Applied Soft Computing Journal 77 (2019) 714–722

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Ring Theory-Based Evolutionary Algorithm and its application to D{0-1} KP

Yichao He [a,*], Xizhao Wang [b], Suogang Gao [c]

[a] College of Information Engineering, Hebei GEO University, Shijiazhuang 050031, China
[b] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[c] College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang 050024, China

HIGHLIGHTS

- Propose a new idea of using ring theory to design evolutionary algorithms.
- Propose a new evolutionary algorithm RTEA by direct product of rings, which is very suitable for solving a class of combinatorial optimization problems.
- Propose a novel approach for solving discounted {0-1} knapsack problem by RTEA, whose performance is better than existing algorithms.

ARTICLE INFO

Article history:
Received 2 February 2018
Received in revised form 24 January 2019
Accepted 31 January 2019
Available online 12 February 2019

Keywords:
Evolutionary algorithms
Residue class ring module *m*
Direct product of rings
Discounted {0-1} knapsack problem

ABSTRACT

In this paper, we propose a new view for designing an evolutionary algorithm by using algebraic theory to solve the combinatorial optimization problem. Using the addition, multiplication and inverse operation of the direct product of rings, we first propose two evolution operators: the global exploration operator (R-GEO) and the local development operator (R-LDO). Then, by utilizing the R-GEO and R-LDO to generate individuals and applying the greedy selection strategy to generate a new population, we propose a new algorithm – the Ring Theory-Based Evolutionary Algorithm (RTEA) – for the combinatorial optimization problem. Moreover, we give a new method for solving the discounted { 0-1} knapsack problem (D{ 0–} KP) by using the RTEA. To verify the performance of the RTEA, we use it and existing algorithms to solve four kinds of large-scale instances of the D{ 0-1} KP. The computational results show that the RTEA performs better than the others, and it is more suitable for solving the D{ 0-1} KP problem. Moreover, it indicates that using algebraic theory to design evolutionary algorithms is feasible and effective.

1. Introduction

The discounted {0-1} knapsack problem (D{0-1}KP) is a novel knapsack problem proposed by Guldan [1]. The "discount" thought is a marketing approach in the field of business, which is a reasonable explanation of international trade patterns and commercial scale phenomena in real-world life. It can be applied to investment decision-making, project selection, and budget control in the field. Guldan [1] first studied the algorithm of the D{0-1}KP, and gave a deterministic algorithm based on dynamic programming. Aiying Rong et al. [2] proposed the definition of the alternative core for the D{0-1}KP by imitating the core concept of the 0-1 knapsack problem (0-1 KP), and combined dynamic programming with the core of the D{0-1}KP to solve it. On the basis of the effective methods for dealing with infeasible solutions, He et al. [3] proposed two efficient algorithms, FirEGA and SecEGA, based on the genetic algorithm (GA) to solve the D{0-1}KP. Recently, they [4] also had a detailed study of the algorithms of the D{0-1}KP and proposed a new deterministic algorithm and two approximation algorithms. Especially, they advanced an efficient algorithm called PSO-GRDKP (it is called GBPSO for short) by using discrete particle swarm optimization [5]. It can effectively solve four kinds of D{0-1}KP instances. In this paper, we advance a new idea by using algebraic theory to design an evolutionary algorithm. Furthermore, we propose a new algorithm for solving the combinatorial optimization problem: the Ring Theory-Based Evolutionary Algorithm (RTEA). Moreover, we propose an effective method for solving the D{0-1}KP using the RTEA. The comparison results of the RTEA, FirEGA, SecEGA and GBPSO show that the RTEA is not only easy to implement but is also the most effective at solving the D{0-1}KP among the four algorithms. This result also indicates that the new design method based on the evolutionary algorithm using ring theory is feasible and effective.

The remainder of the paper is organized as follows. In Section 2, we briefly introduce the basic concepts of rings, the direct product of rings, the residue class ring module *m* and the direct product of

* Corresponding author.
E-mail addresses: heyichao@hgu.edu.cn (Y. He), xizhaowang@ieee.org (X. Wang), sggaomail@163.com (S. Gao).

https://doi.org/10.1016/j.asoc.2019.01.049
1568-4946/

rings composed by it. Then, the definition and mathematical model of the D{0-1}KP are summarized. In Section 3, we propose an evolutionary algorithm, the RTEA, by using the direct product of rings. It can be used to solve the combinatorial optimization problem whose feasible solution is an integer vector in $\{0, 1, \dots, m_1 - 1\} \times \dots \times \{0, 1, \dots, m_n - 1\}$. In Section 4, by applying the NROA [3] to handle the infeasible solutions, we propose a new method for solving the D{0-1}KP using the RTEA. In Section 5, the reasonable values of the parameter P_m in the RTEA are first determined. Then, we use the RTEA, GBPSO, FirEGA and SecEGA to solve the four kinds of large-scale D{0-1}KP instances, and show that the performance of the RTEA is the best among the four algorithms. This shows that the algebraic method for designing the evolutionary algorithm is not only feasible but also effective. Finally, we summarize the full text and state further research directions for the future.

2. Preliminaries

2.1. Ring and direct product of rings

A ring is an algebraic system with two binary operations, which consists of an Abel group with an additive operation and a multiplicative semigroup. The direct product of rings is a special ring, in which every element is an ordered m-tuple. The direct product of rings can also be seen as a method of constructing a new ring by using many rings. For describing the principle of the RTEA algorithm, simple introductions of the ring and the product of rings are given below, and for more details one can refer to [6] and [7].

Definition 1 ([6,7]). A ring $\langle R, +, \bullet\rangle$ is a nonempty set R together with two binary operations $+$ and $\bullet$, which we call addition and multiplication, defined on R such that the following axioms hold:

(1) $\forall a, b \in R, a + b = b + a$;

(2) $\forall a, b, c \in R, (a + b) + c = a + (b + c)$

(3) $\exists 0 \in R$ such that $\forall a \in R, a + 0 = 0 + a$, where 0 is called an additive identity;

(4) $\forall a \in R, \exists b \in R$ such that $a + b = b + a = 0$, where b is called the inverse of a, and it is written as $-a$;

(5) $\forall a, b, c \in R, (a \bullet b) \bullet c = a \bullet (b \bullet c)$; and

(6) $\forall a, b, c \in R, a\bullet(b+c) = a\bullet b+a\bullet c, (b+c)\bullet a = b\bullet a+c\bullet a$.

Let $Z_m = \{[0], [1], \dots, [m-1]\}$ be a collection of residue classes of module m, where $[j] = \{x \in Z | x \equiv j (mod\ m)\}, 0 \le j \le m - 1$, Z is the set of all integers, and m is an integer that is more than 1. Define two binary operations $\bigoplus$ and $\bigodot$ on Z_m as following:

$$[i]\bigoplus[j] = [(i+j) mod\ m], \quad [i]\bigodot[j] = [(ij) mod\ m], \quad \forall[i], [j] \in Z_m.$$

Thus, Z_m is a ring with operations $\bigoplus$ and $\bigodot$, and it is said that module m residues in the class ring. In Z_m, the additive identity is $[0]$, and $-[0] = [0], -[j] = [m - j]$, for $j = 1, 2, \dots, m - 1$.

Definition 2 ([6,7]). If R_i are rings, $i \in I = \{1, 2, \dots, n\}$, then $\prod_{i\in I} R_i = R_1 \times R_2 \times \dots \times R_n$ is a ring with operations defined by $\langle a_1, a_2, \dots, a_n\rangle \bigoplus \langle b_1, b_2, \dots, b_n\rangle = \langle a_1+b_1, a_2+b_2, \dots, a_n+b_n\rangle$, and $\langle a_1, a_2, \dots, a_n\rangle \bigodot \langle b_1, b_2, \dots, b_n\rangle = \langle a_1b_1, a_2b_2, \dots, a_nb_n\rangle$. We call $\prod_{i\in I} R_i$ as the direct product of $R_i, i \in I$.

Obviously, $\prod_{i=1}^{n} Z_{m_i} = Z_{m_1} \times Z_{m_2} \times \dots \times Z_{m_n}$ is a direct product of rings by Definition 2, which is made up of n rings Z_{m_i} $(1 \le i \le n)$. The additive identity is $\langle[0], [0], \dots, [0]\rangle$, and its inverse is itself . For any nonzero element $\langle[a_1], [a_2], \dots, [a_n]\rangle$, its inverse is $\langle-[a_1], -[a_2], \dots, -[a_n]\rangle$. In fact, because $\langle[1], [1], \dots, [1]\rangle \bigodot \langle[a_1], [a_2], \dots, [a_n]\rangle = \langle[a_1], [a_2], \dots, [a_n]\rangle \bigodot\langle[1], [1], \dots, [1]\rangle = \langle[a_1], [a_2], \dots, [a_n]\rangle$, element $\langle[1], [1], \dots, [1]\rangle$ is the multiplicative identity of $\prod_{i=1}^{n} Z_{m_i}$.

2.2. Mathematical models of D{0-1}KP

Definition 3 ([2]). Given a set of n item groups, suppose that each group $i(i = 0, 1, \dots, n - 1)$ consists of three items: $3i$, $3i + 1$ and $3i + 2$. The item $3i$ has weight w_{3i} and profit p_{3i}, and the item $3i + 1$ has weight w_{3i+1} and profit p_{3i+1}. The first two items $3i$ and $3i + 1$ are paired to derive the third item $3i + 2$ with profit $p_{3i+2} = p_{3i}+p_{3i+1}$ and the discounted weight w_{3i+2}, which satisfies $w_{3i+2} < w_{3i} + w_{3i+1}$, $w_{3i} < w_{3i+2}$ and $w_{3i+1} < w_{3i+2}$. In each group, at most one of the three items can be selected to be placed in the knapsack with capacity C. The problem is how to select items loaded into the knapsack such that the total profit is maximized under the condition that the total weight of the selected items does not exceed C.

The first mathematical model of the D{0-1}KP come from [2], and we will not repeat it here. Now, we introduce the second mathematical models of the D{0-1}KP [3] as follows.

Without the loss of generality, let $p_j, w_j (0 \le j \le 3n - 1)$ and C be integers, and $w_{3i+2} \le C (0 \le i \le n - 1)$, $\sum_{i=0}^{n-1} w_{3i+2} > C$. Let $X = (x_0, x_1, \dots, x_{n-1}) \in \{0, 1, 2, 3\}^n$, where $x_i = 0 (0 \le i \le n - 1)$ denotes that no item of the ith group is loaded into the knapsack, $x_i = 1$ indicates that item $3i$ is put into the knapsack, $x_i = 2$ denotes that item $3i + 1$ is loaded into the knapsack, and $x_i = 3$ indicates that item $3i + 2$ is put into the knapsack. Let $\lceil x\rceil$ be the ceiling function, which is the minimum integer that is not less than x. Hence, the second mathematical model of D{0-1}KP is described as follows:

$$max\ f(X) = \sum_{i=0}^{n-1} \lceil x_i/3\rceil p_{3i+|x_i-1|} \tag{1}$$

$$s.t. \sum_{i=0}^{n-1} \lceil x_i/3\rceil w_{3i+|x_i-1|} \le C \tag{2}$$

$$x_i \in \{0, 1, 2, 3\}, i = 0, 1, \dots, n - 1. \tag{3}$$

When solving the D{0-1}KP using the evolutionary algorithm under the second mathematical model, the encoding of an individual is an n-dimensional integer vector $X = (x_0, x_1, \dots, x_{n-1}) \in \{0, 1, 2, 3\}^n$. Consequently, any n-dimensional integer vector $X \in \{0, 1, 2, 3\}^n$ is a potential solution of the D{0-1}KP. Only when X satisfies (2) is it a feasible solution, and otherwise it is an infeasible solution.

3. Ring theory-based evolutionary algorithm

Note that there is a bijection $H : \prod_{i=1}^{n} Z_{m_i} \to Z[m_1, m_2, \dots, m_n]$ from $\prod_{i=1}^{n} Z_{m_i}$ to $Z[m_1, m_2, \dots, m_n] = \{0, 1, \dots, m_1 - 1\} \times \{0, 1, \dots, m_2 - 1\} \times \dots \times \{0, 1, \dots, m_n - 1\}$, for any element $\bar{X} = \langle[x_1], [x_2], \dots, [x_n]\rangle \in \prod_{i=1}^{n} Z_{m_i}, 0 \le x_i \le m_i - 1$, such that

$$H(\langle[x_1], [x_2], \dots, [x_n]\rangle) = (x_1, x_2, \dots, x_n) \in Z[m_1, m_2, \dots, m_n].$$

Here m_i is an integer and large than 1, and $i = 1, 2, \dots, n$. Therefore, we can use the element $\bar{X} = \langle[x_1], [x_2], \dots, [x_n]\rangle$ in $\prod_{i=1}^{n} Z_{m_i}$ to denote the vector $X = (x_1, x_2, \dots, x_n)$ in $Z[m_1, m_2, \dots, m_n]$. Thus, by drawing support from the addition, multiplication and inverse operations of $\prod_{i=1}^{n} Z_{m_i}$, we develop two evolutionary operators on $Z[m_1, m_2, \dots, m_n]$, and propose a new evolutionary algorithm called the Ring Theory-Based Evolutionary Algorithm (RTEA), which can be used to solve the combinatorial optimization problem whose feasible solution is an n-dimensional integer vector in $Z[m_1, m_2, \dots, m_n]$.

Let $Y_1 = (y_{11}, y_{12}, \dots, y_{1n})$, $Y_2 = (y_{21}, y_{22}, \dots, y_{2n})$, $Y_3 = (y_{31}, y_{32}, \dots, y_{3n})$ and $Y_4 = (y_{41}, y_{42}, \dots, y_{4n})$ be four different n-dimensional integer vectors randomly selected from $Z[m_1, m_2, \dots, m_n]$. We can use Y_1, Y_2, Y_3 and Y_4 to generate new

n-dimensional integer vectors $X = (x_1, x_2, \ldots, x_n) \in Z[m_1, m_2, \ldots, m_n]$ using the following equation:

$$x_i = \begin{cases} \{y_{1i} + y_{4i} \times [y_{2i} + (m_i - y_{3i})]\}(mod\ m_i), & if\ rand(i) \le 0.5; \\ \{y_{1i} + [y_{2i} + (m_i - y_{3i})]\}(mod\ m_i), & otherwise. \end{cases} \tag{4}$$

Here, $i = 1, 2, \ldots, n$, $rand(i)$ is a random real number in the interval (0,1).

For example, suppose $Y_1 = (3, 1, 3, 2, 1)$, $Y_2 = (0, 2, 0, 1, 3)$, $Y_3 = (1, 2, 3, 1, 1)$ and $Y_4 = (1, 2, 3, 2, 0)$ are four different integer vectors randomly selected from $Z[4, 4, 4, 4, 4] = \{0, 1, 2, 3\}^5$, and $rand(1) = 0.21$, $rand(2) = 0.74$, $rand(3) = 0.43$, $rand(4) = 0.18$, and $rand(5) = 0.91$. Then, we can generate a new 5-dimensional integer vector $X = (x_1, x_2, x_3, x_4, x_5) = (2, 1, 2, 2, 3)$ by using Eq. (4), where

$$\begin{aligned} x_1 &= [y_{11} + y_{41} \times (y_{21} + (4 - y_{31}))](mod\ 4) \\ &= [3 + 1(0 + (4 - 1))](mod\ 4) = 2; \\ x_2 &= [y_{12} + (y_{22} + (4 - y_{32}))](mod\ 4) \\ &= [1 + (2 + (4 - 2))](mod\ 4) = 1; \\ x_3 &= [y_{13} + y_{43} \times (y_{23} + (4 - y_{33}))](mod\ 4) \\ &= [3 + 3(0 + (4 - 3))](mod\ 4) = 2; \\ x_4 &= [y_{14} + y_{44} \times (y_{24} + (4 - y_{34}))](mod\ 4) \\ &= [2 + 2(1 + (4 - 1))](mod\ 4) = 2; \\ x_5 &= [y_{15} + (y_{25} + (4 - y_{35}))](mod\ 4) \\ &= [1 + (3 + (4 - 1))](mod\ 4) = 3. \end{aligned}$$

In fact, using Eq. (4) to produce a new vector X is essentially equivalent to the following process: First, generate $\bar{T} = \langle [t_1], [t_2], \ldots, [t_n] \rangle$ by using $\bar{Y_4} = \langle [y_{41}], [y_{42}], \ldots, [y_{4n}] \rangle$ which is randomly selected from $\prod_{i=1}^{n} Z_{m_i}$ and the multiplicative identity $\bar{1} = \langle [1], [1], \ldots, [1] \rangle$ of $\prod_{i=1}^{n} Z_{m_i}$, in which every component $[t_i](1 \le i \le n)$ is randomly selected from $\{[y_{4i}], [1]\}$ with equal probability. Second, according to the following Eq. (5), use three randomly selected elements $\bar{Y_1} = \langle [y_{11}], [y_{12}], \ldots, [y_{1n}] \rangle$, $\bar{Y_2} = \langle [y_{21}], [y_{22}], \ldots, [y_{2n}] \rangle$, $\bar{Y_3} = \langle [y_{31}], [y_{32}], \ldots, [y_{3n}] \rangle$ and $\bar{T}$ to generate a new element $\bar{X} = \langle [x_1], [x_2], \ldots, [x_n] \rangle$ of $\prod_{i=1}^{n} Z_{m_i}$. Finally, map $\bar{X}$ to vector $X = (x_1, x_2, \ldots, x_n)$ using the bijection H.

$$\bar{X} = \bar{Y_1} \bigoplus [\bar{T} \bigodot (\bar{Y_2} \bigoplus -\bar{Y_3})] \tag{5}$$

Because Y_1, Y_2, Y_3 and Y_4 are randomly selected from $Z[m_1, m_2, \ldots, m_n]$ in Eq. (4), the procedure that generates a new n-dimensional integer vector X by Eq. (4) is essentially a random operator. On the other hand, the procedure generated new n-dimensional integer vector X indicates that the vector X has the ability of global learning. That is, it is generated by learning from four different elements that are randomly selected from the whole search space. Hence, we call this procedure the Ring-based Global Exploration Operator (R-GEO), and represent it as $X = \boldsymbol{R-GEO}(Y_1, Y_2, Y_3, Y_4)$ in the following. Obviously, R-GEO is a global stochastic search operator in the discrete space $Z[m_1, m_2, \ldots, m_n]$. It reflects the global exploration ability of the evolutionary algorithm. However, only the global exploration ability is not enough, but is must also have the local exploration ability in order to keep the balance of the global search and local search in the evolutionary algorithm [8]. Therefore, we give the following new evolutionary operator based on the inverse operation to implement the local exploration. The new operator is named the Ring-based Local Development Operator (R-LDO).

Let $X = (x_1, x_2, \ldots, x_n) \in Z[m_1, \ldots, m_n]$. P_m is a given real number in the interval (0,1), and we named it the local search probability. Then, the pseudo-code of R-LDO is described as follows:

Algorithm 1. $R - LDO$
Input: $X = (x_1, x_2, \ldots, x_n)$ and P_m;
Output: $X = (x_1, x_2, \ldots, x_n)$.

```
for i ← 1 to n do
    if (rand1 < P_m) then
        if (rand2 < 0.5 and x_i ≠ 0) then x_i ← m_i − x_i;
        else x_i ← rand({0, 1, ..., m_i − 1} − {x_i});
        end if
    endif
end for
```

In R-LDO, $rand1$ and $rand2$ are two random real numbers in the interval (0,1), respectively. The value of P_m is usually set in the interval (0, 0.5]. $rand(\{0, 1, \ldots, m_i - 1\} - \{x_i\})$ is an integer that is generated randomly from the set $\{0, 1, \ldots, m_i - 1\} - \{x_i\}$. It is easy to see that the essence of R-LDO is also a random operator, and the time complexity of R-LDO is $O(n)$.

In summary, we use R-GEO and R-LDO to generate new individuals, and utilize the greedy selection strategy to select individuals to form the new generation population. Hereby, we propose the following algorithm RTEA based on the general framework of the evolutionary algorithm [9] .

Let $P(t) = \{X_k(t) | 1 \le k \le NP\}$ be the tth generation population of the RTEA and let $X_k(t) = (x_{k1}(t), \ldots, x_{kn}(t)) \in Z[m_1, \ldots, m_n]$ be the kth individual in $P(t)$, where $0 \le t \le MIT$, MIT is the maximum iterative number, and NP is the size of population. We use $Fit(X_k(t))$ to denote the fitness value of individual $X_k(t)$, and use $B = (b_1, b_2, \ldots, b_n) \in Z[m_1, \ldots, m_n]$ to denote the best individual in the population $P(MIT)$. Then, for the combinatorial optimization problem $\prod$: $max\, f(X), X \in Z[m_1, \ldots, m_n]$, the pseudo-code of the RTEA is described as follows:

Algorithm 2. *RTEA*
Input: An instance of $\prod$; parameters NP, MIT, and P_m;
Output: B and $f(B)$.

```
1  Generate initial population P(0) = {X_k(0) | 1 ≤ k ≤ NP}
   randomly;
2  Compute Fit(X_k(0)), for 1 ≤ k ≤ NP;
3  for t ← 1 to MIT do
4      for k ← 1 to NP do
5          Y ← R − GEO (X_p1(t − 1), X_p2(t − 1), X_p3(t − 1),
           X_p4(t − 1));
6          Y ← R − LDO(Y, P_m);
7          if Fit(Y) > Fit(X_k(t − 1)) then X_k(t) ← Y else
           X_k(t) ← X_k(t − 1);
8      end for
9  end for
10 Determine B in P(MIT);
11 return(B, f(B)).
```

In the RTEA, $X_{p1}(t-1)$, $X_{p2}(t-1)$, $X_{p3}(t-1)$, and $X_{p4}(t-1)$ are four different individuals randomly selected from population $P(t-1)$ in Step 5. Usually, the time complexity of computing $Fit(X)$ is $O(n)$. Then, the time complexity of Step 1, Step 2 and Steps 4–8 are $O(n * NP)$. The time complexity of the RTEA is $O(MIT * n * NP)$. Note that MIT and NP are both a linear function of n, and it indicates that $O(MIT * n * NP) = O(n^3)$.

4. Using RTEA to solve D{0-1}KP

When we use the RTEA to solve the D{0-1}KP under the second mathematical model, the encoding of the individual is an n-dimensional integer vector $X = (x_0, x_1, \ldots, x_{n-1}) \in Z[4, 4, \ldots, 4] = \{0, 1, 2, 3\}^n$. Since the D{0-1}KP is a constrained optimization problem, many infeasible solutions would be inevitably generated in the RTEA. This is not the beneficial to calculating the fitness value of an individual, but it also reduces the efficiency

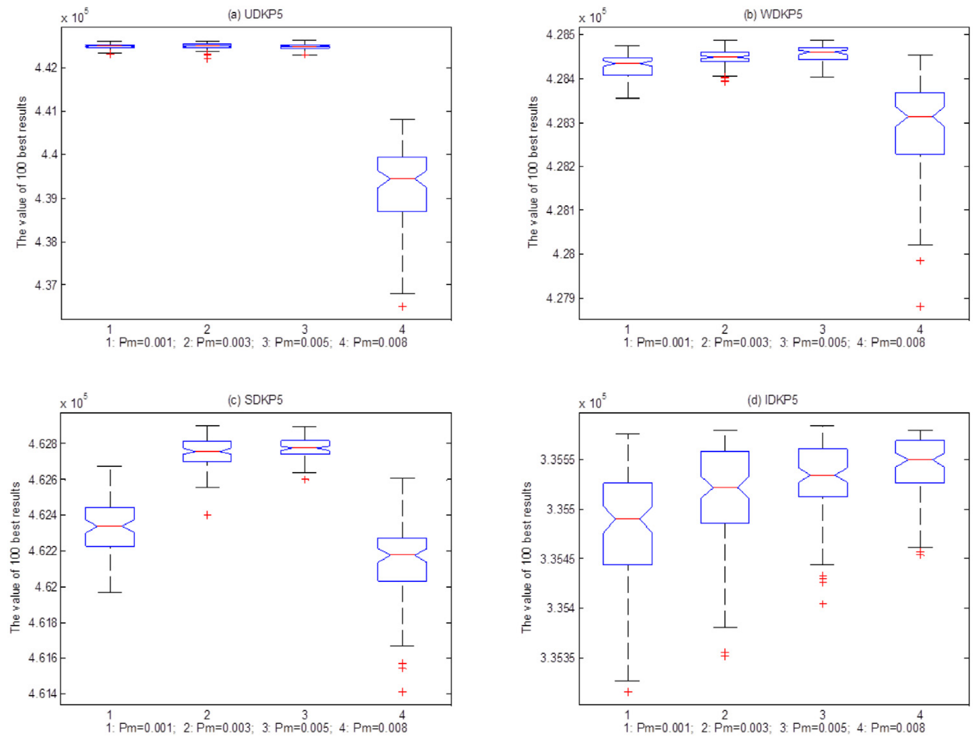

Fig. 1. Box plot of 4 instances: (a) UDKP5, (b) WDKP5, (c) SDKP5, (d) IDKP5.

of the algorithm. Therefore, we use the algorithm NROA [3] (The pseudo-code of NORA is given in Appendix) to repair and optimize all individuals in the RTEA, and utilize the objective function value of the feasible solution as the fitness value of the individual corresponding to it. Accordingly, we present a new method for solving D{0-1}KP using the RTEA.

Let $H[0\ldots 3n-1]$ be an array, which is used to store the index of all items according to the descending order of $\frac{p_j}{w_j}, 0 \le j \le 3n-1$. That is, $\frac{p_{H[0]}}{w_{H[0]}} \ge \frac{p_{H[1]}}{w_{H[1]}} \ge \cdots \ge \frac{p_{H[3n-1]}}{w_{H[3n-1]}}$. Let $B = (b_0, b_1, \ldots, b_{n-1}) \in \{0, 1, 2, 3\}^n$ denote the best individual in the population $P(MIT)$. Then, the pseudo-code of the RTEA for the D{0-1}KP is illuminated as follows:

Algorithm 3. *RTEA for* $D\{0-1\}KP$

Input: An instance of $D\{0-1\}KP$; parameters NP, MIT, and P_m;

Output: B and $f(B)$.

```
Determine H[0...3n−1] with p_H[0]/w_H[0] ≥ p_H[1]/w_H[1] ≥ ... ≥ p_H[3n−1]/w_H[3n−1];
Generate initial population P(0) = {X_k(0) = (x_k0(0), ...,
  x_k,n−1(0))| x_kj(0) ∈ {0, 1, 2, 3}, 1 ≤ k ≤ NP, 0 ≤ j ≤ n − 1}
  randomly;
for k ← 1 to NP do
    (X_k(0), f(X_k(0))) ← NROA(X_k(0), H[0...3n − 1]);
end for
for t ← 1 to MIT do
    for k ← 1 to NP do
        Y ← R − GEO (X_p1(t − 1), X_p2(t − 1), X_p3(t − 1),
        X_p4(t − 1));
        Y ← R − LDO(Y, P_m);
        (Y, f(Y)) ← NROA (Y, H[0...3n − 1]);
        if f(Y) > f(X_k(t − 1)) then X_k(t) ← Y else
        X_k(t) ← X_k(t − 1);
    end for
end for
Determine B in P(MIT);
return(B, f(B)).
```

Obviously, the time complexity of Algorithm 3. is $O(nlogn) + O(MIT * n * NP) = O(n^3)$.

5. Computational experiments

For verifying the performance of the RTEA, we first determine the optimal value of parameter P_m in the RTEA according to the computational results of some D{0-1}KP instances. Then, we use the RTEA, GBPSO, FirEGA and SecEGA to solve the four kinds of large-scale D{0-1}KP instances [3] whose scale ($3n$) is from 300 to 3000 (URL: https://www.researchgate.net/project/Four-kinds-of-D0-1-KP-instances), and evaluate the performance of the RTEA by comparing it with their computational results.

All experiments are performed on an Acer Aspire E1-570G notebook computer with an Intel(R) Core(TM) i5-3337U CPU-1.8 GHz and 4 GB DDR3 memory (3.82 GB available). The operating system is Microsoft Windows 8. All the algorithms are implemented using Visual C++ 6.0, and use MATLAB7.10.0.499 (R2010a) for drawing.

5.1. Determine the reasonable value of parameter P_m

To determine the reasonable value of parameter P_m of the RTEA, we solve four groups of D{0-1}KP instances, including *DKP1, *DKP3, *DKP5, *DKP7, and *DKP9 (where * represents symbol U, W, S, and I, respectively), when the value of parameter P_m is respectively 0.001, 0.003, 0.005, and 0.008. For each given value of P_m, every instance is calculated independently 100 times by using the RTEA, and determine the reasonable value of P_m by comparing

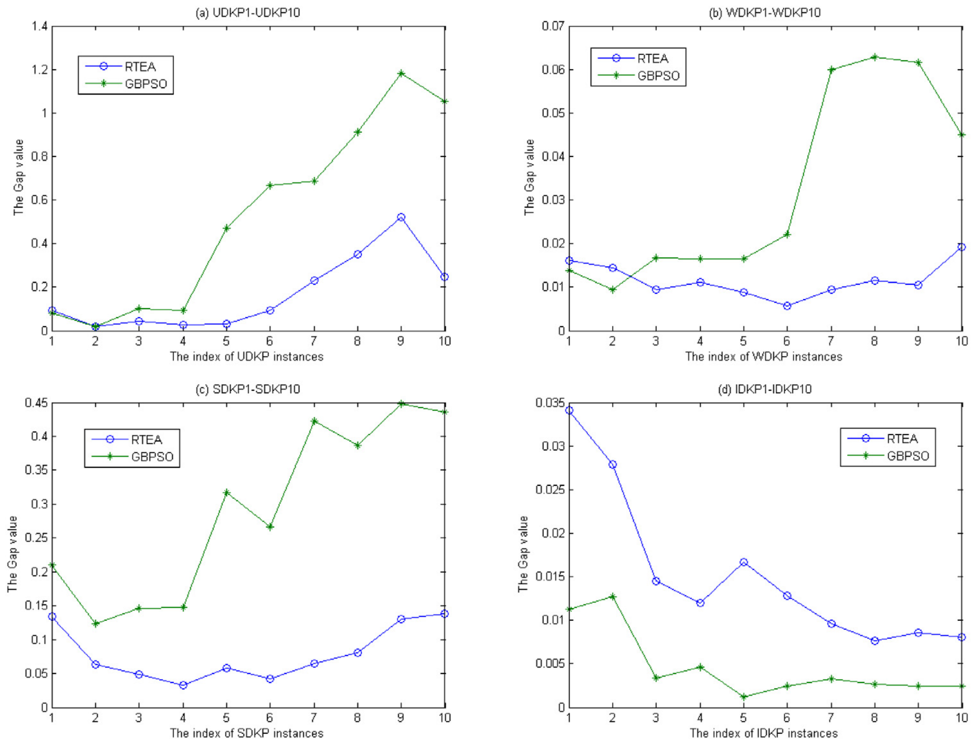

Fig. 2. Fitting curves of *Gap*: (a) UDKP1–UDKP10, (b) WDKP1–WDKP10, (c) SDKP1–SDKP10, (d) IDKP1–IDKP10.

Table 1
The mean ranks obtained by Kruskal–Wallis test.

P_m	0.001	0.003	0.005	0.008
UDKP5's mean ranks	251.3850	259.1150	241	50.5000
WDKP5's mean ranks	196.2400	254.9900	289.6250	61.1450
SDKP5's mean ranks	129.2450	289.6250	310.6800	72.4500
IDKP5's mean ranks	128.7100	190.4900	223.8800	258.9200

Table 2
The value of parameters of all algorithms.

Algorithm	Size of population	Other parameters
RTEA	$NP = 20$	$P_m = 0.005$
FirEGA	$NP = 50$	$P_c = 0.8, P_m = 1.0(UDKP), P_m = 0.01(Others)$
SecEGA	$NP = 50$	$P_c = 0.8, P_m = 0.01$
GBPSO	$NP = 20$	$w = 1.0, C_1 = C_2 = 2.0, [-A, A] = [-5.0, 5.0]$

Table 3
Time upper bound for solving D{0-1}KP instances.

Instance	*DKP1	*DKP2	*DKP3	*DKP4	*DKP5
Time (s)	0.5	1.5	3.5	6.0	10.5
Instance	*DKP6	*DKP7	*DKP8	*DKP9	*DKP10
Time (s)	14.0	19.0	24.0	30.0	36.5

the computational results of those instances. To save space, we only give the computational results of the four representative instances UDKP5, WDKP5, SDKP5, and IDKP5, and use them to illustrate what P_m values make RTEA the best performance. The test results of Kruskal–Wallis test [10–12] that is used to compare different P_m value are given in Table 1 and Fig. 1. In Table 1, the mean rank of test data are given, and Fig. 1 displays box plots of the 100 best results when $P_m = 0.001$, 0.003, 0.005, and 0.008. According to the above test results, $P_m = 0.005$ is the more appropriate.

5.2. Computation and comparison

When computing the four kinds of large-scale D{0-1}KP instances, parameters of the RTEA are set as: the size of population is $NP = 20$, and the parameter $P_m = 0.005$. The parameters of the FirEGA, SecEGA, and GBPSO come from [3] and [4] respectively. For ease of comparison, the value of parameters of all algorithms are listed in Table 2.

Because the operations in an iteration of the RTEA, FirEGA, SecEGA, and GBPSO are different from each other, the cost of an iteration of them are quite different. For example, the consumed time of an iteration of GBPSO is 3 times more than that of the RTEA. It is unfair to evaluate the performance of the four algorithms based on the number of iterations. Therefore, on the premise that the performance of each algorithm is fully exhibited, we set a reasonable upper bound for the solution time (in seconds) of each instance. They are listed in Table 3, in which the symbol $*$ represents the letters U, W, S, and I, respectively.

Tables 4–7 summarize the comparison among the RTEA,FirEGA, SecEGA and GBPSO based on the five different performance criteria, namely, the best result in the 100 independent calculations results (*BEST*), the average of the 100 results (*Mean*), the worst result of the 100 results (*Worst*), the standard deviation (*StD*) and the *Gap* [13] between the *Mean* and *OPT*, where *OPT* is the optimal value of instance. The formula of computing the *Gap* is the following Eq. (6).

$$Gap = \frac{|OPT - Mean|}{OPT} \times 100(\%) \tag{6}$$

From Tables 4–7, we can see that the *Worst* of the RTEA and GBPSO are both better than the *BEST* of FirEGA and SecEGA except for IDKP1 and IDKP2. Even for IDKP1 and IDKP2, the *BEST*, *Mean* and *Worst* of the RTEA and GBPSO are better than those of FirEGA and SecEGA. It shows that the performance of FirEGA and SecEGA are far worse than those of the RTEA and GBPSO. It is unnecessary to compare the performances of the RTEA and GBPSO with those

Table 4
Comparison of RTEA, GBPSO, FirEGA and SecEGA for UDKP1–UDKP10.

Index	Instance	*OPT*	Algorithm	*BEST*	*Mean*	*Worst*	*StD*	*Gap*
1	UDKP1	85 740	RTEA	**85 740**	85 661	**85 477**	63.48	0.092
			GBPSO	**85 740**	**85 669**	85 459	67.51	0.082
			FirEGA	80 593	79 103	77 935	690.01	7.741
			SecEGA	78 287	76 807	75 156	798.95	10.418
2	UDKP2	163 744	RTEA	**163 744**	**163 710**	**163 592**	30.53	0.021
			GBPSO	**163 744**	**163 710**	163 566	33.08	0.021
			FirEGA	155 039	151 662	149 875	1044.95	7.379
			SecGA	148 043	145 548	143 833	883.43	11.112
3	UDKP3	269 393	RTEA	**269 393**	**269 273**	**269 089**	60.12	0.045
			GBPSO	269 340	269 124	268 504	129.52	0.100
			FirEGA	246 698	240 886	237 980	1491.97	10.582
			SecEGA	228 823	225 492	222 486	1353.58	16.296
4	UDKP4	347 599	RTEA	**347 574**	**347 507**	**347 336**	45.08	0.026
			GBPSO	347 541	347 267	346 786	147.99	0.096
			FirEGA	321 605	317 319	314 486	1426.85	8.712
			SecEGA	305 796	299 978	297 606	1435.46	13.700
5	UDKP5	442 644	RTEA	**442 627**	**442 499**	**442 303**	66.74	0.033
			GBPSO	441 693	440 555	439 151	464.36	0.472
			FirEGA	405 409	399 620	395 367	1692.23	9.720
			SecEGA	376 147	370 808	367 574	1611.71	16.230
6	UDKP6	536 578	RTEA	**536 503**	**536 067**	**534 370**	440.87	0.095
			GBPSO	534 571	532 997	531 429	707.31	0.667
			FirEGA	486 556	478 726	474 015	2233.61	10.782
			SecGA	447 438	442 499	438 809	1765.28	17.533
7	UDKP7	635 860	RTEA	**635 481**	**634 402**	**631 836**	614.55	0.229
			GBPSO	632 919	631 497	629 352	746.35	0.686
			FirEGA	568 119	560 948	556 938	2441.80	11.781
			SecEGA	529 753	521 401	518 407	1813.04	18.001
8	UDKP8	650 206	RTEA	**649 514**	**647 934**	**645 934**	659.77	0.349
			GBPSO	646 602	644 282	641 659	877.06	0.911
			FirEGA	590 137	585 286	580 684	2078.87	9.985
			SecEGA	550 645	546 678	543 836	1449.36	15.923
9	UDKP9	718 532	RTEA	**716 760**	**714 787**	**711 259**	903.15	0.521
			GBPSO	712 591	710 039	707 289	967.62	1.182
			FirEGA	655 172	649 636	645 012	2023.64	9.588
			SecEGA	613 581	602 215	605 835	2003.75	16.188
10	UDKP10	779 460	RTEA	**778 692**	**777 524**	**775 519**	573.73	0.248
			GBPSO	773 678	771 246	768 946	1027.40	1.054
			FirEGA	712 270	706 575	701 545	2013.43	9.351
			SecEGA	665 459	658 908	655 645	1723.80	15.466

of the FirEGA and SecEGA. Therefore, in the following we only compare the performances of the RTEA and GBPSO.

Because the evolutionary algorithm is a class of stochastic approximation algorithm, one usually measures the performance of two different algorithms by their average performance and stability. For the former, the fitting curve of *Gap* is a simple and effective method. As the fitting curve of *Gap* is closer to the horizontal axis, the average performance of the algorithm is better. For the stability, the *StD* value can well reflect the stability of the algorithm.

Fig. 2 shows that the computational results of RTEA are better than those of GBPSO for the instances of UDKP, WDKP, and SDKP. Although the computational results of the RTEA for the IDKP instances are worse than those of GBPSO, the disparity between them is small. Therefore, for all instances of the D{0-1}KP, the performance of the RTEA is better than that of GBPSO.

To clearly illustrate the stability of the algorithm, Fig. 3 shows the *StD* histogram of the RTEA and GBPSO. For the instances of WDKP, UDKP, and SDKP, the stability of the RTEA is better than that of GBPSO. However, the stability of GBPSO is better than the RTEA for the instances of IDKP.

In fact, from Tables 4–7 we also easily see that the maximum value of *Gap* of the RTEA is 0.521, and for GBPSO it is 1.182. This shows that the stability of the RTEA is better than that of GBPSO.

In order to further point out that RTEA more outperforms than GBPSO for solving D{0-1}KP, we use the Wilcoxon rank sum test [10–12] with the level of significance $\alpha = 0.005$ to test for differences between the RTEA and GBPSO. Table 8 reports the results of rank sum tests of the instances UDKP1–UDKP10, WDKP1–WDKP10, SDKP1–SDKP10, and IDKP1–IDKP10, respectively. In Tables 8, "1" indicates that RTEA is better than GBPAO at the 99.5% confidence. On the contrary, it is represented as "−1". In addition, "0" shows that two algorithms have similar performance.

It is not difficult to see from Table 8 that RTEA outperforms GBPSO for the 26 instances of D{0-1}KP, and they have no significant difference for the 3 instances of D{0-1}KP. RTEA is inferior to GBPSO for the 11 instances of D{0-1}KP. Therefore, the performance of RTEA is excellent than that of GBPSO for the D{0-1}KP problem.

From the above comparison, it is not difficult to see that for the D{0-1}KP problem the RTEA has the best performance, followed by GBPSO, and they are far better than those of FirEGA and SecEGA. This shows that the design method of evolutionary algorithms based on ring theory is not only feasible but is also effective.

6. Conclusions and further works

In this paper, a new method for designing evolutionary algorithms based on ring theory is advanced. By using the direct product of rings, we proposed a new evolutionary algorithm called the RTEA to solve the combinatorial optimization problems whose feasible solution is an n-dimensional integer vector in $Z[m_1, \ldots, m_n]$. To verify the performance of the RTEA, we use it to solve four

Table 5
Comparison of RTEA, GBPSO, FirEGA and SecEGA for WDKP1–WDKP10.

Index	Instance	*OPT*	Algorithm	*BEST*	*Mean*	*Worst*	*StD*	*Gap*
1	WDKP1	83 098	RTEA	**83 098**	83 085	83 036	7.88	0.016
			GBPSO	**83 098**	**83 087**	**83 058**	6.85	0.014
			FirEGA	82 803	82 693	82 592	52.04	0.487
			SecEGA	80 014	79 022	78 096	473.67	4.905
2	WDKP2	138 215	RTEA	**138 215**	138 195	138 130	15.42	0.014
			GBPSO	**138 215**	**138 202**	**138 133**	18.26	0.009
			FirEGA	137 704	137 584	137 356	63.23	0.457
			SecGA	133 315	132 276	131 337	415.62	4.297
3	WDKP3	256 616	RTEA	**256 616**	**256 592**	**256 523**	15.87	0.009
			GBPSO	**256 616**	256 573	256 493	24.60	0.017
			FirEGA	254 120	253 657	253 307	173.01	1.153
			SecEGA	238 331	235 721	234 025	873.58	8.143
4	WDKP4	315 657	RTEA	**315 655**	**315 622**	**315 566**	18.44	0.011
			GBPSO	315 653	315 605	315 493	32.05	0.017
			FirEGA	313 966	312 849	311 998	484.76	0.890
			SecEGA	293 640	290 851	288 764	950.06	7.859
5	WDKP5	428 490	RTEA	**428 485**	**428 452**	**428 377**	18.22	0.009
			GBPSO	428 484	428 419	428 303	34.54	0.017
			FirEGA	426 311	424 548	423 058	798.53	0.920
			SecEGA	393 617	390 014	387 992	1059.83	8.980
6	WDKP6	466 050	RTEA	**466 045**	**466 023**	**465 983**	13.18	0.006
			GBPSO	466 019	465 947	465 828	45.22	0.022
			FirEGA	463 185	461 672	457 718	1107.57	0.940
			SecGA	429 208	425 112	423 269	1058.37	8.784
7	WDKP7	547 683	RTEA	**547 682**	**547 631**	**547 553**	23.03	0.009
			GBPSO	547 565	547 355	547 138	87.73	0.060
			FirEGA	544 019	541 949	538 126	1224.68	1.047
			SecEGA	501 557	496 134	493 845	1230.94	9.412
8	WDKP8	576 959	RTEA	**576 946**	**576 893**	**576 813**	25.77	0.011
			GBPSO	576 800	576 597	576 339	87.49	0.063
			FirEGA	573 427	571 559	563 253	1495.36	0.936
			SecEGA	530 971	523 203	520 350	2157.09	9.317
9	WDKP9	650 660	RTEA	**650 646**	**650 592**	**650 391**	39.17	0.010
			GBPSO	650 502	650 259	649 938	107.90	0.062
			FirEGA	647 477	644 820	630 086	2056.06	0.898
			SecEGA	598 343	586 770	583 854	2315.50	9.819
10	WDKP10	678 967	RTEA	**678 957**	**678 836**	**678 500**	78.57	0.019
			GBPSO	678 862	678 662	678 401	91.12	0.045
			FirEGA	675 452	673 008	668 239	1441.96	0.878
			SecEGA	620 230	606 215	609 964	3090.86	10.715

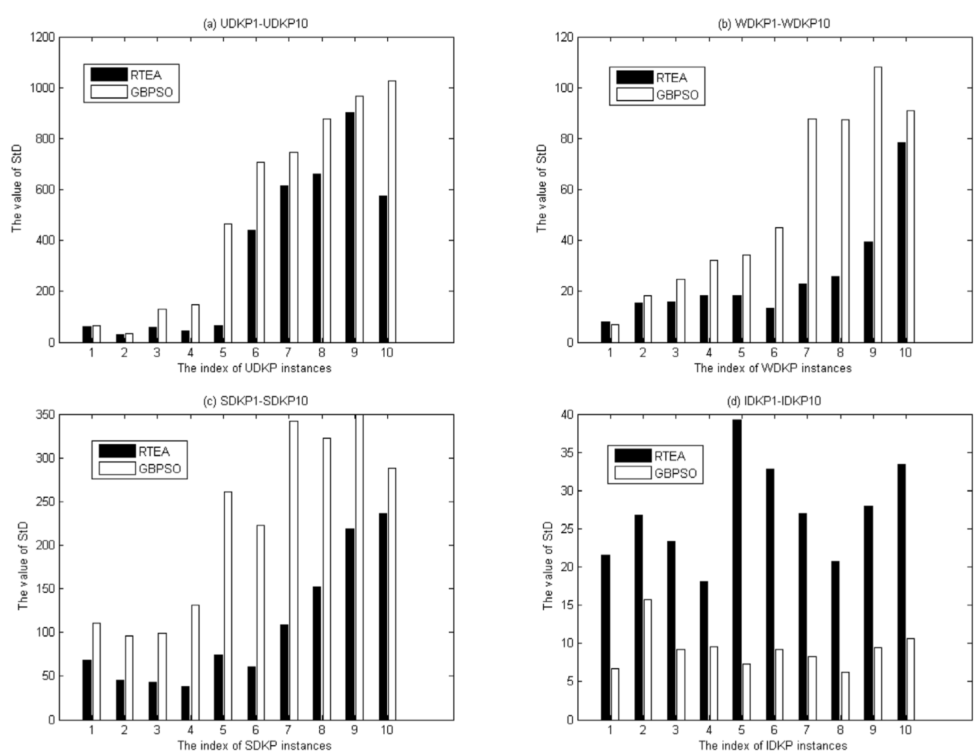

Fig. 3. The *StD* histogram: (a) UDKP1–UDKP10, (b) WDKP1–WDKP10, (c) SDKP1–SDKP10, (d) IDKP1–IDKP10.

Table 6
Comparison of RTEA, GBPSO, FirEGA and SecEGA for SDKP1–SDKP10.

Index	Instance	*OPT*	Algorithm	*BEST*	*Mean*	*Worst*	*StD*	*Gap*
1	SDKP1	94 459	RTEA	**94 449**	**94 333**	**94 132**	67.88	0.134
			GBPSO	**94 449**	94 261	93 818	110.26	0.210
			FirEGA	93 235	93 171	93 070	42.15	1.364
			SecEGA	89 769	88 832	87 463	594.91	5.958
2	SDKP2	160 805	RTEA	**160 786**	**160 702**	**160 588**	45.52	0.064
			GBPSO	160 777	160 607	160 253	95.83	0.123
			FirEGA	159 159	159 004	158 859	61.54	1.120
			SecGA	153 821	152 059	150 753	489.39	5.439
3	SDKP3	238 248	RTEA	**238 215**	**238 133**	**237 992**	43.12	0.048
			GBPSO	238 158	237 900	237 606	99.13	0.146
			FirEGA	235 454	235 241	235 043	79.86	1.262
			SecEGA	224 997	223 580	221 918	543.38	6.157
4	SDKP4	340 027	RTEA	**339 988**	**339 918**	**339 801**	38.25	0.032
			GBPSO	339 830	339 526	339 156	131.69	0.147
			FirEGA	336 353	335 963	335 709	122.41	1.195
			SecEGA	318 510	315 513	313 747	851.14	7.209
5	SDKP5	463 033	RTEA	**462 929**	**462 766**	**462 593**	74.40	0.058
			GBPSO	462 107	461 566	460 906	260.87	0.317
			FirEGA	452 900	447 587	444 255	1974.99	3.336
			SecEGA	420 238	416 964	413 933	1291.65	9.950
6	SDKP6	466 097	RTEA	**466 036**	**465 903**	**465 712**	61.21	0.042
			GBPSO	465 378	464 856	464 171	222.96	0.266
			FirEGA	459 254	458 893	458 584	162.94	1.546
			SecGA	430 738	427 304	425 504	1031.12	8.323
7	SDKP7	620 446	RTEA	**620 238**	**620 048**	**619 741**	108.31	0.064
			GBPSO	618 753	617 827	616 602	342.71	0.422
			FirEGA	599 361	592 279	579 673	3949.03	4.540
			SecEGA	561 224	556 083	552 007	1926.26	10.3747
8	SDKP8	670 697	RTEA	**670 431**	**670 151**	**669 702**	151.77	0.081
			GBPSO	668 821	668 107	667 341	322.92	0.386
			FirEGA	661 276	660 104	659 367	426.06	1.579
			SecEGA	611 644	606 263	603 774	1446.94	9.607
9	SDKP9	739 121	RTEA	**738 707**	**738 162**	**737 660**	219.06	0.130
			GBPSO	736 589	735 805	734 871	349.52	0.449
			FirEGA	729 135	727 544	727 064	343.67	1.566
			SecEGA	674 885	667 900	664 580	1614.04	9.636
10	SDKP10	765 317	RTEA	**764 821**	**764 263**	**763 665**	236.47	0.138
			GBPSO	762 603	761 980	761 258	288.41	0.436
			FirEGA	756 205	753 394	750 757	985.46	1.558
			SecEGA	708 935	695 557	691 994	2956.08	9.115

kinds of large-scale D{0-1}KP instances. A comparison among the calculation results of FirEGA, SecEGA and GBPSO indicates that the RTEA not only is better than GBPSO but is also much better than FirEGA and SecEGA. It indicates that the design method based on the evolutionary algorithm based on ring theory is feasible and effective.

Note that $m_i \in Z[m_1, \ldots, m_n]$ $(1 \leq i \leq n)$ is greater than 1 and does not require them to be the same each other. The RTEA can be used not only to solve the D{0-1}KP problem but can also be used to solve other combinatorial optimization problems, such as the set covering problem [14], the multiple-choice knapsack problem [15], and the satisfiability problem [16]. Therefore, in the future, we will search whether or not the evolution operators R-GEO and R-LDO have the versatility and effectiveness for solving other problems, whether or not the RTEA can be used to solve the numerical optimization problem, and so on. All those problems are worth discussing one by one in the future.

Acknowledgments

This article was supported by the Basic Research Project of Knowledge Innovation Program in Shenzhen, China (JCYJ20150324140036825), China Postdoctoral Science Foundation (2015M572361), the National Natural Science Foundations of China (61503252, 71371063 and 11471097), the Scientific Research Project Program of Colleges and Universities in Hebei Province, China (ZD2016005), and Natural Science Foundation of Hebei Province, China (F2016403055).

Appendix

Let array $H[0 \ldots 3n-1]$ satisfy $\frac{p_{H[0]}}{w_{H[0]}} \geq \frac{p_{H[1]}}{w_{H[1]}} \geq \cdots \geq \frac{p_{H[3n-1]}}{w_{H[3n-1]}}$, and $X = (x_0, x_1, \ldots, x_{n-1}) \in Z[4, 4, \ldots, 4] = \{0, 1, 2, 3\}^n$. Then, the pseudo-code of NROA [3] is described as follows:

Algorithm 4. *NROA*
Input: Infeasible solution $X = (x_0, x_1, \ldots, x_{n-1})$ and $H[0 \ldots 3n-1]$;
Output: feasible solution $X = (x_0, x_1, \ldots, x_{n-1})$ and $f(X)$.
$fweight \leftarrow 0; fvalue \leftarrow 0;$
for $i \leftarrow 0$ to $3n-1$ do
  $k \leftarrow \lfloor H[i]/3 \rfloor; r \leftarrow H[i](mod\ 3);$
  if $(x_k = r+1)$ and $(fweight + w_{H[i]} \leq C)$ then
    $fweight \leftarrow fweight + w_{H[i]};$
    $fvalue \leftarrow fvalue + p_{H[i]};$
  end if
  if $(x_k = r+1)$ and $(fweight + w_{H[i]} > C)$ then
  $x_k \leftarrow 0;$
end for
for $i \leftarrow 0$ to $3n-1$ do
  $k \leftarrow \lfloor H[i]/3 \rfloor; r \leftarrow H[i](mod\ 3);$
  if $(x_k = 0)$ and $(fweight + w_{H[i]} \leq C)$ then
    $fweight \leftarrow fweight + w_{H[i]};$
    $fvalue \leftarrow fvalue + p_{H[i]}; x_k \leftarrow r+1;$
  end if
end for
return$(X, fvalue)$.

Table 7
Comparison of RTEA, GBPSO, FirEGA and SecEGA for IDKP1–IDKP10.

Index	Instance	*OPT*	Algorithm	*BEST*	*Mean*	*Worst*	*StD*	*Gap*
1	IDKP1	70 106	RTEA	**70 106**	70 082	70 037	21.55	0.034
			GBPSO	**70 106**	70 098	70 077	6.68	0.011
			FirEGA	**70 106**	**70 099**	**70 090**	7.23	0.009
			SecEGA	68 663	68 000	67 369	328.44	3.004
2	IDKP2	118 268	RTEA	**118 268**	118 235	118 146	26.72	0.028
			GBPSO	**118 268**	**118 253**	**118 202**	15.69	0.013
			FirEGA	118 169	117 869	117 625	102.59	0.337
			SecGA	114 434	113 385	112 307	7446.67	4.129
3	IDKP3	234 804	RTEA	**234 804**	234 770	234 703	23.34	0.014
			GBPSO	**234 804**	**234 796**	**234 759**	9.20	0.003
			FirEGA	234 497	233 997	233 666	175.42	0.344
			SecEGA	220 096	217 982	216 313	835.83	7.164
4	IDKP4	282 591	RTEA	282 583	282 557	282 485	18.05	0.012
			GBPSO	**282 591**	**282 578**	**282 554**	9.55	0.005
			FirEGA	282 148	280 695	278 881	827.63	0.671
			SecEGA	263 238	260 425	258 922	933.40	7.844
5	IDKP5	335 584	RTEA	335 580	335 528	335 314	39.20	0.017
			GBPSO	**335 584**	**335 580**	**335 546**	7.34	0.001
			FirEGA	335 004	333 484	329 621	1173.90	0.626
			SecEGA	309 573	306 878	304 881	907.19	8.554
6	IDKP6	452 463	RTEA	452 450	452 405	452 311	32.80	0.013
			GBPSO	**452 463**	**452 452**	**452 425**	9.19	0.002
			FirEGA	451 680	449 863	446 704	1161.52	0.575
			SecGA	414 090	411 367	408 788	1099.31	9.083
7	IDKP7	489 149	RTEA	489 142	489 102	488 948	26.99	0.009
			GBPSO	**489 149**	**489 133**	**489 105**	8.27	0.003
			FirEGA	488 009	485 592	476 385	2294.28	0.727
			SecEGA	451 528	444 316	442 133	1280.31	9.166
8	IDKP8	533 841	RTEA	533 833	533 800	533 724	20.75	0.008
			GBPSO	**533 839**	**533 827**	**533 808**	6.25	0.003
			FirEGA	533 035	529 984	514 196	2308.11	0.723
			SecEGA	490 494	481 831	478 035	2215.66	9.743
9	IDKP9	528 144	RTEA	**528 144**	528 099	528 015	27.95	0.009
			GBPSO	528 140	**528 131**	**528 094**	9.47	0.002
			FirEGA	526 410	523 982	511 651	2216.13	0.788
			SecEGA	489 661	477 001	471 848	3656.22	9.684
10	IDKP10	581 244	RTEA	581 238	581 197	581 091	33.39	0.008
			GBPSO	**581 244**	**581 230**	**581 194**	10.61	0.002
			FirEGA	578 903	576 772	568 903	1905.18	0.769
			SecEGA	535 541	521 604	516 445	4265.07	10.261

Table 8
The results of Wilcoxon rank sum test with a level of significance $\alpha = 0.005$.

Instance	Result	Instance	Result	Instance	Result	Instance	Result
UDKP1	0	WDKP1	0	SDKP1	1	IDKP1	−1
UDKP2	0	WDKP2	−1	SDKP2	1	IDKP2	−1
UDKP3	1	WDKP3	1	SDKP3	1	IDKP3	−1
UDKP4	1	WDKP4	1	SDKP4	1	IDKP4	−1
UDKP5	1	WDKP5	1	SDKP5	1	IDKP5	−1
UDKP6	1	WDKP6	1	SDKP6	1	IDKP6	−1
UDKP7	1	WDKP7	1	SDKP7	1	IDKP7	−1
UDKP8	1	WDKP8	1	SDKP8	1	IDKP8	−1
UDKP9	1	WDKP9	1	SDKP9	1	IDKP9	−1
UDKP10	1	WDKP10	1	SDKP10	1	IDKP10	−1
“1”	8		8		10		0
“0”	2		1		0		0
“−1”	0		1		0		10

Note that $fvalue$ is the objective function value $f(X)$ in the second mathematical model of D{0-1}KP. Obviously, the time complexity of NROA is $O(n)$.

References

[1] B. Guldan, Heuristic and Exact Algorithms for Discounted Knapsack Problems (Master thesis), University of Erlangen-Nrnberg, Germany, 2007.
[2] A. Rong, J.R. Figueira, Kathrin Klamroth, Dynamic programming based algorithms for the discounted {0-1} knapsack problem, Appl. Math. Comput. 218 (2012) 6921–6933.
[3] Y. He, X. Wang, W. Li, X. Zhang, Y. Chen, Research on genetic algorithms for discounted {0-1} knapsack problem, Chinese J. Comput. 39 (12) (2016) 2614–2630.
[4] Y. He, X. Wang, Y. He, S. Zhao, W. Li, Exact and approximate algorithms for discounted 0-1 knapsack problem, Inform. Sci. 369 (2016) 634–647.
[5] J. Kennedy, R.C. Eberhart, A discrete binary version of the particle swarm optimization, in: Proceedings of 1997 Conference on System, Man, and Cybernetices, 1997, pp. 4104–4109.
[6] J.B. Fraleigh, A First Course in Abstract Algebra, seventh ed., Pearson Education Inc., New York, 2008.
[7] J.J. Rotman, A. M, Algebra, Pearson Education Inc., New York, 2002.
[8] D. Ashlock, Evolutionary Computation for Modeling and Optimization, Springer, New York, 2006.
[9] A. Draa, On the performances of the flower pollination algorithm - qualitative and quantitative analyses, Appl. Soft Comput. 34 (2015) 349–371.
[10] S. Garca, D. Molina, M. Lozano, F. Herrera, A study on the use of nonparametric tests for analyzing the evolutionary algorithms behaviour: A case study on the CEC2005 special session on real parameter optimization, J. Heuristics 15 (2009) 617–644.
[11] P. Sprent, N.C. Smeeton, Applied Nonparamteric Statistical Methods (4th), CRC Press, Boca Raton, 2007.
[12] Joaqun Derrac, Salvador Garca, Daniel Molina, Francisco Herrer, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[13] M.H. Kashan, A.H. Kashan, Nasim Nahavandi, A novel differential evolution algorithm for binary optimization, Comput. Optim. Appl. 55 (2013) 481–513.
[14] Y. Yu, X. Yao, Z. Zhou, On the approximation ability of evolutionary optimization with application to minimum set cover, Artificial Intelligence 180–181 (2010) 20–33.
[15] K. Khalili-Damghani, M. Nojavan, M. Tavana, Solving fuzzy multidimensional multiple-choice knapsack problems: The multi-start partial bound enumeration method versus the efficient epsilon-constraint method, Appl. Soft Comput. 13 (4) (2013) 1627–1638.
[16] D. Boughaci, B. Benhamou, H. Drias, Scatter search and genetic algorithms for MAX-SAT problems, J. Math. Modell. Algorithms Oper. Res. 7 (2) (2008) 101–124.