文章编号: 1001-0920 (2016) 02-0310-07 DOI: 10.13195/j.kzyjc.2014.1742

全局竞争和声搜索算法

夏红刚[1,2], 欧阳海滨[1], 高立群[1], 孔祥勇[1]

(1. 东北大学 信息科学与工程学院，沈阳 110004；2. 沈阳大学 信息工程学院，沈阳110044)

摘 要: 提出一种全局竞争和声搜索 (GCHS) 算法, 给出随机局部平均和声和全局平均和声的概念, 建立竞争搜索机制, 实现每次迭代产生两个和声向量并进行竞争选择. 设计自适应全局调整和局部学习策略, 平衡算法的局部搜索和全局搜索, 详细分析参数 HMS、HMCR 和 PAR 对算法优化性能的影响. 数值结果表明, GCHS 算法在精度、收敛速度和鲁棒性方面比和声搜索算法及最近文献中提出的 7 种优秀改进和声搜索算法要好.

关键词: 和声搜索算法；竞争搜索机制；自适应全局调整；局部学习策略

中图分类号: TP301.6 **文献标志码:** A

Global competitive harmony search algorithm

XIA Hong-gang[1,2], *OUYANG Hai-bin*[1], *GAO Li-qun*[1], *KONG Xiang-yong*[1]

(1. College of Information Science and Engineering，Northeastern University，Shenyang 110004，China；2. School of Information Engineering，Shenyang University，Shenyang 110044，China. Correspondent：OUYANG Hai-bin, E-mail：oyhb1987@163.com)

Abstract: A global competitive harmony search algorithm(GCHS) is proposed. In this algorithm, the conceptions of stochastic local mean and global mean are given. The competition search mechanism is built to realize two harmony vectors are competition selection, and the two harmony vectors are both generated in the each iteration. The adaptive global pitch adjustment and local learning strategy are designed to balance the global search and local search. The effects that the parameter HMS, HMCR and PAR have on the performance of the GCHS algorithm are also analyzed in detail. The numerical results show the superiority of the proposed GCHS algorithm in terms of accuracy, convergence speed, and robustness when compared with the harmony search algorithm and other seven state-of-the-art harmony search algorithms.

Keywords: harmony search algorithm；competition search；adaptive global pitch adjustment；local learning

0 引 言

最优化是指从所有可能的解中挑选出合理可行的最优解, 能够找到最优解的方法是最优化方法. 由于现实生活中, 各个领域经常碰到时间最少、成本花费最低、风险最小或经济利益最大、质量最好、速度最快等复杂问题, 这些复杂问题的解决迫切需要有效的优化方法. 因此, 最优化方法一直是学者们研究的热点. 尤其, 一些新的启发式算法如遗传算法、粒子群优化算法、差分进化算法、蚁群算法、人工蜂群算法、量子进化算法及和声搜索算法, 吸引了学者们的研究兴趣.

和声搜索算法 (HS) 是由 Geem 等[1]提出的一种随机搜索算法, 它凭借独特的随机搜索方式, 在许多实际优化问题中得到了广泛应用, 如电力系统环境经济调度问题[2]、混合稳态自适应模糊控制器的设计[3]、地下水管理[4]和动态经济负载调度问题[5]. 由于 HS 算法的优化性能和效率存在很大的提升空间, 许多学者进行了大量研究并提出了一些改进算法, 如改进和声搜索算法 (IHS)[6]、全局最好和声搜索算法 (GHS)[7]、改进全局和声搜索算法 (IGHS)[8]、全局动态和声搜索算法 (GDHS)[9]、新颖的全局和声搜索算法 (NGHS)[10]、带有动态控制参数的和声搜索算法 (NDHS)[11]和探索和声搜索算法 (EHS)[12]. 在一定程度上, 这些改进算法提高了和声搜索算法的优化性能, 但仍存在着搜索盲目、易陷入局部最优、收敛精度不高的缺陷.

本文分析了 HS 算法存在的不足, 提出一种全局竞争和声搜索 (GCHS) 算法. 该算法提出了自适应全

收稿日期: 2014-11-15；**修回日期:** 2015-02-09.

基金项目: 国家自然科学基金项目(61403174).

作者简介: 夏红刚(1980−), 男, 博士生, 从事智能优化与系统建模的研究；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.

局调整和局域学习策略, 并取代了 HS 算法的基音调整, 改善了算法的搜索空间, 采用竞争选择模式, 舍去了贪婪选择策略, 增加算法跳出局部最优的可能性. 最后通过仿真测试, 表明了所提出算法的有效性.

1　HS 算法及其改进算法

1.1　HS 算法

算法 1 (HS 算法)[9]　HS 算法的基本步骤如下.

Step 1: 算法参数初始化. 给定和声记忆库大小 (HMS)、和声记忆库考虑概率(HMCR)、基音调整概率 (PAR)、和声微调步长 (BW) 和最大迭代次数 K.

Step 2: 给定范围 $[x_i^L, x_i^U]$, 其中 x_i^L 和 x_i^U 分别为第 i 维变量的下限和上限. 根据下式随机初始化, 产生 HMS 个和声向量存入和声记忆库 (HM):

$$x_i = x_i^L + \text{rand} \times (x_i^U - x_i^L). \tag{1}$$

Step 3: 基于 HMCR、PAR 和 BW 进行即兴创作, 产生新的和声向量, 具体伪代码如下所示:

For i =1 to N

If rand $<$ HMCR 和声记忆库考虑

$r \in \{1, 2, \cdots, \text{HMS}\}$, x_i^{new}=x_i^r;

If rand $<$ PAR 基音调整

$x_i^{\text{new}} = x_i^r \pm \text{rand} \times \text{BW}$

End If

Else 随机变异

$x_i^{\text{new}} = x_i^L + \text{rand} \times (x_i^U - x_i^L)$

End If

End For

其中: N 为问题的维数, 在进行基音调整时, 随机数大于 0.5 取 + 号, 小于 0.5 取 − 号.

Step 4: 更新和声记忆库. 判断新和声 x^{new} 是否优于当前 HM 内最差和声 x^{worst}, 若是则用 x^{new} 代替 x^{worst}.

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.

1.2　几种经典的改进和声搜索算法

算法 2 (IHS 算法)　Mahdavi 等将基音调整概率 PAR 设计为随着迭代次数的增加呈线性递增, 基音调整步长 BW 随着迭代次数的增加呈指数递减[6], 具体表达如下:

$$\text{PAR}_k = \text{PAR}_{\min} + \frac{(\text{PAR}_{\max} - \text{PAR}_{\min})}{K} \times k, \tag{2}$$

$$\text{BW}_k = \text{BW}_{\max} \exp\left(\frac{\text{In}(\text{BW}_{\min}/\text{BW}_{\max})}{K} \times k\right). \tag{3}$$

其中: k 为当前迭代次数, K 为最大迭代次数, $\text{PAR}_{\max}$ 和 $\text{PAR}_{\min}$ 分别为最大基音调整概率和最小基音调整概率, $\text{BW}_{\max}$ 和 $\text{BW}_{\min}$ 分别为最大微调步长和最小微调步长. IHS 算法调整了参数, 降低了算法对参数的敏感性, 提高了搜索的有效性. 但是, IHS 算法的参数调节具有一定盲目性, 不能有效适应搜索的进程.

算法 3 (GHS 算法)　Omran 等[7]提出了一种全局最好和声搜索算法 (GHS). GHS 算法运用和声记忆库中的最好和声 $x^{\text{best}} = (x_1^{\text{best}}, x_2^{\text{best}}, \cdots, x_N^{\text{best}})$ 来产生新和声 x^{new}, 其表达式为

$$x_i^{\text{new}} = x_i^{\text{best}}. \tag{4}$$

GHS 算法舍去了步长 BW 的设置, 提高了搜索的快速性, 但容易陷入局部极值.

算法 4 (NGHS 算法)　Zou 等提出了一种新颖的全局和声搜索算法, 并成功应用于工作分配问题[13]、系统可靠性问题[14]和 0-1 背包问题[15]. NGHS 算法在即兴创作过程中引入了位置更新和变异操作, 取代了 HS 算法即兴创作过程中的和声记忆考虑、基音调整和随机产生操作, 舍去了贪婪准则, 直接用新和声取代和声记忆库中的最差和声. 位置更新和变异操作分别为

$$x_i^{\text{new}} = x_i^{\text{worst}} + \text{rand} \times (x_r - x_i^{\text{best}}), \tag{5}$$

$$x_r = \min(\max(2 \times x_i^{\text{best}} - x_i^{\text{worst}}, x^L), x^U), \tag{6}$$

$$x_i^{\text{new}} = x^L + \text{rand} \times (x^U - x^L). \tag{7}$$

其中: x_i^{best} 和 x_i^{worst} 分别为最好和声和最差和声的第 i 维分量, x_i^{new} 为新和声的第 i 维分量, $\min(\max(A, B), C)$ 表示先取 A 和 B 中较大的, 再与 C 比较取较小的. NGHS 算法应用搜索进程中的有用信息, 提高了搜索的有效性, 尽管结合了小概率的变异操作, 但由于搜索的空间逐渐狭小, 容易陷入当前全局最优.

2　全局竞争和声搜索算法 (GCHS)

2.1　HS 算法分析

HS 算法具有良好的优化潜力, 但其本身也存在一些不足, 鉴于本课题组对和声搜索的仿真研究, 总结如下:

1) HS 算法参数多且难以有效设置. 这些参数包括和声记忆库大小 HMS、和声记忆库考虑概率 HMCR、基音调整概率 PAR 和基音调整步长 BW. 参数一旦初始化后, 迭代过程中不再改变, 这样难以适应搜索进程的需要, 容易使算法搜索盲目, 效率低. 文献 [13] 从数学理论上推导了 HS 算法搜索能力与参数的关系, 并证明 BW 的设置直接影响算法的收敛性. BW 如何确定一直是研究 HS 算法的一个难题, BW 过大导致搜索过于广泛, BW 太小又容易陷入局部极值.

2) HS 算法缺乏指导信息, 搜索效率低. 从 HS 算

法的Step 3可知, 即兴创作主要包括3个操作: 和声记忆库考虑、基音调整和随机选择. 假设即兴创作产生一个D维的新和声向量, 考虑整个即兴创作都基于随机数的变化来进行, 那么该新和声向量结构可能有7种, 由1个操作单独产生的有3种, 任意两个操作产生的有3种, 还有一种由3个操作产生. 新和声向量由3部分组成, 设总维数为D, 有$(1-\text{HMCR})\times D$维和声分量由随机选择产生, HMCR PAR $\times D$维和声分量由基音调整得到, HMCR $(1-\text{PAR})\times D$维和声分量来自和声记忆库. 由即兴创作可以看出, 随机选择产生的和声分量类似于随机初始化, 从和声记忆库选择的和声分量是迭代保存下来的, 而基音调整得到的和声分量是通过步长调整得到的, 这3部分和声分量都缺乏指导性信息, 带有较强的随机性和盲目性, 难以有效地搜索到较好的解.

3) 贪婪选择. 使算法易陷入局部最优HS算法利用新产生的解与和声记忆库中的最差的解相比较, 如果较优则取代. 这种选择方式实质上为贪婪选择策略, 容易错过一些具有优化潜力的新解, 极有可能造成和声记忆库经过多次迭代也没有得到更新, 陷入了停滞状态.

针对HS算法的不足, 本文提出一种全局竞争和声搜索算法, 首先给出全局平均和声和随机局部平均和声的概念, 然后具体介绍自适应全局调整、局部学习和竞争选择.

2.2 全局平均和声和随机局部平均和声

定义1 全局平均和声是指整个和声记忆库中所有和声每一维和声分量的平均值所组合成的和声向量. 设全局平均和声为

$$\bar{x}^{gl}=(\bar{x}_1^{gl},\bar{x}_2^{gl},\cdots,\bar{x}_i^{gl},\cdots,\bar{x}_N^{gl}),$$

由定义可知

$$\bar{x}_i^{gl}=\frac{1}{\text{HMS}}\sum_{m=1}^{\text{HMS}}x_i^m. \tag{8}$$

定义2 随机局部平均和声是指从和声记忆库中随机选择p个和声向量每一维和声分量的平均值所组成的和声向量. 设随机局部平均和声为

$$\bar{x}^{pl}=(\bar{x}_1^{pl},\bar{x}_2^{pl},\cdots,\bar{x}_i^{pl},\cdots,\bar{x}_N^{pl}),$$

由定义可知

$$\bar{x}_i^{pl}=\frac{1}{p}\sum_{m=1}^{p}x_i^{u_m}, \tag{9}$$

其中随机选择的p个和声的标号为$u_1,u_2,\cdots,u_p$. 具体产生步骤如下.

Step 1: 确定随机选择的数目

$$p=2+\text{round}(\text{rand}(\text{HMS}-2)),$$

保证$p>2$.

Step 2: 利用原有和声数目产生随机序列, 如HMS $=5$, 一个随机序列为$u=(3,1,4,5,2)$.

Step 3: 利用式(9)计算随机局部平均和声的每一维分量, 最终得到随机局部平均和声.

2.3 自适应全局调整

由于步长BW难以设置, 且HS算法缺乏指导信息, 受群智能思想的启发, 提出一种自适应全局调整操作, 具体表达式为

$$x_i^{\text{new}}=$$
$$x_i^{\text{new}}+(2\text{rand}-1)\times(x_i^{\text{best}}-C\times\bar{x}_i^{gl}). \tag{10}$$

其中: x_i^{best}为最好和声的第i维分量; x_i^{new}为新和声的第i维分量; rand为0～1之间的随机数; C为全局学习因子, 且$C=\text{round}(1+\text{rand})$, C只取1或2, 目的是更好地利用全局平均和声进行广泛而有效的搜索; round(x)为对x取整; $\bar{x}_i^{gl}$为全局平均和声的第i维分量.

自适应全局调整舍去了步长BW的设置, 引入全局最好和声和全局平均和声, 增加了算法的指导信息. 一方面通过全局平均和声检测当前和声记忆库的平均水平, 另一方面利用当前全局最好和声指导算法的进一步搜索, 增强算法搜索的有效性. 基于当前全局最好和声与全局平均和声的差异, 不断改善每一个新和声的质量, 这样和声记忆库中和声将不断向全局最好和声移动, 全局平均和声也不断提高, 减少了搜索的盲目性.

2.4 局域学习

为了增强算法的局部搜索能力, 使算法在不同的搜索区域都进行开采和探索, 本文提出一种局域学习策略, 有

$$x_i^{\text{new}}=$$
$$\begin{cases}x_i^{\text{new}}+r_1\times(\bar{x}_i^{pl}-x_i^{\text{new}}),\ r_2<0.5;\\ x_i^{\text{new}}-r_1\times(\bar{x}_i^{pl}-x_i^{\text{new}}),\ r_2\geqslant 0.5.\end{cases} \tag{11}$$

其中: r_1和r_2均为0～1之间的随机数; $\bar{x}_i^{pl}$为随机局部平均和声的第i维分量.

局域学习通过随机局部平均和声的引导, 在不同的搜索区域进行探索. 虽然随机选择的和声不能完全表征整个和声记忆库的信息, 但反映了某一个局部搜索区间的信息, 这样可以增加算法的多样性, 扩展搜索空间. 此外, 局域学习是自适应全局调整的一个补充, 使算法能够利用整体信息的全面性, 也考虑到局部信息的变化.

2.5 竞争选择

HS算法每次迭代只产生一个和声向量, 并且与和声记忆库中的最差和声向量进行比较, 如果较优则

取代, 这使得HS算法搜索效率低, 且容易陷入局部极值. 本文考虑建立竞争选择机制, 利用自适应全局调整和局域学习各产生一个和声向量, 两个和声分别带有全局信息和局部信息的特点, 通过这两个和声向量的竞争与比较, 选择较优的取代和声记忆库中的最差和声向量. 竞争选择较好地融合了自适应全局调整和局域学习操作, 平衡了算法的全局搜索和局部搜索. 同时, 每次迭代直接更新和声记忆库, 加快了和声记忆库信息更新的速率.

2.6 GCHS算法流程

GCHS算法与HS算法流程基本一致, 主要的不同为即兴创作部分, GCHS算法的具体流程如下.

Step 1: 算法和问题参数初始化.

Step 2: 和声记忆库初始化.

Step 3: 即兴创作, 产生新和声.

For i =1 to N

If rand $<$ HMCR 和声记忆库考虑

$r \in \{1, 2, \cdots, \mathrm{HMS}\}$, $x_i^{\mathrm{new2}} = x_i^r$, $x_i^{\mathrm{new1}} = \bar{x}_i^{pl}$;

If rand $<$ PAR 基音调整

$x_i^{\mathrm{new1}} = x_i^{\mathrm{new1}} + (2 \times \mathrm{rand} - 1) \times (x_i^{\mathrm{best}} - C \times \bar{x}_i^{gl})$

If rand <0.5

$x_i^{\mathrm{new2}} = x_i^{\mathrm{new2}} + \mathrm{rand} \times (\bar{x}_i^{pl} - x_i^{\mathrm{new2}})$

Else

$x_i^{\mathrm{new2}} = x_i^{\mathrm{new2}} - \mathrm{rand} \times (\bar{x}_i^{pl} - x_i^{\mathrm{new2}})$

End If

End If

Else 随机变异

$x_i^{\mathrm{new}} = x_i^L + \mathrm{rand} \times (x_i^U - x_i^L)$

$x_i^{\mathrm{new2}} = x_i^L + \mathrm{rand} \times (x_i^U - x_i^L)$

End If

End For

Step 4: 通过竞争选择, 更新和声记忆库.

Step 5: 判断终止条件, 如果当前迭代次数 k 等于最大迭代次数 K, 则算法停止, 否则重复执行Step 3和Step 4.

3 实验结果和分析

3.1 实验准备

实验应用8个经典测试函数对全局竞争和声搜索算法进行测试, 测试函数的维数、搜索空间和最优值如表1所示, 表达式如下:

f_1: Sphere function

$$\min f_1 = \sum_{i=1}^{N} x_i^2;$$

f_2: Schwefel's problem 2.22

$$\min f_2 = \sum_{i=1}^{N} |x_i| + \prod_{i=1}^{N} |x_i|;$$

f_3: Rastrigrin function

$$\min f_3 = \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i) + 10);$$

f_4: Griewank function

$$\min f_4 = \frac{1}{4\,000} \sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos\Big(\frac{x_i}{\sqrt{i}}\Big) + 1;$$

f_5: Ackley's function

$$\min f_5 = 20 + e - 20\exp\Big(-0.2\sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}}\Big) - \exp\Big(\frac{\sum_{i=1}^{N} \cos(2\pi x_i)}{N}\Big);$$

f_6: Schwefel's problem 2.26

$$\min f_6 = -\sum_{i=1}^{N} (x_i \sin(\sqrt{|x_i|}));$$

f_7: Schafferf 7 function

$$\min f_7 = \sum_{\mathrm{i}=1}^{n-1} [(x_i^2 + x_{i+1}^2)^{0.25} (\sin(50(x_i^2 + x_{i+1}^2)^{0.1})^2 + 1)];$$

f_8: Zakharov function

$$\min f_8 = \sum_{i=1}^{N} x_i^2 + \Big(\sum_{i=1}^{N} 0.5 i x_i\Big)^2 + \Big(\sum_{i=1}^{N} 0.5 i x_i\Big)^4.$$

表 1　测试函数的维数、搜索空间和最优值

函数	维数	搜索空间	最优解	最优值
f_1	30	[−100, 100]	(0, 0, ⋯, 0)	0
f_2	30	[−100, 100]	(0, 0, ⋯, 0)	0
f_3	30	[−600, 600]	(0, 0, ⋯, 0)	0
f_4	30	[−32, 32]	(0, 0, ⋯, 0)	0
f_5	30	[−100, 100]	(0, 0, ⋯, 0)	0
f_6	30	[−512, 512]	(420.96, ⋯, 420.968 7)	−12 596
f_7	30	[−100, 100]	(0, 0, ⋯, 0)	0
f_8	30	[−100, 100]	(0, 0, ⋯, 0)	0

8个函数中, f_1、f_2 和 f_8 为单模态函数, f_3、f_4、f_5、f_6 和 f_7 为多模态函数, 且 f_6 具有较多的局部极值点, 难以搜索到全局最优值, f_8 混合了维数和变量的乘积, 尤其在维数增加时, 求解的难度会迅速增加, 难以获得较优的解.

3.2 参数HMS、HMCR和PAR对GCHS算法优化性能的影响

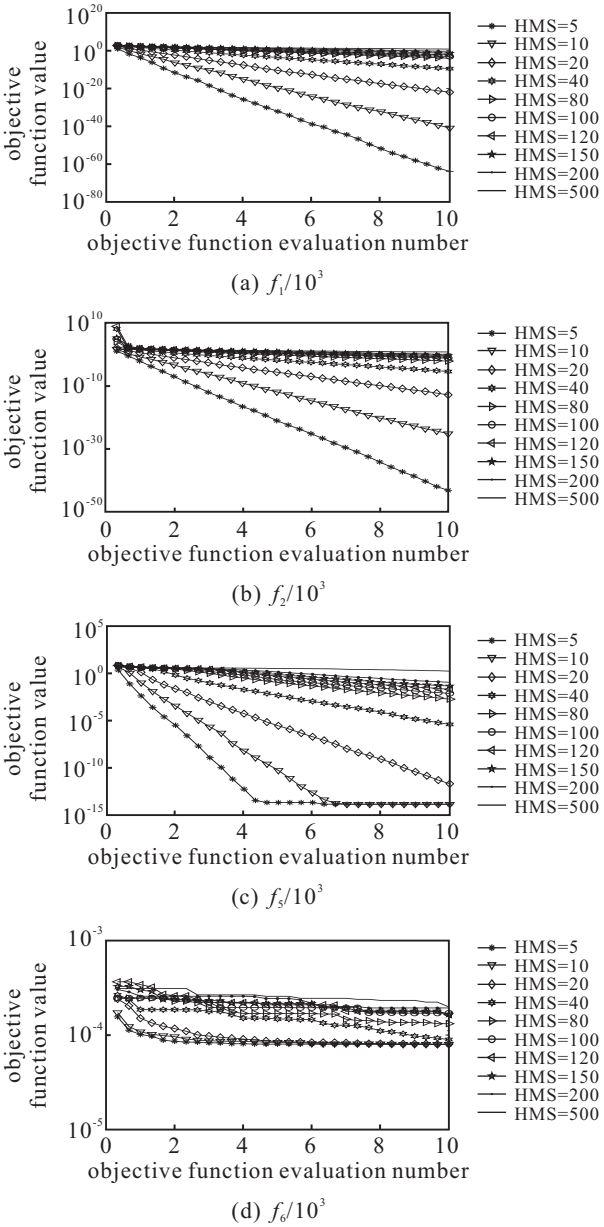
算法的优化效果在一定程度上会受到参数的影响, 算法对不同参数的敏感性是不同的, 因此研究设置合适的参数是算法研究必不可少的一部分. 针对GCHS算法, 研究和声记忆库大小(HMS)、和声记忆

库考虑概率(HMCR)和基音调整概率(PAR)等参数的设置. 以平均最优值作为指标, 利用单模态函数 f_1、f_2 和多模态函数 f_5、f_6 测试HMS、HMCR和PAR变化对GCHS算法性能的影响. 参数的影响测试采用控制变量法, 固定其他参数值, 讨论所研究的参数对算法的影响. 讨论HMS对GCHS算法优化性能的影响时, HMCR = 0.99, PAR = 0.5, HMS从5变化到500, 函数维数均取30, 目标函数评价次数为10 000, 采用Matlab 7.1实验仿真软件独立运行算法程序30次, 结果如表2所示. 给出了各个函数在不同HMS值条件下的优化曲线, 如图1所示.

表 2 HMS变化对GCHS算法的影响

HMS	f_1	f_2	f_5	f_6
5	7.69e-65	1.97e-42	1.07e-14	−1.26e+04
10	2.63e-41	7.85e-26	1.07e-14	−1.26e+04
20	1.47e-21	1.70e-13	1.77e-11	−1.25e+04
40	2.37e-10	3.77e-06	3.46e-06	−1.11e+04
80	8.18e-05	7.25e-03	2.14e-03	−7.13e+03
100	8.81e-04	5.22e-02	6.70e-03	−5.45e+03
200	3.38e-01	1.19e+00	1.16e-01	−5.19e+03
500	7.92e+00	7.12e+00	1.72e+00	−4.86e+03

由表2可见, 随着HMS值的增大, 算法所得到的值逐渐变差, 表明HMS的值不宜过大, 也不能太小, 建议在5～40之间取值. 由图1可见, 当HMS = 5时, GCHS算法的优化效果最好. 在前期具有快速的搜索效率, 到了后期能够有效地收敛, 因此, 本文建议HMS取5. 为测试HMCR和PAR对GCHS算法优化性能的影响, 选用 f_1 和 f_5, HMS = 5, 维数取30, 目标函数评价次数为10 000, 独立运行算法程序30次, 优化结果如表3和表4所示. 在测试HMCR和PAR值对GCHS算法影响时, 分别采用8组不同的HMCR值和PAR值进行组合, 共进行64组仿真. 当PAR值确定时, 在0.01～0.99之间, 随着HMCR值的变大, 优化的值越来越小, 越来越好, 尤其在HMCR取0.99时, GCHS算法所得到的平均最优值均较好, 这与文献[1,6]所建议的HMCR值应大于0.9相符合, 因此本文HMCR的取值为0.99. 当HMCR值固定时, PAR的取值在0.5附近所得到的值相对而言较好. 从整体看, PAR = 0.5, HMCR = 0.99是较好的参数选择, 但对于不同的问题有待进一步考虑适当的调整.



(a) $f_1/10^3$ (b) $f_2/10^3$ (c) $f_5/10^3$ (d) $f_6/10^3$

图 1 HMS值对GCHS算法的影响

表 3 HMCR和PAR对GCHS算法的影响(f_1)

PAR	HMCR							
	0.01	0.1	0.25	0.5	0.75	0.9	0.95	0.99
0.05	4.09e+04	3.60e+04	2.84e+04	1.32e+04	2.78e+03	2.91e+01	1.68e-02	6.54e-20
0.1	4.21e+04	3.52e+04	2.58e+04	1.33e+04	1.68e+03	2.25e+01	1.21e-02	5.79e-31
0.25	4.17e+04	3.45e+04	2.55e+04	1.31e+04	2.29e+03	1.32e+01	3.62e-05	1.36e-49
0.4	4.20e+04	3.58e+04	2.87e+04	1.20e+04	2.25e+03	1.27e+01	4.41e-06	3.26e-60
0.5	4.04e+04	3.91e+04	2.86e+04	1.41e+04	3.22e+03	5.07e+00	2.61e-05	6.31e-63
0.75	4.22e+04	3.62e+04	3.06e+04	1.45e+04	2.23e+03	1.51e+01	9.70e-05	5.13e-65
0.9	3.85e+04	3.66e+04	2.62e+04	1.64e+04	2.54e+03	2.13e+01	4.74e-04	6.47e-58
0.99	4.46e+04	4.12e+04	3.15e+04	1.42e+04	2.68e+03	1.09e+01	1.08e-03	5.62e-54

表 4　HMCR 和 PAR 对 GCHS 算法的影响 (f_5)

PAR	HMCR							
	0.01	0.1	0.25	0.5	0.75	0.9	0.95	0.99
0.05	1.96e+01	1.95e+01	1.87e+01	1.68e+01	9.62e+00	2.72e+00	2.13e-02	2.17e-11
0.1	1.97e+01	1.95e+01	1.87e+01	1.67e+01	9.74e+00	2.71e+00	3.77e-03	3.55e-14
0.25	1.96e+01	1.94e+01	1.91e+01	1.68e+01	1.03e+01	1.75e+00	1.15e-03	1.42e-14
0.4	1.95e+01	1.93e+01	1.88e+01	1.64e+01	9.72e+00	1.63e+00	5.82e-04	1.42e-14
0.5	1.96e+01	1.95e+01	1.88e+01	1.65e+01	1.02e+01	1.71e+00	1.75e-03	5.33e-15
0.75	1.98e+01	1.95e+01	1.92e+01	1.67e+01	8.97e+00	2.36e+00	2.75e-03	7.11e-15
0.9	1.98e+01	1.93e+01	1.90e+01	1.68e+01	1.05e+01	2.38e+00	5.70e-03	7.11e-15
0.99	1.97e+01	1.97e+01	1.91e+01	1.70e+01	1.15e+01	2.49e+00	6.54e-03	7.11e-15

表 5　9 种和声搜索算法所获得的 $f_1 \sim f_8\,(N=30)$ 的优化结果

函 数		算 法								
		HS	IHS	GHS	IGHS	GDHS	NGHS	NDHS	EHS	GCHS
f_1	mean	1.82e+00	4.04e-11	3.67e-03	2.64e-11	3.86e-04	3.25e-19	**0.00e+00**	2.67e-01	**0.00e+00**
	SD	5.66e-01	9.91e-12	3.93e-03	2.45e-12	6.52e-05	2.78e-19	**0.00e+00**	2.20e-03	**0.00e+00**
f_2	mean	6.08e+00	2.76e-01	1.83e-01	3.15e+00	7.20e-02	8.90e-10	5.03e-143	2.06e+00	**2.21e-211**
	SD	1.34e+00	3.88e-01	9.11e-02	2.26e+00	1.02e-02	6.71e-10	6.05e-144	2.69e-01	**0.00e+00**
f_3	mean	3.75e+00	3.38e+00	7.24e-03	6.95e-01	1.49e-03	**0.00e+00**	3.25e-01	7.85e-01	**0.00e+00**
	SD	3.83e-01	9.00e-01	1.39e-02	6.47e-01	3.57e-05	**0.00e+00**	6.51e-01	5.18e-01	**0.00e+00**
f_4	mean	1.01e+00	7.41e-03	6.14e-04	1.60e-02	**3.91e-04**	1.85e-02	3.80e-02	5.14e-01	4.92e-03
	SD	3.56e-02	5.32e-03	9.52e-04	7.63e-03	**1.94e-04**	5.10e-03	2.80e-02	4.92e-02	9.85e-03
f_5	mean	9.24e-01	4.88e-06	9.44e-03	2.29e-01	5.44e-03	1.36e-10	1.51e-02	2.02e-01	**7.11e-15**
	SD	2.81e-01	2.47e-07	1.35e-02	1.59e-01	9.80e-05	6.30e-11	3.02e-02	3.08e-02	**0.00e+00**
f_6	mean	−1.26e+04	−1.26e+04	−1.26e+04	**−1.26e+04**	−1.26e+04	−1.26e+04	−1.21e+04	−1.26e+04	**−1.26e+04**
	SD	1.43e+00	4.46e-12	6.76e-03	**1.05e-12**	6.07e-06	2.35e-12	6.55e-01	2.15e-01	2.50e-03
f_7	mean	2.66e+01	1.83e+01	8.74e+00	2.59e+01	2.57e+00	2.05e-02	**0.00e+00**	1.94e+01	**0.00e+00**
	SD	5.17e+00	3.59e+00	4.97e+00	8.79e+00	1.71e-01	2.35e-02	**0.00e+00**	1.91e+00	**0.00e+00**
f_8	mean	5.70e+03	6.30e+03	6.57e+02	1.05e-10	1.39e-03	6.55e+03	5.55e-01	4.28e+04	**2.86e-25**
	SD	1.47e+03	5.06e+02	7.72e+02	2.43e-11	3.62e-04	3.09e+03	3.22e-01	1.88e+03	**5.40e-25**

3.3　不同 HS 算法的比较

为了显示 GCHS 算法的有效性, 将 GCHS 算法与 HS 算法[1]及其改进算法 (如 IHS 算法[6]、GHS 算法[7]、IGHS 算法[8]、GDHS 算法[9]、NGHS 算法[10]、NDHS 算法[11]和 EHS 算法[12]) 进行比较, 各算法参数均采用文献中给定的设置. 在 GCHS 算法中, HMS = 5, HMCR = 0.99, PAR = 0.5. 最大目标函数评价次数为 105 次, 所得结果如表 5 所示. 其中: mean 表示平均最优值, SD 表示标准差. 进行 t-test 检验实验, 检验水平在 0.005, 测试本文算法与其他算法的显著性水平, 结果如表 6 所示. 其中: + 表示 t-test 测试结果是显著的, 表明本文算法所得出的结果有明显提高; - 表示测试结果不显著, 表明本文算法没有其他算法好; NA 表示差别不大, 表明本文算法与其他算法所得结果差不多.

表 6　9 种 HS 算法所获得的 $f_1 \sim f_8\,(N=30)$ 的优化结果

算 法	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
GCHS/HS	+	+	+	+	+	+	+	+
GCHS/IHS	+	+	+	NA	+	NA	+	+
GCHS/GHS	+	+	+	NA	+	NA	+	+
GCHS/IGHS	+	+	+	NA	+	NA	+	+
GCHS/GDHS	+	+	+	NA	+	NA	+	+
GCHS/NGHS	+	+	+	NA	+	NA	+	+
GCHS/NDHS	NA	+	+	+	+	NA	NA	+
GCHS/EHS	+	+	+	+	+	NA	+	+

由表 5 可知: HS 算法优化出的结果均不理想; GHS 算法对 f_1 和 f_4 寻优能够达到精度 10^{-2} 以上, 但对于其他函数无法寻找到较优的值, 表明单纯应用全局最优解并不能有效地指导算法向较好的解空间进行搜索; IHS 算法对参数 BW 进行动态

调整, 有效提高了HS算法的优化性能, 能够搜索到f_1、f_4、f_5和f_6较好的值, 但对于余下的4个函数, 算法几乎陷入了局部最优, 表明单纯地动态调整参数没有从根本上改善算法的搜索能力; NGHS算法能够有效地寻找到f_1、f_2、f_3、f_5、f_6的较优解, 尤其f_3所搜索到结果是全局最优值, 但NGHS算法无法优化出函数f_4、f_7和f_8较优的解; NDHS算法找到了f_1和f_7的全局最优解, 并且能优化出f_2较好的结果, 但搜索余下的函数时, NDHS算法仍难以搜索到接近最优解的解; IGHS算法是改进的和声搜索算法, 它利用高斯分布和均匀分布对随机和声和全局和声进行扰动和探索, 能快速搜索到f_1、f_6和f_8的近似最优值, 但优化其他函数得到的结果并不理想; GDHS算法是一种改进HS算法, 采用搜索邻域动态变化形式, 加快搜索的有效性, 整体结果有了较大的进步, 但优化精度并不高; 本文所提出的GCHS算法优化出的结果除了f_4外, 其他优化结果均优于其他算法或相差不多, 尤其本文算法能够寻找到f_1、f_2、f_3和f_7的全局最优值或靠近全局最优值的值. 对于f_4, 本文算法的搜索精度也能达到10^{-3}. 综上所述, 就优化精度和稳定性而言, 本文算法比其他8种HS算法搜索效果更好, 且由表6可见, GCHS算法寻找到的结果显著优于其他8种和声搜索算法, 进一步验证了所提出算法的有效性, 表明算法具有良好的优化潜力.

4 结 论

本文提出了一种全局竞争和声搜索算法, 主要有3个方面的改进: 1) 引入了随机局部平均和声和全局平均; 2) 建立了竞争搜索机制, 实现优胜劣汰的理念; 3) 设计了自适应全局调整和局部学习策略, 平衡算法的局部搜索和全局搜索, 并分析了参数对算法的影响, 测试了算法的有效性. 数值结果表明, GCHS算法在精度、收敛速度和鲁棒性方面优于和声搜索算法和文献中提出的优秀改进和声搜索算法.

参考文献(References)

[1] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: Harmony search[J]. Simulation, 2001, 76(2): 60-68.

[2] Sivasubramani S, Swarup K S. Environmental economic dispatch using multi-objective harmony search algorithm[J]. Electrical Power & Energy Systems, 2011, 81(9): 1778-1785.

[3] Sharma K D, Chatterjee A, Rakshit A. Design of a hybrid stable adaptive fuzzy controller employing lyapunov theory and harmony search algorithm[J]. IEEE Trans on Control Systems Technology, 2010(99): 1-8.

[4] Ayvaz M T. Application of Harmony Search algorithm to the solution of groundwater management models[J]. Advances in Water Resources, 2009, 32(6): 916-924.

[5] Ravikumar V, Pandi Bijaya, Ketan Panigrahi. Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm[J]. Expert Systems with Applications, 2011, 38(7): 8509-8514.

[6] Mahdavi M , Fesanghary M, Damangir E. An improve harmony search algorithm for solving optimization problems[J]. Applied Mathematics and Computation, 2007, 188(2): 1567-1579.

[7] Omran M G H, Mahdavi M. Global-best harmony search[J]. Applied Mathematics and Computation, 2008, 198(2): 643-656.

[8] Mohammed E A. An improved global-best harmony search algorithm[J]. Applied Mathematics and Computation, 2013, 222(1): 94-106.

[9] Khalili M, Kharrat R, Salahshoor K, et al. Global dynamic harmony search algorithm: GDHS[J]. Applied Mathematics and Computation, 2014, 228(1): 195-219.

[10] 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.

[11] Chen J , Pan Q K, Li J Q. Harmony search algorithm with dynamic control parameters[J]. Applied Mathematics and Computation, 2012, 219(2): 592-604.

[12] Das S, Mukhopadhyay A, Roy A, et al. Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2011, 41(1): 89-106.

[13] Zou D X, Gao L Q, Li S, et al. A novel global harmony search algorithm for task assignment problem[J]. J of Systems and Software, 2010, 83(10): 1678-1688.

[14] Zou D X, Gao L Q, Li S, et al. Solving 0-1 knapsack problem by a novel global harmony search algorithm[J]. Applied Soft Computing, 2011, 11(2): 1556-1564.

[15] Zou D X, Gao L Q, Wu J H, et al. A novel global harmony search algorithm for reliability problems[J]. Computers & Industrial Engineering, 2010, 58(2): 307-316.

(责任编辑：郑晓蕾)