/1" into the proposed memetic algorithm to explore the search space effectively.

In the original version of "DE/current-to-best/1" (Das and Suganthan 2011), the "best" denotes the best individual of entire population, with which this original version provides more exploitation and less exploration because individuals can be attracted towards the best individual of entire population very quickly (Das et al. 2009). Thus, motivated by Das et al. (2009), we present a modified version of "DE/current-to-best/1" using the the ring neighborhood topology. In our modified version of "DE/current-to-best/1", the "best" represents the best individual among an individual and its neighborhood, by which this modified version is expected to lead to an effective exploration. Next, we describe the modified version of "DE/current-to-best/1" as follows.

$$\mathbf{v}_i = \mathbf{x}_i + F * (\mathbf{x}_{i_{best}} - \mathbf{x}_i) + F * (\mathbf{x}_{r1} - \mathbf{x}_{r2}) \tag{16}$$

where $\mathbf{x}_{i_{best}}$ is the best individual among $\mathbf{x}_i$ and its neighborhood.

With the formula (16) and (7), a trial vector $\mathbf{u}_i$ is generated and is compared with $\mathbf{x}_i$ to determine whether it replaces the $\mathbf{x}_i$ or not. Next, we describe the selection mechanism of the DE variant as follows.

1. if the generated trial vector $\mathbf{u}_i$ is a feasible solution and is superior to all the feasible solutions, then the

worst feasible solution is replaced by the generated trial vector;

2. if the generated trial vector $\mathbf{u}_i$ is an infeasible solution, then all the infeasible solutions will be compared with $\mathbf{u}_i$ under the concept of pareto dominance, and if there are no infeasible solutions that dominates $\mathbf{u}_i$, then select the infeasible solutions that is dominated by $\mathbf{u}_i$; and then find the solution in the selected ones which has the maximal amount of constraint violation and is replaced by $\mathbf{u}_i$; but if there are no infeasible solutions that is dominated by $\mathbf{u}_i$, then find the solution in all the infeasible ones which has the maximal amount of constraint violation and is replaced by $\mathbf{u}_i$.

Under the selection mechanism, feasible solutions and infeasible solutions which have the better objective function value and low amount of constraint violation respectively are preferable.

Algorithm description of the DE variant is presented in *Algorithm* 2.

Algorithm 2 Procedures of the modified version of "DE/current-to-best/1"

step 1: input *NP* individuals, denoted as *P*;
step 2: **for** i=1:*NP* **do**
step 3: randomly select different subscript r_1, r_2 within [1, *NP*]-i-i_{best};
/* i_{best} is the subscript of the best individual among $\mathbf{x}_i$ and its neighborhood*/
step 4: v_i=*mutation*(P_{r_1,r_2}, F);
/* use the formula (16) to generate the mutant vector */
step 5: u_i=*crossover*(v_i, P_i, C_r);
/* use the formula (7) to generate the trial vector */
step 6: determine whether x_i is replaced by u_i or not according to the selection mechanism, as presented in the Section 5.2;
step 7: **end for**

Finally, with the above detailed descriptions, we combine the novel IWO variant with the modified version of "DE/current-to-best/1" for the aim of exerting the efficient exploitation and the effective exploration to tackle COPs and the entire process of the proposed memetic algorithm, denoted as IWO_DE/Ring, is presented in *Algorithm* 3.

Algorithm 3 The entire procedures of IWO_DE/Ring

step 1: t=1;
step 2: initialize population P_0;
step 3: organize the population to be a ring neighborhood by virtue of the set of indices of individuals in the population;
step 4: F_0=Evaluate(P_0);
step 5: **while** terminal condition is false **do**
step 6: execute *Algorithm* 1 to generate population P_t;
step 7: F_t= Evaluate(P_t);
step 8: execute *Algorithm* 2 to generate population P_t again;
step 9: F_t= Evaluate(P_t);
step 10: t=t+1;
step 11: **end while**

Table 1 The parameter values of IWO_DE/Ring

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
β	The parameter of diversity mutation	Between 0 and 1

6 Experimental results

6.1 Experimental setup

The proposed memetic algorithm IWO_DE/Ring is performed on 13 well-known benchmark test functions that are taken from Liang et al. (2006) and four engineering constrained optimization problems that are presented in Aguirre et al. (2007), Cagnina et al. (2008). Performance of IWO_DE/Ring is compared with several state-of-the-art constrained optimization algorithms.

We execute 25 independent runs on IWO_DE/Ring under the maximal 200,000 function evaluations (FEs) and the tolerance value ϵ in formula (3) is set to 0. Furthermore, IWO_DE/Ring has several parameters. For the modified version of "DE/current-to-best/1", the scaling factor F, and the crossover probability parameter C_r. For the novel IWO variant, the initial and maximal number of population, the minimal and maximal number of seeds, the parameter β of diversity mutation. Details of the parameter values are presented in Table 1. Additionally, the neighborhood radius of individuals is defined as 3, that is, the neighborhood size of each individual is 6, as discussed and suggested in Das et al. (2009).

6.2 Results on benchmark test functions

First we introduce and summarize characteristics of the 13 well-known benchmark test functions in Table 2.

It is very clear that Table 2 contains various types of test functions, such as quadratic, nonlinear, polynomial, cubic and linear. Besides, the test functions have various number of decision variables n and constraints and also have different types of constraints, including linear equality constraints (LE), linear inequality constraints (LI), nonlinear equality constraints (NE) and nonlinear inequality constraints (NI). Additionally, ρ is the estimated percentage of the feasible space out of the whole search space and a is the number of active constraints at the best known optimal solution.

According to experimental setup presented in Sect. 6.1, we report the results of IWO_DE/Ring on benchmark test

Table 2 Characteristics of the 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

functions in Table 3. It can be seen from Table 3 that the best results obtained by IWO_DE/Ring is very approximate to the known optimal results. In addition, IWO_DE/Ring can find the best optimal result consistently on test functions over 25 runs except for test function g02. With respect to g02, IWO_DE/Ring cannot find the best result consistently but we can notice that the mean result is very close to the known optimal result. Besides, the standard deviation of most test functions is very small. These observations all indicate that the performance of IWO_DE/Ring is stable and robust when handling these benchmark test functions.

It is noteworthy to mention that equality constraints have not been converted into inequality constraints in this paper as we set the tolerance value in formula (3) into 0, which explains why the best optimal results of test functions with the equality constraints are slightly different from what are presented in Liang et al. (2006).

6.3 Convergence analysis on test functions

In this section, we present the convergence graphs of test functions, which is suggested in Liang et al. (2006), to visualize the convergence rate of test functions under our proposed approach. Two axes of the convergence graphs show $log_{10}(f(\mathbf{x}) - f(\mathbf{x}^*))$ vs. FEs, where $\mathbf{x}$ is the best result obtained after a certain number of FEs and $\mathbf{x}^*$ is the known optimal result. Noteworthily, the results that satisfy $f(\mathbf{x}) - f(\mathbf{x}^*) \leq 0$ are not plotted here simply because the logarithmic function is inapplicable to zero or negative value.

Clearly, we can observe from Figs. 4, 5, 6, 7 that the convergence to the known optimal result of test functions is very fast and most test functions have been converged after around 1×10^5 FEs.

6.4 Performance comparison on benchmark test functions

6.4.1 Comparison with several state-of-the-art algorithms

Comparisons are carried out between IWO_DE/Ring and five state-of-the-art constrained optimization algorithms to fur-

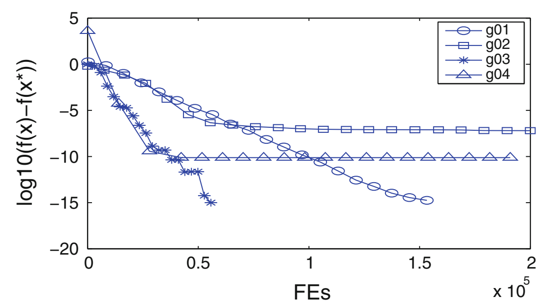


Fig. 4 Convergence graph for g05, g06, and g07

Table 3 Results of IWO_DE/Ring on benchmark test functions

f	Optimal	Best	Median	Mean	Worst	Std. dev.
g01	−15.0000000000	−15.0000000000	−15.0000000000	−15.0000000000	−15.0000000000	1.2E−15
g02	−0.8036191042	−0.8036190923	−0.7881153288	−0.7880207077	−0.7506732926	1.5E−02
g03	−1.0000000000	−1.0000000000	−1.0000000000	−1.0000000000	−1.0000000000	2.0E−16
g04	−30,665.5386717834	−30,665.53867178332	−30,665.53867178332	−30,665.53867178332	−30,665.53867178332	3.7E−12
g05	5,126.4981095952	5,126.4981095953	5,126.4981095953	5,126.4981095953	5,126.4981095953	1.3E−12
g06	−6,961.8138755802	−6,961.8138755802	−6,961.8138755802	−6,961.8138755802	−6,961.8138755802	0.0E+00
g07	24.3062090681	24.3062090682	24.3062090682	24.3062090682	24.3062090684	5.1E−11
g08	−0.0958250415	−0.0958250414	−0.0958250414	−0.0958250414	−0.0958250410	9.1E−11
g09	680.6300573745	680.6300573744	680.6300573744	680.6300573744	680.6300573744	4.1E−13
g10	7,049.2480205286	7,049.2480205287	7,049.2480205288	7,049.2480205371	7,049.2480205989	1.8E−08
g11	0.7500000000	0.7500000000	0.7500000000	0.7500000000	0.7500000000	0.0E+00
g12	−1.0000000000	−1.0000000000	−1.0000000000	−1.0000000000	−1.0000000000	1.1E−11
g13	0.0539498477	0.0539498478	0.0539498478	0.0539498478	0.0539498478	1.6E−17

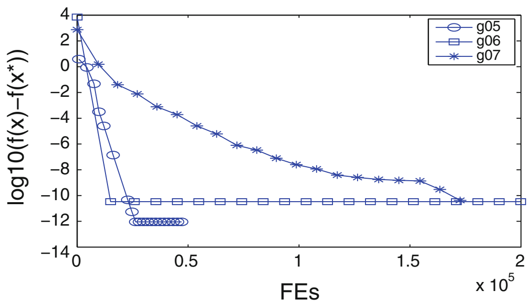


Fig. 5 Convergence graph for g05, g06, and g07

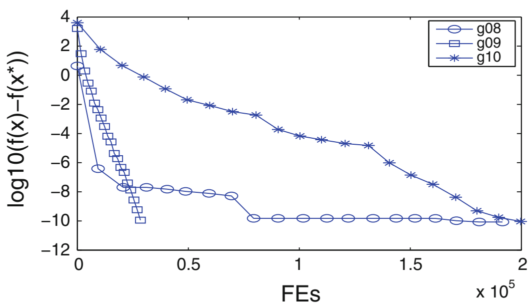


Fig. 6 Convergence graph for g08, g09 and g10

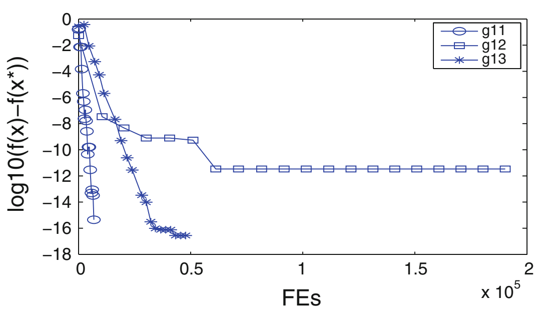


Fig. 7 Convergence graph for g11, g12, and g13

ther verify the efficiency of IWO_DE/Ring. These compared algorithms include a modified artificial bee colony algorithm (Karaboga and Akay 2011), a penalty genetic algorithm based on rough set theory (Lin 2013), an electromagnetism-like mechanism algorithm (Zhang et al. 2013), an agent-based memetic algorithm (Ullah et al. 2009), and a biogeography-based optimization algorithm (Boussaid et al. 2012). For convenience, we denote these compared algorithms as MABC (Karaboga and Akay 2011), RPEA (Lin 2013), ICEM (Zhang et al. 2013), AMA (Ullah et al. 2009) and CBBO-DM (Boussaid et al. 2012), respectively.

The comparative results among algorithms have been listed in Table 4, where results of the five compared algorithms are taken from relevant literatures. From Table 4, IWO_DE/Ring has the nearly equivalent ability of solving g01, g03, ,g04, g08, g11 and g12 when compared against the five algorithms. However, the best result of IWO_DE/Ring is better than four compared algorithms for g02 and equal to that of ICEM, but the mean result of IWO_DE/Ring for g02 is worse than these compared algorithms. For g05, the best result of IWO_DE/Ring is superior to that of AMA and RPGA, and the mean result of IWO_DE/Ring is better than AMA, RPGA, and MABC while obtaining the same mean result compared with CBBO-DM. For g06, the performance of IWO_DE/Ring on the best result is only better than AMA and better than AMA and RPGA on the mean result. IWO_DE/Ring outperforms AMA, CBBO-DM, RPGA and MABC on both the best and mean result for g07 and g10. The difference on performance for g09 between IWO_DE/Ring and the compared algorithms is small except for ICEM but the result of standard deviation gives IWO_DE/Ring more superiority. Finally, for g13, IWO_DE/Ring is better than MABC on both the best and mean result. Meanwhile, IWO_DE/Ring is also superior to AMA, ICEM on the mean result for g13.

It is important to mention that IWO_DE/Ring tackles these test functions under the maximal 200,000 FEs but the FEs of AMA, CBBO-DM, RPEA, ICEM and MABC is 350,000, 350,000, 350,000, 350,000 and 240,000 respectively. Furthermore, COPs with equality constraints are usually considered more difficult to solve as feasible regions of such problems are usually very small compared with the whole search space. One common method to handle COPs with equality constraints is to convert the inequality constraints to equality constraints constraints by using a small tolerance value, as shown in formula (3). In this paper, however, the equality constraints have not been converted into inequality constraints, as the tolerance value ϵ in formula (3) is set to 0. Meanwhile, the tolerance value ϵ in formula (3) was set to different small numbers in all compared algorithm before solving these COPs, which means the compared algorithms have already been given some privilege even before the experiments. Under these circumstances, we believe the efficiency of IWO_DE/Ring is very competitive compared with all other five algorithms.

6.4.2 Comparison with the previous work (Cai et al. 2013)

In this section, comparison is conducted between the proposed IWO_DE/Ring and the original IWO_DE (Cai et al. 2013) to further demonstrate the performance of IWO_DE/Ring. We adopt the performance criteria, namely success performance which was suggested in Liang et al. (2006), to evaluate performance between IWO_DE/Ring and IWO_DE. However, equality constraints had been converted into inequality constraints in the original IWO_DE, by setting the tolerance value ϵ in formula (3) to 0.0001, which made them easier to handle. To conduct a fair comparison, we run IWO_DE/Ring again over COPs with equality constraints (g03, g05, g11 and g13), but this time, the tolerance value ϵ is set to 0.0001.

Table 4 Comparison among algorithms on benchmark test functions

f		AMA	CBBO-DM	RPGA	ICEM	MABC	IWO-DE/Ring
g01	Best	−15.000	−15.000	−15.000	−15.000	−15.000	−15.000
	Mean	−15.000	−15.000	−15.000	−15.000	−15.000	−15.000
	Std. dev.	0.0E+00	8.2E−14	0.0E+00	0.0E+00	0.0E+00	1.2E−15
g02	Best	−0.803549	−0.803557	−0.803612	−0.803619	−0.803598	−0.803619
	Mean	−0.803500	−0.802774	−0.794453	−0.802896	−0.792412	−0.788021
	Std. dev.	2.2E−05	2.7E−03	8.2E−03	2.0E−03	1.2E−02	1.5E−02
g03	Best	−1.000	−1.000	−1.000	−1.0005	−1.000	−1.000
	Mean	−1.000	−1.000	−1.000	−1.0005	−1.000	−1.000
	Std. dev.	6.6E−06	6.0E−16	8.8E−05	1.28E−07	0.0E+00	2.0E−16
g04	Best	−30665.538	−30665.539	−30665.539	−30665.539	−30665.539	−30665.539
	Mean	−30665.537	−30665.539	−30665.539	−30665.539	−30665.539	−30665.539
	Std. dev.	4.3E−04	1.7E−11	2.1E−05	1.44E−11	0.0E+00	3.7E−16
g05	Best	5126.512	5126.498	5126.544	5126.497	5126.484	5126.498
	Mean	5148.966	5126.498	5352.188	5126.497	5185.714	5126.498
	Std. dev.	6.4E+01	2.2E−04	246.2	3.32E−13	7.5E+01	1.3E−12
g06	Best	−6961.807	−6961.814	−6961.814	−6961.814	−6961.814	−6961.814
	Mean	−6961.804	−6961.814	−6961.284	−6961.814	−6961.813	−6961.814
	Std. dev.	2.3E−03	4.6E−12	1.0E−11	0.0E+00	0.2E−02	0.0E+00
g07	Best	24.315	24.326	24.333	24.306	24.330	24.306
	Mean	24.315	24.345	24.387	24.306	24.473	24.306
	Std. dev.	1.1E−01	1.3E−02	2.8E−02	1.05E−14	1.9E−01	5.1E−11
g08	Best	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825
	Mean	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825
	Std. dev.	4.2E−17	2.8E−17	2.1E−17	2.63E−17	0.0E+00	9.1E−11
g09	Best	680.645	680.630	680.631	680.630	680.634	680.630
	Mean	680.671	680.630	680.634	680.630	680.640	680.630
	Std. dev.	9.2E−03	4.3E−13	1.7E−03	0.0E+00	0.4E−02	4.1E−13
g10	Best	7281.957	7059.802	7049.861	7049.248	7053.904	7049.248
	Mean	7479.064	7075.832	7131.084	7049.248	7224.407	7049.248
	Std. dev.	9.8E+01	8.5	67.2	3.96E−12	1.3E+02	1.8E−08
g11	Best	0.750	0.750	0.749	0.7499	0.750	0.750
	Mean	0.750	0.750	0.749	0.7499	0.750	0.750
	Std. dev.	3.0E−08	0.0E+00	1.2E−07	0.0E+00	0.0E+00	0.0E+00
g12	Best	−1.000	−1.000	*NA*	−1.000	−1.000	−1.000
	Mean	−1.000	−1.000	*NA*	−1.000	−1.000	−1.000
	Std. dev.	0.0E+00	0.0E+00	*NA*	0.0E+00	0.0E+00	1.1E−11
g13	Best	0.053947	*NA*	*NA*	0.053942	0.760	0.0539498
	Mean	0.054020	*NA*	*NA*	0.439162	0.968	0.0539498
	Std. dev.	4.8E−05	*NA*	*NA*	3.7E−01	5.6E−02	1.6E−17

NA denotes the results are not available

The comparative results have been listed in Table 5. Based on the observation from Table 5, IWO_DE/Ring drastically outperforms the original IWO_DE for g01, g03, 04, g05, g06, g07, g09, g10 and g13. It's also slightly better than the original IWO_DE for g08 and g11 in terms of success performance. Furthermore, as for test functions with equality constraints, such as g03, g05, g11 and g13, IWO_DE/Ring is better than IWO_DE for g03, g05, g11 and g13, although worse for g02 and g12.

Meanwhile, The statistical *t*-test on results of each problem for the total experimental runs that record the number of FEs for successful run (Liang et al. 2006) is conducted under the 95 % confidence level between IWO_DE/Ring and IWO_DE and the statistical results are shown in Table 6. It

Table 5 Comparing IWO_DE/Ring with IWO_DE

f	Success performance	
	IWO_DE/Ring	IWO_DE
g01	40,983	53,634
g02	144,592	66,692
g03	12,272	16,484
g04	14,886	22,537
g05	12,812	25,025
g06	7,450	10,770
g07	45,275	93,403
g08	2,755	2,990
g09	13,455	23,990
g10	95,788	182,112
g11	1,862	1,976
g12	1,484	1,402
g13	9,655	17,827

Table 6 Statistical results of *t*-test between IWO_DE/Ring and IWO_DE

f	t-value	p-value	Significance
g01	11.7089	1.1309E−15	Extremely significant
g02	0.8026	0.4261	Not significant
g03	7.0154	6.9646E−09	Extremely significant
g04	19.6358	1.3931E−24	Extremely significant
g05	28.5149	9.5396E−32	Extremely significant
g06	19.9308	7.3510E−25	Extremely significant
g07	18.3320	2.5759E−23	Extremely significant
g08	0.4005	0.6905	Not significant
g09	24.2887	1.2895E−28	Extremely significant
g10	25.1119	2.9190E−29	Extremely significant
g11	1.2388	0.2215	Not significant
g12	0.5420	0.5903	Not significant
g13	5.4895	1.4950E−06	Extremely significant

can be obviously observed from Table 6 that IWO_DE/Ring outperforms IWO_DE significantly in 9 out of the 13 test functions. However, the performance of IWO_DE/Ring and IWO_DE is not significant difference for g02, g08, g11 and g12.

In addition, comparison graphs of convergence rate between IWO_DE/Ring and IWO_DE for g01, g03, g07 and g10 are shown in Figs. 8, 9, 10, 11. Obviously, the convergence rate of IWO_DE/Ring is faster than that of IWO_DE for the compared test functions.

6.5 Engineering optimization problems

To further evaluate the performance of IWO_DE/Ring, we execute IWO_DE/Ring on four real-world engineering constrained optimization problems. These four problems are taken from Aguirre et al. (2007), Cagnina et al. (2008) namely,

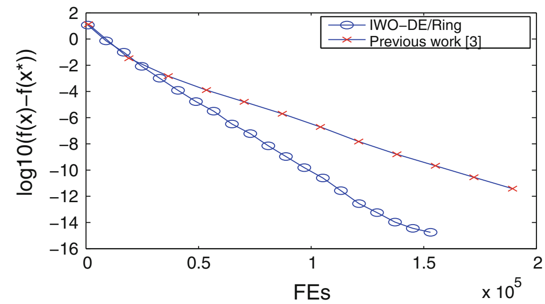


Fig. 8 Comparison with convergence rate for g01

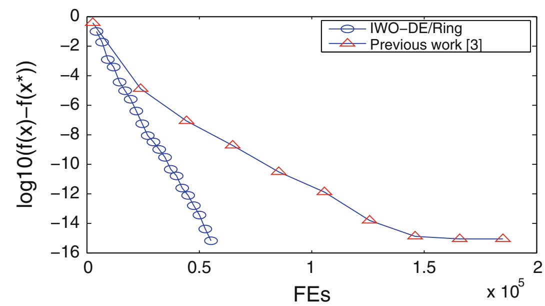


Fig. 9 Comparison with convergence rate for g03

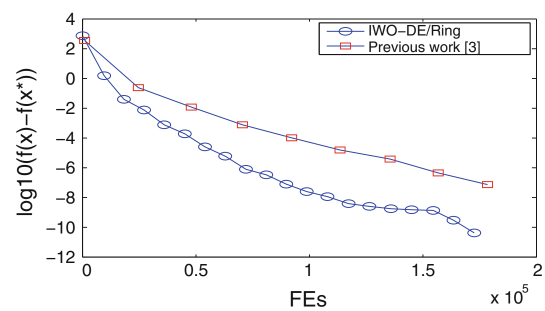


Fig. 10 Comparison with convergence rate for g07

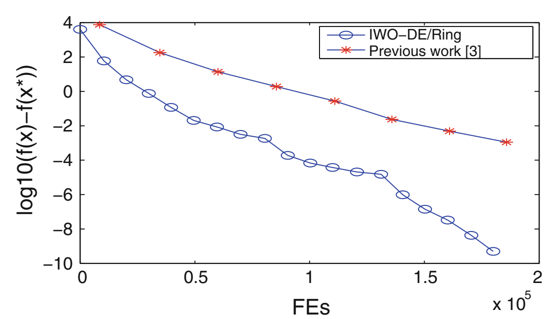


Fig. 11 Comparison with convergence rate for g10

1. Welded Beam design problem.
2. Speed Reducer design problem.
3. Tension/Compression Spring design problem.

Table 7 Results of IWO_DE/Ring on engineering constrained optimization problems

f	Best	Median	Mean	Worst	Std. dev.	FEs
WBP	1.7248523086	1.7248523086	1.7248523086	1.7248523086	1.1E−15	80,000
SRP	2,994.4710661468	2,994.4710661468	2,994.4710661468	2,994.4710661468	1.9E−12	80,000
T/CRP	0.0126652328	0.0126652328	0.0126652328	0.0126652333	8.7E−11	100,000
PVP	6,059.7143350484	6,059.7143350484	6,059.7143350484	6,059.7143350484	9.3E−13	20,000

4. Pressure Vessel design problem.

For convenience, the four problems are denoted as WBP, SRP, T/CSP adn PVP respectively.

6.6 Results on engineering problems

All experimental setups remain the same as what are used in Sect. 6.1, except for the the number of FEs. In this section, IWO_DE/Ring is performed on WBP and SRP by using 80000 FEs, T/CRP by using 100,000 FEs and PVP by using 20,000 FEs. The experimental results are shown in Table 7. It is very obvious that IWO_DE/Ring has successfully tackled all four engineering problems, as shown in Table 7.

6.7 Convergence analysis on engineering problems

For real-world engineering optimization problems, the efficiency of the algorithm is always a very important factor. We further illustrate the convergence graphs of these engineering problems to show the efficiency of IWO_DE/Ring.

The convergence graphs are illustrated in Figs. 12, 13, 14, 15. Apparently, the convergence of IWO_DE/Ring for these engineering optimization problems is very fast and IWO_DE/Ring has converged rapidly to the current known optimal results before the given maximum FEs. Therefore, it can be concluded that IWO_DE/Ring is capable of solving the engineering COPs with high efficiency.

6.8 Performance comparison on engineering problems

this section compares IWO_DE/Ring with several state-of-the-art algorithms on WBP, SRP, T/CSP and PVP respectively. Furthermore, we rerun IWO_DE/Ring under the 30,000 FEs for WBP, SRP and T/CSP respectively and all the parameters are set to the same as that have been presented in Sect. 6.1 in order to have a fair comparison.

For WBP, IWO_DE/Ring is compared against four algorithms from Aguirre et al. (2007), Coello and Becerra (2004), He and Wang (2007), Zhang et al. (2013) and we denote the four compared algorithm as COPSO (Aguirre et al. 2007), CEA (Coello and Becerra 2004), HPSO

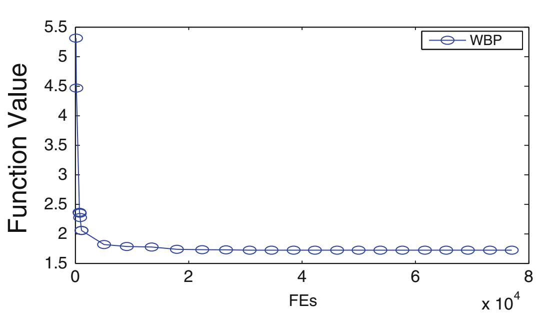


Fig. 12 Convergence graph for WBP

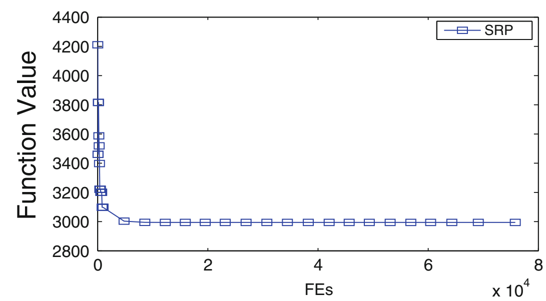


Fig. 13 Convergence graph for SRP

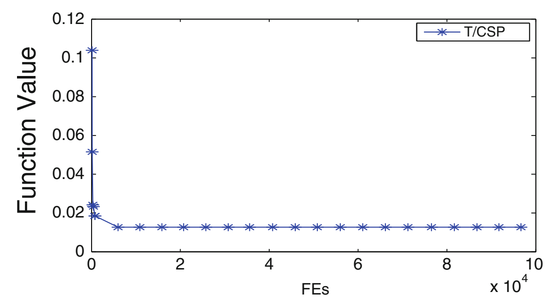


Fig. 14 Convergence graph for T/CSP

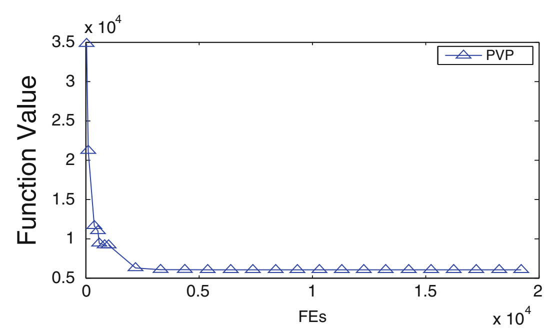


Fig. 15 Convergence graph for PVP

Table 8 Comparison among algorithms on WBP

Algorithm	WBP			
	Best	Mean	Std. dev.	FEs
COPSO	1.724852	1.724881	1.3E−05	30,000
CEA	1.724852	1.971809	4.4E−01	50,020
ICEM	1.724852	1.724852	8.9E−12	80,000
HPSO	1.724852	1.749040	4.0E−02	81,000
IWO-DE/Ring	1.724853	1.726270	2.5E−03	30,000

Table 9 Comparison among algorithms on SRP

Algorithm	SRP			
	Best	Mean	Std. dev.	FEs
COPSO	2,996.372448	2,996.408525	2.9E−02	30,000
HEA-ACT	2,994.499107	2,994.613368	7.0E−02	40,000
SC	2,994.744241	3,001.758,264	4.0	54,456
ISOD	2,996.356689	2,996.367,220	8.2E-03	24,000
IWO-DE/Ring	2,994.471068	2,994.471088	2.6E-05	30,000

Table 10 Comparison among algorithms on T/CSP

Algorithm	T/CSP			
	Best	Mean	Std. dev.	FEs
COPSO	0.012665	0.012666	1.3E−06	30,000
ICEM	0.012665	0.012665	3.7E−08	80,000
HEA-ACT	0.012665	0.012665	1.4E−09	40,000
HPSO	0.012665	0.012707	1.6E−05	81,000
IWO-DE/Ring	0.012665	0.012665	2.1E−08	30,000

Table 11 Comparison among algorithms on PVP

Algorithm	PVP			
	Best	Mean	Std. dev.	FEs
COPSO	6,059.7143	6,071.0133	15.10	30,000
DELC	6,059.7143	6,059.7143	2.1E−11	20,000
ICEM	6,059.7143	6,059.7143	9.1E−13	80,000
HPSO	6,059.7143	6,099.9323	86.2	81,000
IWO-DE/Ring	6,059.7143	6,059.7143	9.3E−13	20,000

(He and Wang 2007) and ICEM (Zhang et al. 2013), respectively. The comparative results have been listed in Table 8.

From Table 8, it can be seen that the mean result obtained by IWO_DE/Ring is better than CEA and HPSO and inferior to COPSO and ICEM. Although the best result IWO_DE/Ring obtains is slightly worse than CEA, ICEM and HPSO, the FEs consumed by IWO_DE/Ring is less than these algorithms.

The comparison of SRP between IWO_DE/Ring and algorithms from Aguirre et al. (2007), Mezura-Montes et al. (2006), Ray and Liew (2003), Wang et al. (2009) that are denoted as COPSO (Aguirre et al. 2007), ISOD (Mezura-Montes et al. 2006), SC (Ray and Liew 2003) and HEA-ACT (Wang et al. 2009) for convenience. Table 9 presents the comparative results among these algorithms.

It is very obvious in Table 9 that IWO_DE/Ring outperforms all the compared algorithms in terms of the quality of results and is also superior to HEA-ACT and SC with respect to the number of FEs.

For T/CSP, the comparison of IWO_DE/Ring is carried out with COPSO (Aguirre et al. 2007), HEA-ACT (Wang et al. 2009), HPSO (He and Wang 2007) and ICEM (Zhang et al. 2013) and the comparative results for T/CSP have been presented in Table 10.

As shown in Table 10, IWO_DE/Ring have approximate efficiency when compared with these compared algorithms for T/CSP both in terms of the best and mean results. Although these compared algorithms has the similar performance on the best and mean result, the number of FEs consumed by ICEM, HEA-ACT and HPSO is larger than that of IWO_DE/Ring.

For PVP, IWO_DE/Ring is compared against COPSO (Aguirre et al. 2007), ICEM (Zhang et al. 2013), HPSO (He and Wang 2007) and DELC (Wang and Li 2010) and then the comparative results are reported in Table 11.

It is clearly shown from Table 11 that IWO_DE/Ring exhibits an approximate performance on the best result in contrast to these compared algorithms and superior to COPSO and HPSO on the mean result. Although IWO_DE/Ring has the same performance with DELC on the best and mean result, DELC is inferior to IWO_DE/Ring with respect to the result of standard deviation and the number of FEs. Meanwhile, although IWO_DE/Ring and ICEM almost display the equivalent statistical results, IWO_DE/Ring takes less computational costs than ICEM.

In summary, IWO_DE/Ring achieves very satisfied results in solving the four engineering COPs. Based on the above comparison, IWO_DE/Ring is fairly superior both in terms of quality of results and computational cost when compared with other algorithms. Furthermore, by increasing the number of FEs on WBP, SRP and T/CSP, the performance of IWO_DE/Ring on WBP, SRP and T/CSP keeps improving (see details in Table 7).

7 Discussion

In this section, several other experiments have been conducted to further analyze the effectiveness of each modification in the proposed IWO_DE/Ring approach.

Table 12 Comparison between IWO_DE/Ring and IWO_DE/Ring_1 on benchmark teat functions

f	Optimal	Algorithm	Best	Mean	Worst	Infeasible run
g03	−1.000	IWO_DE/Ring	−1.000	−1.000	−1.000	0
		IWO_DE/Ring_1	–	–	–	25
g05	5126.498	IWO_DE/Ring	5126.498	5126.498	5126.498	0
		IWO_DE/Ring_1	5,223.141	5,223.141	5,223.141	24
g07	24.3062	IWO_DE/Ring	24.3062	24.3062	24.3062	0
		IWO_DE/Ring_1	24.3080	24.3371	24.4263	0
g09	680.6301	IWO_DE/Ring	680.6301	680.6301	680.6301	0
		IWO_DE/Ring_1	680.6301	680.6302	680.6307	0
g10	7,049.248	IWO_DE/Ring	7,049.248	7,049.248	7,049.248	0
		IWO_DE/Ring_1	7,049.416	7,088.103	7,329.069	0
g11	0.75	IWO_DE/Ring	0.75	0.75	0.75	0
		IWO_DE/Ring_1	0.75	0.75	0.75	24
g13	0.0539498	IWO_DE/Ring	0.0539498	0.0539498	0.0539498	0
		IWO_DE/Ring_1	–	–	–	25

– denotes no feasible solutions can be obtained

7.1 Effectiveness of the modification in “DE/current-to-best/1”

As presented in Sect. 3.2, we modify the original version of “DE/current-to-best/” under the ring neighborhood topology for the sake of achieving an effective exploration. In this section, we perform another algorithm (denoted as IWO_DE/Ring_1), in which the original version of “DE/current-to-best/1” rather than the modified version is incorporated with the IWO variant, in order to demonstrate the effectiveness of the modification in “DE/current-to-best/1”.

To achieve a fair comparison, the experimental parameters of IWO_DE/Ring_1 is set to the same as that are shown in Sect. 6.1. The comparative results are reported in Table 12. In this table, we only report the results that have significant difference between IWO_DE/Ring and IWO_DE/Ring_1 for clarity. It can be observed obviously from Table 12 that IWO_DE/Ring outperforms IWO_DE/Ring_1 on g05, g07 and g10. Although the best result of IWO_DE/Ring_1 for g09 is very close to that of IWO_DE/Ring, the mean and worst results of IWO_DE/Ring_1 are inferior to that of IWO_DE/Ring. More importantly, IWO_DE/Ring_1 is unable to solve g03 and g13 over 25 runs. Additionally, for g05 and g11, there are 24 infeasible runs out of 25 runs respectively when adopting IWO_DE/Ring_1.

From the comparative results, we can draw the conclusion that the modification in “DE/current-to-best/1” plays an important and positive role in addressing COPs.

7.2 Search ability of the novel variant of IWO

In this paper, a novel IWO variant is proposed to achieve a fine-grained local search with adaptation. In this section, in order to show the search ability of the proposed IWO variant, another algorithm is used (denoted as IWO_DE/Ring_2) for comparison. In IWO_DE/Ring_2, the IWO variant has not been employed and only the modified version of “DE/current-to-best/1” is adopted.

Similarly, the parameters of IWO_DE/Ring_2 keeps what have been set in Sect. 6.1. For clarity, we summarize the experimental results that have significant difference between IWO_DE/Ring and IWO_DE/Ring_2 in Table 13. It can be observed that IWO_DE/Ring_2 exhibits the same performance on the best results of g01, g04 and g09 when compared with IWO_DE/Ring, but IWO_DE/Ring is better than IWO_DE/Ring_2 from the mean and worst results of g01, g04 and g09. Furthermore, IWO_DE/Ring outperforms IWO_DE/Ring_2 on g02, g07 and g10. In addition, IWO_DE/Ring_2 only obtains one feasible run out of 25 experimental runs for g03 and the obtained results of g03 are also worse than that obtained by IWO_DE/Ring.

Therefore, based on the experimental results from Table 13, the novel IWO variant really play an indispensable role to enhance the exploitative ability for our proposed approach.

7.3 Investigation of the effectiveness using the ring neighborhood topology

This paper adopts the ring neighborhood topology as the structure of population to provide a good balance between exploration and exploitation in the improved memetic algorithm and then a series of experiments have been demonstrated that the improved memetic algorithm exhibits a greater competitive results. Therefore, we employ another organizational form of neighborhood, namely the form of k-Nearest Neighbor, with the aim of deeply investigating the effectiveness of the ring neighborhood topology.

Table 13 Comparison between IWO_DE/Ring and IWO_DE/Ring_2 on benchmark teat functions

f	Optimal	Algorithm	Best	Mean	Worst	Infeasible run
g01	−15.000	IWO_DE/Ring	−15.00000	−15.00000	−15.00000	0
		IWO_DE/Ring_2	−14.99996	−13.88246	−11.81973	0
g02	−0.8036191	IWO_DE/Ring	−0.8036191	−0.7897140	−0.7596150	0
		IWO_DE/Ring_2	−0.4822592	−0.4013197	−0.2939232	0
g03	−1.000	IWO_DE/Ring	−1.000	−1.000	−1.000	0
		IWO_DE/Ring_2	−0.998	−0.993	−0.987	24
g04	−30,665.539	IWO_DE/Ring	−30,665.539	−30,665.539	−30,665.539	0
		IWO_DE/Ring_2	−30,665.539	−30,664.983	−30,662.843	0
g07	−24.3062	IWO_DE/Ring	−24.3062	−24.3062	−24.3062	0
		IWO_DE/Ring_2	24.4816	25.1562	27.9720	0
g09	680.6301	IWO_DE/Ring	680.6301	680.6301	680.6301	0
		IWO_DE/Ring_2	680.6301	680.6842	681.2988	0
g10	7,049.248	IWO_DE/Ring	7,049.248	7,049.248	7,049.248	0
		IWO_DE/Ring_2	7,053.475	7,119.900	7,310.366	0

Table 14 Comparing IWO_DE/Ring with IWO_DE/kNN

f	Success performance	
	IWO_DE/Ring	IWO_DE/kNN
g01	40,983	46,460
g02	144,592	1,483,675
g03	50,093	79,840
g04	14,886	15,656
g05	30,394	27,988
g06	7,450	6,765
g07	45,275	54,556
g08	2,755	2,369
g09	13,455	13,961
g10	95,788	842,130
g11	12,583	6,902
g12	1,484	1,341
g13	46,056	46,167

We denote the comparing algorithm using the k-Nearest Neighbor as the IWO_DE/kNN and through the k-Nearest Neighbor, the neighborhood of individuals are determined by the euclidian distance in the decision space. The all experimental parameters are the same as that are presented in Sect. 6.1 for achieving a fair comparison with IWO_DE/Ring.

We adopt the the performance criteria, namely success performance (Liang et al. 2006), to evaluate performance between IWO_DE/Ring and IWO_DE/kNN and the comparative results have been listed in Table 14. From the Table 14, it is shown that IWO_DE/Ring have achieved greater performance on g02 and g10 when compared with IWO_DE/kNN. Meanwhile, IWO_DE/Ring is superior to IWO_DE/kNN for g01, g03, g04, g07, g09 and g13. However, there are five test functions, namely g05, g06, g08, g11 and g12, whose performance obtained by IWO_DE/kNN are better than that obtained by IWO_DE/Ring. Hence, it can be concluded, to a certain extent, that the ring neighborhood topology indeed makes a difference in the proposed algorithm. Noteworthily, the results of success performance for g03, g05, g11 and g13 are different from that in Table 5 because the equality constraints are not converted into inequality constraints in IWO_DE/Ring and IWO_DE/kNN.

In summary, with the above discussions, the novel IWO variant and the modification in "DE/current-to-best/1" using the ring neighborhood topology both exert important and positive effect during the search process.

8 Conclusion

This paper proposes an improved memetic algorithm with ring neighborhood topology to solve COPs based on our previously proposed IWO_DE framework in Cai et al. (2013). To further improve the IWO_DE framework, a novel IWO variant with a neighborhood-based dispersal operator is proposed relying on the ring neighborhood topology. The proposed operator depends on the estimation of each solution's neighborhood fitness information to determine the dispersal degree, which leads to a more fine-grained local search. Furthermore, a modified version of "DE/current-to-best/1" is incorporated to further improve the performance of the IWO_DE approach.

Experimental results show that the proposed memetic algorithm is competent to handle various types of COPs, and its performance on many aspects outperforms the previous work (Cai et al. 2013) and several state-of-the-art algorithms. Our future work includes the investigation of IWO as local search engine in depth so that a more competitive variant of IWO can be proposed for COPs.

Acknowledgments The authors would like to thank the related associate editor and the anonymous reviewers for their time and valuable suggestions. This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant 61300159, 61175073 and 51375287, by the Natural Science Foundation of Jiangsu Province under grant BK20130808, by the Research Fund for the Doctoral Program of Higher Education of China under grant 20123218120041 and by the Fundamental Research Funds for the Central Universities of China under grant NZ2013306.

References

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

Boussaid I, Chatterjee A, Siarry P, Ahmed-Nacer M (2012) Biogeography-based optimization for constrained optimization problems. Comput Operat Res 39:3293–3304

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

Cai X, Hu Z, Fan Z (2013) A novel memetic algorithm based on invasive weed optimization and differential evolution for constrained optimization. Soft Comput 17(10):1893–1910

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 (2000) Use of a self-adaptivepenaltyapproach for engineering optimization problems. Comput Ind 41(2):113–127

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

Coello CAC, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36(2):219–236

Coello CAC, Mezura-Montes E (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 16((3):193–203

Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. IEEE Trans Evol Comput 13(3):526–553

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

Gong W, Cai Z, Ling CX (2011) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. Soft Comput 15:645–665

Handoko SD, Kwoh CK, Ong YS (2010) Feasibility structure modeling: a effective chaperone for constrained memetic algorithms. IEEE Trans Evol Comput 14(5):740–758

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

Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Trans Evol Comput 7(2):204–223

Karaboga D, Akay B (2011) A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems. Appl Soft Comput 11:3021–3031

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

Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of the Conference on Evolutionary Computation, pp 1931–1938

Kennedy J, Mendes R (2002) Topological structure and particle swarm performance. In: Proceedings of the 4th Congress Evolutionary Computation, pp 1671–1676

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

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

Li X (2010) Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Trans Evol Comput 14(1):150–169

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, Technical Report

Lin C (2013) A rough penalty genetic algorithm for constrained optimization. Inf Sci 241:119–137

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

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, Coello CAC, Velzquez-Reyes J (2006) Increasing successful offspring and diversity in differential evolution for engineering design. In: Proceedings of the seventh international conference on adaptive computing in design and manufacture, pp 131–139

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

Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Caltech concurrent computation program, C3P, Report 826

Nema S, Goulermas JY, Sparrow G, Helman P (2011) A hybrid cooperative search algorithm for constrained optimization. Struct Multidisc Optim 43:107–119

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

Nguyen QH, Ong YS, Krasnogor N (2007) A study on the design issues of memetic algorithm. IEEE Congress onEvolutionary Computation, pp 2390–2397

Omran MGH, Engelbrecht AP, Salman A (2006) Using the ring neighborhood topology with self-adaptive differential evolution. Lect Notes Comput Sci 4221:976–979

Ong YS, Lim MH, Chen X (2010) Memetic computationpast, present & 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

Ray T, Liew K (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7(4):386–396

Roy S, Islam SM, Das S, Ghosh S (2013) Multimodal optimization by artificial weed colonies enhanced with localized group search optimizers. Appl Soft Comput 13:27–46

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

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

Sun J, Garibaldi JM, Krasnogor N, Zhang Q (2013) An intelligent multi-restart memetic algorithm for box constrained global optimisation. Evol Comput 21(1):107–147

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

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

Tasgetiren MF, Suganthan PN (2006) A multi-populated differential evolution algorithm for solving constrained optimization problem. IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, pp 33–40

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

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

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

Wang L, Li L (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidisc Optim 41:947–963

Wang Y, Cai Z, Guo G, Zhou Y (2007) Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. IEEE Trans Syst Man Cybern Part B Cybern 37(3):560–575

Wang Y, Cai Z, Zhou Y, Zeng W (2008) An adaptive tradeoff model for constrained evolutionary optimization. IEEE Trans Evol Comput 12(1):80–92

Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidisc Optim 37:395–413

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

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

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

Zhang C, Li X, Gao L, Wu Q (2013) An improved electromagnetism-like mechanism algorithm for constrained optimization. Expert Syst Appl 40:5621–5634