

# 基于遗传算法求解折扣 {0-1} 背包问题的研究

贺毅朝<sup>1)</sup> 王熙照<sup>2)</sup> 李文斌<sup>3)</sup> 张新禄<sup>4)</sup> 陈焱瑛<sup>1)</sup>

<sup>1)</sup>(石家庄经济学院信息工程学院 石家庄 050031)

<sup>2)</sup>(深圳大学计算机与软件学院 广东 深圳 518060)

<sup>3)</sup>(石家庄经济学院网络与信息安全实验室 石家庄 050031)

<sup>4)</sup>(河北师范大学数学与信息科学学院 石家庄 050024)

**摘 要** 目前,求解折扣 {0-1} 背包问题(D{0-1}KP)的主要算法是基于动态规划的具有伪多项式时间的确定性算法,当 D{0-1}KP 实例中各项的价值系数与重量系数在大范围内取值时缺乏实用性.文中基于杰出者保留策略遗传算法(EGA)求解 D{0-1}KP,首先建立了 D{0-1}KP 的两个新的数学模型;然后,为了利用 EGA 和第一数学模型求解 D{0-1}KP,提出了一种处理非正常编码个体的贪心修复与优化算法 GROA,并将其与 EGA 相结合给出了求解 D{0-1}KP 的第一遗传算法 FirEGA;紧接着,利用 EGA 和第二数学模型求解 D{0-1}KP,提出了处理非正常编码个体的另一种有效算法 NROA,并将其与 EGA 相结合给出了求解 D{0-1}KP 的第二遗传算法 SecEGA;最后,利用四类大规模 D{0-1}KP 实例,确定了 FirEGA 和 SecEGA 的交叉概率与变异概率的合理取值,比较了两个算法的实际求解性能.对四类实例的计算结果表明:FirEGA 和 SecEGA 都非常适于求解大规模的难 D{0-1}KP 实例,均能够得到一个近似比非常接近于 1 的近似解,并且 FirEGA 的平均求解性能比 SecEGA 的更优.

**关键词** 折扣 {0-1} 背包问题;遗传算法;非正常编码个体;贪心策略;修复与优化

**中图法分类号** TP18 **DOI 号** 10.11897/SP.J.1016.2016.02614

## Research on Genetic Algorithms for the Discounted {0-1} Knapsack Problem

HE Yi-Chao<sup>1)</sup> WANG Xi-Zhao<sup>2)</sup> LI Wen-Bin<sup>3)</sup> ZHANG Xin-Lu<sup>4)</sup> CHEN Yi-Ying<sup>1)</sup>

<sup>1)</sup>(College of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang 050031)

<sup>2)</sup>(College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, Guangdong 518060)

<sup>3)</sup>(Laboratory of Network and Information Security, Shijiazhuang University of Economics, Shijiazhuang 050031)

<sup>4)</sup>(College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang 050024)

**Abstract** At present, the deterministic algorithm based on dynamic programming is the main method for solving the discounted {0-1} knapsack problem (D{0-1}KP). But its complexity is pseudo polynomial time, and when the value coefficients and the weight coefficients of the D{0-1}KP instance are in a large range, the deterministic algorithm is no longer practical. In this paper, we use genetic algorithm with elitist reservation strategy (EGA) to solve the D{0-1}KP. Firstly, we establish two new mathematical models of the D{0-1}KP. Secondly, in order to use EGA to solve the D{0-1}KP based on the first mathematical model, we propose a greedy repair and optimization algorithm (GROA) to deal with the non-normal coding individual. Combining EGA with GROA, we give the first genetic algorithm (FirEGA) for solving the D{0-1}KP. Thirdly, for solving the D{0-1}KP by EGA and the second mathematical model, we propose another

收稿日期:2015-03-10;在线出版日期:2015-11-17. 本课题得到国家自然科学基金(71371063)、深圳市科技计划项目(JCYJ2015032414-0036825)和河北省高等学校科研基金(ZD2016005, Z2013110)资助. 贺毅朝,男,1969年生,教授,中国计算机学会(CCF)高级会员,主要研究领域为进化算法理论与应用、算法设计与分析、计算复杂性理论和 Group Testing 理论. E-mail: heyichao119@163.com. 王熙照,男,1963年生,博士,教授,IEEE Fellow,主要研究领域为机器学习、智能计算和遗传算法. 李文斌,男,1974年生,博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为人工智能及其应用和智能计算. 张新禄,男,1968年生,副教授,主要研究方向为智能计算和 Group Testing 理论. 陈焱瑛,女,1971年生,博士,教授,主要研究领域为智能计算、地球探测与信息技术.

algorithm named NROA, which is an greedy repair and optimization algorithm too, to deal with the non-normal coding individual, and give the second genetic algorithm (SecEGA) for solving the D{0-1}KP based on EGA and NROA. Finally, we ascertain the reasonable values of the crossover probability and the mutation probability of the FirEGA and SecEGA on the basis of the computational results of four kinds instances of D{0-1}KP. The computational results show that FirEGA and SecEGA are fit for solving the large instance of the hard D{0-1}KP, and they can obtain an approximation solution whose approximation rate is close to 1. Furthermore, the average performance of FirEGA is more efficient than SecEGA.

**Keywords** discounted {0-1} knapsack problem; genetic algorithm; non-normal coding individual; greedy strategy; repair and optimization

## 1 引言

背包问题(Knapsack problem, KP)<sup>[1-2]</sup>是计算机科学中的一个重要的 NP-Complete 问题,也是一个经典的组合优化问题,在投资决策与资源分配等方面具有重要的应用背景<sup>[3-4]</sup>. 0-1 背包问题(0-1 Knapsack Problem, 0-1 KP)<sup>[5]</sup>是最基本的 KP 问题,它的一般描述为:从  $n$  个具有价值系数与重量系数的物品(或项)中,选择若干个装入一个具有载重限制的背包,如何选择才能使装入物品的重量系数之和在不超过背包载重前提下价值系数之和达到最大? KP 问题有许多扩展形式,如有界背包问题(Bounded Knapsack Problem, BKP)、无界背包问题(Unbounded Knapsack Problem, UKP)、多维背包问题(Multidimensional Knapsack Problem, MKP)、多选择背包问题(Multiple-Choice Knapsack Problem, MCKP)和二次背包问题(Quadratic Knapsack Problem, QKP)等<sup>[6-7]</sup>,这些扩展形式大多已得到了广泛的研究与应用.

近年来,许多新颖的 KP 问题被相继提出,如随机背包问题(Stochastic Knapsack Problem, SKP)<sup>[8-9]</sup>、动态背包问题(Dynamic Knapsack Problem, DKP)<sup>[10-13]</sup>、具有单连续变量的 0-1 背包问题(0-1 Knapsack Problem with a single Continuous variable, KPC)<sup>[14-15]</sup>和折扣{0-1}背包问题(Discounted {0-1} Knapsack Problem, D{0-1}KP)<sup>[16-17]</sup>等,并取得了一些研究成果,例如 Lin 等人<sup>[8]</sup>深入研究了 SKP 中所涉及的交换策略与动态定价问题; Dizdar 等人<sup>[9]</sup>研究了 SKP 在税收最大化中的应用问题; Goldberg 和 Smith<sup>[10]</sup>提出了背包载重在两个固定值之间振荡变化的 DKP——时变背包问题(Time-Varying Knapsack

Problems, TVKP),并且利用二倍体遗传算法(Genetic Algorithms, GA)求解 TVKP 问题; Hadad 和 Lewis 等人<sup>[11-12]</sup>利用多倍体 GA 求解 TVKP 问题,并比较了几种多倍体方法求解 TVKP 的优劣; He 等人<sup>[13]</sup>将 TVKP 推广为随机时变背包问题(Randomized Time-Varying Knapsack Problems, RTVKP),并分别利用动态规划法、贪心算法和 GA 求解 RTVKP 问题; Marchand 和 Wolsey<sup>[14]</sup>首先提出了 KPC 问题,并分析了它的若干性质; Lin 等人<sup>[15]</sup>则给出了求解 KPC 问题的确定性算法; Guldán<sup>[16]</sup>首先提出了 D{0-1}KP 问题,并给出了求解它的动态规划法; Rong 等人<sup>[17]</sup>研究了 D{0-1}KP 的核(Core)问题,并基于动态规划与核相结合求解 D{0-1}KP. 对于 D{0-1}KP 问题,基于动态规划的确定性算法<sup>[16-17]</sup>是伪多项式时间的,当问题规模较大并且各项的价值系数与重量系数在较大范围内取值时,会导致算法耗费大量的求解时间而变得不实用,因此探讨如何利用进化算法(Evolutionary algorithms, EAs)快速求解 D{0-1}KP 是一个值得研究的问题.

本文利用遗传算法求解 D{0-1}KP 问题,在第 2 节中,介绍 D{0-1}KP 的定义和第一数学模型,然后建立它的两个新数学模型. 在第 3 节简单介绍具有杰出者保留策略遗传算法(EGA)<sup>[18-19]</sup>的原理,给出算法伪代码描述. 在第 4 节中,首先指出个体采用 0-1 向量编码方法时导致非正常编码个体(即编码不对应问题可行解的个体)的概率至少为  $1 - (1/2)^n$ ,然后基于贪心策略提出一种处理这种个体的修复与优化算法,并将它应用于 EGA 给出求解 D{0-1}KP 问题的第一遗传算法(记为 FirEGA). 在第 5 节中,针对个体采用整型向量的编码方法,首先给出一种新的变异算子,接着基于贪心策略提出一种新的修复与优化算法,然后将它们应用于 EGA 给出求解

D{0-1}KP 问题的第二遗传算法(记为 SecEGA). 在第 6 节中,首先根据四类 D{0-1}KP 实例的计算结果确定了 FirEGA 和 SecEGA 的交叉概率与变异概率的合理取值,然后通过对四类大规模 D{0-1}KP 实例的计算比较 FirEGA 和 SecEGA 的求解性能,计算结果表明:FirEGA 和 SecEGA 均适于求解大规模的难 D{0-1}KP 实例,从获得最好结果的能力来看它们的性能基本相当,从平均求解结果来看 FirEGA 的性能比 SecEGA 更胜一筹.最后,总结全文并展望今后进一步的研究方向.

## 2 D{0-1}KP 的定义与数学模型

D{0-1}KP 问题<sup>[16-17]</sup>中的“打折”思想源于商业领域,通常打折是商家进行促销的一种强有力的手段.例如,某商店对于商品 A 或商品 B 进行促销,如果顾客同时购买商品 A 和 B 则可以享受一个打折优惠.假设商品 A 和商品 B 的售价分别是 30 元和 50 元,商品 A 和 B 被同时购买的售价为 70 元.显然,顾客可以有四种选择:只购买商品 A,只购买商品 B,同时购买商品 A 与 B,或者商品 A 与 B 均不购买.D{0-1}KP 准确地反映了现实生活中的实际问题,能够更好地为项目决策、投资和预算控制等领域提供合理的理论模型.

下面先给出 D{0-1}KP 问题的定义和已有数学模型,然后建立它的两个新的数学模型.

**定义 1**<sup>[17]</sup>. 给定  $n$  个均含有 3 个项(或物品)的项集,项集  $i(0 \leq i \leq n-1)$  中含有的 3 个项分别记为  $3i, 3i+1, 3i+2$ ,其中前两个项  $3i$  和  $3i+1$  具有的价值系数分别为  $p_{3i}$  和  $p_{3i+1}$ ,具有的重量系数分别为  $w_{3i}$  和  $w_{3i+1}$ ;前两个项合并在一起构成第 3 个项  $3i+2$ ,它具有的价值系数为  $p_{3i+2} = p_{3i} + p_{3i+1}$ ,具有的折扣重量系数为  $w_{3i+2}$ ,满足  $w_{3i+2} < w_{3i} + w_{3i+1}$  并且  $w_{3i} < w_{3i+2}, w_{3i+1} < w_{3i+2}$ .对于每个项集  $i(0 \leq i \leq n-1)$ ,项  $3i, 3i+1, 3i+2$  中至多有一个可以被选择装入背包中.如何选择各项装入背包使得装入背包的所有项的重量系数之和在不超过背包载重  $C$  的前提下价值系数之和达到最大?

记 D{0-1}KP 的规模为项的个数  $3n$ ,则规模为  $3n$  的 D{0-1}KP 实例由价值系数集  $P = \{\{p_{3i}, p_{3i+1}, p_{3i+2}\} | 0 \leq i \leq n-1\}$ 、重量系数集  $W = \{\{w_{3i}, w_{3i+1}, w_{3i+2}\} | 0 \leq i \leq n-1\}$  和背包载重  $C$  构成.不失一般性,设  $p_j, w_j(0 \leq j \leq 3n-1)$  和  $C$  均为正整

数,并且  $w_{3i+2} \leq C(0 \leq i \leq n-1), \sum_{i=0}^{n-1} w_{3i+2} > C$ .由此,Guldan<sup>[16]</sup>给出了 D{0-1}KP 的第一数学模型,其描述如下:

$$\max f(X) = \max \sum_{i=0}^{n-1} (x_{3i} p_{3i} + x_{3i+1} p_{3i+1} + x_{3i+2} p_{3i+2}) \quad (1)$$

$$\text{subject to } x_{3i} + x_{3i+1} + x_{3i+2} \leq 1, i=0, 1, \dots, n-1 \quad (2)$$

$$\sum_{i=0}^{n-1} (x_{3i} w_{3i} + x_{3i+1} w_{3i+1} + x_{3i+2} w_{3i+2}) \leq C \quad (3)$$

$$x_{3i}, x_{3i+1}, x_{3i+2} \in \{0, 1\}, i=0, 1, \dots, n-1 \quad (4)$$

其中,二元变量  $x_j(0 \leq j \leq 3n-1)$  表示项  $j$  是否被装入背包中,即  $x_j = 1$  表示项  $j$  被装入了背包中, $x_j = 0$  表示项  $j$  没有被装入背包.显然,任意的 0-1 向量  $\mathbf{X} = [x_0, x_1, \dots, x_{3n-1}] \in \{0, 1\}^{3n}$  仅仅表示 D{0-1}KP 的一个潜在解,只有当它同时满足了约束条件(2)和(3)时才是可行解.

根据定义 1,下面我们建立 D{0-1}KP 的第二、第三数学模型.

设  $\mathbf{X} = [x_0, x_1, \dots, x_{n-1}] \in \{0, 1, 2, 3\}^n$  为一个  $n$  维整型向量,则 D{0-1}KP 的第二数学模型定义如下:

$$\max f(X) = \max \sum_{i=0}^{n-1} \lceil x_i/3 \rceil p_{3i+|x_i-1|} \quad (5)$$

$$\text{subject to } \sum_{i=0}^{n-1} \lceil x_i/3 \rceil w_{3i+|x_i-1|} \leq C \quad (6)$$

$$x_i \in \{0, 1, 2, 3\}, i=0, 1, \dots, n-1 \quad (7)$$

其中, $\lceil x \rceil$  为顶函数,整型变量  $x_i(0 \leq i \leq n-1)$  表示项集  $i$  中是否存在项被装入了背包中, $x_i = 0$  表示项集  $i$  中没有项被装入背包, $x_i = 1$  表示项  $3i$  被装入了背包中, $x_i = 2$  表示项  $3i+1$  被装入了背包中, $x_i = 3$  表示项  $3i+2$  被装入了背包中.同样地,任意整型向量  $\mathbf{X} = [x_0, x_1, \dots, x_{n-1}] \in \{0, 1, 2, 3\}^n$  仅仅表示 D{0-1}KP 的一个潜在解,只有当它满足了约束条件(6)时才是可行解.

此外,我们还可以建立 D{0-1}KP 的第三数学模型.设  $U = \{k | 0 \leq k \leq 3n-1 \text{ 且 } k \text{ 是整数}\}, S \subseteq U$  且  $S \neq \emptyset$ ,则 D{0-1}KP 的第三数学模型定义如下:

$$\max f(S) = \max \sum_{i \in S} p_i \quad (8)$$

$$\text{subject to } \sum_{i \in S} w_i \leq C \quad (9)$$

$$\lfloor i/3 \rfloor \neq \lfloor j/3 \rfloor, \forall i, j \in S \quad (10)$$

其中, $\lfloor x \rfloor$  为底函数;集合  $S$  是由装入背包中各项的下标构成,即项  $j$  被装入背包中当且仅当  $j \in S$ .显

然,任意满足  $S \subseteq U$  且  $S \neq \emptyset$  的集合  $S$  只是  $D\{0-1\}$  KP 的一个潜在解,只有它同时满足约束条件(9)与(10)时才是可行解。

### 3 遗传算法

遗传算法(Genetic Algorithms, GA)<sup>[19-25]</sup> 是 Holland 教授于 1975 年借鉴生物进化规律提出的一种进化算法,其主要特点是不需要计算目标函数的导数和梯度,也不要求目标函数具有连续性,并且算法具有内在的隐含并行性和全局寻优能力<sup>[23-24]</sup>。目前,GA 已被广泛应用于数值优化、组合优化、机器学习、图像识别、神经网络和模糊控制等众多领域<sup>[20-25]</sup>。Rudolph<sup>[18]</sup> 指出标准 GA 不具有全局收敛性,但是在 GA 中引入杰出者保留策略以后则具有了全局渐进收敛性。在本文下面的论述中,所涉及的遗传算法均是指杰出者保留策略遗传算法(EGA)。

遗传算法利用交叉算子(Crossover operator)、变异算子(Mutation operator)和选择算子(Selection operator)等 3 类遗传算子实现进化操作,常见的交叉算子有单点交叉、两点交叉和多点交叉,常用的变异算子为基本变异算子和逆转算子,而选择算子一般是按照适应度比例法进行选择。有关遗传算子的详细介绍和实现请参考文献[21-24],限于篇幅不再赘述。下面以求解最大约束优化问题  $\max f(X), X \in \{0,1\}^d$  ( $d$  为问题的规模或维数)为例,给出个体采用二进制向量编码,并利用单点交叉算子、基本变异算子和适应度比例法实现的 EGA 原理与算法伪代码描述。

记  $\mathbf{P}(t) = \{X_i(t) | 1 \leq i \leq N\}$  为 EGA 的第  $t$  ( $t \geq 0$ ) 代种群,  $X_i(t) = [x_{i1}(t), x_{i2}(t), \dots, x_{id}(t)] \in \{0,1\}^d$  为  $\mathbf{P}(t)$  中的第  $i$  个个体,  $N$  为种群规模,  $d$  为问题的维数。令  $\text{fit}(X_i(t))$  表示个体  $X_i(t)$  的适应度,  $B(t) = [b_1(t), b_2(t), \dots, b_d(t)] \in \{0,1\}^d$  为第  $t$  次迭代进化后产生的杰出者。

EGA 在第  $t+1$  次迭代进化时,首先将单点交叉算子作用于  $\mathbf{P}(t)$  中个体产生一个临时种群  $\mathbf{P}_1(t) = \{Y_i(t) = [y_{i1}(t), y_{i2}(t), \dots, y_{id}(t)] \in \{0,1\}^d | 1 \leq i \leq N\}$ ;然后将基本变异算子作用于  $\mathbf{P}_1(t)$  中个体产生第 2 个临时种群  $\mathbf{P}_2(t) = \{Z_i(t) = [z_{i1}(t), z_{i2}(t), \dots, z_{id}(t)] \in \{0,1\}^d | 1 \leq i \leq N\}$ ;接着,计算  $\mathbf{P}_2(t)$  中个体  $Z_i(t)$  ( $1 \leq i \leq N$ ) 的适应度  $\text{fit}(Z_i(t))$ ,并根据适

应度的大小在  $\mathbf{P}_2(t) \cup \{B(t)\}$  中确定  $B(t+1)$ ;最后,利用选择算子按照适应度比例法从  $\mathbf{P}_2(t)$  中选择个体构成新一代种群  $\mathbf{P}(t+1)$ 。

分别记“ $\mathbf{P}_1(t) \leftarrow \text{CROperator}(\mathbf{P}(t), p_c)$ ”、“ $\mathbf{P}_2(t) \leftarrow \text{MUOperator}(\mathbf{P}_1(t), p_m)$ ”和“ $\mathbf{P}(t+1) \leftarrow \text{SEOperator}(\mathbf{P}_2(t))$ ”表示上述 3 类遗传算子的实现过程,其中  $p_c$  为交叉概率,  $p_m$  为变异概率,记  $\text{MaxIt}$  为算法的迭代进化次数,则 EGA 的算法伪代码描述如算法 1。

#### 算法 1. EGA.

输入:  $\max f(X), X \in \{0,1\}^d$  的实例数据,参数  $N, p_c, p_m$  和  $\text{MaxIt}$

输出: 近似解(或最优解)  $B(t)$  及其目标函数值  $f(B(t))$

1. Generate initial population  $\mathbf{P}(0) = \{X_i(0) | 1 \leq i \leq N\}$  randomly;
2. Compute fitness  $\text{fit}(X_i(0))$  of individual  $X_i(0)$  ( $1 \leq i \leq N$ );
3. Determine  $B(0)$  by  $\text{fit}(X_i(0))$  ( $1 \leq i \leq N$ ) in  $\mathbf{P}(0)$ ;
4.  $t \leftarrow 0$ ;
5. WHILE ( $t \leq \text{MaxIt}$ )
6.  $\mathbf{P}_1(t) \leftarrow \text{CROperator}(\mathbf{P}(t), p_c)$ ;
7.  $\mathbf{P}_2(t) \leftarrow \text{MUOperator}(\mathbf{P}_1(t), p_m)$ ;
8. Compute fitness  $\text{fit}(Z_i(t))$  of  $Z_i(t) \in \mathbf{P}_2(t)$ , ( $1 \leq i \leq N$ );
9. Determine  $B(t+1)$  by  $\text{fit}(Z_i(t))$  in  $\mathbf{P}_2(t) \cup \{B(t)\}$ ;
10.  $\mathbf{P}(t+1) \leftarrow \text{SEOperator}(\mathbf{P}_2(t))$ ;
11.  $t \leftarrow t+1$ ;
12. END WHILE
13. RETURN( $B(t), f(B(t))$ ).

在 EGA 中,步 1、6 和 7 的时间复杂度均为  $O(dN)$ ,步 3 和 9 的时间复杂度均为  $O(N)$ ,步 10 的时间复杂度为  $O(N^2)$ ;不妨记  $O(T)$  为计算个体适应度的时间复杂度,则步 2 和 8 的时间复杂度均为  $N \times O(T)$ ;因此 EGA 的算法时间复杂度为  $O(dN) + N \times O(T) + O(N) + \text{MaxIt} \times [2O(dN) + N \times O(T) + O(N) + O(N^2)] = O(\text{MaxIt} \times N \times (d + T + N))$ 。注意到  $N$  和  $\text{MaxIt}$  通常都是关于  $d$  的线性函数,当  $T$  为  $d$  的多项式函数时,EGA 是一个具有多项式时间复杂度的进化算法。

### 4 求解 D{0-1} KP 的第一遗传算法

在利用 EAs 求解问题时,选择数学模型的关键在于个体的编码表示与模型中的可行解(或潜在解)是否一致以及 EAs 的进化算子是否适用、有效且易

于实现. 由于基本 GA 中个体编码是二进制向量, 因此 EGA 适合利用第一数学模型进行求解. 对于第二数学模型, 为了利用 EGA 进行求解, 个体编码必须为  $\{0, 1, 2, 3\}^n$  上的整型向量, 并且还需给出适用于这种编码的有效变异算子. 在第三数学模型中, 问题的潜在解是一个基数不固定的集合  $S$ , 这将导致个体的编码长度不固定, 不利于交叉算子和变异算子的实现. 为此, 本文主要基于第一、二数学模型研究如何利用 EGA 求解 D{0-1}KP 问题.

#### 4.1 个体的二进制编码与适应度

在基于第一数学模型利用 EGA 求解 D{0-1}KP 问题时, 个体  $\mathbf{X}$  的编码为一个二进制向量  $\mathbf{X} = [x_0, x_1, \dots, x_{3n-1}] \in \{0, 1\}^{3n}$ , 其中当  $x_j = 1$  ( $0 \leq j \leq 3n-1$ ) 时表示项  $j$  被装入背包, 当  $x_j = 0$  时表示项  $j$  没有被装入背包. 这种编码方法虽然简单易行, 便于 EGA 的交叉算子和变异算子的实现, 但是也存在一个明显的缺点, 即产生非正常编码个体的概率很高. 事实上, 下述结论表明当个体采用二进制向量编码时几乎所有个体均为非正常编码个体.

**定理 1.** 当个体  $\mathbf{X}$  采用二进制向量编码时,  $\mathbf{X}$  为非正常编码个体的概率不小于  $1 - (1/2)^n$ .

证明. 令  $\mathbf{X} = [x_0, x_1, \dots, x_{3n-1}] \in \{0, 1\}^{3n}$  为一个随机产生的个体, 其分量  $x_{3i}, x_{3i+1}, x_{3i+2}$  ( $0 \leq i \leq n-1$ ) 的取值组合有 8 种情形: 000, 001, 010, 011, 100, 101, 110 和 111, 因此  $x_{3i}, x_{3i+1}, x_{3i+2}$  中至多有一个值是 1 的概率为  $1/2$ , 故个体  $\mathbf{X}$  不满足约束条件(2)的概率为  $1 - (1/2)^n$ . 此外, 即使个体  $\mathbf{X}$  满足了约束条件(2), 它也不一定满足约束条件(3), 因此个体  $\mathbf{X}$  不满足约束条件(2)或(3)的概率至少为  $1 - (1/2)^n$ . 注意到 EGA 的初始种群是随机产生的, 交叉算子与变异算子均具有随机性, 所产生的个体自然也是随机的. 故当个体  $\mathbf{X}$  采用  $3n$  维二进制向量编码时,  $\mathbf{X}$  为非正常编码个体的概率不小于  $1 - (1/2)^n$ . 证毕.

显然, 如果 EGA 中存在大量非正常编码个体, 必将导致算法的求解效率极低; 此外, 当 EGA 的所有个体均为正常编码个体时, 利用目标函数值

$f(\mathbf{X}) = \sum_{i=0}^{n-1} (x_{3i} p_{3i} + x_{3i+1} p_{3i+1} + x_{3i+2} p_{3i+2})$  作为个体  $\mathbf{X}$  的适应度来比较个体之间的优劣是非常方便的. 因此, 将 EGA 中的非正常编码个体转化为正常编码个体, 不仅可以提高算法的求解效率, 还可以使用  $f(\mathbf{X})$  作为个体的适应度, 可谓一举两得.

#### 4.2 基于贪心策略的修复与优化法

在利用 EAs 求解约束优化问题时, 处理非正常编码个体的常见方法有罚函数法 (Penalty function approach)、修复法 (Repair approach)、纯正法 (Purist approach) 和分离法 (Separatist approach)<sup>[26,27]</sup>, 各种方法均有利弊, 而且不具有通用性. Michalewicz<sup>[28]</sup> 在利用 EGA 求解 0-1 KP 问题时分别采用基于贪心策略<sup>[5,7]</sup>的修复法和罚函数法处理非正常编码个体, 并指出贪心修复法比罚函数法处理非正常编码个体的效果更好. 贺毅朝等人<sup>[13,29]</sup> 在贪心修复的基础上引入优化处理, 提出了消除非正常编码个体的贪心修复与优化法, 进一步提升了 EGA 求解 0-1 KP 和 RTVKP 的性能.

事实上, 罚函数法是根据个体违反约束条件的程度, 利用一个适当的惩罚项对目标函数进行“惩罚”, 其本质是给出个体适应度测度的一种相对合理的定义方法, 并不能改善个体的编码质量, 自然也不会减少非正常编码个体的数量, 当算法中存在大量非正常编码个体时不宜使用此方法. 由定理 1 易知: 当 EGA 的个体采用二进制向量编码时, 几乎所有个体均为非正常编码个体, 因此罚函数法是不适用的. 为此, 下面借鉴文献<sup>[13,29]</sup> 中的设计思路, 基于贪心策略提出一种处理非正常编码个体的贪心修复与优化算法 (记为 GROA).

设  $\mathbf{P} = \{\{p_{3i}, p_{3i+1}, p_{3i+2}\} \mid 0 \leq i \leq n-1\}$  为 D{0-1}KP 的价值系数集,  $\mathbf{W} = \{\{\omega_{3i}, \omega_{3i+1}, \omega_{3i+2}\} \mid 0 \leq i \leq n-1\}$  为重量系数集,  $C$  为背包载重. 将  $3n$  个项根据价值系数密度 (即  $p_j/\omega_j$ ,  $0 \leq j \leq 3n-1$ ) 由大到小进行排序, 并按照排序后的顺序将各项的下标存入数组  $H[0 \dots 3n-1]$  中. 令  $\text{Flag}[0 \dots n-1]$  为一个布尔数组, 用于标识各项集的状态, 即当  $\text{Flag}[j] = 1$  时表示项集  $j$  ( $0 \leq j \leq n-1$ ) 中恰有一项被装入背包, 当  $\text{Flag}[j] = 0$  时表示项集  $j$  中没有项被装入背包. 设  $\mathbf{X} = [x_0, x_1, \dots, x_{3n-1}] \in \{0, 1\}^{3n}$  为任一个体,  $\mathbf{Y} = [y_0, y_1, \dots, y_{3n-1}] \in \{0, 1\}^{3n}$  为一个二进制向量, 则 GROA 的伪代码描述如算法 2.

##### 算法 2. GROA.

输入: 个体  $\mathbf{X} = [x_0, x_1, \dots, x_{3n-1}]$  和数组  $H[0 \dots 3n-1]$

输出: 二元向量  $\mathbf{Y} = [y_0, y_1, \dots, y_{3n-1}]$  和  $f(\mathbf{Y})$  的值

1. FOR  $i \leftarrow 0$  TO  $3n-1$  DO  $y_i \leftarrow 0$ ;
2. FOR  $i \leftarrow 0$  TO  $n-1$  DO  $\text{Flag}[i] \leftarrow 0$ ;
3.  $fweight \leftarrow 0$ ;  $fvalue \leftarrow 0$ ;  $i \leftarrow 0$ ;
4. WHILE ( $fweight < C \wedge i \leq 3n-1$ )
5. IF ( $x_{H[i]} = 1$ )  $\wedge$  ( $fweight + \omega_{H[i]} \leq C$ )  
 $\wedge$  ( $\text{Flag}[\lfloor H[i]/3 \rfloor] = 0$ ) THEN

```

6.       $fweight \leftarrow fweight + w_{H[i]}$ ;
7.       $y_{H[i]} \leftarrow 1$ ;  $Flag[\lfloor H[i]/3 \rfloor] \leftarrow 1$ ;
8.  END IF
9.   $i \leftarrow i + 1$ ;
10. END WHILE
11. FOR  $i \leftarrow 0$  TO  $3n - 1$  DO
12.  IF  $(fweight + w_{H[i]} \leq C) \wedge$ 
       $(Flag[\lfloor H[i]/3 \rfloor] = 0)$  THEN
13.     $fweight \leftarrow fweight + w_{H[i]}$ ;
14.     $y_{H[i]} \leftarrow 1$ ;  $Flag[\lfloor H[i]/3 \rfloor] \leftarrow 1$ ;
15.  END IF
16. END FOR
17. FOR  $i \leftarrow 0$  TO  $3n - 1$  DO  $fvalue \leftarrow fvalue + y_i \times p_i$ ;
18. RETURN  $(Y, fvalue)$ .

```

当输入的  $\mathbf{X}$  为非正常编码个体时, GROA 首先利用步 4~10 将其修复为正常编码个体, 并存储于  $\mathbf{Y}$  中, 然后再利用步 11~16 对  $\mathbf{Y}$  做进一步的优化. 在步 17 中所计算的  $fvalue$  为个体  $\mathbf{Y}$  的适应度  $f(\mathbf{Y})$ . 当输入的  $\mathbf{X}$  为正常编码个体时, GROA 实质上仅是对其进行优化处理. 显然, GROA 的算法时间复杂度为  $O(n)$ .

#### 4.3 求解 D{0-1}KP 的第一遗传算法

由上述分析可知: 在基于第一数学模型利用 EGA 求解 D{0-1}KP 问题时, 如果使用 GROA 对每一代种群中的每一个个体进行修复与优化, 既能够保证算法中不存在非正常编码个体, 又可利用  $f(\mathbf{X}) = \sum_{i=0}^{n-1} (x_{3i} p_{3i} + x_{3i+1} p_{3i+1} + x_{3i+2} p_{3i+2})$  作为个体  $\mathbf{X}$  的适应度. 由此, 下面利用 GROA 处理非正常编码个体, 利用单点交叉算子、变异算子和适应度比例法等 3 个遗传算子实现进化操作, 给出求解 D{0-1}KP 的第一遗传算法(记为 FirEGA).

记“ $H[0 \cdots 3n-1] \leftarrow \text{QuickSort}(\{p_j/\omega_j \mid p_j \in \mathbf{P}, \omega_j \in \mathbf{W}, 0 \leq j \leq 3n-1\})$ ”表示  $3n$  个项按照  $p_j/\omega_j$  ( $0 \leq j \leq 3n-1$ ) 由大到小的顺序排序后将各项下标依次存入数组  $H[0 \cdots 3n-1]$  中, 设  $MaxIt$  为迭代进化次数, 则 FirEGA 的算法伪代码描述如算法 3.

##### 算法 3. FirEGA.

输入:  $\mathbf{P} = \{ \{ p_{3i}, p_{3i+1}, p_{3i+2} \} \mid 0 \leq i \leq n-1 \}$ ,  $\mathbf{C}, \mathbf{W} = \{ \{ \omega_{3i}, \omega_{3i+1}, \omega_{3i+2} \} \mid 0 \leq i \leq n-1 \}$ , 参数  $N, p_c, p_m$  和  $MaxIt$

输出: 近似解(或最优解)  $B(t)$  及其目标函数值  $f(B(t))$

```

1.  $H[0 \cdots 3n-1] \leftarrow \text{QuickSort}(\{p_j/\omega_j \mid p_j \in \mathbf{P}, \omega_j \in \mathbf{W}, 0 \leq j \leq 3n-1\})$ ;
2. Generate initial population  $\mathbf{P}(0) = \{ X_i(0) \mid 1 \leq i \leq N \}$  randomly;

```

```

3. FOR  $i \leftarrow 1$  TO  $N$  DO
4.    $(X_i(0), f(X_i(0))) \leftarrow \text{GROA}(X_i(0), H[0 \cdots 3n-1])$ ;
5. END FOR
6. Determine  $B(0)$  by  $f(X_i(0))$  ( $1 \leq i \leq N$ ) in  $\mathbf{P}(0)$ ;
    $t \leftarrow 0$ ;
7. WHILE  $(t \leq MaxIt)$ 
8.    $\mathbf{P}_1(t) \leftarrow \text{CROperator}(\mathbf{P}(t), p_c)$ ;
9.    $\mathbf{P}_2(t) \leftarrow \text{MUOperator}(\mathbf{P}_1(t), p_m)$ 
10.  FOR  $i \leftarrow 1$  TO  $N$  DO
11.     $(Z_i(t), f(Z_i(t))) \leftarrow \text{GROA}(Z_i(t), H[0 \cdots 3n-1])$ ;
12.  END FOR
13.  Determine  $B(t+1)$  by  $f(Z_i(t))$  in  $\mathbf{P}_2(t) \cup \{B(t)\}$ ;
14.   $\mathbf{P}(t+1) \leftarrow \text{SEOperator}(\mathbf{P}_2(t))$ 
15.   $t \leftarrow t + 1$ ;
16. END WHILE
17. RETURN  $(B(t), f(B(t)))$ .

```

在 FirEGA 中, 步 1 可以利用快速排序算法 (QuickSort)<sup>[5,7]</sup> 实现, 其时间复杂度为  $O(n \log n)$ ; 显然, 步 2 与步 3~5 的时间复杂度均为  $O(Nn)$ , 步 6 的时间复杂度为  $O(N)$ ; 在步 7~16 中, 步 8、9 和步 10~12 的时间复杂度均为  $O(Nn)$ , 步 13 的时间复杂度为  $O(N)$ , 步 14 的时间复杂度为  $O(N^2)$ ; 注意到  $N$  和  $MaxIt$  是关于  $n$  的线性函数, 故 FirEGA 的时间复杂度为  $O(n \log n) + 2O(Nn) + O(N) + MaxIt \times (3O(Nn) + O(N) + O(N^2)) = O(n^3)$ , FirEGA 是一个复杂度为多项式时间的进化算法.

## 5 求解 D{0-1}KP 的第二遗传算法

在本节中, 基于第二数学模型利用 EGA 求解 D{0-1}KP 问题, 首先分析个体采用  $\{0, 1, 2, 3\}^n$  上整型向量编码的优缺点, 并给出一种适用于整型向量编码的新变异算子; 然后借鉴 GROA 的算法设计思想, 提出一种适用于这种编码方法的新贪心修复与优化算法(记为 NROA), 并利用 EGA 与 NROA 相结合给出求解 D{0-1}KP 问题的第二遗传算法(记为 SecEGA).

### 5.1 个体的整型编码与变异算子

在利用 EGA 和第二数学模型求解 D{0-1}KP 问题时, 由于个体  $\mathbf{X}$  采用  $\{0, 1, 2, 3\}^n$  上的整型向量进行编码, 即  $\mathbf{X} = [x_0, x_1, \dots, x_{n-1}] \in \{0, 1, 2, 3\}^n$ , 只要  $\mathbf{X}$  满足式(6)即为问题的一个可行解, 因此容易检验个体是否为非正常编码个体. 但是, 这种编码

方法也存在两点不足:其一是不可避免地产生非正常编码个体;其二是 EGA 的基本变异算子对于这种编码方法不再适用.下面针对整型向量编码方法,首先给出一种新的变异算子,并证明此编码方法的一个性质,然后提出一种处理非正常编码个体的新算法.

注意到个体  $\mathbf{X}=[x_0, x_1, \dots, x_{n-1}] \in \{0, 1, 2, 3\}^n$  中分量有 4 种可能的取值,借鉴 EGA 基本变异算子的设计思想,基于等概率随机变异原则提出一种适用于  $\{0, 1, 2, 3\}^n$  上整型向量编码的变异算子 (Modified Mutation Operator, MMOperator).

设  $\mathbf{P}=\{X_i | X_i \in \{0, 1, 2, 3\}^n, 1 \leq i \leq N\}$  为算法的当前种群,其规模为  $N$ ,  $\mathbf{X}_i=[x_{i0}, x_{i1}, \dots, x_{i,n-1}] \in \{0, 1, 2, 3\}^n$  为  $\mathbf{P}$  中第  $i$  ( $1 \leq i \leq N$ ) 个个体.记  $\text{rand}[0, 1]$  为区间  $[0, 1]$  上的一个随机实数,  $\text{rand}(S)$  表示从集合  $S$  中随机选择一个元素,  $p_m$  为变异概率,则 MMOperator 的伪代码由下述过程描述.

#### 过程 1. MMOperator.

输入: 种群  $\mathbf{P}=\{X_i | X_i \in \{0, 1, 2, 3\}^n, 1 \leq i \leq N\}$  和变异概率  $p_m$

输出: 对  $\mathbf{P}$  中每一个体进行变异操作后产生的新种群

1. FOR  $i \leftarrow 1$  TO  $N$  DO
2. FOR  $j \leftarrow 0$  TO  $n-1$  DO
3. IF ( $\text{rand}[0, 1] < p_m$ ) THEN  $x_{ij} \leftarrow \text{rand}(\{0, 1, 2, 3\} - \{x_{ij}\})$ ;
4. END FOR
5. END FOR
6. RETURN( $\mathbf{P}$ ).

显然,MMOperator 将种群  $\mathbf{P}$  的第  $i$  个个体  $X_i$  中的任一分量  $x_{ij}$  ( $0 \leq j \leq n-1$ ) 以概率  $p_m$  随机变异为  $\{0, 1, 2, 3\} - \{x_{ij}\}$  中的任一个元素,其算法时间复杂度为  $O(Nn)$ .

事实上,当个体采用  $\{0, 1, 2, 3\}^n$  上的  $n$  维整型向量进行编码时具有以下性质.

**定理 2.** 当个体  $\mathbf{X}=[x_0, x_1, \dots, x_{n-1}]$  中每一维分量以等概率随机取  $\{0, 1, 2, 3\}$  中的值时,  $\mathbf{X}$  所对应的潜在解使得欲装入背包中项的重量系数之和  $\mathbf{W}(\mathbf{X})$  几乎总是满足  $\mathbf{W}(\mathbf{X}) > \frac{1}{2} \sum_{i=0}^{n-1} \omega_{3i+2}$ .

证明. 不妨设  $S_0 = \{i | x_i = 0, x_i \in \mathbf{X}\}$ ,  $S_1 = \{i | x_i = 1, x_i \in \mathbf{X}\}$ ,  $S_2 = \{i | x_i = 2, x_i \in \mathbf{X}\}$ ,  $S_3 = \{i | x_i = 3, x_i \in \mathbf{X}\}$ , 则在等概率的条件下  $|S_0| = |S_1| = |S_2| = |S_3|$ . 于是,当  $x_i$  ( $0 \leq i \leq n-1$ ) 以等概率随机取  $\{0, 1, 2, 3\}$  中的值时几乎总有  $\sum_{i \in S_1} \omega_{3i} = \frac{1}{4} \sum_{i=0}^{n-1} \omega_{3i}$ ,

$$\sum_{i \in S_2} \omega_{3i+1} = \frac{1}{4} \sum_{i=0}^{n-1} \omega_{3i+1}, \sum_{i \in S_3} \omega_{3i+2} = \frac{1}{4} \sum_{i=0}^{n-1} \omega_{3i+2}, \text{ 故而}$$

$$\mathbf{W}(\mathbf{X}) = \sum_{i \in S_1} \omega_{3i} + \sum_{i \in S_2} \omega_{3i+1} + \sum_{i \in S_3} \omega_{3i+2} = \frac{1}{4} \sum_{i=0}^{n-1} (\omega_{3i+1} + \omega_{3i+2}) + \frac{1}{4} \sum_{i=0}^{n-1} \omega_{3i+2} > \frac{1}{2} \sum_{i=0}^{n-1} \omega_{3i+2}. \quad \text{证毕.}$$

这样,当个体采用  $\{0, 1, 2, 3\}^n$  上的  $n$  维整型向量编码时,对于背包载重满足  $C < \frac{1}{2} \sum_{i=0}^{n-1} \omega_{3i+2}$  的所有  $D\{0-1\}$  KP 实例,几乎总有  $\mathbf{W}(\mathbf{X}) > C$ ,从而算法随机产生的个体几乎均为非正常编码个体,显然不宜利用罚函数法进行处理.为此,下面提出一种适用于  $\{0, 1, 2, 3\}^n$  上整型向量编码个体的贪心修复与优化算法 (NROA) 来处理非正常编码个体.

## 5.2 整型向量的贪心修复与优化法

类似于 GROA 的算法设计思路,在 NROA 中,如果输入个体  $\mathbf{X}$  是非正常编码个体,首先将其修复为正常编码个体,然后再对修复后的结果做进一步的优化,并计算修复与优化后  $\mathbf{X}$  对应的目标函数值作为其适应度.如果  $\mathbf{X}$  是正常编码个体,则仅对  $\mathbf{X}$  进行优化与计算其适应度.

令集合  $\mathbf{P}$ 、集合  $\mathbf{W}$ 、整数  $C$  及数组  $H[0 \cdots 3n-1]$  与 4.2 节中相同,并设  $\mathbf{X}=[x_0, x_1, \dots, x_{n-1}] \in \{0, 1, 2, 3\}^n$  表示个体的  $n$  维整型向量编码,  $H[i] \pmod{3}$  表示对  $H[i]$  进行模 3 取余运算,则 NROA 的算法伪代码描述如算法 4.

#### 算法 4. NROA.

输入: 个体  $\mathbf{X}=[x_0, x_1, \dots, x_{n-1}]$  和数组  $H[0 \cdots 3n-1]$

输出: 修复与优化后的  $\mathbf{X}=[x_0, x_1, \dots, x_{n-1}]$  及其适应度  $f(\mathbf{X})$

1.  $fweight \leftarrow 0$ ;  $fvalue \leftarrow 0$ ;
2. FOR  $i \leftarrow 0$  TO  $3n-1$  DO
3.  $k \leftarrow \lfloor H[i]/3 \rfloor$ ;  $r \leftarrow H[i] \pmod{3}$ ;
4. IF  $(x_k = r+1) \wedge (fweight + \omega_{H[i]} \leq C)$  THEN
5.  $fweight \leftarrow fweight + \omega_{H[i]}$ ;
6.  $fvalue \leftarrow fvalue + p_{H[i]}$ ;
7. ELSE IF  $(x_k = r+1) \wedge (fweight + \omega_{H[i]} > C)$  THEN  $x_k \leftarrow 0$ ;
8. END IF
9. END FOR
10. FOR  $i \leftarrow 0$  TO  $3n-1$  DO
11.  $k \leftarrow \lfloor H[i]/3 \rfloor$ ;  $r \leftarrow H[i] \pmod{3}$ ;
12. IF  $(x_k = 0) \wedge (fweight + \omega_{H[i]} \leq C)$  THEN
13.  $fweight \leftarrow fweight + \omega_{H[i]}$ ;
14.  $fvalue \leftarrow fvalue + p_{H[i]}$ ;  $x_k \leftarrow r+1$ ;
15. END IF

15. END FOR  
16. RETURN ( $X, fvalue$ ).

在 NROA 中,步 2~8 的作用是将非正常编码个体  $\mathbf{X}=[x_0, x_1, \dots, x_{n-1}]$  修复为正常编码个体,并将修复后的  $\mathbf{X}$  作为可行解计算装入背包的项的价值系数之和与重量系数之和;当个体  $\mathbf{X}$  为正常编码个体时,仅计算  $\mathbf{X}$  对应的项的价值系数之和与重量系数之和.在步 9~15 中,NROA 对个体  $\mathbf{X}$  做进一步的优化处理,以提高其编码质量;此外,在步 16 输出的  $fvalue$  是修复与优化后个体  $\mathbf{X}$  的适应度  $f(\mathbf{X}) = \max \sum_{i=0}^{n-1} \lceil x_i/3 \rceil p_{3i+|x_i-1|}$ .显然,NROA 的算法时间复杂度也为  $O(n)$ .

### 5.3 求解 D{0-1}KP 的第二遗传算法

在 SecEGA 中,所使用的 3 类遗传算子分别为两点交叉算子、变异算子 MMOperator 和基于适应度比例法的选择算子. SecEGA 的算法实现流程与 FirEGA 基本相同,即首先随机生成初始种群  $\mathbf{P}(0)$ ,利用 NROA 对  $\mathbf{P}(0)$  中所有个体进行修复与优化,并确定  $B(0)$ .在 SecEGA 的第  $t+1(t \geq 0)$  次迭代进化中,首先由两点交叉算子作用于  $\mathbf{P}(t)$  产生  $\mathbf{P}_1(t)$ ,再由变异算子 MMOperator 作用于  $\mathbf{P}_1(t)$  产生  $\mathbf{P}_2(t)$ ;然后利用 NROA 对  $\mathbf{P}_2(t)$  中所有个体进行修复与优化,并在  $\mathbf{P}_2(t) \cup \{B(t)\}$  中确定  $B(t+1)$ ;最后,利用适应度比例法由  $\mathbf{P}_2(t)$  产生  $\mathbf{P}(t+1)$ .反复进行迭代操作,直到满足迭代终止条件为止,输出  $B(t)$  和  $f(B(t))$  并结束算法.

不妨记“ $\mathbf{P}_1(t) \leftarrow \text{TROperator}(\mathbf{P}(t), p_c)$ ”和“ $\mathbf{P}_2(t) \leftarrow \text{MMOperator}(\mathbf{P}_1(t), p_m)$ ”分别表示两点交叉算子和变异算子的实现过程,仍然记“ $H[0 \dots 3n-1] \leftarrow \text{QuickSort}(\{p_j/\omega_j \mid p_j \in \mathbf{P}, \omega_j \in \mathbf{W}, 0 \leq j \leq 3n-1\})$ ”表示  $3n$  个项按照  $p_j/\omega_j (0 \leq j \leq 3n-1)$  由大到小的顺序排序后将其下标依次存入数组  $H[0 \dots 3n-1]$  中,记  $MaxIt$  为算法的迭代进化次数,则 SecEGA 的算法伪代码描述如算法 5.

#### 算法 5. SecEGA.

输入:  $\mathbf{P} = \{\{p_{3i}, p_{3i+1}, p_{3i+2} \mid 0 \leq i \leq n-1\}, \mathbf{C}, \mathbf{W} = \{\{\omega_{3i}, \omega_{3i+1}, \omega_{3i+2} \mid 0 \leq i \leq n-1\},$  参数  $N, p_c, p_m, MaxIt$

输出: 近似解(或最优解) $B(t)$ 及其目标函数值  $f(B(t))$

1.  $H[0 \dots 3n-1] \leftarrow \text{QuickSort}(\{p_j/\omega_j \mid p_j \in \mathbf{P}, \omega_j \in \mathbf{W}, 0 \leq j \leq 3n-1\})$ ;
2. Generate initial population  $\mathbf{P}(0) = \{X_i(0) \mid 1 \leq i \leq N\}$  randomly;
3. FOR  $i \leftarrow 1$  TO  $N$  DO

4.  $(X_i(0), f(X_i(0))) \leftarrow \text{NROA}(X_i(0), H[0 \dots 3n-1])$ ;
5. END FOR
6. Determine  $B(0)$  by  $f(X_i(0)) (1 \leq i \leq N)$  in  $\mathbf{P}(0)$ ;  
 $t \leftarrow 0$ ;
7. WHILE ( $t \leq MaxIt$ )
8.  $\mathbf{P}_1(t) \leftarrow \text{TROperator}(\mathbf{P}(t), p_c)$ ;
9.  $\mathbf{P}_2(t) \leftarrow \text{MMOperator}(\mathbf{P}_1(t), p_m)$ ;
10. FOR  $i \leftarrow 1$  TO  $N$  DO
11.  $(Z_i(t), f(Z_i(t))) \leftarrow \text{NROA}(Z_i(t), H[0 \dots 3n-1])$ ;
12. END FOR
13. Determine  $B(t+1)$  by  $f(Z_i(t))$  in  $\mathbf{P}_2(t) \cup \{B(t)\}$ ;
14.  $\mathbf{P}(t+1) \leftarrow \text{SEOperator}(\mathbf{P}_2(t))$
15.  $t \leftarrow t+1$ ;
16. END WHILE
17. RETURN ( $B(t), f(B(t))$ ).

在 SecEGA 中,由于 **QuickSort** 的时间复杂度为  $O(n \log n)$ ,NROA 的时间复杂度为  $O(n)$ ,两点交叉算子和改进变异算子的时间复杂度均为  $O(nN)$ ,选择算子的时间复杂度为  $O(N^2)$ ,类似于 FirEGA 的分析易知,SecEGA 的时间复杂度也为  $O(n^3)$ ,即它也是一个复杂度为多项式时间的进化算法.

虽然 SecEGA 与 FirEGA 的算法时间复杂度均为  $O(n^3)$ ,但是由于 SecEGA 的个体编码长度仅为 FirEGA 的  $1/3$ ,在实现 3 类遗传算子时其速度更快,所以对于同一个 D{0-1}KP 实例,在相同的迭代次数下 SecEGA 比 FirEGA 所耗费的时间更少.

## 6 实例计算与比较

在本节中,首先给出了四类 D{0-1}KP 实例——不相关 D{0-1}KP 实例(Uncorrelated instances of D{0-1}KP,UDKP)、弱相关 D{0-1}KP 实例(Weakly correlated instances of D{0-1}KP,WDKP)、强相关 D{0-1}KP 实例(Strongly correlated instances of D{0-1}KP,SDKP)和逆向强相关 D{0-1}KP 实例(Inverse strongly correlated instances of D{0-1}KP, IDKP)<sup>[17,30]</sup>的参数设置,并据此对每一类分别随机生成 10 个不同规模的实例;然后,利用若干实例的计算结果所对应的箱线图(Boxplot)分析并确定 FirEGA 和 SecEGA 的交叉概率  $p_c$  和变异概率  $p_m$  的合理取值;最后,利用对四类大规模 D{0-1}KP 实例的各种计算结果比较 FirEGA 和 SecEGA 的求解性能.



本文所使用微型计算机为 Acer Aspire E1-570G 笔记本, 硬件配置为 Intel(R) Core(TM) i5-3337u CPU-1.8GHz, 4GB DDR3 内存(3.82GB 可用), 操作系统为 Microsoft Windows 8. 利用 C++ 语言进行编程, 编译环境为 Visual C++6.0, 并利用 MATLAB 7.10.0.499 (R2010a) 绘制箱线图.

### 6.1 四类 D{0-1}KP 实例

四类 D{0-1}KP 实例的规模分别为  $300 \leq 3n \leq 3000$ , 实例编号和参数设置如下所述(具体数据请参考下面网址中 pdf 文档<sup>①</sup>或邮件索取 Word 文档).

(1) UDKP 实例. 实例编号为 UDKP1~10, 参数设置为  $w_{3i} \in_R [2, 1000]$ ,  $w_{3i+1} \in_R [2, 1000]$ , 且  $w_{3i} < w_{3i+1}$ ,  $w_{3i+2} \in_R [w_{3i+1} + 1, w_{3i} + w_{3i+1} - 1]$ ,  $p_{3i} \in_R [1, 1000]$ ,  $p_{3i+1} \in_R [1, 1000]$ , 且  $p_{3i} < p_{3i+1}$ ,  $p_{3i+2} = p_{3i} + p_{3i+1}$ .

(2) WDKP 实例. 实例编号为 WDKP1~10, 参数设置为  $w_{3i} \in_R [101, 1000]$ ,  $w_{3i+1} \in_R [101, 1000]$ ,  $w_{3i} < w_{3i+1}$ ,  $w_{3i+2} \in_R [w_{3i+1} + 1, w_{3i} + w_{3i+1} - 1]$ ,  $p_{3i} \in_R [w_{3i} - 100, w_{3i} + 100]$ ,  $p_{3i+1} \in_R [w_{3i+1} - 100, w_{3i+1} + 100]$ , 且  $p_{3i} < p_{3i+1}$ ,  $p_{3i+2} = p_{3i} + p_{3i+1}$ .

(3) SDKP 实例. 实例编号为 SDKP1~10, 参数设置为  $w_{3i} \in_R [2, 1000]$ ,  $w_{3i+1} \in_R [2, 1000]$ , 且  $w_{3i} < w_{3i+1}$ ,  $w_{3i+2} \in_R [w_{3i+1} + 1, w_{3i} + w_{3i+1} - 1]$ ,  $p_{3i} = w_{3i} + 100$ ,  $p_{3i+1} = w_{3i+1} + 100$ ,  $p_{3i+2} = p_{3i} + p_{3i+1}$ .

(4) IDKP 实例. 实例编号为 IDKP1~10, 参数设置为  $p_{3i} \in_R [2, 1000]$ ,  $p_{3i+1} \in_R [2, 1000]$ ,  $p_{3i} < p_{3i+1}$  且  $p_{3i+2} = p_{3i} + p_{3i+1}$ ,  $w_{3i} = p_{3i} + 100$ ,  $w_{3i+1} =$

$p_{3i+1} + 100$ ,  $w_{3i+2} \in_R [w_{3i+1} + 1, w_{3i} + w_{3i+1} - 1]$ .

其中,  $x \in_R [A, B]$  表示  $x$  是随机取自区间  $[A, B]$

上的整数; 实例的背包载重取为  $C = \left\lfloor \alpha \sum_{i=0}^{n-1} w_{3i+2} \right\rfloor$ ,  $\alpha$  是  $[0.45, 0.75]$  上的一个随机实数.

### 6.2 确定 FirEGA 中 $p_m$ 和 $p_c$ 的合理取值

在 FirEGA 中, 正常编码个体的二进制编码中“0”的个数至少是“1”的个数的 2 倍, 显然交叉算子对于改变个体的这种结构形式作用不大. 但为了能够获得更优秀的个体, FirEGA 必须打破个体的这种结构形式, 因此只能借助于变异算子来实现这一突破. 据此我们推断: 在 FirEGA 中变异概率  $p_m$  的取值变化对算法的求解性能将起到关键的作用, 而交叉概率  $p_c$  对算法的影响并不会太大.

为了验证以上推断, 下面分别利用四类 D{0-1}KP 实例对  $p_c$  和  $p_m$  的每一种组合形式进行计算, 以确定  $p_c$  和  $p_m$  的合理取值. 限于篇幅, 下面仅给出规模  $3n = 900$  的实例 UDKP3、WDKP3、SDKP3 和 IDKP3 的计算结果及其所对应的箱线图.

对  $p_c$  分别取值 0.2, 0.5 和 0.8, 对  $p_m$  分别取值 0.0, 0.01, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 共构成 42 种不同的组合形式  $(p_c, p_m)$ , 给每种组合形式赋予一个序号 id, 并将它们在表 1 中列出. 对于每一种组合形式  $(p_c, p_m)$ , 利用 FirEGA 对实例 UDKP3、WDKP3、SDKP3 和 IDKP3 分别独立计算 30 次, 所得到的 30 个最好值的统计结果分布情况由图 1~图 4 中的箱线图给出.

表 1 FirEGA 中参数  $p_c$  和  $p_m$  构成的 42 种组合形式  $(p_c, p_m)$  及其序号 id

id	$(p_c, p_m)$	id	$(p_c, p_m)$	id	$(p_c, p_m)$	id	$(p_c, p_m)$	id	$(p_c, p_m)$	id	$(p_c, p_m)$
1	(0.2, 0.0)	8	(0.2, 0.4)	15	(0.5, 0.0)	22	(0.5, 0.4)	29	(0.8, 0.0)	36	(0.8, 0.4)
2	(0.2, 0.01)	9	(0.2, 0.5)	16	(0.5, 0.01)	23	(0.5, 0.05)	30	(0.8, 0.01)	37	(0.8, 0.05)
3	(0.2, 0.03)	10	(0.2, 0.6)	17	(0.5, 0.03)	24	(0.5, 0.06)	31	(0.8, 0.03)	38	(0.8, 0.06)
4	(0.2, 0.05)	11	(0.2, 0.7)	18	(0.5, 0.05)	25	(0.5, 0.07)	32	(0.8, 0.05)	39	(0.8, 0.07)
5	(0.2, 0.1)	12	(0.2, 0.8)	19	(0.5, 0.1)	26	(0.5, 0.08)	33	(0.8, 0.1)	40	(0.8, 0.08)
6	(0.2, 0.2)	13	(0.2, 0.9)	20	(0.5, 0.2)	27	(0.5, 0.09)	34	(0.8, 0.2)	41	(0.8, 0.09)
7	(0.2, 0.3)	14	(0.2, 1.0)	21	(0.5, 0.3)	28	(0.5, 1.0)	35	(0.8, 0.3)	42	(0.8, 1.0)

从图 1~图 4 可以看出: 对于实例 UDKP3, 当  $p_m = 1.0$  时 FirEGA 的求解效果最优; 而对于实例 WDKP3、SDKP3 和 IDKP3, 当  $p_m = 0.01$  时 FirEGA 不仅求解效果很好, 而且算法的稳定性极佳. 此外, 从图中还容易看出  $p_c$  的取值变化确实对 FirEGA 的求解效果影响不大. 这样, 在利用 FirEGA 求解 UDKP 类实例时, 设置  $p_c = 0.5$  或  $0.8$  且  $p_m = 1.0$  是合理的选择; 而对于 WDKP 类、SDKP 类和 IDKP 类实例, 设置  $p_c = 0.5$  或  $0.8$  且  $p_m = 0.01$  是合理的选择.

由于 EGA 是一类对参数敏感的进化算法, 对不同类型的问题甚至同一类型问题的不同实例类, 其参数的选取往往会出现差异, 所以 FirEGA 求解

① UDKP: <http://wenku.baidu.com/view/35a54e7503d8ce2-f06623a6>  
WDKP: <http://wenku.baidu.com/view/845b040e168884-868762d6bf>  
SDKP: <http://wenku.baidu.com/view/e62b015f4b73f24-2336c5fc8>  
IDKP: <http://wenku.baidu.com/view/d07492ada58da01-16c1749d7>

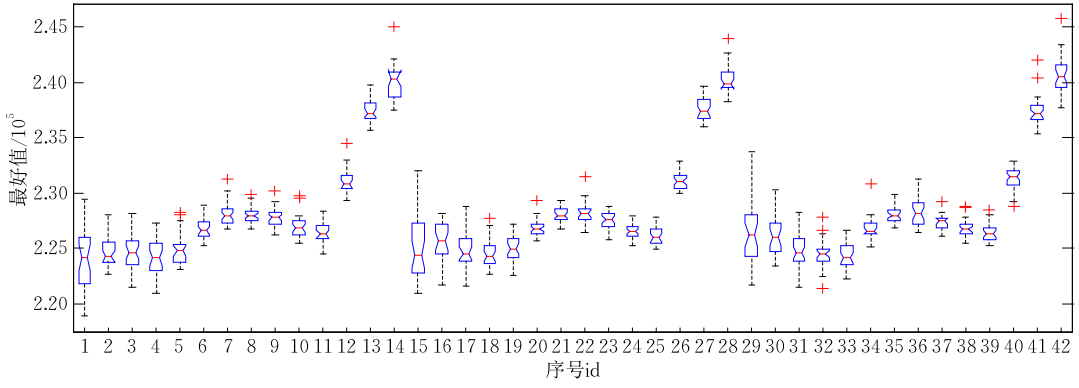


图 1 在 42 种组合形式下 FirEGA 求解 UDKP3 的性能比较

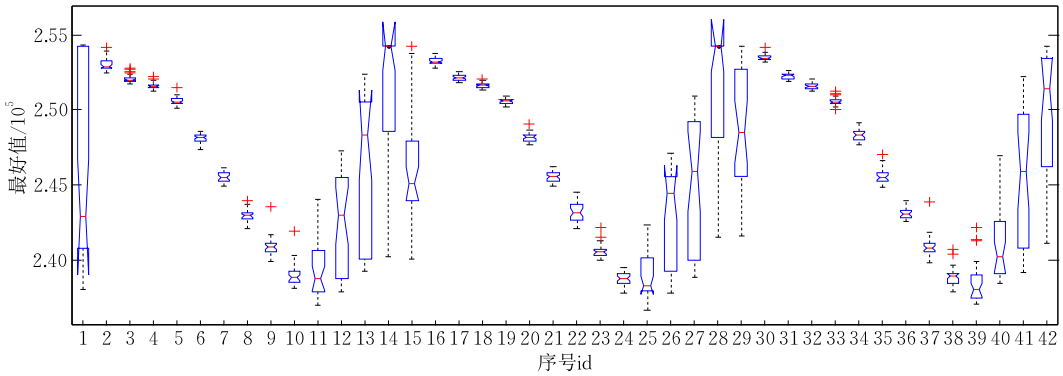


图 2 在 42 种组合形式下 FirEGA 求解 WDKP3 的性能比较

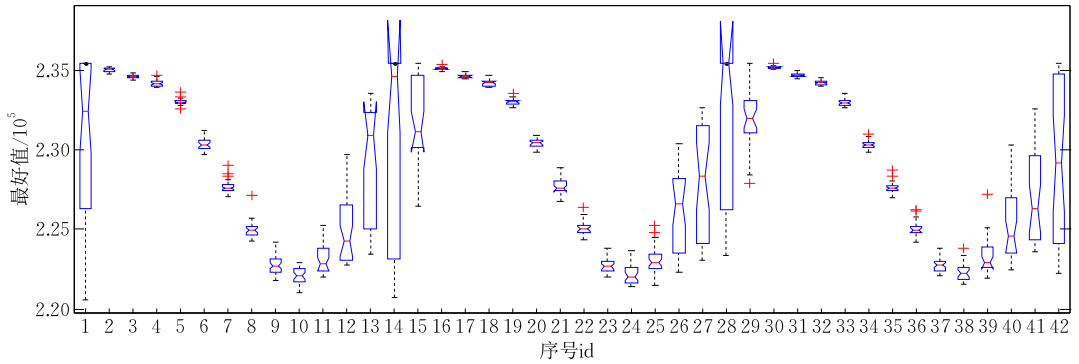


图 3 在 42 种组合形式下 FirEGA 求解 SDKP3 的性能比较

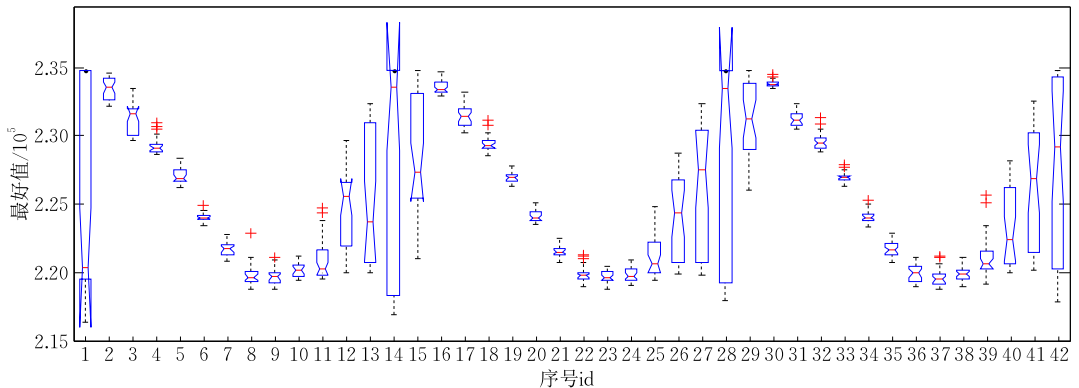


图 4 在 42 种组合形式下 FirEGA 求解 IDKP3 的性能比较

UDKP 类时变异概率  $p_m$  的取值与其它实例类不同是正常的, 这是由 UDKP 类实例的价值系数与重量系数在取值区间  $[A, B]$  上是随机均匀分布导致的.

### 6.3 确定 SecEGA 中 $p_m$ 和 $p_c$ 的合理取值

在 SecEGAD 中, 无论是两点交叉算子, 还是新的变异算子, 既可能使新产生的个体变得更好, 也可能变得更差, 无法判断它们能否改善个体的质量, 因此无法推断交叉概率  $p_c$  和变异概率  $p_m$  对算法求解性能的影响. 为此, 下面分别利用四类 D{0-1}KP 实例对  $p_c$  和  $p_m$  的每一种组合形式进行计算, 以确定它们的合理取值. 限于篇幅, 下面仅给出实例 UDKP3、WDKP3、SDKP3 和 IDKP3 的计算结果及其所对应的箱线图.

对  $p_m$  分别取值 0.0, 0.01, 0.05, 0.2, 0.5 和

0.8, 对  $p_c$  分别取值 0.1, 0.3, 0.5, 0.7, 0.8, 0.9 和 1.0, 共构成 42 种不同的组合形式  $(p_m, p_c)$ , 其序号仍用 id 表示, 并在表 2 中列出所有的组合形式及其序号 id. 对于每一种组合形式  $(p_m, p_c)$ , 利用 SecEGA 对实例 UDKP3、WDKP3、SDKP3 和 IDKP3 分别独立计算 30 次, 所有 30 个最好值的统计结果分布情况由图 5~图 8 中的箱线图给出.

从图 5~图 8 的箱线图分布情况可以看出: 对于四类 D{0-1}KP 实例, 当  $p_m=0.01$  时 SecEGA 的求解效果普遍较好; 对于固定的  $p_m=0.01$ , 当  $p_c=0.8$  或 1.0 时算法的求解效果相对更优. 由此可知, 在利用 SecEGA 求解各类 D{0-1}KP 实例时, 设置交叉概率  $p_c=0.8$  或 1.0, 变异概率  $p_m=0.01$  是比较合理的.

表 2 SecEGA 的参数  $p_m$  和  $p_c$  构成的 42 种组合形式  $(p_m, p_c)$  及其序号 id

id	$(p_m, p_c)$	id	$(p_m, p_c)$	id	$(p_m, p_c)$	id	$(p_m, p_c)$	id	$(p_m, p_c)$	id	$(p_m, p_c)$
1	(0.0, 0.1)	8	(0.01, 0.1)	15	(0.05, 0.1)	22	(0.2, 0.1)	29	(0.5, 0.1)	36	(0.8, 0.1)
2	(0.0, 0.3)	9	(0.01, 0.3)	16	(0.05, 0.3)	23	(0.2, 0.3)	30	(0.5, 0.3)	37	(0.8, 0.3)
3	(0.0, 0.5)	10	(0.01, 0.5)	17	(0.05, 0.5)	24	(0.2, 0.5)	31	(0.5, 0.5)	38	(0.8, 0.5)
4	(0.0, 0.7)	11	(0.01, 0.7)	18	(0.05, 0.7)	25	(0.2, 0.7)	32	(0.5, 0.7)	39	(0.8, 0.7)
5	(0.0, 0.8)	12	(0.01, 0.8)	19	(0.05, 0.8)	26	(0.2, 0.8)	33	(0.5, 0.8)	40	(0.8, 0.8)
6	(0.0, 0.9)	13	(0.01, 0.9)	20	(0.05, 0.9)	27	(0.2, 0.9)	34	(0.5, 0.9)	41	(0.8, 0.9)
7	(0.0, 1.0)	14	(0.01, 1.0)	21	(0.05, 1.0)	28	(0.2, 1.0)	35	(0.5, 1.0)	42	(0.8, 1.0)

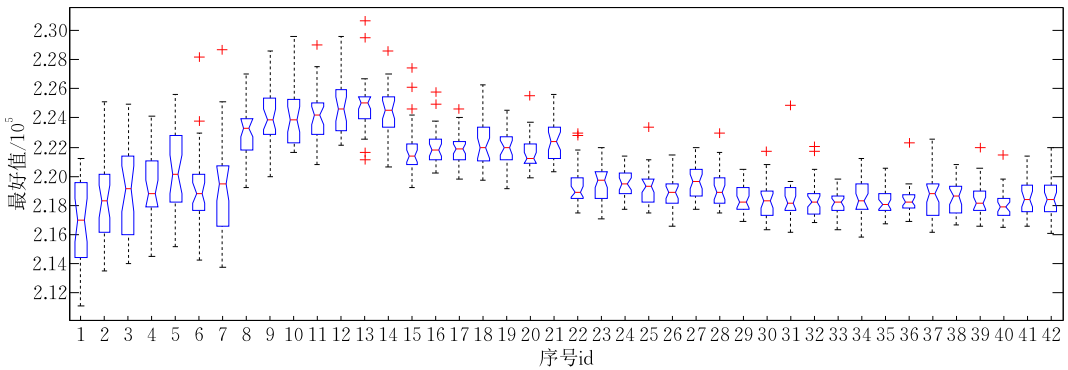


图 5 在 42 种组合形式下 SecEGA 求解 UDKP3 的性能比较

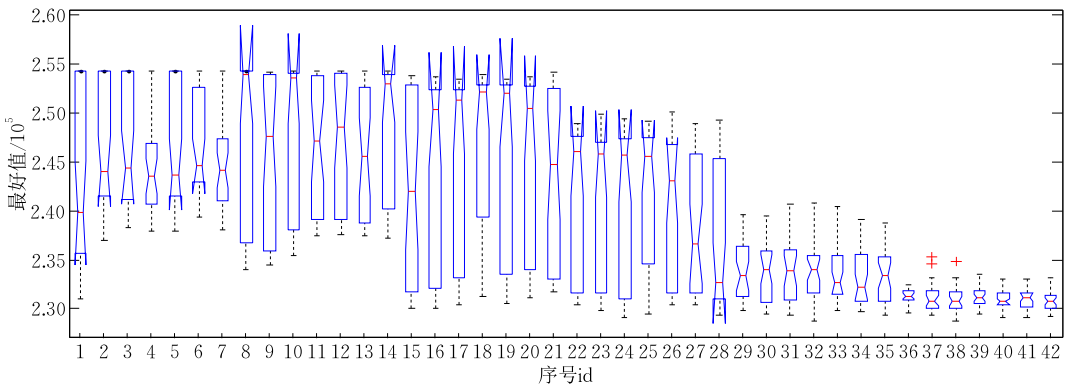


图 6 在 42 种组合形式下 SecEGA 求解 WDKP3 的性能比较

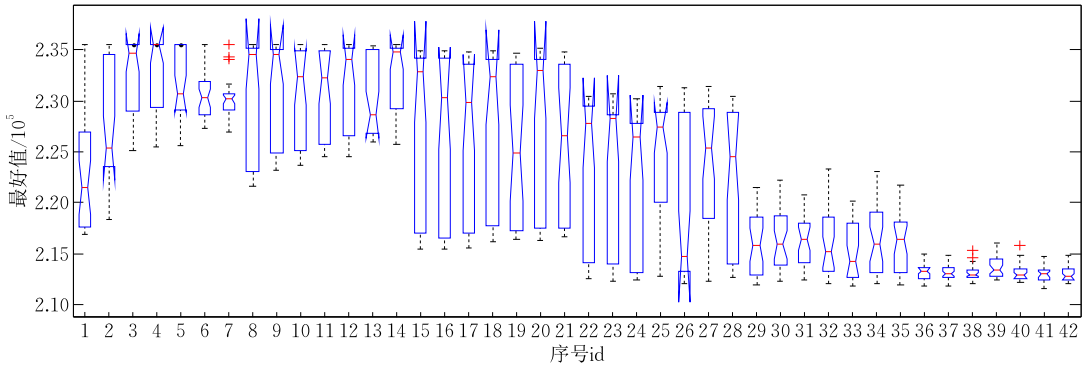


图 7 在 42 种组合形式下 SecEGA 求解 SDKP3 的性能比较

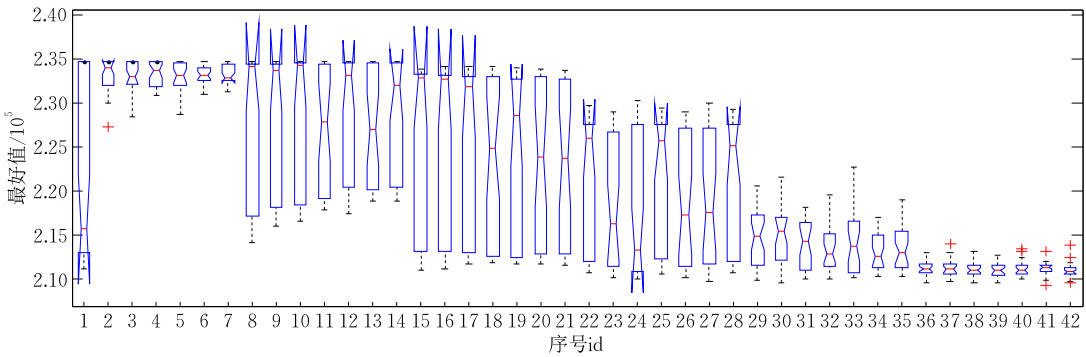


图 8 在 42 种组合形式下 SecEGA 求解 IDKP3 的性能比较

### 6.4 计算结果比较与分析

在利用 FirEGA 求解四类 D{0-1}KP 实例时，对于 UDKP 类实例，设置  $p_c = 0.8$  且  $p_m = 1.0$ ；对于 WDKP 类、SDKP 类和 IDKP 类实例，设置  $p_c = 0.8$  且  $p_m = 0.01$ 。在利用 SecEGA 求解四类 D{0-1}KP 实例时，设置  $p_c = 0.8$  且  $p_m = 0.01$ 。除此以外，在所有的计算中，两个算法的种群规模均设置为 50，迭代次数均等于实例的规模（即  $3n$ ）。利用 FirEGA 和 SecEGA 求解四类 D{0-1}KP 实例的计算结果见表 3~表 6，其中 Opt 为由动态规划法（记为 DPDKP）计算出的实例最优值；Best、Worst 和 Mean 分别为 FirEGA 和 SecEGA 在 30 次独立计算中得到的所有结果的最好值、最差值及所有结果的数学期望；Opt/

Best、Opt/Worst 和 Opt/Mean 分别表示以上各值的近似比<sup>[5,7]</sup>；Time1 和 Time2 分别表示 FirEGA 和 SecEGA 对于每个 D{0-1}KP 实例 30 次独立计算的平均耗费时间（单位：s）。

从表 3 可以看出：FirEGA 求解 UDKP 实例所得最好值的近似比基本保持在 1.10 左右，平均值和最差值的近似比均不超过 1.1441；SecEGA 求解 UDKP 实例所得最好值和平均值的近似比基本保持在 1.20 左右，即使最差值的近似比也不超过 1.2285。从表 4 可以看出：FirEGA 求解 WDKP 实例所得到的最好值的近似比不超过 1.0094，平均值的近似比不超过 1.0131，最差值的近似比不超过 1.0282；SecEGA 求解 WDKP 实例所得最好值的近

表 3 FirEGA 与 SecEGA 求解 D{0-1}KP 实例 UDKP1~10 的结果比较

算法	FirGA								SecEGA						
	DPDKP	Opt	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst	Time1	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst
UDKP1	85740	80101	1.0703	79325.3	1.0809	78499	1.0922	0.1511	77974	1.0996	76808.3	1.1163	75624	1.1338	0.1323
UDKP2	163744	152969	1.0704	151045.2	1.0841	149732	1.0936	0.5802	148042	1.1061	146310.5	1.1192	144113	1.1362	0.5323
UDKP3	269393	244291	1.1028	241061.2	1.1175	239114	1.1266	1.3516	230485	1.1688	225232.4	1.1961	222118	1.2128	1.1584
UDKP4	347599	319680	1.0873	316503.4	1.0982	313141	1.1100	2.4068	306358	1.1346	301700.8	1.1521	299059	1.1623	2.1230
UDKP5	442644	403908	1.0959	399525.2	1.1079	396937	1.1151	3.9228	375619	1.1784	371688.8	1.1909	368445	1.2014	3.3147
UDKP6	536578	483350	1.1101	478779.5	1.1207	474558	1.1307	5.5336	447231	1.1998	442556.4	1.2125	438762	1.2229	4.7106
UDKP7	635860	564656	1.1261	559815.4	1.1358	555763	1.1441	7.6609	531192	1.1970	523809.2	1.2139	517579	1.2285	6.2522
UDKP8	650206	590237	1.1016	584264.3	1.1129	580258	1.1205	9.9590	560932	1.1592	555100.7	1.1713	545509	1.1919	8.5761
UDKP9	718532	652354	1.1014	646592.2	1.1113	642965	1.1175	12.6306	619444	1.1599	615990.4	1.1665	609077	1.1797	10.8808
UDKP10	779460	708744	1.0998	703947.8	1.1073	700702	1.1124	15.4784	689248	1.1309	684872.4	1.1381	662742	1.1761	13.4265

表 4 FirEGA 与 SecEGA 求解 D{0-1}KP 实例 WDKP1~10 的结果比较

算法	DPDKP			FirGA					SecEGA						
	结果	Opt	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst	Time1	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst
WDKP1	83098	82722	1.0045	82539.6	1.0068	82454	1.0078	0.1474	82664	1.0053	81118.3	1.0244	80284	1.0351	0.1307
WDKP2	138215	137712	1.0037	137225.8	1.0072	136983	1.0090	0.5620	137712	1.0037	135951.3	1.0167	134490	1.0277	0.5620
WDKP3	256616	254234	1.0094	253294.4	1.0131	252909	1.0147	1.3214	254315	1.0090	247826.2	1.0355	236444	1.0853	1.2244
WDKP4	315657	314107	1.0049	312343.1	1.0106	310665	1.0161	2.3064	314227	1.0046	308628.9	1.0228	293697	1.0748	2.2345
WDKP5	428490	426783	1.0040	424384.2	1.0097	421584	1.0164	3.5991	426783	1.0040	421415.1	1.0168	391633	1.0941	3.4382
WDKP6	466050	463870	1.0047	460750.4	1.0115	455201	1.0238	5.3218	463435	1.0056	461779.8	1.0092	446741	1.0432	5.0059
WDKP7	547683	544059	1.0067	541505.3	1.0114	535551	1.0227	7.6337	544776	1.0053	537821.2	1.0183	501094	1.0930	6.5758
WDKP8	576959	574201	1.0048	571594.9	1.0094	565119	1.0210	9.7813	574862	1.0036	567507.7	1.0167	527204	1.0944	8.8046
WDKP9	650660	647012	1.0056	644298.2	1.0099	639241	1.0179	12.5521	648402	1.0035	643444.6	1.0112	589018	1.1047	11.3073
WDKP10	678967	677359	1.0024	673776.0	1.0077	660332	1.0282	15.7707	677118	1.0027	672912.3	1.0090	610343	1.1124	14.3101

表 5 FirEGA 与 SecEGA 求解 D{0-1}KP 实例 SDKP1~10 的结果比较

算法	DPDKP			FirGA					SecEGA						
	结果	Opt	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst	Time1	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst
SDKP1	94459	93316	1.0122	93192.8	1.0136	93064	1.0150	0.1495	93009	1.0156	91684.0	1.0303	90256	1.0466	0.1307
SDKP2	160805	159116	1.0106	158936.7	1.0118	158798	1.0126	0.5844	159107	1.0107	156557.3	1.0271	154241	1.0426	0.5266
SDKP3	238248	235372	1.0122	235204.4	1.0129	235015	1.0138	1.3094	235474	1.0118	231287.4	1.0301	224872	1.0595	1.1709
SDKP4	340027	336369	1.0109	335844.7	1.0125	335524	1.0134	2.3386	336597	1.0102	330437.4	1.0290	318638	1.0671	2.1225
SDKP5	463033	451184	1.0263	447335.9	1.0351	444252	1.0423	3.5163	444748	1.0411	435933.3	1.0622	415923	1.1133	3.2387
SDKP6	466097	459236	1.0149	458746.1	1.0160	458427	1.0167	5.2570	458509	1.0165	453973.9	1.0267	430286	1.0832	4.7950
SDKP7	620446	607200	1.0218	602797.7	1.0293	600496	1.0332	7.2951	598648	1.0364	592672.4	1.0469	571469	1.0857	6.1576
SDKP8	670697	661104	1.0145	659844.6	1.0164	659120	1.0176	9.8929	662465	1.0124	653459.8	1.0264	610664	1.0983	8.1947
SDKP9	739121	728443	1.0147	727364.5	1.0162	726872	1.0169	12.8355	730036	1.0124	726324.1	1.0176	671623	1.1005	10.7512
SDKP10	765317	755189	1.0134	752931.0	1.0165	749879	1.0206	15.5267	756662	1.0114	750716.4	1.0194	697520	1.0972	13.6472

表 6 FirEGA 与 SecEGA 求解 D{0-1}KP 实例 IDKP1~10 的结果比较

算法	DPDKP			FirGA					SecEGA						
	结果	Opt	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst	Time1	Best	Opt/Best	Mean	Opt/Mean	Worst	Opt/Worst
IDKP1	70106	<b>70106</b>	1.0000	70078.0	1.0004	70022	1.0012	0.1406	70101	1.0001	70010.4	1.0014	69947	1.0023	0.1391
IDKP2	118268	118034	1.0020	117544.3	1.0062	117249	1.0087	0.5365	118232	1.0003	116921.7	1.0115	115384	1.0250	0.6063
IDKP3	234804	234508	1.0013	233896.3	1.0039	233447	1.0059	1.2527	234698	1.0005	228166.1	1.0291	218621	1.0740	1.2584
IDKP4	282591	281804	1.0028	280536.6	1.0073	278179	1.0159	2.2339	282484	1.0004	273731.2	1.0324	259485	1.0890	2.2860
IDKP5	335584	335068	1.0015	332180.2	1.0102	328661	1.0211	3.5691	335580	1.00001	331886.1	1.0111	309964	1.0827	3.5616
IDKP6	452463	451498	1.0021	449781.0	1.0060	446456	1.0135	5.0966	452016	1.0010	447358.4	1.0114	412539	1.0968	4.9877
IDKP7	489149	487675	1.0030	484305.8	1.0100	475476	1.0288	6.8666	488840	1.0006	483569.7	1.0115	445999	1.0967	6.8920
IDKP8	533841	531872	1.0037	529372.8	1.0084	524404	1.0180	8.9463	533686	1.0002	531844.8	1.0038	508560	1.0497	9.8470
IDKP9	528144	525460	1.0051	522243.5	1.0113	501428	1.0533	11.2157	527942	1.0004	523597.8	1.0087	476760	1.1078	11.8425
IDKP10	581244	578897	1.0041	575128.5	1.0106	551772	1.0534	13.7694	580910	1.0006	572178.7	1.0158	520255	1.1172	14.9866

似比不超过 1.009, 平均值的近似比不超过 1.0355, 最差值的近似比不超过 1.1124. 从表 5 可以看出: FirEGA 求解 SDKP 实例所得到的最好值的近似比均不超过 1.0263, 平均值的近似比不超过 1.0351, 最差值的近似比也均不超过 1.0423; SecEGA 求解 SDKP 实例所得最好值的近似比不超过 1.0411, 平均值的近似比不超过 1.0622, 最差值的近似比也不超过 1.1133. 从表 6 可以看出: FirEGA 求解 IDKP 实例所得最好值的近似比不超过 1.0051, 平均值的近似比不超过 1.0113, 最差值的近似比不超过 1.0534; SecEGA 求解各 IDKP 实例所得到的最好值的近似比不超过 1.001, 平均值的近似比不超过 1.0324, 最差值的近似比不超过 1.1172.

由表 3~表 6 还可以看出: SecEGA 的求解速度明显比 FirEGA 快, 而且它求得各实例的最好结果与 FirEGA 不相上下, 对于某些实例甚至比 FirEGA 得到的最好结果更优; 但是, 从平均求解性能来看, SecEGA 的求解效果却明显不如 FirEGA 更优.

令  $F(t) = \frac{1}{30} \sum_{i=1}^{30} f(B_i(t))$ , 其中  $f(B_i(t))$  为算法在第  $i$  次独立计算某实例的第  $t$  次迭代进化后所得杰出者对应的函数值, 当  $t$  在区间  $[0, MaxIt]$  上均匀取值时 (包括 0 和  $MaxIt$ ), 连接各坐标点  $(t, F(t))$  所构成的曲线即为算法求解各实例的平均收敛曲线, 其中  $MaxIt$  为算法的迭代进化次数.

为了更直观地比较 FirEGA 和 SecEGA 的平均

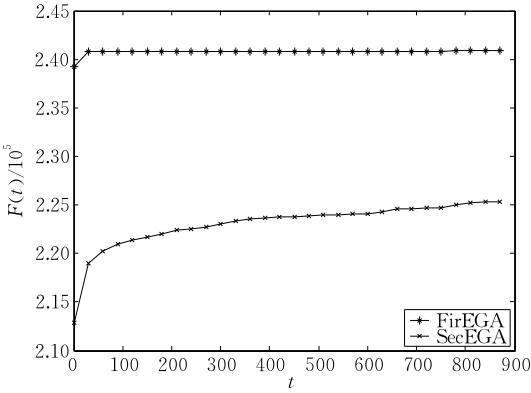


图 9 求解 UDKP3 的平均收敛曲线比较

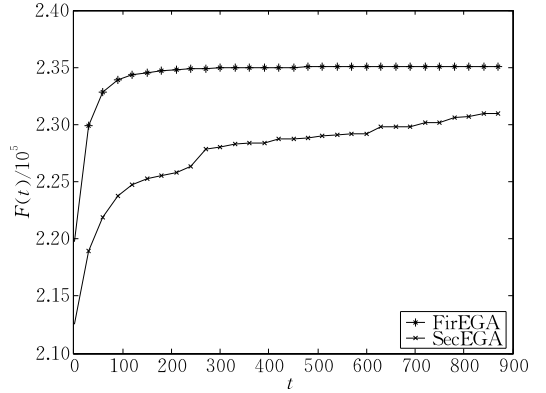


图 13 求解 SDKP3 的平均收敛曲线比较

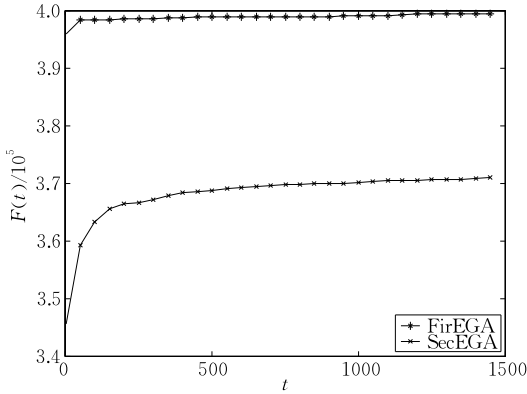


图 10 求解 UDKP5 的平均收敛曲线比较

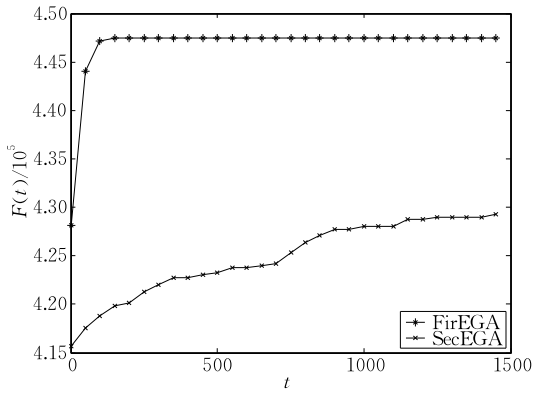


图 14 求解 SDKP5 的平均收敛曲线比较

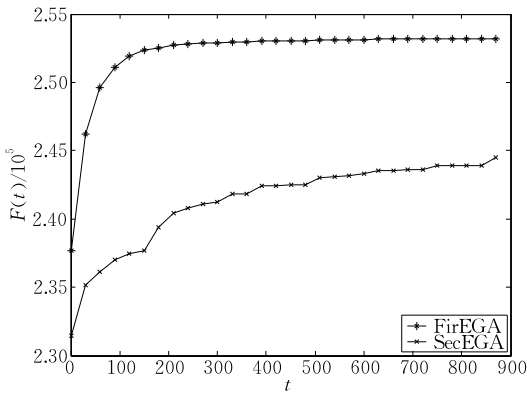


图 11 求解 WDKP3 的平均收敛曲线比较

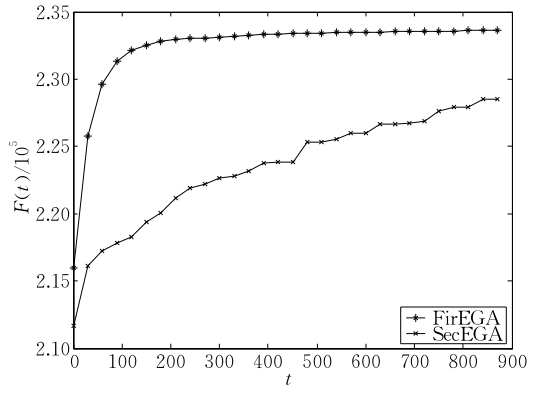


图 15 求解 IDKP3 的平均收敛曲线比较

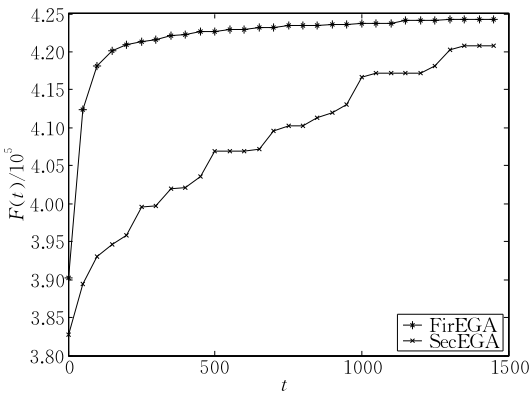


图 12 求解 WDKP5 的平均收敛曲线比较

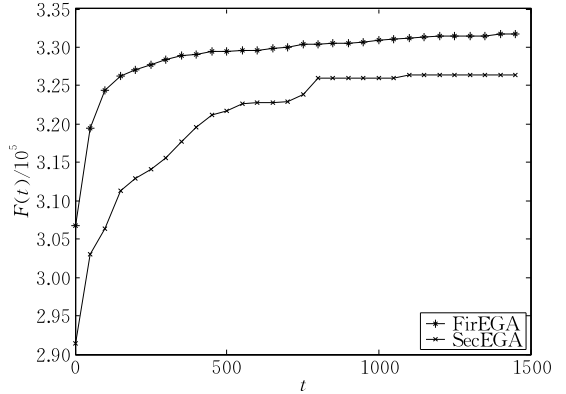


图 16 求解 IDKP5 的平均收敛曲线比较

求解性能,在图 9~图 16 给出它们 30 次独立计算实例 UDKP3、UDKP5、WDKP3、WDKP5、SDKP3、SDKP5、IDKP3 和 IDKP5 的平均收敛曲线。

由图 9~图 16 可以看出:FirEGA 的平均收敛速度非常快,基本上在不超过  $0.2 \times MaxIt$  的迭代次数内即可获得极好的结果;SecEGA 的平均收敛速度明显比 FirEGA 的慢,其平均收敛曲线呈现出一种缓慢上升的态势,并且它的平均求解结果也明显比 FirEGA 的差。

事实上,由于基于动态规划的确定性算法是伪多项式时间的,当  $D\{0-1\}$  KP 实例的规模较大且各项的价值系数和重量系数均在较大范围(例如  $[A, B] = [101, 10^{18}]$ )内取值时,需要耗费大量的求解时间而缺乏实用性。但是这些因素对于 FirEGA 和 SecEGA 是没有实质性影响,因为可以对  $D\{0-1\}$  KP 实例中各项的价值系数、重量系数和背包载重同时乘以一个很小的系数  $\epsilon$  ( $0 < \epsilon < 1$ ),从而得到一个与原实例具有相同最优解(最优值相差  $1/\epsilon$  倍)的新  $D\{0-1\}$  KP 实例,利用 FirEGA 或 SecEGA 求解这个新实例即可。因此,虽然 FirEGA 和 SecEGA 不一定能够求得  $D\{0-1\}$  KP 问题的最优解,但是对于规模大且各项价值系数和重量系数均在大范围内随机取值的难  $D\{0-1\}$  KP 实例,能够快速求得一个近似比接近于 1 的近似解,这恰好符合实际应用中更注重算法的求解速度,而对于求解结果只要是一个具有满意近似比的近似解即可的要求,所以 FirEGA 和 SecEGA 均为适于求解  $D\{0-1\}$  KP 问题的有效且实用的进化算法。

基于对四类  $D\{0-1\}$  KP 实例的计算、比较和分析可以得出以下结论:

对于项的价值系数和重量系数均在大范围内随机取值的四类  $D\{0-1\}$  KP 实例,FirEGA 和 SecEGA 均能够快速求得一个近似比接近于 1 的近似解,因此是适于求解大规模难  $D\{0-1\}$  KP 实例的有效且实用的进化算法。此外,从算法求得的最好结果来看,FirEGA 和 SecEGA 的求解能力基本相当;从算法的平均求解性能来看,FirEGA 的求解效果明显优于 SecEGA。

## 7 结束语

本文研究如何利用遗传算法求解  $D\{0-1\}$  KP 问题,在提出该问题的两个新数学模型的基础上,个体分别采用 0-1 向量和整型向量的编码方法,给出了求

解  $D\{0-1\}$  KP 的两个进化算法 FirEGA 和 SecEGA,并针对非正常编码个体,分别基于贪心策略给出了两个有效的处理算法;通过对四类大规模  $D\{0-1\}$  KP 实例的计算结果比较与分析表明:FirEGA 和 SecEGA 均不受实例中各项的价值系数、重量系数和背包载重的大小影响,对于大规模的难  $D\{0-1\}$  KP 实例,能够快速求得一个近似比接近于 1 的近似解,因此均为求解大规模难  $D\{0-1\}$  KP 实例的实用进化算法。

由于  $D\{0-1\}$  KP 提出的时间较短,其研究结果还相对较少,且已有确定性算法不适于求解大规模的难  $D\{0-1\}$  KP 实例,因此设计求解  $D\{0-1\}$  KP 的高效进化算法是一个值得深入研究的问题。此外,如何基于  $D\{0-1\}$  KP 的第三数学模型设计有效的进化算法也是一个值得探讨的问题。今后将对以上两问题做进一步的探讨与研究,以获得求解  $D\{0-1\}$  KP 的具有更好近似比的进化算法。

**致 谢** 审稿人提出了宝贵意见,对提高论文水平有很大的帮助;编辑付出了辛勤工作,在此一并致谢!

## 参 考 文 献

- [1] Arindama Singh. Elements of Computation Theory. London, Britain; Springer, 2009
- [2] Du D-Z, Ko K-I. Theory of Computational Complexity. New York, US; Wiley-Interscience, 2000
- [3] Cord J. A method for allocating funds to investment projects when returns are subject to uncertainty. Management Science, 1964, 10(2): 335-341
- [4] Goldberg D E, Smith R E. Nonstationary function optimization using genetic algorithms with dominance and diploidy// Proceedings of the International Conference on Genetic Algorithms. Hillsdale, Canada, 1987: 59-68
- [5] Cormen T H, Leiserson C E, Rivest R L, Stein C. Introduction to Algorithms, 2nd Edition. Cambridge, Britain; The MIT Press, 2001
- [6] Kellerer H, Pferschy U, Pisinger D. Knapsack Problems. Berlin, Germany; Springer, 2004
- [7] Alsuwaiyel M H. Algorithms Design Techniques and Analysis. Singapore; World Scientific Publishing Company, 2009
- [8] Lin G Y, Lu Y, Yao D D. The stochastic knapsack revisited: Switch-over policies and dynamic pricing. Operations Research, 2008, 56(4): 945-957
- [9] Dizdar D, Gershkov A, Moldovanu B. Revenue maximization in the dynamic knapsack problem. Theoretical Economics, 2011, 6(2): 157-184

- [10] Goldberg D E, Smith R E. Nonstationary function optimization using genetic algorithms with dominance and diploidy // Proceedings of the International Conference on Genetic Algorithms. Hillsdale, USA, 1987: 59-68
- [11] Hadad B S, Eick C F. Supporting polyploidy in genetic algorithms using dominance vectors // Proceedings of the 6th International Conference on Evolutionary Computation. San Francisco, USA, 1997: 223-234
- [12] Lewis J, Hart E, Ritchie G. A comparison of dominance mechanisms and simple mutation on non-stationary problems // Proceedings of the Parallel Problem Solving from Nature, North-Holland, Amsterdam, 1998: 139-148
- [13] He Yi-Chao, Zhang Xin-Lu, Li Wen-Bin, et al. Algorithms for randomized time-varying knapsack problems. Journal of Combinatorial Optimization, 2016, 31(1): 95-117
- [14] Marchand H, Wolsey L A. The 0-1 knapsack problem with a single continuous variable. Math Program, 1999, 85(1): 15-33
- [15] Lin G, Zhu W, Ali M M. An exact algorithm for the 0-1 knapsack problem with a single continuous variable. Journal of Global Optimization, 2011, 50(4): 657-673
- [16] Guldan B. Heuristic and Exact Algorithms for Discounted Knapsack Problems [M. S. dissertation]. University of Erlangen-Nürnberg, Germany, 2007
- [17] Rong Ai-Ying, Figueira J R, Klamroth K. Dynamic programming based algorithms for the discounted {0-1} knapsack problem. Applied Mathematics and Computation, 2012, 218(12): 6921-6933
- [18] Rudolph G. Convergence analysis of canonical genetic algorithms. IEEE Transactions on Neural Networks, 1994, 5(1): 96-101
- [19] Goldberg D E. Genetic Algorithms in Search, Optimization, and Machine Learning. MA, USA: Addison-Wesley, 1989
- [20] Holland H J. Adaptation in Natural and Artificial Systems. Cambridge, USA: The MIT Press, 1992
- [21] Schmitt L M. Theory of genetic algorithms. Theoretical Computer Science, 2001, 259(1-2): 1-61
- [22] Mitchell M. An Introduction to Genetic Algorithms. Cambridge, USA: The MIT Press, 1996
- [23] Chen Guo-Liang, Wang Xi-Fa, Zhuang Zhen-Quan, et al. Genetic Algorithm and Its Applications. Beijing: Posts & Telecom Press, 2003(in Chinese)  
(陈国良, 王熙法, 庄镇泉等. 遗传算法及其应用. 北京: 人民邮电出版社, 2003)
- [24] Liu Yong, Kang Li-Shan, Chen Yu-Ping. Non-Numerical Parallel Algorithms (2)—Genetic Algorithm. Beijing: Science Press, 2003(in Chinese)  
(刘勇, 康立山, 陈毓屏. 非数值并行算法(二)——遗传算法. 北京: 科学出版社, 2003)
- [25] Michalewicz Z, et al. A modified genetic algorithm for optimal control problems. Computers & Mathematics with Applications, 1992, 23(12): 83-94
- [26] Runarsson T P, Yao X. Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284-294
- [27] Coello C A C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithm—A survey of the state of art. Computer Methods in Applied Mechanics & Engineering, 2002, 191: 1245-1287
- [28] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. Evolutionary Computation, 1996, 4(1): 1-32
- [29] He Yi-Chao, Song Jian-Min, Zhang Jing-Min, et al. Research on genetic algorithms for solving static and dynamic knapsack problems. Application Research of Computers, 2015, 32(4): 1011-1015(in Chinese)  
(贺毅朝, 宋建民, 张敬敏等. 利用遗传算法求解静态与动态背包问题的研究. 计算机应用研究, 2015, 32(4): 1011-1015)
- [30] Pisinger D. Where are the hard knapsack problems? Computers & Operations Research, 2005, 32(9): 2271-2284



**HE Yi-Chao**, born in 1969, professor. His main research areas are the theory and applications of evolutionary algorithm, the design and analysis of algorithms, computational complexity theory and group testing theory.

**WANG Xi-Zhao**, born in 1963, Ph.D., professor, IEEE Fellow. His main research areas are machine learning, intelligent computation and genetic algorithm.

## Background

Discounted {0-1} knapsack problem (D{0-1} KP) is a new extension problem of the classical knapsack problem. It

**LI Wen-Bin**, born in 1974, Ph.D., professor. His main research areas are artificial intelligence and its applications, intelligent computation.

**ZHANG Xin-Lu**, born in 1968, associate professor. His main research areas are intelligent computation and group testing theory.

**CHEN Yi-Ying**, born in 1971, Ph.D., professor. Her main research areas are intelligent computation, earth exploration and information technology.

can be applied in commerce, investment decision, resource allocation, computational complexity theory, cryptography



and applied mathematics etc., and it will become a hot problem in evolutionary computation. Since the exact solution of  $D\{0-1\}$  KP is not really necessary in practice, efficient approximation algorithms for finding approximate solution is of practical importance. So, it's more significant to design efficient evolutionary algorithm for solving the large scale and hard instances of  $D\{0-1\}$  KP.

The authors of this paper have conducted the research in evolutionary algorithms and NP-hard problems direction from 2004, and have been supported by the National Natural Science Foundation of China (71371063), the Basic Research Project of Knowledge Innovation Program in Shenzhen (JCYJ20150324140036825), the University Scientific Research Fund of Hebei Province (ZD2016005, Z2013110). The research team has done some creative work and published some papers on international or domestic journals and conference proceedings.

The second author WANG Xi-Zhao is the leader in our research team. He takes charge of the project "Data Mining for Imbalanced Big Data in Electronic Health Records" (NSF

CNS-71371063) at present. He concentrated on the research of evolutionary algorithms for combinational optimization problem and cooperatively published several papers on differential evolution and genetic algorithm etc.

Because  $D\{0-1\}$  KP is a new combinational optimization problem, there only exists one mathematic model at present, and the complexity of exact algorithm based on dynamic programming is pseudo polynomial time, it is not practical for the large scale and hard instances of  $D\{0-1\}$  KP with huge value coefficients and weight coefficients. In this paper, we establish two new mathematic models of  $D\{0-1\}$  KP, which provide new theory foundational for designing algorithm. Furthermore, we propose two genetic algorithms based on different mathematic models of  $D\{0-1\}$  KP, and solve the problem about non-normal coding individual in evolutionary algorithms. The extensive simulations demonstrate that the proposed evolutionary algorithms— FirEGA and SecEGA can achieve excellent approximation solutions to large scale and hard instances of  $D\{0-1\}$  KP.