

求解 0-1 背包问题的混合贪婪遗传算法

陈 桢^{1*}, 钟一文², 林 娟¹

(1. 福建农林大学 计算机与信息学院, 福州 350002; 2. 智慧农林福建省高等学校重点实验室(福建农林大学), 福州 350002)
(* 通信作者电子邮箱 chenzhen@fafu.edu.cn)

摘 要: 求解 0-1 背包问题(KP)的最优解的时候, 传统遗传算法(GA)的局部求精能力不足而简单局部搜索算法的全局探索能力有限, 针对上述问题, 将这两个算法整合并提出了混合贪婪遗传算法(HGGA)。在 GA 全局搜索框架下增加局部搜索模块, 并改进传统仅基于物品价值密度的修复算子, 增加基于物品价值的贪婪混合选项, 从而加速寻优过程。HGGA 一方面引导种群在进化的优质解空间中展开精细搜索, 另一方面依靠 GA 的经典操作算子开拓全局搜索空间, 从而达到算法求精能力和开拓能力的良好平衡。HGGA 分别在三组数据上做了测试, 结果表明在第一组 15 个测试用例中的 12 个上, HGGA 能够百分百找到最优解, 成功率达到 80%; 在第二组小规模数据集上, HGGA 的性能明显好于其他同类 GA 和其他元启发算法; 在第三组大规模数据集上, HGGA 较其他元启发式算法具有更好的稳定性和高效性。

关键词: 0-1 背包问题; 混合贪婪遗传算法; 求精能力; 求泛能力; 混合贪婪算子; 局部搜索

中图分类号: TP18 **文献标志码:** A

Hybrid greedy genetic algorithm for solving 0-1 knapsack problem

CHEN Zhen^{1*}, ZHONG Yiwen², LIN Juan¹

(1. College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou Fujian 350002, China;
2. Key Laboratory of Smart Agriculture and Forestry (Fujian Agriculture and Forestry University), Fujian Province University, Fuzhou Fujian 350002, China)

Abstract: When solving the optimal solutions of 0-1 Knapsack Problems (KPs), the traditional Genetic Algorithm (GA) has insufficient local refinement ability and the simple local search algorithm has limited global exploration ability. Aiming at these problems, two algorithms were integrated to the Hybrid Greedy Genetic Algorithm (HGGA). Under the GA global search framework, local search module was added, and the traditional repair operator based only on item value density was improved, the greedy hybrid option based on item value was added, so as to accelerate the optimization process. In HGGA, the population was led to carry out fine search in the excellent solution space of evolution, and the classical operators of GA were relied on to expand the global search space, so as to achieve a good balance between the refinement ability and the development ability of the algorithm. HGGA was tested on three sets of data. The results show that in the first set of 15 test cases, HGGA is able to find the optimal solution on 12 cases, with a success rate of 80%; on the second small-scale dataset, the performance of HGGA is obviously better than those of other similar GA and other meta-heuristic algorithms; on the third large-scale dataset, HGGA is more stable and efficient than other meta-heuristic algorithms.

Key words: 0-1 Knapsack Problem (KP); Hybrid Greedy Genetic Algorithm (HGGA); refinement ability; generalization ability; hybrid greedy operator; local search

0 引言

背包问题(Knapsack Problem, KP)^[1]属于经典的组合优化问题, 属于 NP 难问题^[2], 该类问题包括许多变种: 0-1KP^[3]、有界 KP、无界 KP、多重约束的 KP、多维 KP、集合并集的 KP、多 KP、多重选择的 KP、二次 KP、动态 KP、折扣 KP^[4]以及其他类型的 KP。在众多变体 KP 中, 最基本的就是 0-1KP。在理论上, 许多整数规划问题的求解都依赖于一个高效的求解 KP 的算法, 因此在实际应用中常常被用于工业、金融等领域, 比如投资决策、货物装载等。

传统求解 0-1KP 的办法是精确算法, 譬如穷举法、动态规划法、递归法、回溯法和分支定界法等^[4]。精确算法可以得到问题的精确解, 但是随着物品数量和背包容量的增加, 算法时间的复杂性呈现出指数级的增长趋势, 计算效率比较低, 因此只适合小规模问题的求解。而元启发式算法作为近似搜索算法, 可在较短时间找到问题的近似有效解, 逐渐成为解决大规模 0-1KP 的主流方法。该类算法大致可分为两类: 以局部搜索为特征的单个体迭代算法, 包括经典的模拟退火 (Simulate Annealing, SA) 算法^[5]、混合贪婪修复算子的噪声检测算法 (Noising Methods with hybrid greedy repair operator, NM)^[6]等;

收稿日期: 2020-05-30; 修回日期: 2020-07-29; 录用日期: 2020-08-05。

基金项目: 福建省自然科学基金资助项目 (2019J01401, 2019J01661); 福建省教育厅中青年教师教育科研项目 (KLA19027A)。

作者简介: 陈桢 (1982—), 女, 福建仙游人, 讲师, 硕士, 主要研究方向: 计算智能; 钟一文 (1968—), 男, 福建上杭人, 教授, 博士, 主要研究方向: 计算智能、生物信息学、科学数据可视化; 林娟 (1980—), 女, 福建福州人, 副教授, 硕士, 主要研究方向: 生物信息学、科学数据可视化。

以协同搜索为特征的基于种群的智能优化算法,包括常见的蚁群优化(Ant Colony Optimization, ACO)^[7]、粒子群优化(Particle Swarm Optimization, PSO)^[8]、遗传算法(Genetic Algorithm, GA)^[9]、新兴的布谷鸟搜索^[10]、帝王蝶优化算法^[11],以及和声搜索算法(Harmony Search Algorithm, HSA)^[12]等,均可见在0-1KP上的应用。

GA作为一种典型的进化型算法,其独特的选择、交叉、变异算子以及概率化的寻优方式,使得算法自动获取解空间的布局,自适应地调整搜索状态,因此算法具有较好的全局协同搜索能力。但由于GA缺乏有效的小范围搜索策略,导致其局部搜索能力较差,在实际应用中,容易产生早熟收敛的问题。为了有效提高GA在全局大范围搜索中跳出局部极值解的能力,同时结合0-1KP的问题特征,本文提出混合贪婪遗传算法(Hybrid Greedy Genetic Algorithm, HGGA):在GA的并行解空间中,增加提高个体的局部求精能力的搜索模块;并针对常用的基于物品价值密度的修复优化算子仅强调物品的性价比,使得总体价值高但相对性价比低的物品难以被选入,从而导致搜索范围缩减及收敛速度降缓的问题,提出同时选择物品价值密度以及物品价值为标准的混合贪婪操作算子,在局部解空间中展开弹性的动态搜索。算法在并行迭代的算法框架中,通过增强GA的精细搜索能力得到算法求精能力和求泛能力的良好平衡。算法在不同问题性质及大规模的测试集上进行了测试,并通过与同类算法的对比,显示出较好的性能。

1 相关工作

1.1 0-1 背包问题

0-1KP可以简单描述为:给定 n 种物品集合 $N=\{1, 2, \dots, n\}$ 和一个背包容量为 C 的背包。每一种物品 i 都有它的重量 $w_i(>0)$ 和价值 $v_i(>0)$ ($i=1, 2, \dots, n$)。求最优装包方案 (x_1, x_2, \dots, x_n) , $x_i \in \{0, 1\}$ ($x_i=0$ 表示不把第 i 个物品放入背包, $x_i=1$ 表示把第 i 个物品放入背包),使得背包的总重量 W 不超过 C 的同时获得的价值 V 达到最大。在选择物品装入背包时,物品只有两种状态:整体装入或不装入。其数学表达式为:

$$V = \max \sum_{i=1}^n v_i x_i$$

$$\text{s.t.} \quad \sum_{i=1}^n w_i x_i \leq C$$

$$x_i \in \{0, 1\}; i = 1, 2, \dots, n$$

0-1KP问题属于带有约束条件的最大值优化问题,对违反约束条件的处理通常有两种方法。第一种是罚函数方法,通过对不可行解增加一个负值罚分从而排查此类解进入下一轮迭代的可能性。另外一种更常见的方法是引入修复优化操作,该操作分为两个步骤:首先是修复操作,即对不可行解按照一定的准则移除物品,直到该不可行解变为可行解为止;接下来是优化操作,即选择合适的物品加入,在出现不可行解之前停止。其中,对于物品的选择标准,通常采用的是基于价值密度的贪婪选择方法,即优先选择单位价值(v_i/w_i)密度大的物品。具体算法描述如算法1和算法2。在两个算法中,均引入了一个基于价值密度降序排列的物品列表 HD ,当不可行解出现时,首先采用贪婪修复算子(OPerator-Repair, OP-R)进行修复,根据物品在 HD 中的排序依次对物品进行考察,如果当前物品在背包中,将其移除,直至背包内物品总量满足约

束条件为止;随后采用贪婪优化算子(OPerator-Optimization, OP-O)对该解进行优化,同样根据 HD 中物品的顺序,优先选择价值密度大的物品进行加入。

算法1 贪婪修复算子(OP-R)。

输入 排好序的物品列表 HD 及不可行解个体 x 。

输出 满足约束条件的解 x' 。

```

1) For  $i=1$  to  $n$                                 //遍历 $HD$ 中每一个物品
2)   If  $W>C$  and  $x_i=1$ 
      //如果当前物品总重量大于背包容量,且当前物品在背包中
3)      $x_i=0$                                        //将该物品移出背包
4)      $W=W-w_i$                                    //修改当前总重量
5)   End If
6) End For

```

算法2 贪婪优化算子(OP-O)。

输入 排好序的物品列表 HD 及可行解个体 x 。

输出 满足约束条件的优化解 x' 。

```

1) For  $i=1$  to  $n$                                 //遍历 $HD$ 中每一个物品
2)   If  $W<C$  and  $x_i=0$ 
      //如果当前物品总重量小于背包容量,且当前物品不在背包中
3)      $x_i=1$                                        //将该物品加入背包
4)      $W=W+w_i$                                    //修改当前总重量
5)   End If
6) End For

```

1.2 求解0-1背包问题的启发式算法

GA作为一个经典的启发式搜索算法,通过对生物遗传和进化过程中选择、交叉、变异机制的模仿,来完成对问题最优解的自适应搜索过程,在解决大规模组合问题方面具有较好的表现。其中在解决0-1KP方面,一般是通过两种方法进行求解:一种是基于罚函数的基本遗传算法(Simple GA, SGA)^[13];一种是基于处理约束条件的修复或修复优化的贪婪算子的改进遗传算法,其中改进遗传算法较SGA优越。Michalewicz^[14]首先研究了利用GA求解0-1KP时个体编码方法的优劣以及处理0-1KP不可行解的罚函数法与修复法,指出:GA的个体采用0-1向量编码比自然数编码的效果更佳,而且利用修复法处理0-1KP不可行解比罚函数法的结果更好。从此,人们在利用GA求解0-1KP时一般均采用0-1向量的编码方式。

GA的搜索能力由选择算子、杂交算子决定,变异算子保证算法能搜索到问题空间的尽可能多的点,从而使其具有搜索全局最优的能力,所以一般求解0-1KP的改进GA也是从这3个算子上做调整和改进,结合基于修复或修复优化的贪婪算子求得最优解。文献[15]将贪婪算子与GA结合起来得到混合遗传算法(Hybrid Genetic Algorithm, HGA)。在HGA中首先采用二进制编码的GA产生新解,然后通过价值密度引导的贪婪算子对不满足约束条件的解进行修复。文献[16]针对HGA中提出的贪婪算子的缺陷提出新的贪婪算子,不仅仅修复不满足约束条件的种群,也优化满足条件的种群,并将此方法与GA相结合构成一种新的算法GGA。文献[17]将基于价值密度的贪婪算法与GA相结合提出一种改进的混合GA来求解0-1背包问题,改进的混合GA通过GA的择优,重复执行选择、交叉和变异以及基于价值密度的贪婪算法的修复优化这样一个过程,使得解无限接近最优解,同时采用精英保留机制来提高算法的收敛速度。文献[18]用GGA改进的贪婪算子提出一种改进混合遗传算法(Improved Hybrid Genetic Algorithm, IHGA),对染色体进行修复和优化,并基于稳态复

制的策略,对GA的选择操作进行改进,给出了随机选择操作。

还有一类改进GA,通过加入模拟退火算法(SA)或改进的SA来增加局部搜索能力,使得通过GA的全局搜索方式下求得的解能够跳出局部极值,并最终趋于全局优化。文献[19-20]分别运用模拟退火和遗传算法相结合的方法求解0-1KP,克服了各自的弱点,提高算法的优化性能、优化效率和可靠性。文献[21]借鉴二重结构编码的解码处理方法设计了一种新解码方法,在保证解可行性的同时修正种群中无对应可行解的个体;采用模拟退火算法和改进的精英选择算子改进SGA。

上述算法在进化效率或最优解搜索能力上都存在一些不足,因此,受SA具有较强局部搜索能力可以弥补GA的局部求精能力不足的启发,同时借鉴文献[6]中贪婪算子的改进方法,本文提出了一种新的混合贪婪遗传算法,使得局部搜索和全局搜索相互结合,取长补短。

2 改进的混合贪婪遗传算法

本文提出的改进算法主要针对以下问题:混合GA的局部求解能力不足、算法倾向早熟收敛;基于价值密度的修复优化方法过分强调价值与重量的比率、无法全面覆盖优质解且降低搜索速度。本文提出结合局部搜索和混合修复优化算子的混合贪婪遗传算法,首先在标准的GA框架下增加局部搜索模块,并采用基于贪婪思想的混合修复优化算子,对个体展开解空间中的精细更新,有机结合协同搜索和局部搜索的优势,从而提高算法的整体性能。

2.1 解的表示及初始化

0-1KP属于组合优化问题,对于此类问题的解,在GA中采用二进制编码方式,即种群中的每一个个体为一个 n 元0-1向量 x ,该向量中的每一位随机取值为0或1,分别表示把此物品放入或不放入背包中。

2.2 全局搜索模块

本文提出的算法依托标准GA的全局开拓能力,执行标准GA的进化框架,在种群随机初始化产生之后,通过交叉、变异和选择操作产生新的种群。

1)交叉操作:对初始化种群中的解随机取两个染色体作为父母个体,依照交叉概率选择个体进行均匀交叉操作,生成与父代个体数目一致的子代个体。

2)变异操作:对于个体按照变异概率进行基位变异,对选中的位置上的0/1编码进行逆向转化,即0变异位1,1变异为0。

3)选择操作:采用择优选择法,将父代和子代的个体统一按照适应值的降序排序,根据该排序选择满足种群数的个体组成新一代的种群。

2.3 局部搜索模块

通过第一阶段的种群进化过程,种群中的个体已覆盖了较广的搜索区域,但对于现有范围的精细搜索,在GA的算法框架内没有较好的办法。因此,在种群总体得到一次进化操作后,对于种群中的每一个个体,增加一个局部搜索模块,使得每一个个体以自身为搜索起点,在其周围展开局部范围内的搜索。具体操作步骤为:对于种群中的每一个个体,随机对一个位进行翻转,采用贪婪的方式接受修改,即如果该操作使

得对应的适应值得到提高,就接受新解以代替旧解;否则沿用旧解,进行下一轮的局部扰动,直到达到预先设置的局部搜索次数。在后续的操作中,经过新一轮的全局搜索过程,采用GA三个基本操作将优质的局部解信息迅速扩散,同时在局部搜索中出现的极易陷入局部最小值,而贪婪接收方式无法轻易跳出极值的问题,通过全局的进化过程,陷入极值的解得以相应调整并往正确的方向进化。进一步地,在总体迭代条件不变的情况下,通过合理平衡全局迭代次数以及局部搜索次数,在种群中的优良信息得以均匀快速地推广同时,在较理想的解空间周边的细致搜索也使得算法找到全局最优解的可能性大大提高。通过这样全局搜索和局部搜索有机结合的方式,算法在全局求泛和局部求精的性能上得到了良好的平衡。具体算法描述如算法3。在算法3中,对于种群的每一个个体,随机选择一维进行扰动,同时进行修复和优化,如果之后的结果好于原先的解,就替代原先的解,继续进行下一轮搜索,直到达到局部搜索次数为止。

算法3 局部搜索算法。

输入 个体 x ,局部搜索次数 $localTimes$ 。

输出 更新后的个体 x 。

- 1) For $iter=1$ to $localTimes$
- 2) 随机生成 $j \in [1, n]$
- 3) 对 j 位上的数据进行翻转,并对个体 $x[id]$ 进行可行解修复生成中间解 $tmpX$
- 4) If ($V(tmpX) > V(x)$)
//新解的适应值大于当前个体的适应值
- 5) $x=tmpX$
- 6) $V(x)=V(tmpX)$
- 7) End If
- 8) End For

2.4 混合贪婪算子

对于在迭代过程中不可行解的处理,通常采用基于价值密度的贪婪修复和优化操作。虽然这种操作方式使得性价比高的物品优先进入背包,但是这种单一的选取方法将导致具有较大价值但相对性价比低的物品无法被选入,而这部分物品的加入可大幅提高算法的全局寻优速度。但是单纯选择价值大的物品同样也会增加算法陷入局部最优解的概率。因此在HGGA中采用基于混合策略贪婪修复算子(Operator-Hybrid, OP-H)。该算子借鉴了文献[6]的混合贪婪修复算子策略,在不可行解时,算法首先采用基于价值密度的准则移除性价比最低的物品,在尽量保持现有优质解信息的同时进行有效率的修复;在优化过程中,采用随机的方式,概率性地选择基于价值密度或价值的策略,弹性地对解进行优化,在提高算法的收敛速度的同时有效避免陷入极值点。与文献[6]不同的地方在于:NM算法在不可行解出现时,直接采用概率方法选择不同的策略同时进行修复和优化,这样会造成价值大的物品有较大概率被移除出背包,从而降低算法收敛速度。具体算法描述如算法4。

算法4 混合贪婪算子(OP-H)。

输入 按照价值密度降序排列的物品列表 HD ,按照价值降序排序的物品列表 HV ,选择概率 p 和不可行解个体 x 。

输出 满足约束条件的优化解 x' 。

- 1) 使用OP-R和 HD 修复 x
- 2) 随机生成 $r \in [0, 1)$
- 3) If $r < p$

- 4) 使用 OP-O 和 *HD* 优化 x
- 5) else
- 6) 使用 OP-O 和 *HV* 优化 x
- 7) End If

在算法 4 中,首先基于价值密度的选择方式调用算法 1 修复初始解 x 中的不可行解,接着随机生成概率 r ,对于修复过的可行解来说,如果 r 小于 p ,采用基于价值密度的选择方法对个体 x 进行优化;反之则优先选择价值大的物品加入 x 中。

在算法运行过程中,一旦产生新解,上述两个算子 OP-R 及 OP-O 就对该新解分别进行修复和优化的工作。其中,贪心修复算子负责将价值密度最低的物品尽快移除,在对违反约束条件的解进行修复的同时提供进一步优化的空间,贪心优化算子负责对修复后的解进行局部优化。两个算子协同工作,在解空间中展开合法、有效的搜索。

2.5 HGGA 框架

HGGA 主要有 3 个重要的组成部分:在初始解随机生成之后,首先依靠 GA 的全局搜索流程,通过经典的交叉、变异、选择操作进化出新一代种群;然后对种群中的每一个个体展开局部范围内的更新;对于不可行解的修复过程采用混合贪心修复算子进行操作。3 个组成部分协同搜索,构造有效解空间,引导整个种群进化,最终找到最优解。具体算法描述如算法 5。

算法 5 HGGA 流程。

输入 函数评价次数 MET , $popSize$, $localTimes$, *HD*, *HV*。

输出 最优解 X_{best} 。

- 1) 随机生成 $popSize$ 个个体组成种群 pop ,使用 OP-H 进行修复;
- 2) 记录当前最好解 X_{best}
- 3) While 当前函数评价次数 $< MET$ do
- 4) 对 pop 进行交叉、变异、选择操作并修复
- 5) For $id=1$ to $popSize$
- 6) 使用混合贪心算子(如算法 4)进行更新 $pop[id]$ 并修复
- 7) EndFor
- 8) 使用局部搜索算子对个体进行局部更新(如算法 3)更新 X_{best}
- 9) EndWhile

3 仿真实验及结果分析

3.1 测试数据集

为了验证 HGGA 的性能,引入在各文献中出现的 3 个数据集测试集分别进行测试。

第 1 个测试数据集来自文献[22],根据问题特征按照物品质量、价值呈无相关、弱相关和强相关分为 3 类,分类特征如表 1 所示。每类中包含 5 个测试实例,分别包含物品数量为 800、1 000、1 200、1 500 和 2 000。在表 2 中分别给出这 3 类测试例子的具体名称、物品数目及理论最优解。

第 2 个测试数据集的两个实例(实例 1、2)是 0-1KP 的经典实例,实例来自文献[18],两个实例的实验次数均为 50 次。 n 为物品数量, C 为背包容量, V 为物品价值, W 为物品重量。

第 3 个测试数据集来自文献[23],数据集是大规模的 0-1KP 产生的。每个物品的价值是从 0.5 到 1 之间随机生成的,相应的物品重量也是在 0.5 到 2 之间随机生成的。

背包的最大容量被设置为上述物品重量之和的 0.75 倍。这些数据都是随机生成器一次生成的,并且用于所有的实验。放入背包的物品数量从 100 到 6 400,包括 100、200、300、500、

700、1 000、1 200、1 500、1 800、2 000、2 600、3 000、3 500、4 900、5 800、6 400。这些实例被标记为 LKP1~LKP16。

表 1 第 1 个大样本数据的分类特征

Tab. 1 Classification characteristics of first dataset of large sample data

相关性	重量 w_i	价值 v_i	容量 c
无相关	rand(10,100)	rand(10,100)	0.75×物品重量
弱相关	rand(10,100)	rand(w_i-10, w_i+10)	0.75×物品重量
强相关	rand(10,100)	w_i+10	0.75×物品重量

表 2 第 1 组 15 个测试例子的参数值

Tab. 2 Parameter values of 15 test cases in first dataset

无相关			弱相关			强相关		
测试 例子	物品 数目	理论最 优解	测试 实例	物品 数目	理论最 优解	测试 实例	物品 数目	理论最 优解
KP01	800	40 686	KP06	800	35 069	KP11	800	40 167
KP02	1 000	50 592	KP07	1 000	43 786	KP12	1 000	49 443
KP03	1 200	61 846	KP08	1 200	53 553	KP13	1 200	60 640
KP04	1 500	77 033	KP09	1 500	65 710	KP14	1 500	74 932
KP05	2 000	102 316	KP10	2 000	118 200	KP15	2 000	99 683

3.2 参数设置及运行环境

在 HGGA 中,采用函数评价次数 MET 作为迭代结束条件,取值为 40 000;在 GA 框架下的参数沿用经典的 GA 参数^[24]设置:交叉概率 cp 取值为 0.1,变异概率 mp 取值为 0.01;对于平衡算法全局与局部搜索性能的种群数 $popSize$ 和局部搜索次数 $localTimes$,在固定的 MET 条件下,采用实验的方式决定。混合贪心算子的选择概率 dp 设为 0.5。每个测试用例上的实验独立运行次数设为 100。

实验的硬件环境为小米笔记本,CPU 型号为 Intel Core i5-8250U,内存 8 GB,软件环境为 64 位 Windows 10 操作系统,算法使用 Java 语言在 Eclipse IDE 中编写。

3.3 参数分析

在相同的函数评价次数下,如何设置种群数和局部搜索次数决定了 HGGA 搜索的深度和广度。如果种群数太大,而局部搜索次数太小,会导致种群在局部范围内搜索精度不足,从而影响算法整体寻优能力;反之如果局部搜索次数太大,而种群数设置太小,会因算法的多样性不足而容易出现早熟收敛的现象。选用 3.1 节中的第 1 组数据集的重量价值呈强相关的 5 个数据集,对于 $popSize$ 和 $localTimes$ 分别设置(100,10)、(50,20)、(20,50)、(10,100)进行测试,并观察 100 次重复实验中 X_{best} 最后一次得到改善时的平均迭代次数。从图 1 可以看出,当种群数和搜索次数取(10,100)时,最优解得到最后一次改进所需要的迭代次数最少,所以这种搭配可以对算法在多样性和收敛性上达到一个比较均衡的平衡。因此,算法最终的 $popSize$ 设置为 10, $localTimes$ 设置为 100。

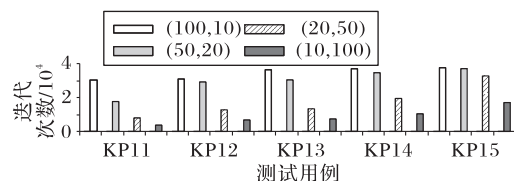


图 1 不同参数组合下算法最后一次得到改进所需的平均迭代次数对比

Fig. 1 Comparison of mean iterations required by the algorithm to have the last improvement under different parameter combinations

对于混合贪婪算子中使用的选择概率 dp 的取值,采用实验统计方式进行。当 dp 取 0 和 1 时,分别代表只采用基于价值和密度和只采用价值作为选择准则的操作算子。测试选用 3.1 节的表 2 的 3 组大样本数据的 15 个测试数据集, dp 分别取 0、0.1、0.2 取到 1,统计 15 个测试用例在不同取值下得到最优解的个数。结果如图 2 所示。

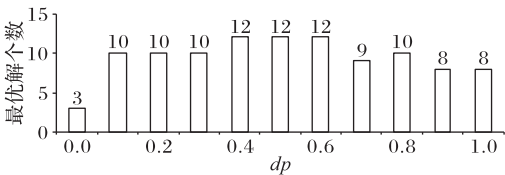


图 2 不同 dp 取值在 15 个测试例子上
取到最优解的个数对比

Fig. 2 Comparison of the number of obtained
optimal solutions on 15 test cases with different dp values

表 3 HGGA 在 15 个不同类型测试例子上的结果

Tab. 3 Results of HGGA on 15 test cases with different types

测试例子	实际差值	最优解	最差解	均值解	中位数解	标准差	迭代次数	时间/s	成功率/%
KP01	0	40686	40685	40685.00	40685	0.13	11133	0.380	10
KP02	0	50592	50592	50592.00	50592	0.00	15603	0.202	100
KP03	0	61846	61845	61845.90	61846	0.37	23060	0.180	96
KP04	0	77033	77033	77033.00	77033	0.00	18507	0.350	100
KP05	0	102316	102316	102316.00	102316	0.00	27426	0.601	100
KP06	0	35069	35069	35069.00	35069	0.00	8671	0.097	100
KP07	0	43786	43786	43786.00	43786	0.00	12050	0.156	100
KP08	0	53553	53552	53552.50	53552	0.50	24263	0.472	46
KP09	0	65710	65709	65709.10	65709	0.20	19909	0.698	12
KP10	0	118200	118200	118200.00	118200	0.00	22551	0.529	100
KP11	0	40167	40167	40167.00	40167	0.00	4297	0.047	100
KP12	0	49443	49443	49443.00	49443	0.00	6857	0.090	100
KP13	0	60640	60640	60640.00	60640	0.00	7281	0.119	100
KP14	0	74932	74932	74932.00	74932	0.00	10549	0.203	100
KP15	0	99683	99683	99683.00	99683	0.00	17054	0.399	100

3.5 与同类算法的比较

为了进一步显示 HGGA 的算法性能,引入不同的测试数据分别与其他 GA 以及近年来表现较好的其他新型算法分别进行纵向及横向的比较。

3.5.1 与 GA 的比较

测试数据采用上述第 2 个测试数据集,背包里的物品数目分别取 50 和 100,所有结果的数据均来源于原文献。其中, HGA 是使用贪心变换与 GA 结合的贪心 GA^[15], IHGA^[18] 是改进的混合 GA。 n 代表物品个数,算法独立运行 50 次, J 为 50 次实验中得到的最好解, JC 代表最好解在 50 次内出现的次数, JZ 代表繁殖代数的均值。 cp 和 mp 分别采用与文献中相同的设置值 0.5 和 0.01。表 4 给出与其他 GA 的比较结果。

从表 4 可以看出, HGA、IHGA 和 HGGA 在背包里的物品数目取 50 时都得到最优的结果, HGGA 在 50 次实验中每次都能求出最优值,而背包里的物品数目取 100 时,只有 HGGA 能得到最优解,而且 50 次实验中都能找到最优解。从这两个结果可得出, HGGA 与同类算法相比,在小规模的测试数据上具有较好的性能。

当 $dp=0$ 时,仅有 3 个测试用例能够得到最优解,当 dp 在 0.4~0.6 时,能够取到的最优解的个数最多。因此,为了增强算法的普适性,在本文中 dp 取值为 0.5。

3.4 算法性能分析

为了测试 HGGA 性能,在第 1 个数据集上进行了测试。算法独立运行 100 次,统计得到的最优解与理论最优解的实际差值、最优解、最差解、均值解、中位数解和标准差,适应值最后一次得到提高时的迭代次数,运行时间以及在 100 次重复中得到最优解的成功率。表 3 给出测试结果。粗体部分为获得最好状态的解。

从表 3 中可以看出,本文提出的算法在所有 15 个测试用例上均能找到最好解,对于其中的 11 个测试用例可以百分百找到最优解。虽然在 KP01、KP08、KP09 上的成功率较低,但是通过观察其找到的均值解、中位数解以及最差解可以看出, HGGA 所能找到的解与最优解非常相近,并且所有测试均在 1 s 内完成 100 次重复性实验,显示了算法的良好性能。

表 4 不同改进遗传算法求解 0-1KP 的结果

Tab. 4 Results of different improved genetic algorithms in
solving 0-1KP

参数	$n=50$			$n=100$		
	HGA	IHGA	HGGA	HGA	IHGA	HGGA
popSize	200	200.00	200.000	200	200.0	200.000
J	3119	3119.00	3119.000	26499	26553.0	26559.000
JC	3	31.00	50.000	4	39.0	50.000
JZ	300	31.80	4.125	500	63.5	8.655

3.5.2 与其他元启发算法的比较

为了进一步测试算法性能,算法与其他同类型的元启发算法在不同规模的问题上进行了比较。进行比较的算法为:基于全局和谐搜索的混合布谷鸟搜索算法(hybrid Cuckoo Search algorithm with Global Harmony Search, CSGHS)^[10]、二元帝王蝶优化(Binary Monarch Butterfly Optimization, BMBO)算法^[11]、混沌帝王蝶优化(Chaotic Monarch Butterfly Optimization, CMBO)算法^[22]、基于对立学习的帝王蝶优化(Opposition-based learning Monarch Butterfly Optimization, OMBO)算法^[25]和混合贪婪修复算子的噪声检测算法(NM)^[6]。

表5~7分别给出了3组数据上的表现。

表5 HGGA与其他元启发算法在求解5个无相关特征0-1KP时的数据对比

Tab. 5 Data comparison of HGGA and other meta-heuristic algorithms in solving 5 0-1KPs without correlation characteristics							
测试例子	理想值	算法	最优解	最差值	均值解	中位数	标准差
KP01	40 686	CSGHS	40 342	40 056	40 182. 00	40 190	68. 87
		BMBO	40 232	39 765	40 035. 00	40 036	105. 80
		CMBO	40 686	40 683	40 684. 00	40 683	105. 80
		OMBO	40 686	40 683	40 684. 00	40 683	0. 86
		NM	40 685	40 684	40 684. 88	40 685	0. 22
		HGGA	40 686	40 685	40 685. 03	40 685	0. 06
KP02	50 592	CSGHS	50 027	49 717	49 846. 00	49 835	84. 36
		BMBO	50 024	49 336	49 699. 00	49 689	135. 30
		CMBO	50 592	50 590	50 590. 00	50 590	0. 49
		OMBO	50 592	50 590	50 590. 00	50 590	0. 70
		NM	50 592	50 592	50 592. 00	50 592	0. 00
		HGGA	50 592	50 592	50 592. 00	50 592	0. 00
KP03	61 846	CSGHS	60 951	60 616	60 788. 00	60 791	79. 79
		BMBO	61 109	60 214	60 677. 00	60 660	165. 80
		CMBO	61 845	61 840	61 841. 00	61 840	1. 38
		OMBO	61 845	61 840	61 842. 00	61 843	1. 82
		NM	61 846	61 845	61 845. 32	61 845	0. 44
		HGGA	61 846	61 845	61 845. 91	61 846	0. 37
KP04	77 033	CSGHS	75 889	75 452	75 639. 00	75 631	112. 30
		BMBO	75 761	75 062	75 464. 00	75 482	193. 30
		CMBO	77 033	77 031	77 031. 00	77 031	0. 31
		OMBO	77 033	77 031	77 031. 00	77 031	0. 56
		NM	77 033	77 032	77 032. 92	77 033	0. 15
		HGGA	77 033	77 033	77 033. 00	77 033	0. 00
KP05	102 316	CSGHS	—	—	—	—	—
		BMBO	—	—	—	—	—
		CMBO	102 316	102 313	102 314. 00	102 313	0. 93
		OMBO	102 316	102 313	102 314. 00	102 313	1. 11
		NM	102 316	102 316	102 316. 00	102 316	0. 00
		HGGA	102 316	102 315	102 315. 99	102 316	0. 02

表5的数据表明,在大样本的物品价值和数量无相关的测试数据中,CMBO、OMBO、NM和HGGA比CSGHS和BMBO在找到最优解上表现得更好,NM在KP01的测试数据里表现略差,CMBO和OMBO在KP03上表现较弱。HGGA在这5个例子中最差解、均值解和中位数解上的表现都比CMBO、OMBO和NM好。在方差 std 的测算上,HGGA的总体表现也是最好的,足见该算法的稳定性。表6显示了这些算法在求解5个无相关特征的0-1KP数据上的平均排名,对于KP05,CSGHS和BMBO没有数据,取KP01~KP04排名的平均值作为KP05的排名取值,并对这些排名数据进行0.05显著性水平条件下的Friedman检验,得到的 p 值等于0.003,说明这些算法在排序值上有显著性差异。

表6 6个算法在求解5个无相关特征的0-1KP时的平均排名

Tab. 6 Average ranking of 6 algorithms in solving 5 0-1KPs without correlation characteristics			
算法	排名	算法	排名
CSGHS	3. 75	OMBO	3. 99
BMBO	5. 75	NM1	2. 77
CMBO	3. 40	HGGA	1. 20

表7的数据表明,在大样本的物品价值和数量呈弱相关的测试数据中,CMBO、OMBO、NM和HGGA比CSGHS和BMBO表现得好。OMBO、HGGA在5个测试算例中都取得最优解,CMBO在KP08,NM在KP09上没有取得最优解。HGGA在最差解、均值解和中位数解上综合表现是胜出的,方差也较小。表8显示了这些算法在求解大样本的物品价值和数量呈弱相关的测试数据上的平均排名。对于KP10,CSGHS和BMBO没有数据,取KP06~KP09排名的平均值作为KP10的排名取值,在0.05显著性水平条件下的Friedman检验得到的 p 值为0.004,说明这些算法在排序值上有显著性差异。

表7 HGGA与其他元启发算法在求解大样本弱相关特征的0-1KP时的数据对比

Tab. 7 Data comparison of HGGA and other meta-heuristic algorithms in solving 0-1KPs with weak correlation characteristics of large samples							
测试例子	理想值	算法	最优解	最差值	均值解	中位数	标准差
KP06	35 069	CSGHS	34 850	34 795	38 824. 00	34 825	14. 00
		BMBO	34 860	34 681	34 786. 00	34 784	35. 01
		CMBO	35 069	35 064	35 067. 00	35 067	1. 45
		OMBO	35 069	35 064	35 067. 00	35 068	1. 47
		NM	35 069	35 069	35 069. 00	35 069	0. 00
		HGGA	35 069	35 069	35 069. 00	35 069	0. 00
KP07	43 786	CSGHS	43 484	43 386	43 440. 00	43 442	22. 39
		BMBO	43 491	43 359	43 412. 00	43 413	31. 36
		CMBO	43 786	43 781	43 784. 00	43 784	1. 34
		OMBO	43 786	43 782	43 785. 00	43 785	1. 03
		NM	43 786	43 785	43 785. 96	43 786	0. 08
		HGGA	43 786	43 786	43 786. 00	43 786	0. 00
KP08	53 553	CSGHS	52 711	52 354	52 556. 00	52 565	76. 89
		BMBO	52 774	52 110	52 425. 00	52 390	158. 20
		CMBO	53 552	53 552	53 552. 00	53 552	0. 00
		OMBO	53 553	53 552	53 552. 00	53 552	1. 82
		NM	53 553	53 552	53 552. 02	53 552	0. 04
		HGGA	53 553	53 552	53 552. 48	53 552	0. 50
KP09	65 710	CSGHS	65 116	64 980	65 045. 00	65 044	38. 14
		BMBO	65 123	64 916	65 022. 00	65 012	56. 38
		CMBO	65 710	65 708	65 709. 00	65 708	0. 58
		OMBO	65 710	65 708	65 709. 00	65 709	0. 52
		NM	65 709	65 709	65 709. 00	65 709	0. 00
		HGGA	65 710	65 709	65 709. 11	65 709	0. 20
KP10	108 200	CSGHS	—	—	—	—	—
		BMBO	—	—	—	—	—
		CMBO	118 200	118 200	118 200. 00	118 200	0. 00
		OMBO	118 200	118 200	118 200. 00	118 200	0. 00
		NM	118 200	118 200	118 200. 00	118 200	0. 00
		HGGA	118 200	118 200	118 200. 00	118 200	0. 00

表8 6个算法在求解5个大样本弱相关特征的0-1KP的数据的平均排名

Tab. 8 Average ranking of 6 algorithms in solving 5 0-1KPs with weak correlation characteristics of large samples			
算法	排名	算法	排名
CSGHS	5. 00	OMBO	3. 00
BMBO	6. 00	NM1	1. 40
CMBO	2. 40	HGGA	2. 58

表9的数据表明,在大样本的物品价值和数量呈强相关的测试数据中,CMBO、OMBO、NM和HGGA比CSGHS和BMBO表现得好。CMBO、OMBO、NM和HGGA在5个测试算

例中都取得最优解。NM 和 HGGA 的方差在 5 个例子里都为 0,说明算法的在这个测试数据集上的表现非常稳定。表 10 显示了这些算法在求解大样本的物品价值和数量呈强相关的测试数据上的平均排名。对于 KP15,取 KP11~KP14 排名的平均值作为 KP15 的排名取值,在 0.05 显著性水平条件下通过 Friedman 检验得到的 p 值为 0.002,说明这些算法在排序值上有显著性差异。

表 9 HGGA 与其他元启发算法在求解大样本强相关特征的 0-1KP 时的数据对比

Tab. 9 Data comparison of HGGA and other meta-heuristic algorithms in solving 0-1KPs with strong correlation characteristics of large samples

测试例子	理想值	算法	最优解	最差值	均值解	中位数	标准差
KP11	40 167	CSGHS	40 147	40 126	40 132	40 130	5.54
		BMBO	40 127	40 107	40 116	40 117	4.52
		CMBO	40 167	40 166	40 167	40 167	0.14
		OMBO	40 167	40 167	40 167	40 167	0.00
		NM	40 167	40 167	40 167	40 167	0.00
		HGGA	40 167	40 167	40 167	40 167	0.00
KP12	49 443	CSGHS	49 403	49 383	49 393	49 393	6.52
		BMBO	49 393	49 353	49 378	49 382	10.12
		CMBO	49 433	49 433	49 422	49 433	2.49
		OMBO	49 443	49 441	49 443	49 443	0.34
		NM	49 443	49 443	49 443	49 443	0.00
		HGGA	49 443	49 443	49 443	49 443	0.00
KP13	60 640	CSGHS	60 587	60 567	60 573	60 570	5.32
		BMBO	60 588	60 530	60 562	60 560	11.98
		CMBO	60 640	60 639	60 640	60 640	0.14
		OMBO	60 640	60 640	60 640	60 640	0.00
		NM	60 640	60 640	60 640	60 640	0.00
		HGGA	60 640	60 640	60 640	60 640	0.00
KP14	74 932	CSGHS	74 858	74 817	74 835	74 832	9.31
		BMBO	74 842	74 772	74 818	74 821	15.80
		CMBO	74 932	74 931	74 932	74 932	0.27
		OMBO	74 932	74 931	74 932	74 932	0.14
		NM	74 932	74 932	74 932	74 932	0.00
		HGGA	74 932	74 932	74 932	74 932	0.00
KP15	99 683	CSGHS	—	—	—	—	—
		BMBO	—	—	—	—	—
		CMBO	99 683	99 672	99 682	99 683	2.23
		OMBO	99 683	99 679	99 683	99 683	0.58
		NM	99 683	99 683	99 683	99 683	0.00
		HGGA	99 683	99 683	99 683	99 683	0.00

表 10 6 种算法在求解 5 个大样本强相关特征的 0-1KP 的数据的平均排名

Tab. 10 Average ranking of 6 algorithms in solving 5 0-1KPs with strong correlation characteristics of large samples

算法	排名	算法	排名
CSGHS	5.25	OMBO	3.25
BMBO	5.75	NM1	1.00
CMBO	4.00	HGGA	1.00

值得说明的是, HGGA 与除 NM 之外的其他比较函数一样,属于基于种群的优化算法,因此针对 0-1KP,它们的算法时间复杂度均取决于外层迭代数目、种群数目及物品数量;而 NM 虽然不属于基于种群的算法,但其迭代过程类似地取决于外层迭代数目、内层局部搜索迭代数目及物品数量;进一步地,虽然 HGGA 加入了局部搜索部分,但这个部分的时间复杂度取决于局部搜索迭代次数、种群数目及物品数量,并不影响

整体算法复杂度,因此这几个参与比较的算法均具有相同的时间复杂度,而 HGGA 可以在更少的函数评价次数内找到优质解。

3.5.3 在大规模数据集上的比较

为了进一步测试算法在大规模数据集上性能,采用上述第三组测试用例,进行比较的算法为:大型 0-1 背包问题的简化二进制和声搜索(Simplified Binary Harmony Search, SBHS)算法^[23],表 11 给出了不同的算法在 16 个测试数据上的表现。表 11 的数据表明,在这 16 个大型 0-1KP 的测试数据中,对于前 6 个数据,SBHS 和 HGGA 的最优解相同,而 HGGA 在最差值、均值解和中位数上的表现都均优于 SBHS。在数据 LKP7-LKP14 中, HGGA 在各指标上的表现都优于 SBHS;在数据 LKP15 中, SBHS 的均值解和中位数解较优;在数据 LKP16 中, SBHS 解的所有表现都会比 HGGA 略好。

在标准差上,除了最后一个测试例子外, HGGA 在每一个测试用例上的表现均优于 SBHS,足见 HGGA 在大规模的测试数据上的稳定性。

表 11 HGGA 与 SBHS 算法在求解大样本强相关特征的 0-1KP 时的数据对比

Tab. 11 Data comparison of HGGA and SBHS algorithms in solving 0-1KPs with strong correlation characteristics of large samples

测试例子	算法	最优解	最差值	均值解	中位数	标准差
LKP1	SBHS	62.08	61.97	62.04	62.04	3.00E-02
	HGGA	62.08	62.06	62.08	62.08	3.00E-03
LKP2	SBHS	129.44	129.27	129.37	129.38	4.00E-02
	HGGA	129.44	129.44	129.44	129.44	1.70E-13
LKP3	SBHS	192.02	191.85	192.01	192.02	3.00E-02
	HGGA	192.02	192.02	192.02	192.02	1.15E-13
LKP4	SBHS	314.23	314.10	314.19	314.20	3.00E-02
	HGGA	314.23	314.22	314.22	314.23	1.48E-03
LKP5	SBHS	448.65	448.46	448.60	448.63	5.00E-02
	HGGA	448.65	448.65	448.65	448.65	1.46E-03
LKP6	SBHS	638.14	638.00	638.09	638.06	4.00E-02
	HGGA	638.14	638.14	638.14	638.14	1.12E-04
LKP7	SBHS	763.81	763.39	763.71	763.72	8.00E-02
	HGGA	763.83	763.82	763.83	763.82	1.85E-03
LKP8	SBHS	964.91	964.70	964.85	964.86	6.00E-02
	HGGA	964.92	964.91	964.92	964.91	3.66E-03
LKP9	SBHS	1 155.65	1 155.35	1 155.57	1 155.58	8.00E-02
	HGGA	1 155.68	1 155.67	1 155.68	1 155.68	2.19E-03
LKP10	SBHS	1 283.92	1 283.26	1 283.79	1 283.81	1.20E-01
	HGGA	1 283.92	1 283.92	1 283.92	1 283.92	7.12E-04
LKP11	SBHS	1 653.72	1 653.43	1 653.64	1 653.66	6.00E-02
	HGGA	1 653.76	1 653.75	1 653.76	1 653.75	2.38E-03
LKP12	SBHS	1 917.49	1 917.23	1 917.42	1 917.44	6.00E-02
	HGGA	1 917.58	1 917.57	1 917.57	1 917.57	6.55E-04
LKP13	SBHS	2 248.27	2 247.77	2 248.12	2 248.14	1.00E-01
	HGGA	2 248.30	2 248.29	2 248.30	2 248.30	1.39E-03
LKP14	SBHS	3 135.71	3 135.29	3 135.58	3 135.59	1.00E-01
	HGGA	3 135.77	3 135.71	3 135.75	3 135.75	1.05E-02
LKP15	SBHS	3 707.39	3 706.98	3 707.29	3 707.31	9.00E-02
	HGGA	3 707.41	3 707.12	3 707.27	3 707.27	0.050481
LKP16	SBHS	4 090.83	4 090.36	4 090.64	4 090.66	1.10E-01
	HGGA	4 090.55	4 089.72	4 090.26	4 090.45	1.40E-01

4 结语

针对 GA 在解决大规模离散优化问题上全局搜索能力强、局部求精能力薄弱的缺点,本文提出混合了局部搜索算子

的改进GA。算法在不变动GA的基本框架下,融合具备局部精细搜索能力的局部搜索方法,使得算法得以依赖GA的强大全局搜索性能将局部搜索中得到的优质解快速扩散;同时设计混合贪婪修复算子,在个体保留优良信息的同时提高算法的寻优速度。算法应用于0-1KP,在不同问题规模及性质的测试集上进行大量的仿真实验,通过与其他同类算法的比较显示出算法的高效性和稳定性,特别是在大规模数据集上的有效性。

今后的工作将进一步挖掘局部搜索和基于种群的新型智能优化算法的有机融合,并扩展到更大的应用范围,例如多维KP、折扣KP等。

参考文献 (References)

- [1] DANTZIG G B. Discrete variable extremum problems [J]. *Operations Research*, 1957, 5(2):266-277.
- [2] HAN K H, KIM J H. Genetic quantum algorithm and its application to combinatorial optimization problem [C]// *Proceedings of the 2000 Congress on Evolutionary Computation*. Piscataway: IEEE, 2000: 1354-1360.
- [3] 张玲玲, 张弘. 0-1 背包问题的预期效率和线性拟合求解[J]. *计算机应用*, 2014, 34(9):2581-2584. (ZHANG L L, ZHANG H. Solution of 0-1 knapsack problem based on expected efficiency and linear fitting[J]. *Journal of Computer Applications*, 2014, 34(9): 2581-2584.)
- [4] 杨洋, 潘大志, 刘益, 等. 折扣0-1 背包问题的简化新模型及遗传算法求解[J]. *计算机应用*, 2019, 39(3):656-662. (YANG Y, PAN D Z, LIU Y, et al. New simplified model of discounted 0-1 knapsack problem and solution by genetic algorithm[J]. *Journal of Computer Applications*, 2019, 39(3):656-662.)
- [5] 韩小雷. 粒子群-模拟退火融合算法及其在函数优化中的应用 [D]. 武汉: 武汉理工大学, 2008:15-23. (HAN X L. The particle swarm-simulated annealing fusion algorithm and its application in function optimization [D]. Wuhan: Wuhan University of Technology, 2008:15-23.)
- [6] ZHAN S, WANG L, ZHANG Z, et al. Noising methods with hybrid greedy repair operator for 0-1 knapsack problem [J]. *Memetic Computing*, 2020, 12(1):37-50.
- [7] 李士勇. 蚁群优化算法及其应用研究进展[J]. *计算机测量与控制*, 2003, 11(12):911-913, 917. (LI S Y. Progresses in ant colony optimization algorithm with applications [J]. *Computer Measurement and Control*, 2003, 11(12):911-913, 917.)
- [8] KENNEDY J, EBERHART R. Particle swarm optimization [C]// *Proceedings of the 1995 International Conference on Neural Networks*. Piscataway: IEEE, 1995: 1942-1948.
- [9] SRINIVAS N, DEB K. Multiobjective optimization using nondominated sorting in genetic algorithms [J]. *Evolutionary Computation*, 1994, 2(3):221-248.
- [10] FENG Y, WANG G, GAO X. A novel hybrid cuckoo search algorithm with global harmony search for 0-1 knapsack problems [J]. *International Journal of Computational Intelligence Systems*, 2016, 9(6):1174-1190.
- [11] FENG Y, WANG G, DEB S, et al. Solving 0-1 knapsack problem by a novel binary monarch butterfly optimization [J]. *Neural Computing and Applications*, 2017, 28(7):1619-1634.
- [12] 欧阳海滨, 高立群, 孔祥勇, 等. 一种求解0-1 背包问题的二进制修正和声搜索算法[J]. *控制与决策*, 2014, 29(7): 1174-1180. (OUYANG H B, GAO L Q, KONG X Y, et al. A binary modified harmony search algorithm for 0-1 knapsack problem[J]. *Control and Decision*, 2014, 29(7): 1174-1180.)
- [13] 王小平, 曹立明. 遗传算法—理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002:136-140. (WANG X P, CAO L M. *Genetic Algorithm - Theory, Application and Software Implementation* [M]. Xi'an: Xi'an Jiaotong University Press, 2002:136-140.)
- [14] MICHALEWICZ Z. *Genetic Algorithm + Data Structure = Evolution Programs: 3rd, Revised and Extended Edition* [M]. Berlin: Springer, 1996:13-103.
- [15] 徐宗本. 计算智能——模拟进化计算[M]. 北京: 高等教育出版社, 2004: 72-87. (XU Z B. *Computational Intelligence - Simulated Evolutionary Computation* [M]. Beijing: Higher Education Press, 2004:72-87.)
- [16] 贺毅朝, 刘坤起, 张翠军, 等. 求解背包问题的贪心遗传算法及其应用[J]. *计算机工程与设计*, 2007, 28(11):2655-2657, 2681. ((HE Y C, LIU K Q, ZHANG C J, et al. Greedy genetic algorithm for solving knapsack problems and its applications[J]. *Computer Engineering and Design*, 2007, 28(11):2655-2657, 2681.)
- [17] 白东玲, 郭绍永. 一种改进的混合遗传算法求解0_1 背包问题[J]. *电子设计工程*, 2013, 21(14): 9-11. (BAI D L, GUO S Y. An improved hybrid genetic algorithm for 0_1 knapsack problem [J]. *Electronic Design Engineering*, 2013, 21(14): 9-11.)
- [18] 刘寒冰, 张亚娟. 求解0-1 背包问题的改进混合遗传算法[J]. *计算机系统应用*, 2015, 24(6):197-201. (LIU H B, ZHANG Y J. Improved hybrid genetic algorithm for solving 0-1 knapsack problem [J]. *Computer Systems and Applications*, 2015, 24(6): 197-201.)
- [19] 金慧敏, 马良. 遗传退火进化算法在背包问题中的应用[J]. *上海理工大学学报*, 2004, 26(6):561-564. (JIN H M, MA L. Genetic annealing evolutionary algorithm applied to the knapsack problem [J]. *Journal of University of Shanghai for Science and Technology*, 2004, 26(6):561-564.)
- [20] 张盛意, 蔡之华, 占志刚. 基于改进模拟退火的遗传算法求解0-1 背包问题[J]. *微电子学与计算机*, 2011, 28(2): 61-64. (ZHANG S Y, CAI Z H, ZHANG Z G. Solving 0-1 knapsack problem based on genetic algorithm with improved simulated annealing [J]. *Microelectronics and Computer*, 2011, 28(2): 61-64.)
- [21] 吕晓峰, 张勇亮, 马羚. 一种求解0-1 背包问题的改进遗传算法[J]. *计算机工程与应用*, 2011, 47(34):44-46, 63. (LYU X F, ZHANG Y L, MA L. Improved genetic algorithm to 0-1 knapsack problem [J]. *Computer Engineering and Applications*, 2011, 47(34):44-46, 63.)
- [22] FENG Y, YANG J, WU C, et al. Solving 0-1 knapsack problems by chaotic monarch butterfly optimization algorithm with Gaussian mutation [J]. *Memetic Computing*, 2018, 10(2):135-150.
- [23] KONG X, GAO L, OUYANG H, et al. A simplified binary harmony search algorithm for large scale 0-1 knapsack problems [J]. *Expert Systems with Applications*, 2015, 42(12): 5377-5355.
- [24] SCHAFFER J D, CARUANA R, ESHELMAN L J, et al. A study of control parameters affecting online performance of genetic algorithms for function optimization [C]// *Proceedings of the 3rd International Conference on Genetic Algorithms*. San Francisco: Morgan Kaufmann Publishers Inc., 1989:51-60.
- [25] FENG Y, WANG G, DONG J, et al. Opposition-based learning monarch butterfly optimization with Gaussian perturbation for large-scale 0-1 knapsack problem [J]. *Computers and Electrical Engineering*, 2018, 67: 454-468.

This work is partially supported by the Natural Science Foundation of Fujian Province (2019J01401, 2019J01661), the Young and Middle-Aged Teachers Education Research Project of Fujian Provincial Department of Education (KLA19027A).

CHEN Zhen, born in 1982, M. S., lecturer. Her research interests include computational intelligence.

ZHONG Yiwen, born in 1968, Ph. D., professor. His research interests computational intelligence, bioinformatics, scientific data visualization.

LIN Juan, born in 1980, M. S., associate professor. Her research bioinformatics, scientific data visualization.