

文章编号: 1001-0920(2013)02-0205-06

## 学习型和谐搜索算法及其在 0-1 背包问题中的应用

李若平<sup>1</sup>, 欧阳海滨<sup>1</sup>, 高立群<sup>1</sup>, 邹德旋<sup>2</sup>

(1. 东北大学 信息科学与工程学院, 沈阳 110819; 2. 徐州师范大学 电气工程及自动化学院, 江苏 徐州 221116)

**摘要:** 针对现有和谐搜索算法存在的不足, 提出一种学习型和谐搜索算法(LHS). 根据目标函数值的变化, 自适应调整和谐记忆考虑概率(HMCR); 引入学习机制, 加快算法的搜索速度; 动态调整基音调整概率(PAR), 增强算法的全局搜索能力. 对 16 个标准函数的测试结果表明, 所提出的 LHS 算法与其他 4 种和谐搜索算法相比具有较好的效果. 最后将改进算法应用于 10 个 0-1 背包问题和 1 个经典的 50 维背包实例, 实验结果表明 LHS 算法优于其他算法.

**关键词:** 和谐搜索算法; 自适应; 学习策略; 搜索速度; 0-1 背包问题

中图分类号: TP273

文献标志码: A

## Learned harmony search algorithm and its application to 0-1 knapsack problems

LI Ruo-ping<sup>1</sup>, OUYANG Hai-bin<sup>1</sup>, GAO Li-qun<sup>1</sup>, ZOU De-xuan<sup>2</sup>

(1. College of Information Science and Engineering, Northeastern University, Shenyang 110819, China; 2. School of Electrical Engineering and Automation, Xuzhou Normal University, Xuzhou 221116, China. Correspondent: OUYANG Hai-bin, E-mail: ouyang1987@163.com)

**Abstract:** For the purpose of avoiding the disadvantage of harmony search algorithm, a learned harmony search(LHS) algorithm is proposed. The adaptive parameter harmony memory consideration rate(HMCR) is designed based on the change of objective function value and the learning strategy is used to accelerate the speed of search. Then pitch adjustment rate(PAR) is adjusted dynamically to enhance the global search. The 16 classic test functions are tested, and the results show that LHS algorithm outperforms the other four harmony search algorithms. Finally, LHS algorithm is applied to 10 0-1 knapsack problems and a classic knapsack example, and the result shows that LHS algorithm is better than other algorithms.

**Key words:** harmony search algorithm; adaptive; learning strategy; search speed; 0-1 knapsack problem

### 0 引言

背包问题(KP)是运筹学和计算机科学中的一个典型优化难题,近年来引起了众多学者的广泛关注. 背包问题不仅具有重要的理论意义,同时具有广泛的实际应用价值,如预算控制、项目选择、材料切割和货物装载等领域都可视为 KP 问题的具体应用<sup>[1]</sup>. 对于背包问题,目前常用的方法有动态规划算法、分支限界算法和贪婪算法等. 随着启发式算法研究的不断深入,近年来许多学者将不同类型的启发式算法应用于 0-1 背包问题的求解,取得了一定的成果<sup>[2-5]</sup>,但仍存在稳定性差、成功率低等问题.

和谐搜索(HS)算法<sup>[6]</sup>是一种新的启发式算法,其思想源于音乐创作. 在音乐创作过程中,乐师们凭借多年音乐创作的经验即兴创作,不断调整音调,最

终得到美妙的和声.

近几年,研究者们探索了基音调整概率(PAR)、和谐记忆考虑概率(HMCR)和基音调整步长(bw)对和谐搜索算法的影响<sup>[7-9]</sup>,并进行了改进. Mahdavi等<sup>[8]</sup>提出了改进的和谐搜索算法(IHS),将参数 PAR 和 bw 设计为动态变化的,这样有利于提高 HS 算法的优化性能. Omran 等<sup>[9]</sup>于 2008 年提出了一种全局最好和谐搜索算法(GHS),用最优和声向量来产生新的和声向量,实验表明该算法具有很强的全局搜索能力和收敛性. Pan 等<sup>[10]</sup>于 2010 年提出了一种自适应全局最好和谐搜索算法(SGHS),引入一种新颖的参数学习周期  $L$ ,使 SGHS 算法能够在进化过程中不断地学习,解决了一定的优化问题. 邹德旋等<sup>[11]</sup>提出了一种新颖的全局和谐搜索算法(NGHS),该算法排除了 HS 算法中

收稿日期: 2012-01-10; 修回日期: 2012-03-28.

基金项目: 国家自然科学基金项目(60674021).

作者简介: 李若平(1964—),男,博士生,从事智能优化与图象处理等研究;高立群(1949—),男,教授,博士生导师,从事智能优化与图象处理等研究.

的 3 个参数: HMCR、PAR 和 bw, 引入了位置更新和变异操作, 进一步提高了算法的收敛性和全局寻优能力. 目前, 和声搜索算法仍然是人们关注和研究的热点之一.

本文提出一种学习型和声搜索算法(LHS). 该算法基于和声搜索算法良好的全局搜索能力, 采取自学习和邻近学习的策略, 大大改善了和声搜索算法存在的不足. 对 16 个标准函数的优化结果表明, 该算法具有很好的优化性能. 将该算法应用于 10 个 0-1 背包问题和 1 个经典背包实例的结果表明, LHS 算法优于其他搜索算法.

## 1 基本和声搜索算法

基本和声搜索算法的流程如下.

Step 1: 算法参数初始化. 给定和声记忆库大小 HMS、最大迭代次数  $K$  以及参数 HMCR、PAR、bw.

Step 2: 给定定义域  $[x_{jL}, x_{jU}]$ , 随机产生 HMS 个和声向量, 并存入和声记忆库 HM, 其中  $x_{jL}$  和  $x_{jU}$  分别为第  $j$  维和声解分量的下限和上限.

Step 3: 基于 HMCR、PAR 和 bw 进行即兴创作, 产生新的和声向量.

Step 4: 更新和声记忆库. 判断新和声向量是否优于当前 HM 内最差和声, 若成立, 则用新和声向量代替当前 HM 内的最差和声.

Step 5: 判断终止准则. 如果当前迭代次数  $k$  大于最大迭代次数  $K$ , 则终止运行 HS 算法, 否则重复执行 Step 3 和 Step 4.

## 2 学习型和声搜索算法

在基本和声搜索算法的基础上, 学习型和声搜索算法主要有 3 点改进, 分别是 HMCR 的自适应更新、学习策略和 PAR 的动态调整. 下面对这 3 点进行详细描述.

### 2.1 HMCR 的自适应更新

在和声搜索算法中, 参数 HMCR 表示从和声记忆库 HM 内选取和声向量的概率, 同时确定在 HM 外搜索的概率. 通过大量的仿真实验, 文献[4]指出较大的 HMCR 有利于局部搜索和算法的快速收敛, 较小的 HMCR 可以增加和声记忆库的多样性, 并建议 HMCR 取较大的值, 如  $\text{HMCR} \geq 0.9$ .

从本质上讲, HMCR 的选取不应依赖于迭代次数, 而应与目标函数值的变化有密切关系. 不失一般性, 假设优化的目的是使目标函数值最小. 当 HM 中不同和声向量对应的目标函数值相差较大时, HMCR 应取较小值, 这样可以增加和声记忆库的多样性, 利于全局大范围搜索和跳出局部最优解的邻域. 当 HM 中不同和声向量对应的目标函数值相差较小时,

HMCR 应取较大值, 以利于和声的精细搜索, 增加寻找到全局最优点的可能性, 并减少算法的搜索时间. 为此, 本文提出一种基于 HM 中不同和声向量目标函数值的差值变化自适应更新参数 HMCR 的方法, 其具体设计如下:

$$\text{HMCR}_k = \begin{cases} 1, & D_k = 0 \text{ or } D_k/D_{k-1} \geq 1; \\ D_k/D_{k-1}, & D_k/D_{k-1} < 1. \end{cases} \quad (1)$$

其中:  $D_k = (E_k - E_{\min})/(E_{\max} - E_k)$ ,  $E_k$ 、 $E_{\min}$  和  $E_{\max}$  分别代表第  $k$  代和声记忆库中和声向量目标函数的平均值、最小值和最大值, 设  $D_0 = 1$ . 迭代初期, 和声记忆库中目标函数的平均值往往偏向最大值, 即  $D_k$  与  $D_{k-1}$  的比值较大, 此时 HMCR 也取较大的值, 有利于局部搜索; 迭代后期, 目标函数值逐渐趋于某一邻域, 此时平均值  $E_k$  的变化相对缓慢, 使得  $D_k$  与  $D_{k-1}$  的比值变小, 较小的 HMCR 值会增加和声的多样性, 提高跳出局部最优的能力.

### 2.2 学习策略

和声搜索算法具有很强的全局搜索能力, 但搜索速度较为缓慢. 针对此不足, 本文提出一种新的学习策略, 具体表达式如下:

$$u_j^k = \frac{N - r_2}{N} \cdot x_{r_1, j}^k + C \cdot \text{rand} \cdot (x_{r_1, j}^k - x_{r_3, j}^k). \quad (2)$$

其中:  $u_j^k$  为候选和声向量  $u^k$  的第  $j$  维变量;  $N$  为和声向量的维数; rand 为 0 到 1 之间的随机数;  $C$  为学习因子;  $x_{r_1, r_2}^k$  为当前代中随机选取的第  $r_1$  个和声向量的第  $r_2$  维变量,  $r_1, r_3 \in \{1, 2, \dots, \text{HMS}\}$ ,  $r_2 \in \{1, 2, \dots, N\}$ .

由式(2)可知, 学习策略前半部分的作用是保持记忆库中和声向量的优势, 通过自我学习, 逐步缩小同一和声向量的变量之间的差距, 从而提高算法收敛性; 后半部分引入邻域学习, 通过与 HM 中另一随机和声向量的解向量比较, 扬长避短, 吸收邻域解向量的优势, 更好地跳出局部最优解. 学习策略既提高了算法的收敛能力, 同时又避免了算法早熟.

### 2.3 PAR 动态调整

改进和声搜索算法<sup>[3]</sup>将 PAR 设计为随迭代次数呈线性递增变化, 具体表达式如下:

$$\text{PAR}(k) = \text{PAR}_{\min} + k \cdot \frac{\text{PAR}_{\max} - \text{PAR}_{\min}}{K}. \quad (3)$$

其中:  $k$  和  $K$  分别为当前迭代次数和最大迭代次数;  $\text{PAR}_{\min}$  和  $\text{PAR}_{\max}$  分别为最小和声调整概率和最大和声调整概率.

从式(3)可以看出, 迭代早期 PAR 取较小的值, 迭代后期 PAR 取较大的值, PAR 在整个搜索过程中变化的速率是恒定的. 实际上, 在迭代早期, PAR 的变化应比较缓慢, 这样更能挖掘局部信息, 而随着迭代次

数的增加, PAR 的值应逐渐加快, 在迭代后期变化快一些可以加快搜索的速度. 基于以上分析, 本文在搜索过程中按下式设定参数 PAR:

$$\begin{aligned} \text{PAR}_k &= m \cdot \exp n \cdot k^2, \\ n &= \frac{1}{K^2 - 1} \cdot \ln \left( \frac{\text{PAR}_{\max}}{\text{PAR}_{\min}} \right), \\ m &= \text{PAR}_{\min} \cdot \exp -n. \end{aligned} \quad (4)$$

其中:  $k$ 、 $K$ 、 $\text{PAR}_{\min}$  和  $\text{PAR}_{\max}$  的具体含义同式(3). 假设  $K = 10\,000$ ,  $\text{PAR}_{\min} = 0.01$ ,  $\text{PAR}_{\max} = 0.99$ , 此时  $\text{PAR}_k$  的变化曲线如图1中实线所示, 虚线表示 IHS 算法中 PAR 的变化.

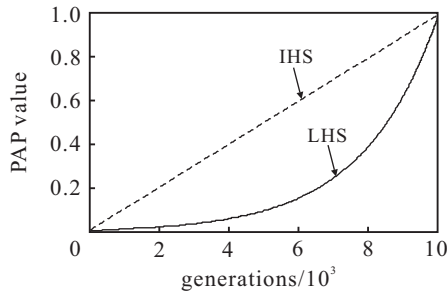


图1 PAR 动态变化曲线

### 2.4 学习型和声搜索算法的具体步骤

Step 1: 算法参数初始化. 设定 HMS、 $K$ 、 $C$ 、 $\text{bw}$ 、 $\text{PAR}_{\min}$  和  $\text{PAR}_{\max}$ , 并令  $D_0 = 1$ .

Step 2: 给定搜索范围  $[x_{jL}, x_{jU}]$ , 随机产生 HMS 个和声向量, 并存入和声记忆库 HM, 其中  $x_{jU}$  和  $x_{jL}$  为第  $j$  维变量的上下限.

Step 3: 即兴创作. 通过和声记忆考虑、基音调整和随机选择操作, 产生候选和声向量  $u^k$ , 即兴创作的具体流程如下:

```

For  $j = 1$  to  $N$  % 维数从1开始进行即兴创作
    If  $\text{rand} < \text{HMCR}_k$  % 和声记忆考虑
         $r_1, r_3 \in \{1, 2, \dots, \text{HMS}\}, r_2 \in \{1, 2, \dots, N\}$ 
         $u_j^k = \frac{(N - r_2)}{N} \cdot x_{r_1, j}^k + C \cdot \text{rand} \cdot (x_{r_1, j}^k - x_{r_3, j}^k)$ 
        If  $\text{rand} < \text{PAR}_k$  % 基音调整
             $u_j^k = u_j^k \pm \text{rand} \cdot \text{bw}$ 
        End If
    Else
         $u_j^k = x_{jL} + \text{rand} \cdot (x_{jU} - x_{jL})$  % 随机选择
    End If
End For % 产生备选和声向量  $u^k$ 
    
```

Step 4: 更新和声记忆库. 判断候选和声向量  $u^k$  的目标函数是否优于当前 HM 内和声向量目标函数的平均值. 若是, 则从当前 HM 内和声向量的目标函数数值比平均值差的和声向量中随机选择一个, 并用  $u^k$  代替它.

Step 5: 判断终止准则. 如果当前迭代次数  $k$  大于

最大迭代次数  $K$ , 则终止运行 LHS 算法, 否则重复执行 Step 3 和 Step 4.

### 2.5 实验及结果分析

选取文献[11]中的16个标准函数进行测试, 实验中各算法的参数值设置如表1所示. 将本文算法与 HS<sup>[6]</sup>、IHS<sup>[8]</sup>、SGHS<sup>[10]</sup>和 NGHS 算法<sup>[11]</sup>进行比较分析, 以验证本文算法的有效性.

表1 各算法的参数设置

| 算法                 | HS    | IHS                          | SGHS                         | NGHS    | LHS   |
|--------------------|-------|------------------------------|------------------------------|---------|-------|
| HMS                | 5     | 5                            | 5                            | 5       | 5     |
| HMCR               | 0.9   | 0.9                          | 0.98                         | —       | —     |
| PAR/pm             | 0.3/- | -/-                          | 0.9/-                        | -/0.005 | -/-   |
| PAR <sub>min</sub> | —     | 0.01                         | —                            | —       | 0.01  |
| PAR <sub>max</sub> | —     | 0.99                         | —                            | —       | 0.99  |
| bw                 | 0.01  | —                            | —                            | —       | 0.001 |
| bw <sub>min</sub>  | —     | 0.0001                       | 0.0005                       | —       | —     |
| bw <sub>max</sub>  | —     | $\frac{x_{jU} - x_{jL}}{20}$ | $\frac{x_{jU} - x_{jL}}{10}$ | —       | —     |
| $L/C$              | —     | —                            | 100                          | —       | 1.5   |

基于计算复杂性考虑, 为了提供一个合理的比较, 设定 IHS、SGHS 和 LHS 的最大迭代次数为 30 000, HS 和 NGHS 分别取 60 000 和 40 000. 每个函数的维数  $N$  为 30, 每种算法独立运行 30 次的优化结果如表 2 所示, 最优结果如黑体字所示.

从表 2 可以看出, 对于函数  $f_2$ 、 $f_7$  和  $f_{13}$ , 本文算法获得的最优值比 NGHS 算法得到的结果差, 但对于余下的 13 个测试函数, 本文算法所搜索到的最优值比其他 4 种和声搜索算法的好. 从平均值和标准差方面看, 除  $f_7$  函数外, 本文算法所获得的结果好于其他 4 种和声搜索算法, 这说明本文算法具有良好的优化精度和稳定性. 此外, 本文算法的平均时间最短, 表明本文算法提高了算法的搜索速度. 总之, 本文算法整体上优于其他算法.

## 3 LHS 算法在 0-1 背包问题中的应用

### 3.1 背包问题描述

0-1 背包问题可描述如下: 已知  $D$  个物体, 第  $j$  ( $j = 1, 2, \dots, D$ ) 个物体的重量和价值分别为  $w_j$  和  $p_j$ , 背包所能容纳的物体总重量上限为  $C$ . KP 问题的目标是在背包能够承受的重量范围内尽可能使背包中的物体价值总和最大化, 具体数学模型如下:

$$\begin{aligned} \max f(X) &= \sum_{j=1}^D p_j x_j; \\ \text{s.t.} \quad &\begin{cases} \sum_{j=1}^D w_j x_j \leq C; \\ x_j = 0 \text{ or } 1, j = 1, 2, \dots, D. \end{cases} \end{aligned} \quad (5)$$

### 3.2 模型转化

0-1 背包问题属于约束离散优化问题, 当用和声算法求解时需对模型进行转化处理.

表 2 5 种和声搜索算法获得的  $f_1 \sim f_{16}$  的优化结果

| 函数    | 算法   | 平均时间            | 最优值              | 平均最优值            | 标准差              | 函数       | 算法   | 平均时间            | 最优值              | 平均最优值            | 标准差             |
|-------|------|-----------------|------------------|------------------|------------------|----------|------|-----------------|------------------|------------------|-----------------|
| $f_1$ | HS   | 4.53e+00        | 5.94e-02         | 6.91e-01         | 7.57e-01         | $f_9$    | HS   | 4.82e+00        | -4.50e+02        | -4.49e+02        | 4.66e-01        |
|       | IHS  | 4.68e+00        | 1.66e+00         | 7.12e+00         | 3.43e+00         |          | IHS  | 5.03e+00        | -4.50e+02        | -4.43e+02        | 4.17e+00        |
|       | SGHS | 3.07e+00        | 5.28e-07         | 1.35e-02         | 1.37e-02         |          | SGHS | 3.33e+00        | -4.50e+02        | -4.50e+02        | 1.66e-02        |
|       | NGHS | 2.24e+00        | 5.73e-18         | 5.99e-16         | 1.00e-15         |          | NGHS | 2.43e+00        | -450             | -450             | 5.06e-14        |
|       | LHS  | <b>2.18e+00</b> | <b>2.16e-249</b> | <b>1.61e-237</b> | <b>0</b>         |          | LHS  | <b>2.30e+00</b> | -450             | -450             | <b>6.84e-14</b> |
| $f_2$ | HS   | 4.44e+00        | 1.78e+02         | 1.02e+03         | 1.86e+03         | $f_{10}$ | HS   | 7.27e+00        | 1.12e+03         | 2.43e+03         | 9.14e+02        |
|       | IHS  | 4.68e+00        | 2.44e+03         | 1.16e+04         | 7.77e+03         |          | IHS  | 5.91e+00        | 4.14e+03         | 7.75e+03         | 3.00e+03        |
|       | SGHS | 3.10e+00        | 3.89e+01         | 4.95e+02         | 1.39e+03         |          | SGHS | 4.18e+00        | 1.76e+02         | 9.71e+02         | 6.31e+02        |
|       | NGHS | 2.29e+00        | <b>2.17e-03</b>  | 1.12e+03         | 2.04e+03         |          | NGHS | 4.15e+00        | -3.99e+02        | -3.07e+02        | 8.57e+01        |
|       | LHS  | <b>2.02e+00</b> | 2.70e+01         | <b>2.74e+01</b>  | <b>4.09E-01</b>  |          | LHS  | <b>3.99e+00</b> | -450             | -450             | <b>2.07e-13</b> |
| $f_3$ | HS   | 4.77e+00        | 2.57e-02         | 9.37e-01         | 9.30e-01         | $f_{11}$ | HS   | 7.66e+00        | 1.32e+07         | 3.32e+07         | 1.02e+07        |
|       | IHS  | 4.89e+00        | 7.57e+00         | 1.18e+01         | 2.40e+00         |          | IHS  | 6.18e+00        | 1.64e+07         | 5.15e+07         | 1.89e+07        |
|       | SGHS | 3.17e+00        | 2.38e-06         | 2.55e+00         | 1.57e+00         |          | SGHS | 4.71e+00        | 5.82e+06         | 1.26e+07         | 4.15e+06        |
|       | NGHS | 2.52e+00        | <b>0</b>         | 1.57e-11         | 7.56e-11         |          | NGHS | 4.54e+00        | 1.15e+06         | 3.41e+06         | 1.58e+06        |
|       | LHS  | <b>2.35e+00</b> | <b>0</b>         | <b>7.70e-16</b>  | <b>1.73e-15</b>  |          | LHS  | <b>3.99e+00</b> | <b>-4.50e+00</b> | <b>-4.50e+02</b> | <b>1.03e-01</b> |
| $f_4$ | HS   | 4.92e+00        | 7.30e-01         | 9.78e-01         | 7.41e-02         | $f_{12}$ | HS   | 7.09e+00        | 1.33e+03         | 3.24e+03         | 1.11e+03        |
|       | IHS  | 4.99e+00        | 1.06e+00         | 1.15e+00         | 6.45e-02         |          | IHS  | 5.67e+00        | 4.86e+03         | 8.59e+03         | 2.86e+03        |
|       | SGHS | 3.28e+00        | 4.10e-02         | 1.67e-01         | 7.86e-02         |          | SGHS | 4.03e+00        | 3.12e+02         | 1.35e+03         | 8.01e+02        |
|       | NGHS | 2.62e+00        | <b>0</b>         | 5.50e-02         | 4.94e-02         |          | NGHS | 4.18e+00        | -4.13e+02        | -2.79e+02        | 8.76e+01        |
|       | LHS  | <b>2.46e+00</b> | <b>0</b>         | <b>3.70e-17</b>  | <b>8.42e-17</b>  |          | LHS  | <b>3.98e+00</b> | -450             | <b>-4.50e+02</b> | <b>9.20e-14</b> |
| $f_5$ | HS   | 4.78e+00        | 8.43e-03         | 6.15e-02         | 1.31e-01         | $f_{13}$ | HS   | 4.87e+00        | 4.88e+02         | 2.67e+03         | 3.19e+03        |
|       | IHS  | 4.93e+00        | 1.04e-01         | 1.84e+00         | 4.52e-01         |          | IHS  | 5.29e+00        | 3.13e+03         | 9.82e+03         | 5.19e+03        |
|       | SGHS | 3.18e+00        | 4.52e-05         | 1.16e-03         | 3.22e-03         |          | SGHS | 3.51e+00        | 4.33e+02         | 2.85e+03         | 4.08e+03        |
|       | NGHS | 2.41e+00        | 7.37e-10         | 5.05e-09         | 4.89e-09         |          | NGHS | 2.80e+00        | <b>3.90e+02</b>  | 1.67e+03         | 2.83e+03        |
|       | LHS  | <b>2.17e+00</b> | <b>7.11e-15</b>  | <b>1.58e-14</b>  | <b>1.45e-14</b>  |          | LHS  | <b>2.44e+00</b> | 4.18e+02         | <b>4.17e+02</b>  | <b>1.05e-01</b> |
| $f_6$ | HS   | 4.72e+00        | 3.18e-01         | 1.53e+00         | 7.02e-01         | $f_{14}$ | HS   | 7.47e+00        | -1.79e+02        | -1.78e+02        | 4.10e-01        |
|       | IHS  | 4.85e+00        | 4.16e-01         | 2.41e+00         | 1.44e+00         |          | IHS  | 6.17e+00        | -1.78e+02        | -1.75e+02        | 1.67e+00        |
|       | SGHS | 3.17e+00        | 1.29e-01         | 4.15e-01         | 1.76e-01         |          | SGHS | 4.68e+00        | -1.79e+02        | -1.79e+02        | 9.36e-02        |
|       | NGHS | 2.51e+00        | 1.34e-09         | 2.04e-08         | 2.33e-08         |          | NGHS | 4.39e+00        | -1.80e+02        | -1.80e+02        | 1.76e-02        |
|       | LHS  | <b>2.19e+00</b> | <b>4.99e-133</b> | <b>1.32e-120</b> | <b>7.21e-120</b> |          | LHS  | <b>3.88e+00</b> | -180             | <b>-1.80e+02</b> | <b>6.61e-14</b> |
| $f_7$ | HS   | 5.32e+00        | 3.72e+00         | 8.21e+00         | 3.53e+00         | $f_{15}$ | HS   | 8.46e+00        | -1.19e+02        | -1.19e+02        | 4.43e-02        |
|       | IHS  | 5.04e+00        | 3.47e+00         | 1.55e+01         | 9.41e+00         |          | IHS  | 7.65e+00        | -1.19e+02        | -1.19e+02        | 4.88e-02        |
|       | SGHS | 3.36e+00        | 1.05e-01         | 6.42e+00         | 1.79e+01         |          | SGHS | 5.63e+00        | -1.20e+02        | -1.20e+02        | 7.84e-02        |
|       | NGHS | 2.88e+00        | <b>3.82e-04</b>  | <b>3.82e-04</b>  | <b>1.01e-12</b>  |          | NGHS | 3.61e+00        | -1.20e+02        | -1.20e+02        | 8.85e-02        |
|       | LHS  | <b>2.52e+00</b> | 9.99e-01         | 9.58e+00         | 5.59e+01         |          | LHS  | <b>3.35e+00</b> | <b>-1.40e+02</b> | <b>-1.40e+02</b> | <b>1.18e-13</b> |
| $f_8$ | HS   | 6.39e+00        | 1.45e+03         | 2.64e+03         | 8.98e+02         | $f_{16}$ | HS   | 5.28e+00        | -3.30e+02        | -3.30e+02        | 3.03e-01        |
|       | IHS  | 5.38e+00        | 3.17e+03         | 6.46e+03         | 1.81e+03         |          | IHS  | 5.04e+00        | -3.28e+02        | -3.25e+02        | 1.53e+00        |
|       | SGHS | 3.75e+00        | 3.89e+02         | 9.57e+02         | 3.98e+02         |          | SGHS | 3.34e+00        | -3.29e+02        | -3.27e+02        | 1.28e+00        |
|       | NGHS | 3.66e+00        | 1.44e+01         | 8.18e+01         | 5.59e+01         |          | NGHS | 2.88e+00        | -330             | <b>-3.30e+02</b> | 8.60e-13        |
|       | LHS  | <b>2.74e+00</b> | <b>9.33e-53</b>  | <b>2.56e-33</b>  | <b>1.40e-32</b>  |          | LHS  | <b>2.73e+00</b> | -330             | <b>-3.30e+02</b> | <b>6.15e-14</b> |

### 3.2.1 约束条件处理

0-1 背包问题是一个带有约束条件限制的最大值优化问题, 约束条件为

$$g = \sum_{j=1}^D w_j x_j - C \leq 0.$$

本文采用一种较为常用的罚函数处理方法, 对不可行解施加惩罚, 而对可行解不施加惩罚, 同时取  $f(x)$  的相反数  $-f(x)$  将最大值问题转化为最小值问题求解. 经过以上两个转化处理后, 0-1 背包问题的目标函数可用下式表示:

$$\min F(x) = -f(x) + \lambda \cdot \max(0, g), \quad (6)$$

其中  $\lambda$  为惩罚系数, 本文取  $10^{20}$ .

### 3.2.2 离散变量处理

在 0-1 背包问题中, 变量均为整数变量且只有 0 和 1 两种取值方式, 而 LHS 算法运行过程中得到的解向量都是实数. 为了将实数变量转换为 0-1 背包问题所能使用的整数变量, 最直接的处理方法是先将变量的取值范围设置在  $0 \sim 1$  之间, 再将通过 LHS 运算后的实数变量按四舍五入原则直接转换为与其最为接近的整数.

### 3.3 实验结果和分析

本文对 0-1 背包问题进行两组实验, 以验证 LHS 算法的可行性和有效性. 实验 1 是对 10 个小维数的背包问题进行求解; 实验 2 是求解 1 个 50 维的背包实例.

**实验 1** 以文献 [12] 中的 10 个 0-1 背包问题为研究对象, 对 LHS、HS、IHS 和 NGHS 算法用 Matlab 7.1 编程实现, 在 Intel (R) Pentium (R) 4 CPU 2.93 GHz 的电脑上独立运行 50 次. 4 种算法的基本参数如表 1 所示, 获得的优化结果如表 3 所示.

表 3 背包问题  $f_1 \sim f_{10}$  的优化结果

| 函数       | 迭代次数  | 算法   | 最好值           | 平均时间           | 平均值           | 标准差      |
|----------|-------|------|---------------|----------------|---------------|----------|
| $f_1$    | 10000 | HS   | 295           | 0.476 3        | 294.3         | 2.21     |
|          | 10000 | IHS  | 295           | 0.486 9        | 294.86        | 0.51     |
|          | 5000  | NGHS | <b>295</b>    | 0.339 8        | <b>295</b>    | <b>0</b> |
|          | 3000  | LHS  | <b>295</b>    | <b>0.208 9</b> | <b>295</b>    | <b>0</b> |
| $f_2$    | 10000 | HS   | 1024          | 0.690 6        | 1024          | 0        |
|          | 10000 | IHS  | 1024          | 0.698 3        | 1024          | 0        |
|          | 10000 | NGHS | 1024          | 0.509 3        | 1024          | 0        |
|          | 5000  | LHS  | 1024          | 0.398 7        | 1024          | 0        |
| $f_3$    | 5000  | HS   | 35            | <b>0.298 5</b> | 35            | 0        |
|          | 5000  | IHS  | 35            | 0.340 5        | 34.58         | 1.68     |
|          | 3000  | NGHS | <b>35</b>     | 0.215 0        | <b>35</b>     | <b>0</b> |
|          | 1000  | LHS  | <b>35</b>     | <b>0.071 8</b> | <b>35</b>     | <b>0</b> |
| $f_4$    | 3000  | HS   | 23            | 0.190 3        | 23            | 0        |
|          | 3000  | IHS  | 23            | 0.229 5        | 23            | 0        |
|          | 3000  | NGHS | 23            | 0.209 8        | 23            | 0        |
|          | 1000  | LHS  | 23            | <b>0.072 5</b> | 23            | 0        |
| $f_5$    | 10000 | HS   | 481.07        | 0.567 5        | 481.07        | 0        |
|          | 10000 | IHS  | 481.07        | 0.589 8        | 478.48        | 10.35    |
|          | 5000  | NGHS | <b>481.07</b> | <b>0.389 5</b> | <b>481.07</b> | <b>0</b> |
|          | 5000  | LHS  | <b>481.07</b> | 0.390 4        | <b>481.07</b> | <b>0</b> |
| $f_6$    | 3000  | HS   | 52            | 0.189 4        | 52            | 0        |
|          | 3000  | IHS  | 52            | 0.200 4        | 50.65         | 1.85     |
|          | 2000  | NGHS | <b>52</b>     | 0.109 6        | <b>52</b>     | <b>0</b> |
|          | 1000  | LHS  | <b>52</b>     | <b>0.079 6</b> | <b>52</b>     | <b>0</b> |
| $f_7$    | 5000  | HS   | 107           | 0.301 5        | 106.8         | 0.61     |
|          | 10000 | IHS  | 107           | 0.449 0        | 103.98        | 4.48     |
|          | 5000  | NGHS | <b>107</b>    | <b>0.280 0</b> | <b>107</b>    | <b>0</b> |
|          | 3000  | LHS  | <b>107</b>    | <b>0.201 8</b> | <b>107</b>    | <b>0</b> |
| $f_8$    | 10000 | HS   | 9767          | 0.698 4        | 9767          | 0        |
|          | 10000 | IHS  | 9767          | 0.708 9        | 9767          | 0        |
|          | 8000  | NGHS | 9767          | 0.550 8        | 9767          | 0        |
|          | 5000  | LHS  | 9767          | <b>0.489 0</b> | 9767          | 0        |
| $f_9$    | 5000  | HS   | 130           | 0.387 0        | 130           | 0        |
|          | 5000  | IHS  | 130           | 0.400 8        | 130           | 0        |
|          | 3000  | NGHS | 130           | 0.230 9        | 130           | 0        |
|          | 3000  | LHS  | 130           | <b>0.230 0</b> | 130           | 0        |
| $f_{10}$ | 10000 | HS   | 1025          | 0.680 9        | 1025          | 0        |
|          | 10000 | IHS  | 1025          | 0.704 6        | 1025          | 0        |
|          | 8000  | NGHS | 1025          | 0.504 8        | 1025          | 0        |
|          | 5000  | LHS  | 1025          | <b>0.469 0</b> | 1025          | 0        |

从表 3 中可以看出: 针对以上 10 个 0-1 背包问题, 4 种和声搜索算法均能找到每个问题的全局最优解; 从平均值看, HS 算法在问题  $f_1$  和  $f_7$  上未能稳定地搜索到全局最优解, 在其他 8 个问题上的全局最优解均能找到; IHS 算法不能稳定寻找到问题  $f_1$ 、 $f_3$ 、 $f_5$ 、 $f_6$  和  $f_7$  的全局最优解, 说明 IHS 算法容易陷入局部最优; 最新提出的 NGHS 算法和本文算法都能够稳定地找到全局最优解, 而且对于所有问题的标准差都为 0, 但本文算法找到最优值所用的平均时间是最短的.

**实验 2** 为进一步说明 LHS 算法的可行性, 选取文献 [13-15] 中给定的 1 个 50 维经典的背包实例进行求解. LHS 算法求得的结果与其他文献报道结果的比较如表 4 所示.

表 4 LHS 算法与其他算法的比较

| 算法                      | 最优值(总价值/总重量)  |
|-------------------------|---------------|
| IACA <sup>[12]</sup>    | 3 103 / -     |
| ISAGA <sup>[13]</sup>   | 3 090 / -     |
| 病毒粒子群算法 <sup>[14]</sup> | 3 103 / 1 000 |
| GDDEA <sup>[15]</sup>   | 3 095 / 1 000 |
| LHS                     | 3 114 / 1 000 |

文献 [13] 指出, 改进蚁群算法 IACA 求出的结果优于基本蚁群算法; 文献 [14] 表明, ISAGA 算法求出的结果优于模拟退火算法 SA 和遗传算法 GA; 文献 [15] 指出, 病毒粒子群算法所取得的结果优于 PSO 算法; 文献 [16] 指出, GDDEA 获得的结果优于 DE 算法. 但从表 4 可以看出, LHS 算法求出的结果比这些文献报道的算法好, 表明 LHS 算法能很好地应用于 0-1 背包问题.

#### 4 结 论

本文针对现有和声搜索算法存在的不足, 提出了一种学习型和声搜索算法 (LHS). 该算法引入一种新的学习搜索机制, 给出了参数 HMCR 和 PAR 新的更新策略. 对 16 个测试函数的仿真实验表明, 与其他 4 种和声搜索算法相比, LHS 算法能够很好地改善 HS 算法的优化性能. 采用 LHS 算法求解 0-1 背包问题的结果表明, LHS 算法与其他 3 种和声搜索算法以及以往文献中的智能优化算法相比, 具有更好的优化效果.

#### 参考文献(References)

- [1] Fayard D, Plateau G. Resolution of the 0-1 knapsack problem comparison of methods[J]. Mathematical Programming, 1975, 8(1): 272-307.
- [2] 闫杨, 汪定伟, 王大志, 等. 求解动态背包问题的多智能体进化算法[J]. 东北大学学报: 自然科学版, 2009, 30(7): 948-951.  
(Yan Y, Wang D W, Wang D Z, et al. Multi-agent-based evolutionary algorithm for dynamic knapsack problem[J]. J of Northeastern University: Natural Science, 2009, 30(7): 948-951.)
- [3] Shi H X. Solution to 0/1 knapsack problem based on improved ant colony algorithm[C]. Int Conf on Information Acquisition. Weihai, 2006: 1062-1066.
- [4] Li Z K, Li N. A novel multi-mutation binary particle swarm optimization for 0/1 knapsack problem[C]. 21st Chinese Control and Decision Conf. Guilin, 2009: 3042-3047.
- [5] Liu Y, Liu C. A schema-guiding evolutionary algorithm for 0-1 knapsack problem[C]. 2009 Int Association of

- Computer Science and Information Technology-Spring Conf. Singapore, 2009: 160-164.
- [6] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: Harmony search[J]. *Simulation*, 2001, 76(2): 60-68.
- [7] Zong Woo Geem, Sim Kwee-Bo. Parameter-setting-free harmony search algorithm[J]. *Applied Mathematics and Computation*, 2010, 217(8): 3881-3889.
- [8] Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems[J]. *Applied Mathematics and Computation*, 2007, 188(2): 1567-1579.
- [9] Omran M G H, Mahdavi M. Global-best harmony search [J]. *Applied Mathematics and Computation*, 2008, 198(2): 643-656.
- [10] Pan Q K, Suganthan P N, Tasgetiren M F, et al. A self-adaptive global best harmony search algorithm for continuous optimization problems[J]. *Applied Mathematics and Computation*, 2010, 216(3): 830-848.
- [11] Zou D X, Gao L Q, Wu J H, et al. Novel global harmony search algorithm for unconstrained problems[J]. *Neurocomputing*, 2010, 73(16/17/18): 3308-3318.
- [12] Zou D X, Gao L Q, Wu J H, et al. Solving 0-1 knapsack problem by a novel global harmony search algorithm[J]. *Applied Soft Computing*, 2011, 11(2): 1556-1564.
- [13] 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, Zhan Z G. Solving 0-1 knapsack problem based on genetic algorithm with improved simulated annealing[J]. *Microelectronics and Computer*, 2011, 28(2): 61-64.)
- [14] 高芳, 崔刚, 吴智博, 等. 求解背包问题的病毒协同进化粒子群算法[J]. *哈尔滨工业大学学报*, 2009, 41(6): 103-107.  
(Gao F, Cui G, Wu Z B, et al. Virus-evolutionary particle swarm optimization algorithm for knapsack problem[J]. *J of Harbin Institute of Technology*, 2009, 41(6): 103-107.)
- [15] 苗世清, 高岳林. 求解 0/1 背包问题的离散差分进化算法[J]. *小型微型计算机系统*, 2009, 30(9): 1828-1830.  
(Miao S Q, Gao Y L. Discrete differential evolution algorithm for solving 0/1 knapsack problem[J]. *J of Chinese Computer Systems*, 2009, 30(9): 1828-1830.)
- [16] 廖灿星, 李行善, 张平, 等. 一种求解背包问题的正太分布蚁群算法[J]. *系统仿真学报*, 2011, 23(6): 1156-1160.  
(Liao C X, Li X S, Zhang P, et al. Improved ant colony algorithm based on normal distribution for knapsack problem[J]. *J of System Simulation*, 2011, 23(6): 1156-1160.)

(上接第204页)

#### 参考文献(References)

- [1] Utkin V I. Sliding modes in control and optimization[M]. Berlin: Springer-Verlag, 1992: 10-11.
- [2] Boiko I, Fridman L. Analysis of chattering in continuous sliding-mode controllers[J]. *IEEE Trans on Automatic Control*, 2005, 50(9): 1442-1446.
- [3] Lee H, Utkin V I. Chattering suppression methods in sliding mode control systems[J]. *Annual Reviews in Control*, 2007, 31(2): 179-188.
- [4] Bartolini G, Ferrara A, Usai E. Chattering avoidance by second-order sliding mode control[J]. *IEEE Trans on Automatic Control*, 1998, 43(2): 241-246.
- [5] Levant A. Higher-order sliding modes, differentiation and output-feedback control[J]. *Int J of Control*, 2003, 76(9/10): 924-941.
- [6] Levant A. Homogeneity approach to high-order sliding mode design[J]. *Automatica*, 2005, 41(5): 823-830.
- [7] Levant A. Quasi-continuous high-order sliding-mode controllers[J]. *IEEE Trans on Automatic Control*, 2005, 50(11): 1812-1816.
- [8] Pukdeboon C, Zinober A S I, Thein M L. Quasi-continuous higher order sliding-mode controllers for spacecraft-attitude-tracking maneuvers[J]. *IEEE Trans on Industrial Electronics*, 2010, 57(4): 1436-1444.
- [9] Levant A. Sliding order and sliding accuracy in sliding mode control[J]. *Int J of Control*, 1993, 58(6): 1247-1263.
- [10] Levant A. Principles of 2-sliding mode design[J]. *Automatica*, 2007, 43(4): 576-586.
- [11] Zhou K, Doyle J C. Essentials of Robust Control[M]. New Jersey: Prentice Hall, 1998: 13-14.