

基于Lagrange插值的学习猴群算法求解折扣{0-1}背包问题

徐小平¹, 徐丽^{1*}, 王峰², 刘龙³

(1. 西安理工大学理学院, 西安 710054; 2. 西安交通大学数学与统计学院, 西安 710049;

3. 西安理工大学自动化与信息工程学院, 西安 710048)

(* 通信作者电子邮箱 1396580639@qq.com)

摘要:折扣{0-1}背包问题(D{0-1}KP)的目的是在不超过背包载重的前提下,使得装入背包的所有物品价值系数之和为最大。针对已有算法在求解规模大、复杂度高的D{0-1}KP时的求解精度低的问题,提出了Lagrange插值的学习猴群算法(LSTMA)。首先,在基本猴群算法的望过程中重新定义了视野长度;其次,在跳过程中引入了种群中最优的个体作为第二个支点,并调整搜索机制;最后,在跳过程之后引入Lagrange插值操作来提高算法的搜索性能。对四类实例的仿真结果表明:LSTMA在求解D{0-1}KP时的求解精度高于对比算法,并且具有良好的鲁棒性。

关键词:折扣{0-1}背包问题;Lagrange插值;猴群算法;学习因子

中图分类号:TP301.6 **文献标志码:**A

Learning monkey algorithm based on Lagrange interpolation to solve discounted {0-1} knapsack problem

XU Xiaoping¹, XU Li^{1*}, WANG Feng², LIU Long³

(1. School of Sciences, Xi'an University of Technology, Xi'an Shaanxi 710054, China;

2. School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an Shaanxi 710049, China;

3. School of Automation and Information Engineering, Xi'an University of Technology, Xi'an Shaanxi 710048, China)

Abstract: The purpose of the Discounted {0-1} Knapsack Problem (D{0-1}KP) is to maximize the sum of the value coefficients of all items loaded into the knapsack without exceeding the weight limit of the knapsack. In order to solve the problem of low accuracy when the existing algorithms solve the D{0-1}KP with large scale and high complexity, the Lagrange Interpolation based Learning Monkey Algorithm (LSTMA) was proposed. Firstly, the length of the visual field was redefined during the look process of the basic monkey algorithm. Then, the best individual in the population was introduced as the second pivot point and the search mechanism was adjusted during the jump process. Finally, the Lagrange interpolation operation was introduced after the jump process to improve the search performance of the algorithm. The simulation results on four types of examples show that LSMTA solves the D{0-1}KP with higher accuracy than the comparison algorithms, and it has good robustness.

Key words: Discounted {0-1} Knapsack Problem (D{0-1}KP); Lagrange interpolation; Monkey Algorithm (MA); learning factor

0 引言

折扣{0-1}背包问题(Discounted {0-1} Knapsack Problem, D{0-1}KP)是经典{0-1}背包问题的一个延伸,其思想源于商业打折促销活动^[1],例如:一家商店对商品A和商品B进行打折促销,A商品单独售价为30元,B商品单独售价50元,当商品A和B一起出售时,售价为70元,即相对于原价优惠了10元。对于顾客而言就有四种选择:单独购买商品A,单独购买商品B,同时购买商品A和商品B,商品A和B均不购买。因此,D{0-1}KP与人们实际生活联系紧密,此外,D{0-1}KP还广泛应用于商业投资、项目决策等问题^[2]。然而,D{0-1}KP规模大、约束条件复杂、属于组合优化类的难解问题,一些传统算

法求解该问题时耗时久,甚至无法找到合理的选配方案。针对这一问题,国内外众多学者利用群智能算法对该问题进行了研究,如粒子群算法(Particle Swarm Optimization, PSO)^[3]、遗传算法(Genetic Algorithm, GA)^[4]、差分进化(Differential Evolution, DE)^[5]算法等。随着问题规模增大,这些算法会出现收敛过早和求解精度降低的问题。

猴群算法(Monkey Algorithm, MA)最早由Zhao等^[6]在2008年提出,其思想源于猴子的爬山行为,设计了爬、望和跳的搜索过程。MA广泛用于求解高维数、多峰值、非线性、不可微的优化问题,其优点在于需要调控的参数较少,容易实现。因此,对于MA的研究受到国内外众多学者的关注。文献[7]提出了动态自适应猴群算法;文献[8]将混沌猴群算法应用于

收稿日期:2020-04-20;修回日期:2020-06-26;录用日期:2020-07-03。 基金项目:国家自然科学基金资助项目(61773016);陕西省创新能力支撑计划项目(2020OPT-023);陕西省自然科学基金基础研究计划项目(2018JQ1089)。

作者简介:徐小平(1973—),男,陕西蓝田人,教授,博士,主要研究方向:智能算法; 徐丽(1995—),女,陕西安康人,硕士研究生,主要研究方向:智能算法; 王峰(1972—),女,河南兰考人,教授,博士,主要研究方向:智能算法; 刘龙(1975—),男,陕西西安人,教授,博士,主要研究方向:机器学习。

有限元模型的修正;文献[9]提出野草猴群算法,并将其应用于传感器优化布置问题;文献[10]提出了基于贪婪策略的猴群算法,用于求解经典{0-1}背包问题。然而,在国内外,猴群算法对D{0-1}KP的研究较少,为进一步提高该算法求解D{0-1}KP时的求解精度,故本文提出Lagrange插值的学习猴群算法(Lagrange Interpolation based Learning Monkey Algorithm, LSTMA)。首先,将望过程中固定的视野长度转换为动态视野长度,同时引入学习因子以提高算法的收敛速度;其次,在跳过程中构造双支点跳机制,以避免陷入局部最优;最后,引入Lagrange插值策略以增强算法的探测能力。仿真结果表明,LSTMA求解该问题时具有较强的鲁棒性和较高的求解精度。

1 折扣{0-1}背包问题模型

D{0-1}KP^[11]可描述为:给定 n 组项集,每个项集 $i(i=0, 1, \dots, n-1)$ 中含有3个项,这三个项分别记为 $3i, 3i+1$ 和 $3i+2$ 。前两项 $3i$ 和 $3i+1$ 的重量系数分别为 w_{3i} 和 w_{3i+1} ,且这两个项的价值系数分别为 p_{3i} 和 p_{3i+1} 。因为第三个项 $3i+2$ 是由前两项组成的,所以第三项 $3i+2$ 的折扣重量系数 $w_{3i+2} < w_{3i} + w_{3i+1}$,且满足 $w_{3i} < w_{3i+2}, w_{3i+1} < w_{3i+2}$,价值系数为 $p_{3i+2} = p_{3i} + p_{3i+1}$ 。三个项至多有一个项被选入容量为 W 的背包。在不超过背包容积 W 的前提下,选择哪些项装入背包,使得装入背包的所有项的价值系数之和最大。其数学模型为:

$$\begin{aligned} \max f(x) &= \max \sum_{i=0}^{n-1} (p_{3i}x_{3i} + p_{3i+1}x_{3i+1} + p_{3i+2}x_{3i+2}) \quad (1) \\ \text{s.t.} \quad &\begin{cases} x_{3i} + x_{3i+1} + x_{3i+2} \leq 1 \\ \sum_{i=0}^{n-1} (w_{3i}x_{3i} + w_{3i+1}x_{3i+1} + w_{3i+2}x_{3i+2}) \leq W; \\ x_{3i}, x_{3i+1}, x_{3i+2} \in \{0, 1\} \end{cases} \\ &i = 0, 1, \dots, n-1 \end{aligned} \quad (2)$$

其中, $x_j (0 \leq j \leq 3n-1)$ 是二元决策变量,决定第 j 项是否被装入背包。当决策变量 x_j 取1时,表示背包中装入项 j ;当 x_j 取0时,表示背包中未装入项 j 。目标函数的一组可行解可表示为 $X = (x_0, x_1, \dots, x_{3n-1})$ 。

2 猴群算法及其改进算法

2.1 基本猴群算法

猴群算法主要分为爬过程、望过程、跳过程3个过程^[12],是模仿猴子爬山行为的仿生学优化算法。

1) 种群初始化。设猴群的规模为 $M, X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 为第 i 只猴子的位置,由式(3)生成:

$$X_i = x_{\min} + (x_{\max} - x_{\min}) \times \text{rand}(1, D); i = 1, 2, \dots, M \quad (3)$$

其中 D 为目标函数的维数。 x_{ij} 为第 i 只猴子在第 j 维的分量,其取值范围为 $[x_{\min}, x_{\max}]$,并且 $j = 1, 2, \dots, D$ 。 $\text{rand}(1, D)$ 是由 $[0, 1]$ 的均匀分布的随机数组成的 D 维向量。

2) 爬过程。爬过程为局部寻优过程,猴子通过迭代以改变当前位置,寻找适应度更高的个体,具体步骤如下:

① 随机生成向量 $\Delta X_i = (\Delta x_{i1}, \Delta x_{i2}, \dots, \Delta x_{iD})$,其中

$$\Delta x_{ij} = \begin{cases} a, & 50\% \text{ 的概率} \\ -a, & 50\% \text{ 的概率} \end{cases} \quad (4)$$

其中 a 为猴群更新的爬步长。

② 设 f 为目标函数,计算

$$f'_{ij}(X_i) = \frac{f(X_i + \Delta X_i) - f(X_i - \Delta X_i)}{2\Delta x_{ij}} \quad (5)$$

向量 $f'_i(X_i) = (f'_{i1}(X_i), f'_{i2}(X_i), \dots, f'_{iD}(X_i))$ 为目标函数在 X_i 处的伪梯度;

③ 令 $X_{\text{new}_i} = X_i + a \times \text{sign}(f'_{ij}(X_i))$,其中, sign 为符号函数;

④ 若 $X_{\text{new}_i} \in [x_{\min}, x_{\max}]$ 且 $f(X_{\text{new}_i}) < f(X_i)$,则 X_{new_i} 替换掉 X_i ;否则 X_i 不变;

⑤ 重复步骤①~④,直至爬迭代次数达到上限 N_g 。

3) 望过程。经过爬过程,每只猴子都达到各自所在山峰的顶端,也就是目标函数达到局部最优。猴子站在山峰顶端向四周眺望,看是否存在比当前山峰更高的山峰,若存在,就跳到更高的山峰,在新的山峰继续进行爬过程。在望过程内定义一个视野长度 b ,决定猴子眺望的最远距离。具体步骤如下:

① 令 $x_{\text{new}_{ij}} = \text{rand}(x_{ij} - b, x_{ij} + b)$,并且 $X_{\text{new}_i} = (x_{\text{new}_{i1}}, x_{\text{new}_{i2}}, \dots, x_{\text{new}_{iD}})$ 。

② 若 $X_{\text{new}_i} \in [x_{\min}, x_{\max}]$ 且 $f(X_{\text{new}_i}) < f(X_i)$,则 X_{new_i} 替换掉 X_i ;否则重复步骤①直至搜寻到满足条件的 X_{new_i} 。

③ 将 X_{new_i} 作为初始位置,再次重复爬过程。

4) 跳过程。跳过程的目的是跳出局部,搜索全局最优。计算所有猴子的重心位置,让所有猴子沿着当前位置指向重心位置的方向,进入新的搜索空间。具体步骤如下:

① 令 $[cc, dd]$ 为跳区间, $\theta = \text{rand}[cc, dd]$ 。定义更新公式为式(6):

$$x_{\text{new}_{ij}} = x_{ij} + \theta(pp_j - x_{ij}) \quad (6)$$

$pp = (pp_1, pp_2, \dots, pp_D)$ 为支点,其中 $pp_j = \frac{1}{M} \sum_{i=1}^M x_{ij}$ 。令

$X_{\text{new}_i} = (x_{\text{new}_{i1}}, x_{\text{new}_{i2}}, \dots, x_{\text{new}_{iD}})$ 。

② 若 $X_{\text{new}_i} \in [x_{\min}, x_{\max}]$ 且 $f(X_{\text{new}_i}) < f(X_i)$,则 X_{new_i} 替换掉 X_i ;否则重复步骤①和②直至搜索到满足条件的 X_{new_i} 。

2.2 基于Lagrange插值的学习猴群算法

在基本猴群算法中,望过程与跳过程的搜索范围分别为局部和全局,是整个猴群算法搜索能力的关键,为了提高算法的性能,这里主要对以上两个过程进行改进。

1) 望过程的改进。在望过程中,将最优个体 $x_{\text{best}} = (x_{\text{best}_1}, x_{\text{best}_2}, \dots, x_{\text{best}_D})$ 与当前个体之差作为视野长度,即: $b_{ij} = x_{\text{best}_j} - x_{ij}$,使得不同个体具有不同的视野长度。对于适应度值较低的猴子,与优秀猴子的距离自然也就越远,需要更长的视野长度,搜索局部最优解。对于适应度值较高的猴子,与优秀猴子的距离自然就更近,需要较短的视野长度,在当前解的局部区域搜索潜在的解。增加学习因子,构造学习望过程,让猴群向当前种群中最优的猴子望去,进行学习,使得望过程更具有目的性,避免盲目搜索。式(7)给出了学习望过程的更新公式:

$$x_{\text{new}_{ij}} = c_1(l_{ij} + b_{ij}) + c_2c_3(uu_{ij} - l_{ij}) \quad (7)$$

其中: $x_{\text{new}_{ij}}$ 为望过程产生的新个体, c_1, c_2 和 c_3 为学习因子, c_1 取值为0.2, c_2 取值为0.5, c_3 为 $[0, 1]$ 内的随机数。 $l_{ij} = x_{ij} - b_{ij}$, $uu_{ij} = x_{ij} + b_{ij}$, $[l_{ij}, uu_{ij}]$ 为视野范围。由式(7)知,猴群受到适应度值高的猴王引导后,在望过程,可以减少算法不必要的盲目搜索,搜索到局部最优解的概率就越大,进而提高算法的搜索效率。具体伪代码如算法1所示。

算法1 学习望过程。

输入 猴群个体 X_i ,学习因子 c_1 和 c_2 ;

输出 更新后的个体 X_{new_i} 。

计算种群中个体的适应度值,找出最优的个体记为 x_{best}

$c_3 = \text{rand}$

```

For  $i = 1$  to  $M$  do
  For  $j = 1$  to  $D$  do
     $b_{ij} = xbest_j - x_{ij}$ 
     $ll_{ij} = x_{ij} - b_{ij}$ 
     $uu_{ij} = x_{ij} + b_{ij}$ 
     $xnew_{ij} = c_1(ll_{ij} + b_{ij}) + c_2c_3(uu_{ij} - ll_{ij})$ 
  End For
End For

```

2) 跳过程的改进。跳过程实现全局寻优,是加速算法寻优效率的关键。基本猴群算法的跳过程只是单纯地规定猴群的支点为种群的重心,并没有给猴群一个趋向较优解的参考支点。为了使较优解产生的概率更高,将猴群的重心作为第一个支点,同时将当前种群中最优的个体 $xbest$ 作为第二个支点,对跳过程进行改进。改进后的更新公式为:

$$xnew_{ij} = ((-1) \times [0.2 + rand]) \times change_one_j \times$$

$$(-\lg(type1)) + change_two_j$$

$$change_one_j = rand \times (pp_j - x_{ij}) \quad (9)$$

$$change_two_j = \frac{type2 \times x_{ij} + type3 \times xbest_j}{type2 + type3} \quad (10)$$

在增加种群多样性的同时,也应当考虑算法的计算效率,故随机地选择式(6)与(8)进行更新,这样做也可以防止算法过早收敛。式(9)中的 pp_j 为支点,与式(6)中支点相同。式(8)和(10)中 $type1$ 、 $type2$ 、 $type3$ 均为 $[0, 1]$ 内的随机数, $\lceil \cdot \rceil$ 为向上取整符号。具体伪代码如算法 2 所示。

算法 2 双支点跳过程。

输入 猴群个体 X_i ;

输出 更新后的个体 $Xnew_i$ 。

计算种群中个体的适应度值,找出最优的个体记为 $xbest$

$\theta = rand(cc, dd)$

For $i = 1$ to M do

For $j = 1$ to D do

$$pp_j = \sum_{i=1}^M x_{ij} / M$$

$$change_one_j = rand \times (pp_j - x_{ij})$$

$$type1 = rand, type2 = rand, type3 = rand$$

$$change_two_j = \frac{type2 \times x_{ij} + type3 \times xbest_j}{type2 + type3}$$

If $rand > 0.5$

$$xnew_{ij} = x_{ij} + \theta \times (pp_j - x_{ij})$$

Else

$$xnew_{ij} = (-1) \times \lceil 0.2 + rand \rceil \times change_one_j \times$$

$$(-\lg(type1)) + change_two_j$$

End If

End For

End For

3) Lagrange 插值策略。Lagrange 插值是优化技术中常用技术手段,该策略可以通过有限的已知数据点来预测未知数据点的信息。因此,可在跳过程之后,引入 Lagrange 插值操作,利用已知三点,在全局最优解的周围挖掘潜在解,从而提高算法的探测能力。式(11)^[13]为 Lagrange 插值的公式:

$$P(x_j) = \sum_{k=0}^2 \left[\prod_{\substack{n=0 \\ n \neq k}}^2 \frac{x_j - x_j^n}{x_j^k - x_j^n} \right] f(x^k) \quad (11)$$

式(11)可转换为式(12):

$$P(x_j) = a(x_j)^2 + bx_j + c \quad (12)$$

则

$$a = [(x_j^2 - x_j^1)f(x^0) + (x_j^0 - x_j^2)f(x^1)]/I +$$

$$(x_j^1 - x_j^0)f(x^2)/I$$

$$b = ((x_j^1)^2 - (x_j^2)^2)f(x^0)/I + ((x_j^2)^2 - (x_j^0)^2)f(x^1)/I +$$

$$((x_j^0)^2 - (x_j^1)^2)f(x^2)/I$$

$$c = (x_j^2 - x_j^1)x_j^1x_j^2f(x^0)/I + (x_j^0 - x_j^2)x_j^0x_j^2f(x^1)/I +$$

$$(x_j^1 - x_j^0)x_j^0x_j^1f(x^2)/I$$

$$I = (x_j^0 - x_j^1)(x_j^1 - x_j^2)(x_j^2 - x_j^0) \quad (16)$$

其中: x^0 为当前种群中最优的个体, x^1 和 x^2 为 x^0 周围的潜在解,式(17)和(18)分别给出了 x_j^1 和 x_j^2 的计算公式:

$$x_j^1 = x_j^0 + \frac{1}{100M} (x_j^{up} - x_j^0)rand(0, 1) \quad (17)$$

$$x_j^2 = x_j^0 + \frac{1}{100M} (x_j^{low} - x_j^0)rand(0, 1) \quad (18)$$

其中: x_j^{up} 和 x_j^{low} 分别为种群在第 j 维的上界和下界。由 x_j^0 、 x_j^1 和 x_j^2 生成的插值点记为 x_j^3 ,式(19)给出了 x_j^3 的计算公式:

$$x_j^3 = \begin{cases} -\frac{b}{2a}, & a > 0 \\ (rand - 0.5) \times (x_j^1 - x_j^2) + \frac{1}{2} (x_j^1 + x_j^2), & a = 0 \wedge b = 0 \\ x_j^{\min} + rand \times c_4 \times (x_j^{\min} - x_j^{\mid}) + \\ \quad rand \times c_5 \times (x_j^{\min} - x_j^{\max}), & \text{其他} \end{cases} \quad (19)$$

其中: x_j^{\min} 、 x_j^{\mid} 和 x_j^{\max} 分别为 x_j^0 、 x_j^1 、 x_j^2 中的最小值、中位数和最大值, $rand$ 为 $(0, 1)$ 内的随机数, $c_4 = c_5 = 0.25$ 。具体伪代码如算法 3 所示。

算法 3 Lagrange 插值策略。

输入 x^0, x^1, x^2 ;

输出 x_j^3 。

$$I = (x_j^0 - x_j^1)(x_j^1 - x_j^2)(x_j^2 - x_j^0)$$

If $I \neq 0$

$$a = (x_j^2 - x_j^1)f(x^0)/I + (x_j^0 - x_j^2)f(x^1)/I +$$

$$(x_j^1 - x_j^0)f(x^2)/I$$

$$b = [((x_j^1)^2 - (x_j^2)^2)f(x^0) + ((x_j^2)^2 - (x_j^0)^2)f(x^1)]/I +$$

$$((x_j^0)^2 - (x_j^1)^2)f(x^2)/I$$

If $a > 0$

$$x_j^3 = -\frac{b}{2a}$$

Else If $a = 0 \wedge b = 0$

$$x_j^3 = \frac{1}{2} (x_j^1 + x_j^2) + (rand - 0.5) \times (x_j^1 - x_j^2)$$

Else

$$x_j^3 = x_j^{\min} + rand \times c_4 \times (x_j^{\min} - x_j^{\mid}) +$$

$$rand \times c_5 \times (x_j^{\min} - x_j^{\max})$$

End If

End If

Else

$$x_j^3 = x_j^{\min} + rand \times c_4 \times (x_j^{\min} - x_j^{\mid}) +$$

$$rand \times c_5 \times (x_j^{\min} - x_j^{\max})$$

End If

3 LSTMA 求解 D {0-1} KP

3.1 求解步骤

LSTMA 在求解 D {0-1} KP 时,需要建立算法与 D {0-1} KP 之间的对应关系,关系如表 1 所示。

表1 LSTMA与D{0-1}KP的对应关系表

Tab. 1 Correspondence table between LSTMA and D{0-1}KP

LSTMA	D{0-1}KP
维数	物品的数量
种群规模	备选方案的数量
一只猴子的位置	式(1)的一组可行解
适应度值	背包内物品的总价值

以式(1)为目标函数,采用二进制编码的方式对可行解进行编码,即 (X, Y) ,其中 X 为连续变量, Y 为离散变量。离散变量可由连续变量生成,该过程被称作离散化,可表示为:当 $X_{ij} > 0$ 时, $Y_{ij} = 1$;当 $X_{ij} \leq 0$ 时, $Y_{ij} = 0$ 。LSTMA在求解D{0-1}KP时,设置 X 的取值范围为 $[-5, 5]$ 。具体求解步骤如下:

步骤1 设置种群规模、迭代次数等参数,并生成初始种群 X 。

步骤2 背包问题的目的是在限定容量的背包内装入总价值量最多的物品,故对每件物品的价值密度进行降序排序。若某项集中有一项被选入背包中,那么该项集被记作 $Set_i = 1$,否则记为 $Set_i = 0$ 。价值密度可定义为:

$$density_j = p_j/w_j \quad (20)$$

步骤3 对种群 X 进行离散化操作,在不超过背包容积 W 的前提下,根据式(1)计算适应度值。

步骤4 进行爬过程更新,若猴群产生较优的个体,则替换原始个体。

步骤5 利用式(7),产生新种群,进行学习望过程。

步骤6 以 pp 为第一个支点,以 x_{best} 为第二个支点,随机选取式(6)和式(8),进行跳过程更新。

步骤7 在完成跳过程之后,挑选出种群中最优的个体,根据式(11)~(19),在最优解的周围产生插值点,进行Lagrange插值操作。

步骤8 重复步骤3~步骤7,直至达到总迭代次数,并将最优值与最优解输出。

LSTMA求解D{0-1}KP的流程如图1所示。

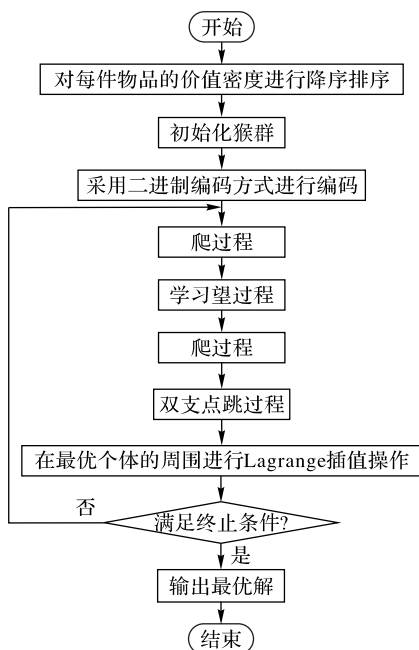


图1 LSTMA求解D{0-1}KP的流程

Fig. 1 Flowchart of LSTMA solving D{0-1}KP

3.2 仿真实验

现选用文献[14]中给出的四类D{0-1}KP的实例进行仿真。包括弱相关D{0-1}KP实例(Weakly correlated instances of D{0-1}KP, WDKP)、不相关D{0-1}KP实例(Uncorrelated instances of D{0-1}KP, UDKP)、逆向强相关D{0-1}KP实例(Inverse strongly correlated instances of D{0-1}KP, IDKP)和强相关D{0-1}KP实例(Strongly correlated instances of D{0-1}KP, SDKP),其中每类数据的维数以300维递增,选取 3×100 和 3×200 维的数据进行仿真。

将LSTMA的仿真结果与野草算法(Invasive Weed Optimization, IWO)^[15]、DE^[5]、PSO^[3]、GA^[4]、MA^[6]、混沌猴群算法(Chaotic Monkey Algorithm, CMA)^[8]、自适应猴群算法(Improved Monkey Algorithm, IMA)^[16]、野草猴群算法(Weed Monkey Algorithm, WMA)^[9]、蜂群优化(Artificial Bee Colony, ABC)算法^[17]及鱼群优化算法(Artificial Fish Swarms Algorithm, AFSA)^[18]进行对比。经过多次实验,LSTMA参数设置如下:爬步长 $a = 0.1$,跳区间 $[c, d] = [-1, 1]$,爬迭代次数 $Ng = 10$,望迭代次数 $Nw = 5$, $c_1 = 0.2$, $c_2 = 0.5$, $c_4 = c_5 = 0.25$ 。11个算法的总迭代次数 $Nc = 500$,种群规模 $M = 20$ 。而IWO^[15]、DE^[5]、PSO^[3]、GA^[4]、MA^[6]、CMA^[8]、IMA^[16]、WMA^[9]、ABC^[17]和AFSA^[18]参数均按原文献设置。LSTMA对IDKP1和IDKP2进行一次求解,绘制寻优曲线如图2和图3所示。

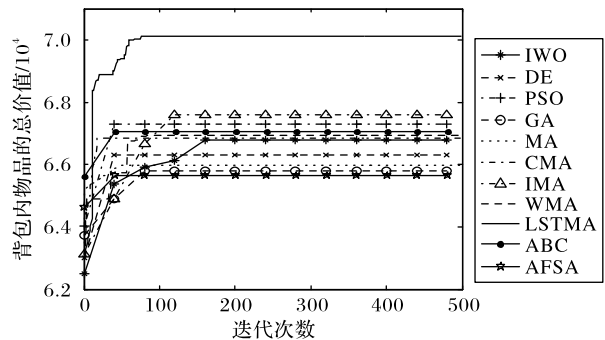


图2 实例IDKP1的寻优曲线

Fig. 2 Optimization curves of example IDKP1

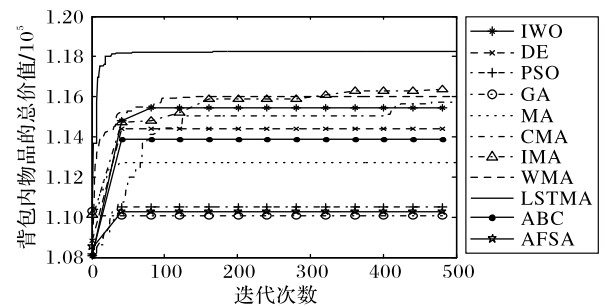


图3 实例IDKP2的寻优曲线

Fig. 3 Optimization curves of example IDKP2

由图2知,LSTMA求解的背包内物品的总价值最高,其最优值为70 037,IMA次之,其最优值为67 581。由图3知,LSTMA的求解精度依然高于对比算法,说明维数的增加,并不影响其寻优效果,并且其寻优曲线在90次迭代左右就趋于收敛状态,而WMA的曲线在150次迭代才趋于收敛,IMA和CMA的曲线在480次还出现不稳定现象。为了说明结果并非偶然,对上述实例进行20次求解,统计其最优值、平均值和方差,结果如表2所示。

表 2 十一种算法求解 D{0-1}KP 的结果
Tab. 2 Results of eleven algorithms in solving D{0-1}KP

实例及理论最优值	算法	最优值	平均值	方差	实例及理论最优值	算法	最优值	平均值	方差
WDKP1 (83 098)	IWO	82 567	81 712	152 443	WDKP2 (138 215)	IWO	135 973	135 157	837 247
	DE	80 047	78 989	297 220		DE	135 094	132 990	1 234 600
	PSO	79 046	77 705	652 082		PSO	131 707	130 080	1 674 100
	GA	78 442	77 030	319 020		GA	132 909	130 813	1 029 900
	MA	79 390	78 499	268 916		MA	132 223	130 961	1 179 916
	CMA	81 957	81 092	375 731		CMA	135 989	134 860	489 640
	IMA	81 597	80 445	261 622		IMA	136 775	136 280	79 981
	WMA	82 559	81 756	296 450		WMA	135 599	134 570	1 301 100
	LSTMA	83 090	83 062	835		LSTMA	138 091	138 017	7 621
	ABC	80 146	79 155	473 782		ABC	132 923	131 801	1 121 618
UDKP1 (85 740)	AFSA	78 722	77 278	325 237	UDKP2 (163 744)	AFSA	132 043	130 446	1 656 530
	IWO	83 278	81 389	1 405 031		IWO	156 630	153 639	4 507 333
	DE	75 381	73 932	1 087 875		DE	152 159	148 000	5 730 000
	PSO	73 054	70 909	1 331 747		PSO	145 469	142 784	2 362 182
	GA	71 590	70 382	874 653		GA	143 844	135 945	12 998 944
	MA	76 311	73 954	2 544 663		MA	146 716	142 060	5 142 500
	CMA	81 471	78 289	2 411 919		CMA	149 126	148 160	262 870
	IMA	82 295	80 525	1 941 485		IMA	154 063	151 920	1 357 000
	WMA	83 080	81 812	1 084 187		WMA	157 325	154 250	6 492 900
	LSTMA	85 588	85 188	52 783		LSTMA	163 020	161 936	371 189
IDKP1 (70 106)	ABC	75 401	74 010	914 508	IDKP2 (118 268)	ABC	145 716	143 996	1 017 389
	AFSA	72 836	72 078	565 083		AFSA	143 844	138 697	132 043
	IWO	66 795	66 608	394 122		IWO	116 146	115 577	222 440
	DE	66 314	65 584	1 657 539		DE	115 821	114 560	1 266 267
	PSO	67 302	67 007	949 446		PSO	111 790	109 898	2 553 611
	GA	65 806	65 904	111 835		GA	110 216	108 962	611 207
	MA	65 961	65 903	40 612		MA	113 403	111 810	830 060
	CMA	66 833	66 840	62 368		CMA	116 994	115 820	415 840
	IMA	67 583	67 583	296 210		IMA	117 897	116 880	435 310
	WMA	66 935	66 912	13 105		WMA	117 318	115 987	1 679 800
SDKP1 (94 459)	LSTMA	70 106	70 101	758	SDKP2 (160 805)	LSTMA	118 268	118 240	779
	ABC	67 050	66 976	21 696		ABC	114 321	113 263	1 093 031
	AFSA	65 950	65 858	7 766		AFSA	110 948	109 863	431 389
	IWO	92 998	91 914	292 126		IWO	157 609	155 886	1 021 991
	DE	92 841	92 265	193 496		DE	154 615	153 030	1 790 300
	PSO	89 824	88 939	332 830		PSO	152 409	149 380	3 335 500
	GA	88 097	87 622	160 525		GA	148 986	146 775	956 149
	MA	90 715	89 592	443 950		MA	152 716	150 560	1 417 100
	CMA	92 749	92 165	230 670		CMA	155 765	153 801	3 158 250
	IMA	93 714	92 697	369 400		IMA	157 892	156 180	1 453 300
	WMA	93 684	92 805	206 220		WMA	156 456	154 270	1 575 400
	LSTMA	94 189	94 068	9 517		LSTMA	160 323	160 025	15 966
	ABC	90 737	90 009	349 642		ABC	152 810	150 935	1 278 297
	AFSA	88 427	88 016	133 486		AFSA	149 986	148 145	583 464

由表 2 知,LSTMA 对于实例 IDKP1 和 IDKP2 的求解可以达到理论最优值,分别为 70 106 和 118 268。对于实例 WDKP、实例 UDKP、实例 SDKP,LSTMA 求解的背包内物品的总价值量明显高于对比算法,并且方差较小,说明 LSTMA 的寻优结果较稳定。为了更直观地比较十一种算法求解效果,选取实例 IDKP1 和 IDKP2 的 20 次结果分别绘制成箱线图,如图 4 和图 5 所示。

由图 4 可知:DE 的中位数最小,并且最大值和最小值差距较大,说明 DE 求解效果不稳定;WMA 的寻优结果较稳定,但是求解的精度不高;LSTMA 求解的背包内的物品总价值最

高,并且搜索结果较稳定。由图 5 知,LSTMA 在求解 D{0-1}KP 时不论是在求解精度方面还是求解的稳定性方面,都明显优于对比算法,算法具有良好的鲁棒性。

4 结语

针对猴群算法求解 D{0-1}KP 时求解精度不高的缺点,提出了 Lagrange 插值的学习猴群算法来求解该问题。首先,在望过程中,引入学习因子,重新定义视野长度,构造学习望过程,提高了算法的收敛速度。其次,在跳过程,将当前种群中最优的个体作为第二个支点,采用双支点进行跳过程更新,增

加了种群的多样性。最后,在跳过程之后,进行Lagrange插值操作,在最优解的周围挖掘潜在解,提高了算法的求解精度。利用LSTMA求解四类D{0-1}KP,结合寻优曲线图和箱线图,可知LSTMA求解的背包内物品的总价值高于对比算法,并且在收敛性方面同样占有优势,这为解决D{0-1}KP提供了一个新途径。

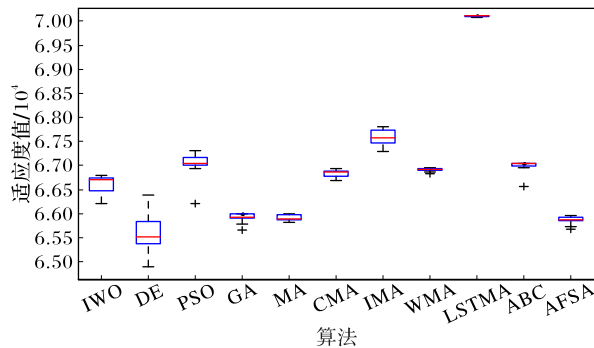


图4 实例IDKP1的箱线图

Fig. 4 Box plots of example IDKP1

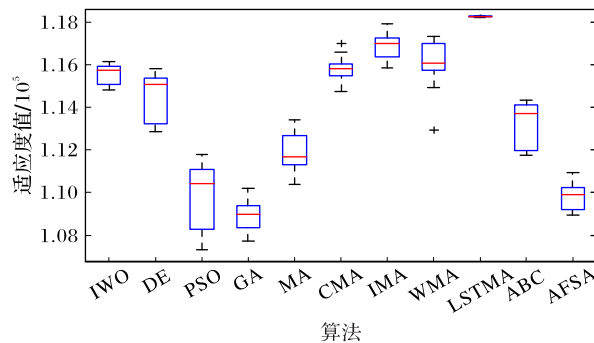


图5 实例IDKP2的箱线图

Fig. 5 Box plots of example IDKP2

参考文献(References)

- [1] FENG Y, WANG G, LI W, et al. Multi-strategy monarch butterfly optimization algorithm for discounted {0-1} knapsack problem[J]. Neural Computing and Applications, 2018, 30(10): 3019-3036.
- [2] RONG A Y, FIGUEIRA J R, KLAMROTH K. Dynamic programming based algorithms for the discounted {0-1} knapsack problem[J]. Applied Mathematics and Computation, 2012, 218(12): 6921-6933.
- [3] HE Y, WANG X, HE Y, et al. Exact and approximate algorithms for discounted {0-1} knapsack problem[J]. Information Sciences, 2016, 369: 634-647.
- [4] 贺毅朝,王熙照,李文斌,等. 基于遗传算法求解折扣{0-1}背包问题的研究[J]. 计算机学报, 2016, 39(12): 2614-2630. (HE Y C, WANG X Z, LI W B, et al. Research on genetic algorithms for the discounted {0-1} knapsack problem[J]. Chinese Journal of Computers, 2016, 39(12): 2614-2630.)
- [5] ZHU H, HE Y, WANG X, et al. Discrete differential evolutions for the discounted {0-1} knapsack problem[J]. International Journal of Bio-Inspired Computation, 2017, 10(4): 219-238.
- [6] ZHAO R, TANG W. Monkey algorithm for global numerical optimization[J]. Journal of Uncertain Systems, 2008, 2(3): 165-176.
- [7] LIANG Z. An improved monkey algorithm with dynamic adaptation[J]. Applied Mathematics and Computation, 2013, 222: 645-657.
- [8] 辛景舟,周建庭,杨俊,等. 基于混沌猴群算法的结构有限元模型修正[J]. 科学技术与工程, 2017, 17(34): 80-85. (XIN J Z, ZHOU J T, YANG J, et al. Finite element model updating based on chaotic monkey algorithm[J]. Science Technology and Engineering, 2017, 17(34): 80-85.)
- [9] 殷红,杜国璋,彭珍瑞,等. 野草猴群算法的传感器优化布置方法研究[J]. 计算机工程与科学, 2018, 40(4): 626-635. (YIN H, DU G Z, PENG Z R, et al. A weed monkey algorithm for optimal sensor placement[J]. Computer Engineering and Science, 2018, 40(4): 626-635.)
- [10] ZHOU Y, CHEN X, ZHOU G. An improved monkey algorithm for a 0-1 knapsack problem[J]. Applied Soft Computing, 2016, 38: 817-830.
- [11] FENG Y, WANG G. Binary moth search algorithm for discounted {0-1} knapsack problem[J]. IEEE Access, 2018, 6: 10708-10719.
- [12] ABIYEV R H, TUNAY M. Experimental study of specific benchmarking functions for modified monkey algorithm[J]. Procedia Computer Science, 2016, 102: 595-602.
- [13] ZHANG K, HUANG Q, ZHANG Y, et al. Hybrid Lagrange interpolation differential evolution algorithm for path synthesis[J]. Mechanism and Machine Theory, 2019, 134: 512-540.
- [14] 吴聪聪,贺毅朝,陈焱瑛,等. 变异蝙蝠算法求解折扣{0-1}背包问题[J]. 计算机应用, 2017, 37(5): 1292-1299. (WU C C, HE Y C, CHEN Y Y, et al. Mutated bat algorithm for solving discounted {0-1} knapsack problem[J]. Journal of Computer Applications, 2017, 37(5): 1292-1299.)
- [15] 杨澜,惠飞,侯俊,等. 基于免疫进化的野草优化随机搜索算法[J]. 计算机技术与应用, 2018, 28(2): 36-39, 44. (YANG L, HUI F, HOU J, et al. An invasive weed optimization algorithm based on immune evolution[J]. Computer Technology and Development, 2018, 28(2): 36-39, 44.)
- [16] 徐小平,张东洁. 一种改进的猴群算法[J]. 计算机系统应用, 2017, 26(6): 193-197. (XU X P, ZHANG D J. Improved monkey algorithm[J]. Computer Systems and Applications, 2017, 26(6): 193-197.)
- [17] OZCAN S, SIMSIR F. A new model based on artificial bee colony algorithm for preventive maintenance with replacement scheduling in continuous production lines[J]. Engineering Science and Technology, an International Journal, 2019, 22(6): 1175-1186.
- [18] LI J, DONG P. Global maximum power point tracking for solar power systems using the hybrid artificial fish swarm algorithm[J]. Global Energy Interconnection, 2019, 2(4): 351-360.

This work is partially supported by the National Natural Science Foundation of China (61773016), the Shaanxi Provincial Innovation Capability Support Program (2020PT-023), the Shaanxi Natural Science Basic Research Program (2018JQ1089).

XU Xiaoping, born in 1973, Ph. D., professor. His research interests include intelligent algorithm.

XU Li, born in 1995, M. S. candidate. Her research interests include intelligent algorithm.

WANG Feng, born in 1972, Ph. D., professor. Her research interests include intelligent algorithm.

LIU Long, born in 1975, Ph. D., professor. His research interests include machine learning.