

一种自适应全局和声搜索算法

翟军昌^{1,2}, 高立群¹, 欧阳海滨¹, 孔祥勇¹

(1. 东北大学 信息科学与工程学院, 沈阳 110004; 2. 渤海大学 基础教研部, 辽宁 锦州 121013)

摘要: 针对新颖全局和声搜索(NGHS)算法过早收敛的问题, 提出自适应全局和声搜索(AGHS)算法. 引入差分向量范数定义和声记忆库多样性, 给出新的位置更新策略, 排除变异操作. 以和声记忆库多样性信息为指导动态产生新和声, 提高算法对解空间信息开发的能力, 避免算法因过早收敛、易陷入局部最优的不足. AGHS算法操作更简单, 需要设置的参数更少, 将其与目前文献中较优的几种改进HS算法、PSO算法和GA算法进行性能测试, 测试结果表明AGHS算法具有较高的寻优精度和较快的收敛速度.

关键词: 和声搜索; 变异; 范数; 多样性; 自适应

中图分类号: TP273

文献标志码: A

An adaptive global harmony search algorithm

ZHAI Jun-chang^{1,2}, GAO Li-qun¹, OUYANG Hai-bin¹, KONG Xiang-yong¹

(1. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. Department of Basic Teaching and Research, Bohai University, Jinzhou 121013, China. Correspondent: ZHAI Jun-chang, E-mail: zhajunchang@163.com)

Abstract: An adaptive global harmony search(AGHS) algorithm is proposed for the problem of premature convergence in novel global harmony search(NGHS) algorithm. A novel position updating strategy is employed in the AGHS algorithm and the mutation operation is excluded with the diversity of harmony memory which is characterized by the norm of differential vector. With the guidance of the diversity of harmony memory, the optimal performance of the algorithm is improved, which avoids the algorithm trapped in local optimal. In addition, the AGHS algorithm is simpler and less parameter needed. The simulation is carried out by using the optimization algorithm of AGHS, PSO, GA and other HS variants that have been proposed. The simulation results show that the AGHS algorithm has higher convergence precision and convergence rate.

Keywords: harmony search; mutation; norm; diversity; adaptive

0 引言

和声搜索(HS)算法是由Geem等^[1]于2001年提出的一种启发式优化算法, 由于其操作简单, HS及其改进算法已经成功地应用于解决工程中的非线性优化问题^[2-6]. 虽然HS算法具有优秀的优化性能, 但其步长调整策略在进化后期盲目搜索, 不能有效地调整解的结构, 使和声记忆库的多样性逐渐消散, 容易陷入局部最优.

为了克服HS算法存在的不足, 部分学者提出采用指数递减、多项式或者其他各种非线性形式的动态参数调整策略, 以提高和声记忆库的多样性, 如改进和声搜索(IHS)算法^[7]、新颖动态和声搜索(NDHS)算法^[8]和动态自适应和声(DSHS)搜索算法^[9]等. 另有

学者提出在HS算法中引入粒子群、差分进化和人工蜂群等优化算法的策略, 或者将这些算法与HS算法进行融合, 如全局最好和声搜索(GHS)算法^[10]、动态多种群粒子群和声搜索(DMS-PSO-HS)算法^[11]和混合和声人工蜂群(HHSABC)算法^[12]等. 其中GHS算法修正了HS算法新和声产生的机理, 通过和声记忆库中最好和声信息产生新和声. 沿着这一研究思路, Pan等^[13]提出了自适应全局最好和声搜索(SGHS)算法, Zou等^[3-4,6,14]提出了新颖全局和声搜索(NGHS)算法, 并用于解决一些带有约束的非线性优化问题. 针对NGHS算法, Valian等^[15]提出了智能全局和声搜索(IGHS-V)算法, 提高了NGHS算法的优化性能. 最近, Xiang等^[16]在GHS算法的基础上提出了改进的全局

收稿日期: 2014-09-04; **修回日期:** 2014-11-11.

基金项目: 国家自然科学基金项目(61273155, 61104106).

作者简介: 翟军昌(1978—), 男, 博士生, 从事智能优化的研究; 高立群(1949—), 男, 教授, 博士生导师, 从事智能优化与图像处理等研究.

最好和声搜索 (IGHS_X) 算法.

虽然不同的设计思想和操作策略引入到和声搜索算法中,使其性能得到了不断的改进,在一定程度上提高了算法的收敛精度和收敛速度.但HS及其改进算法仍然存在收敛速度慢、易陷入局部最优等问题,而且在目前改进的HS算法中参数设置过多,需要通过大量的仿真实验或者经验设置调整,降低了算法在实际中的适用性.

为了进一步提高HS算法的优化性能,本文提出了一种自适应全局和声搜索 (AGHS) 算法.该算法在NGHS算法的基础上,排除了变异操作,给出了新的位置更新策略.通过引入和声记忆库多样性作为指导信息,自适应产生新和声,提高了算法全局搜索和局部搜索的能力,避免了算法易陷入局部最优的不足,而且新算法的参数变得更少,提高了算法的适用性.最后通过仿真实验验证了所提出算法的有效性.

1 HS和NGHS算法

1.1 HS算法

HS算法的寻优思想源于模拟音乐家的即兴创作过程,通过经验搜索和有效的探索得到全局最优解或接近全局最优解.算法首先产生HMS个初始解,并放入和声记忆库中;然后对解的各个分量分别以概率HMCR在和声记忆库内进行搜索,以 $1-HMCR$ 的概率在和声记忆库外搜索,期望获得新解的对应分量,在和声记忆库内进行搜索时,当随机搜索到某一分量后,对该分量以概率PAR进行扰动;最后由搜索后得到的各个分量构成新解,若新解优于和声记忆库中的最差解,则用新解替换和声记忆库中最差解.如此循环,直到满足终止条件为止.

HS算法的参数包括:和声记忆库大小(HMS)、和声记忆库考虑概率(HMCR)、基音调整步长(bw)、基音调整概率(PAR)和迭代次数(J).HS算法操作步骤包括:1)初始化优化问题和算法的参数;2)初始化和声记忆库;3)即兴创作产生一个新和声;4)更新和声记忆库;5)判断终止准则.关于HS算法的详细操作步骤见文献[1].

1.2 NGHS算法

NGHS算法排除了HS算法中HMCR、PAR和bw三个参数,与HS算法的不同主要有两点:1)通过引入位置更新和变异操作即兴创作产生新和声;2)和声记忆库的更新方式.

1.2.1 位置更新

在NGHS算法中,利用最差和声向量分量关于最优和声分量的对称区间进行位置更新,具体操作为

$$x_R = 2x_j^{\text{best}} - x_j^{\text{worst}}, \quad (1)$$

$$x'_j = x_j^{\text{worst}} + \text{rand}() \times (x_R - x_j^{\text{worst}}). \quad (2)$$

其中: x_R 为 x_j^{worst} 关于 x_j^{best} 的对称点, x_j^{best} 和 x_j^{worst} 分别为最优和声和最差和声向量的第 j 维分量, x'_j 为新生成和声向量的第 j 维分量.由式(2)可知, x'_j 产生在 x_j^{worst} 关于 x_j^{best} 对称的某个区域内.

1.2.2 变异

NGHS算法位置更新操作容易陷入局部最优,因此在小于变异概率 P_m 下引入了变异操作,即

$$x'_j = x_j^L + \text{rand}() \times (x_j^U - x_j^L), \quad (3)$$

其中 x_j^L 和 x_j^U 为第 j 维和声分量的下界和上界.

1.2.3 更新和声记忆库

在NGHS中,和声记忆库的更新方式与基本HS算法不同,新产生的和声直接替换和声记忆库中的最差和声,即使和声记忆库中的最差和声优于新产生的和声也将被替换.

在NGHS算法中,新和声向量的分量在最差和声分量与最优和声分量的对称区间内随机产生.位置更新操作虽然能提高算法的收敛性,但算法中新和声直接更新最差和声,导致和声记忆库中的和声快速向当前最优和声靠近,容易使算法陷入局部最优,出现算法早熟现象.虽然通过小概率的方式引入变异操作增加了和声记忆库的多样性,但是并未考虑到和声记忆库的多样性变化情况,不利于算法对局部信息的开发,从而容易漏掉一些对算法有指导作用的和声信息.

2 自适应全局和声搜索 (AGHS) 算法

本文提出的自适应全局和声搜索 (AGHS) 算法,通过引入和声记忆库多样性给出了新的位置更新策略.以和声记忆库的多样性信息为指导,利用自适应因子实现位置动态自适应更新,排除了NGHS算法中的变异操作即兴创作产生新和声.

2.1 和声记忆库多样性

在和声记忆库中,最优和声向量与最差和声向量的差异越小,表明和声记忆库的和声彼此越接近,和声记忆库的多样性越差,反之则和声记忆库的多样性越好.为了刻画和声记忆库的多样性,引入向量的范数定义和声记忆库的多样性 D_i ,即

$$D_i = \|HM_{\text{best}} - HM_{\text{worst}}\|. \quad (4)$$

其中: D_i 为第 i 代和声记忆库的多样性, HM_{best} 和 HM_{worst} 分别为第 i 代和声记忆库中最优和声和最差和声向量, $\|\cdot\|$ 为向量的1范数.

若 D_i 取值越大,则表明当前和声记忆库中最优和声与最差和声差异越大,即和声记忆库中和声的多样性越好.若 D_i 的取值越小,则表明当前和声记忆库最优和声与最差和声差异越小,即多样性较差.在算

法迭代早期, 和声记忆库中最优和声与最差和声的差异较大, 因此 D_i 也相对较大. 随着算法迭代次数的增加, 和声记忆库中最优和声与最差和声的差异较小, 需要对最优和声的邻域信息进行精细的搜索, 以避免算法陷入局部最优.

2.2 自适应因子

根据和声记忆库中和声的多样性信息, 定义自适应因子 (AF), 即

$$AF = \begin{cases} 1, & D_i = 0 \text{ or } D_i/D_{i-1} \geq 1; \\ D_i/D_{i-1}, & D_i/D_{i-1} \leq 1. \end{cases} \quad (5)$$

其中: D_i 和 D_{i-1} 分别为第 i 和 $i-1$ 代和声记忆库的多样性, $D_0 = 1$.

自适应因子反映了当前和声记忆库的和声多样性的变化情况. 若 AF 越小, 则表明和声记忆库的多样性越差, 需要对最优和声的邻域信息进行精细的搜索, 反之需要提高算法的全局搜索性能.

2.3 即兴创作产生新和声

在 AGHS 算法中, 新和声通过新的位置更新策略即兴创作产生, 将 NGHS 算法中变异操作排除. 新位置更新策略的关键是 x_j^{worst} 关于 x_j^{best} 的对称点 x_R 的选取, 该操作操作由两部分构成, 一是保留 NGHS 算法中的操作策略; 二是根据和声记忆库的多样性作为指导, 利用自适应因子 AF 进行动态调整选取对称点. 对于两种不同的对称点选择策略, 通过引入位置更新概率 P_R 对其进行决策, 从而实现不同的位置更新策略. 下面对位置更新概率和位置更新策略作具体介绍.

2.3.1 位置更新概率

位置更新概率 P_R 定义如下:

$$P_R = (1/N)^{(i/J)}. \quad (6)$$

其中: N 为解空间维数, i 为当前迭代次数, J 为进化代数. 由式 (6) 的定义可知, 位置更新概率 P_R 在算法迭代初期取值较大, 随着算法迭代次数不断增加, 取值逐渐变小.

2.3.2 位置更新策略

在 AGHS 算法中, x_j^{worst} 关于 x_j^{best} 的对称点 x_R 按如下操作策略执行: 1) 以概率 P_R 保留原算法中对称点 x_R 的选取操作, 即式 (1); 2) 以概率 $1 - P_R$ 执行新的选取对称点 x_R 操作, 即

$$x_R = x_j^{\text{best}} - AF \times (x_j^{\text{best}} + x_j^{\text{worst}}), \quad (7)$$

其中 AF 为自适应因子.

由于最差和声向量总是向最优和声向量靠近, 在优化后期大部分和声向量差异较小, 此时和声记忆库的多样性较差, 容易使算法陷入局部最优. 在 $1 - P_R$ 的概率下引入式 (7) 的操作, 由式 (5) 和 (7) 可知, 算法

可以根据和声记忆库的多样性变化情况, 实现对称点自适应动态调整.

在算法中引入新的对称点选取策略后, 根据式 (2) 的计算可知, 新和声采用两种动态步长的位置更新策略, 使新和声以不同的策略动态产生. 在优化早期, 参数 P_R 的取值较大, 可以保证算法以较大的概率进行全局搜索, 使算法保持 NGHS 算法快速收敛的特性. 在优化后期参数 P_R 的取值逐渐变小, 使算法以较大的概率对局部信息进行更精细的搜索, 提高算法跳出局部最优的能力. 此外, 在新的位置更新策略中, 参数 P_R 随着算法的迭代不断变化, 不需要人为指定或者根据经验设置.

2.3.3 排除变异操作

在 NGHS 算法中通过小概率的方式引入变异操作, 其目的是保持和声记忆库的多样性, 但该操作没有和声记忆库多样性信息作指导, 不能反映当前和声记忆库多样性变化情况.

本文算法将 NGHS 算法中的变异操作排除, 在位置更新过程中以和声记忆库的多样性变化情况作为指导产生新和声, 不仅使算法具有更好的适应性, 而且减少了算法中经验参数设置的数量.

2.4 算法操作步骤

AGHS 算法的操作步骤如下.

Step 1: 初始化优化问题和算法的参数, 设置和声记忆库大小 HMS 和最大迭代次数 J .

Step 2: 初始化和声记忆库. 确定第 j 个分量的范围 $[x_j^L, x_j^U]$, 随机产生 HMS 个和声向量存入和声库 HM 中.

Step 3: 即兴创作产生一个新和声

for each dimension $j \in [1, N]$ do

if $\text{rand}() \leq P_R$

$$x_R = 2x_j^{\text{best}} - x_j^{\text{worst}}$$

else

$$x_R = x_j^{\text{best}} - AF \times (x_j^{\text{best}} + x_j^{\text{worst}})$$

end if

if $x_R > x_j^U$

$$x_R = x_j^U$$

else if $x_R < x_j^L$

$$x_R = x_j^L$$

end

$$x'_j = x_j^{\text{worst}} + \text{rand}() \times (x_R - x_j^{\text{worst}})$$

end for

Step 4: 更新和声记忆库. 新和声直接更新和声记忆库中的最差和声.

Step 5: 判断终止准则. 如果当前迭代次数等于最

大迭代次数 J , 则终止运行 AGHS 算法, 否则重复执行 Step 3 和 Step 4.

3 算法分析

HS 算法与其他启发式优化算法一样, 算法的参数设置和复杂的操作对算法的性能影较大. 下面对 SGHS^[13]、NGHS^[14]、IGHS_V^[15]和 IGHS_X^[16]四种算法与 HS 算法^[1]和 AGHS 算法的参数、复杂度作简单分析, 并对 AGHS 算法的收敛性作简单分析.

3.1 算法参数分析

HS 算法和其他 5 种改进的 HS 算法中, 需要通过经验设置的主要参数数量分别为: HS 算法 5 个, SGHS 算法 6 个, NGHS 算法 3 个, IGHS_V 算法 4 个, IGHS_X 算法 9 个. 这些参数需要通过大量的仿真实验进行调整, 或者通过经验进行设置. 本文提出的 AGHS 算法在排除了 NGHS 算法的变异操作后, 变异概率 P_m 也随之被排除掉. 虽然 AGHS 算法增加了自适应因子 AF 和位置更新概率 P_R 两个参数, 但这两个参数分别依据和声记忆库多样性和迭代次数自动变化, 不需要经验设置. 本文算法实际需要主观设置的参数只有 2 个, 即 HMS 和 J .

事实上, AGHS 算法所需人为设置参数数量在现有所有改进 HS 算法中是最少的, 因此算法受主观因素的影响也相对最小, 具有更好的适用性和稳定性, 更适合工程实际应用.

3.2 算法复杂度分析

算法复杂度包括空间复杂度和时间复杂度, 下面就算法的空间复杂度和时间复杂度作简单分析.

从算法的空间复杂度看, AGHS 算法除了要存储两个经验参数 HMS 和 J 外, 还要存储和声记忆库多样性 D_i 、 D_{i-1} , 自适应因子 AF 和位置更新概率 P_R 直接通过计算获得; AGHS 算法运行时产生的中间变量与 NGHS 算法相同. 虽然 AGHS 算法与 NGHS 算法相比要存储的参数多了一个参数和标量 D_i 、 D_{i-1} , 但是少了变异概率 P_m , 因此 AGHS 与 NGHS 算法的空间复杂度相差不大. 其他几种算法需要存储的变量比 AGHS 算法要多, 尤其是 IGHS_X 算法, 不仅需要存储的参数较多, 而且其和声记忆库所占用的存储空间是其他几种算法的 4 倍.

从算法的时间复杂度看, 几种算法的关键操作步骤中, 对和声记忆库初始化的计算量相同, 但在即兴创作产生新和声时不同. NGHS 算法只有位置更新和变异两个关键操作, 每个操作只有一步, NGHS 算法在常见的几种改进 HS 算法中计算量最少, IGHS_X 算法计算量最大. AGHS 算法位置更新操作只有一步, 与 NGHS 算法相同, 虽然增加了对和声记忆库多样

性的计算, 但排除了变异操作, 因此在时间复杂度上, AGHS 算法与 NGHS 算法基本相同.

3.3 AGHS 算法收敛性分析

与其他几种改进 HS 算法相比, AGHS 算法引入新的位置更新操作, 通过位置更新概率决策执行不同的位置更新策略, 使算法在迭代初期具有较好的全局搜索能力, 在迭代后期具有较强的局部搜索能力, 提高了算法跳出局部最优的能力. 而且两种不同的位置更新策略, 都是通过当前和声记忆库中最优和声向量作为引导动态产生新和声, 可以保证新和声以一种单调递增的方式产生. 因此, 新算法的更新策略实质上仍然是一种贪婪的选择策略, 引入新的位置更新策略后可以保证算法的收敛性.

4 仿真实验

4.1 仿真实验准备

为了验证 AGHS 算法的性能, 本文将其与 HS 算法^[1]、SGHS 算法^[13]、NGHS 算法^[14]、IGHS_V 算法^[15]、IGHS_X 算法^[16]、基本粒子群算法 PSO^[17]和遗传算法 GA^[18]进行优化性能测试. 在 Windows XP 下, 以 Matlab 7.6 为仿真实验环境. 实验中选取优化算法 10 个经典标准测试函数, 具体表达如下:

f_1 : Sphere Function

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

f_2 : Rosenbrock Function

$$\min f_2(x) = \sum_{i=1}^{N-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2];$$

f_3 : Rastrigin Function

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

f_4 : Griewank Function

$$\min f_4(x) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1;$$

f_5 : Ackley Function

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

f_6 : Schwefel's problem 2.2.2 Function

$$\min f_6(x) = \sum_{i=1}^N |x_i| + \prod_{i=1}^N |x_i|;$$

f_7 : Roated hyper-ellipsoid Function

$$\min f_7(x) = \sum_{i=1}^N \left(\sum_{j=1}^N x_j \right)^2;$$

f_8 : Schwefel's problem 1.2 with noise

$$\min f_8(x) = \sum_{i=1}^N \left(\sum_{j=1}^i x_j \right)^2 (1 + 0.4|N(0, 1)|) - 450;$$

f_9 : Zakharov Function

$$\min f_9(x) = \sum_{i=1}^N x_i^2 + \left(\sum_{i=1}^N 0.5ix_i \right)^2 + \left(\sum_{i=1}^N 0.5ix_i \right)^4;$$

f_{10} : Powell Function

$$\min f_{10}(x) = \sum_{i=1}^{N/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4.$$

10 个测试函数的维数、搜索空间和最优值如表 1 所示。

表 1 函数维数、搜索空间和最优值

函数	维数 ²	搜索空间	最优值
f_1	100	$[-100, 100]^n$	0
f_2	100	$[-100, 100]^n$	0
f_3	100	$[-10, 10]^n$	0
f_4	100	$[-600, 600]^n$	0
f_5	100	$[-32, 32]^n$	0
f_6	100	$[-100, 100]^n$	0
f_7	100	$[-100, 100]^n$	0
f_8	100	$[-100, 100]^n$	-450
f_9	100	$[-5, 10]^n$	0
f_{10}	48	$[-4, 5]^n$	0

4.2 仿真结果与分析

本文 AGHS 算法是对文献 [14] 中 NGHS 算法的改进, 对于参数 HMS 的设置文献 [14] 已经作了分析, 为了保证算法对比的公平性, 仿真实验中, AGHS 算法取 HMS = 5, 与文献 [14] 设置相同. 其他几种算法的参数均选择参考文献中的最优设置. 具体设置为: HS 算法, HMS = 5, HMCR = 0.95, PAR = 0.33, bw = 0.01; SGHS 算法, HMS = 5, HMCR_m = 0.98, PAR_m = 0.9, bw_{min} = 0.0005, bw_{max} = $(x_i^U - x_i^L)/10$, LP = 0; NGHS 算法, HMS = 5, P_m = 0.005; IGHS_V 算法, HMS = 5, HMCR = 0.9950, PAR = 0.4; IGHS_X 算法, HMS = 20, HMCR_{min} = 0.9, HMCR_{max} = 0.99, PAR_{min} = 0.9, PAR_{max} = 0.95, $\xi = 0.1$, $\alpha = 0.25$, $\beta = 0.05$; PSO 算法, 学习因子 C₁ 和 C₂ 分别取 2, 种群大小取 40; GA 算法, 杂交概率 P_c = 0.9, 变异概率 P_m = 0.04, 种群大小取 40.

由于 PSO 和 GA 算法对目标函数的计算次数不仅与迭代次数有关, 而且与种群的大小有关, 这一点与 HS 算法的操作机理不同. 为了保证算法对比的公平, 实验中将每种算法对目标函数的最大评价次数作为迭代终止条件. 其中 HS 及改进 HS 算法迭代 80 000 次, 与文献 [14-15] 相同, PSO 和 GA 算法分别迭代 2 000 次, 所有算法对目标函数的评价次数均为

80 000 次. 每种算法独立运行 30 次, 分别用 B 代表最优值, W 代表最差值, M 代表平均值, S 代表方差, 对 10 个函数的测试结果如表 2 所示, 粗体字代表算法优化得到的最好值.

由表 2 对函数 f_1 、 f_3 、 f_4 和 f_5 的优化结果可见, GA 算法对 4 个函数的优化结果略优于 HS、SGHS、NGHS 和 PSO 算法, IGHS_V、IGHS_X 和 AGHS 三种改进 HS 算法的优化精度均得到了有效的提高. 其中 IGHS_V 算法对函数 f_3 和 f_4 优化可以得到最优解, IGHS_X 算法对函数 f_1 、 f_3 和 f_4 优化均可以得到最优解, 对函数 f_1 和 f_4 优化得到的最差值和平均值与最优解相同. 相比之下, AGHS 算法在高维空间中对函数 f_1 、 f_3 、 f_4 和 f_5 优化的结果明显优于其他几种算法, 不仅可以得到最优解, 而且得到的最差值和平均值均与最优解相同.

由表 2 对函数 f_2 和 f_6 的优化结果可见, IGHS_V 和 IGHS_X 算法的优化效果最好, 且 IGHS_X 算法对函数 f_6 优化得到的最优值、最差值、平均值和方差均为 0. PSO 算法的优化精度优于 HS 算法, 但与其他几种改进 HS 算的优化结果相比稍差. SGHS 算法和 NGHS 算法得到的最优值比较接近, 但 NGHS 算法得到的平均值和方差都优于 SGHS 算法. GA 算法的优化精度优于 SGHS 和 NGHS 算法的优化结果. AGHS 算法得到的结果虽稍差于 IGHS_V 算法和 IGHS_X 算法, 但对函数 f_6 优化得到的平均值数量级达到了 10^{-208} .

由表 2 对函数 f_7 和 f_8 的优化结果可见, 除了 AGHS 算法外, 其他几种算法的优化结果均与最优解的偏差较大. 其中 PSO 算法和 IGHS_X 算法得到的最优值略优于 HS、SGHS、NGHS、IGHS_V 和 GA 算法的优化结果. AGHS 算法对函数 f_7 和 f_8 的优化结果明显优于其他几种算法, 而且对函数 f_8 优化得到的最优值、最差值和平均值均与最优解相同.

实验中, 函数 f_9 和 f_{10} 的搜索区间是非对称的. 由表 2 对函数 f_9 和 f_{10} 优化的结果可见, 在非对称区间中寻优时, AGHS 算法得到的结果均优于其他几种算法的优化结果. 由此可知, AGHS 算法在非对称区间内寻优具有较好的优化性能. 总体而言, AGHS 算法除了对函数 f_2 寻优精度稍差外, 对其他几个函数优化得到的结果均得到了有效的提升. 此外, AGHS 算法对 10 个函数优化得到的方差都是最小的, 且在高维向量空间中, AGHS 算法与其他几种算法相比, 优化的精度变化相对较小, 比其他几种算法更稳定.

为了对比几种 HS 算法的收敛精度和收敛速度, 给出 6 种 HS 算法对函数 f_1 、 f_3 、 f_4 、 f_7 和 f_{10} 优化迭代 80 000 次时的最优值进化曲线, 如图 1 ~ 图 5 所示.

表 2 标准函数测试结果

函数	指标	HS	SGHS	NGHS	IGHS_V	IGHS_X	PSO	GA	AGHS
f_1	B	6.58e+02	6.22e-02	1.54e-03	2.08e-171	0	4.48e+00	1.94e-09	0
	W	1.27e+03	9.55e-01	6.41e-03	3.92e-155	0	9.09e+00	4.33e-09	0
	M	8.90e+02	3.39e-01	3.68e-03	1.43e-156	0	6.79e+00	2.94e-09	0
	S	1.40e+02	2.48e-01	1.28e-03	7.17e-156	0	9.80e-01	5.09e-10	0
f_2	B	2.54e+06	4.08e+02	2.30e+02	1.23e-28	1.23e-28	1.82e+03	9.48e+01	9.79e+01
	W	6.11e+06	1.06e+04	7.66e+03	9.68e+01	9.50e+01	5.50e+03	1.94e+02	9.88e+01
	M	4.23e+06	1.17e+03	1.06e+03	6.76e+01	7.59e+01	2.99e+03	1.36e+02	9.85e+01
	S	8.81e+05	2.15e+03	1.65e+03	4.50e+01	3.86e+01	9.45e+02	3.26e+01	3.47e-01
f_3	B	1.03e+02	3.04e+01	2.09e+00	0	0	1.26e+02	5.78e+01	0
	W	1.63e+02	1.56e+02	9.09e+00	3.05e+02	3.30e+02	2.05e+02	7.97e+01	0
	M	1.37e+02	5.21e+01	6.05e+00	4.89e+01	1.62e+01	1.65e+02	7.07e+01	0
	S	1.43e+01	2.50e+01	1.83e+00	8.96e+01	6.57e+01	2.20e+01	5.98e+00	0
f_4	B	7.84e+00	2.06e-01	7.33e-04	0	0	9.97e-01	1.19e-09	0
	W	1.25e+01	6.12e-01	8.18e-02	2.22e-16	0	1.09e+00	2.31e-09	0
	M	9.97e+00	3.85e-01	1.82e-02	7.40e-18	0	1.05e+00	1.65e-09	0
	S	1.16e+00	1.02e-01	2.13e-02	4.05e-17	0	2.07e-02	2.77e-10	0
f_5	B	4.36e+00	2.69e-04	5.93e-03	3.55e-15	7.11e-15	6.04e+00	7.31e-06	0
	W	5.59e+00	2.21e-01	1.16e-02	7.11e-15	1.42e-14	8.64e+00	1.43e-05	0
	M	5.11e+00	4.81e-02	8.92e-03	4.50e-15	7.82e-15	7.24e+00	9.63e-06	0
	S	2.66e-01	5.45e-02	1.28e-03	1.60e-15	2.17e-15	6.06e-01	1.54e-06	0
f_6	B	1.01e+02	2.73e+00	2.16e-01	2.48e-128	0	2.00e+02	9.01e-05	5.74e-211
	W	1.98e+02	6.48e+00	4.40e-01	3.75e-121	0	1.80e+03	1.27e-04	1.50e-207
	M	1.40e+02	4.32e+00	3.18e-01	2.66e-122	0	9.27e+02	1.04e-04	1.02e-208
	S	1.81e+01	8.42e-01	5.19e-02	7.49e-122	0	3.69e+02	1.08e-05	0
f_7	B	6.57e+04	1.94e+04	1.21e+04	2.83e+04	4.26e+03	2.36e+03	3.15e+05	0
	W	1.24e+05	3.90e+04	2.30e+04	2.49e+05	2.47e+05	9.18e+03	4.33e+05	1.51e-241
	M	9.80e+04	2.98e+04	1.53e+04	1.21e+05	1.20e+05	3.76e+03	3.67e+05	5.02e-243
	S	1.28e+04	5.47e+03	2.29e+03	5.78e+04	6.50e+04	1.20e+03	2.62e+04	0
f_8	B	1.30e+05	1.49e+05	1.14e+05	1.88e+04	3.95e+03	9.73e+03	3.30e+05	-450
	W	2.18e+05	2.73e+05	1.87e+05	3.34e+05	2.70e+05	3.84e+04	4.99e+05	-450
	M	1.66e+05	1.94e+05	1.44e+05	1.84e+05	1.49e+05	2.30e+04	4.33e+05	-450
	S	2.33e+04	3.14e+04	2.04e+04	7.49e+04	7.33e+04	6.45e+03	4.10e+04	0
f_9	B	3.41e+02	1.07e+02	1.35e+02	1.27e-01	3.06e-04	7.33e+02	1.38e+03	4.21e-247
	W	6.25e+02	4.05e+02	2.71e+02	1.44e+01	5.95e+00	3.44e+03	1.66e+03	1.94e-216
	M	4.57e+02	1.85e+02	1.92e+02	2.77e+00	7.00e-01	2.17e+03	1.53e+03	7.19e-218
	S	7.41e+01	6.31e+01	3.25e+01	3.76e+00	1.59e+00	6.45e+02	6.50e+01	0
f_{10}	B	2.03e-01	4.04e-02	9.99e-03	3.39e-70	1.17e-07	8.23e-01	4.15e-02	1.59e-205
	W	5.57e-01	1.12e-01	1.78e-02	3.82e-05	7.16e-04	3.22e+00	2.04e-01	3.22e-197
	M	3.80e-01	6.77e-02	1.45e-02	1.27e-06	6.17e-05	1.73e+00	8.92e-02	1.20e-201
	S	8.83e-02	1.85e-02	1.83e-03	6.97e-06	1.35e-04	6.57e-01	3.36e-02	0

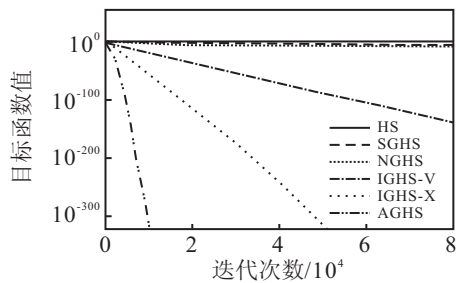


图 1 函数 f_1 最优值进化曲线

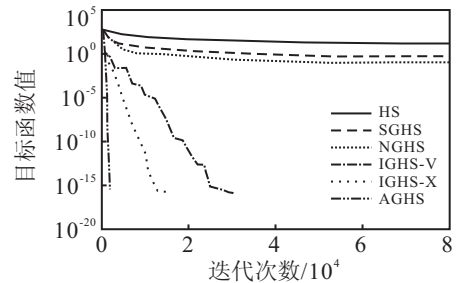


图 3 函数 f_4 最优值进化曲线

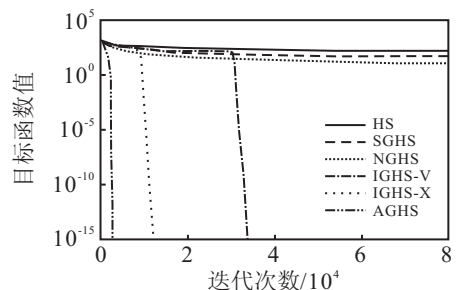


图 2 函数 f_3 最优值进化曲线

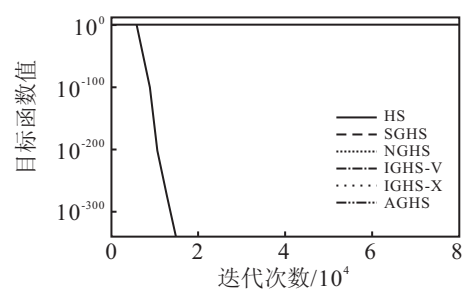


图 4 函数 f_7 最优值进化曲线

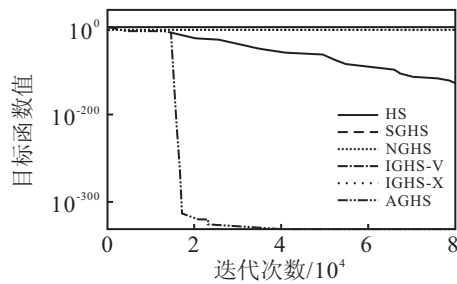


图 5 函数 f_{10} 最优值进化曲线

由图 1~图 5 可见, 无论从收敛速度还是收敛精度看, AGHS 算法的优化效果都明显优于其他几种改进 HS 算法. 此外, 通过函数的最优值进化曲线图可以看出, 本文算法在迭代 10 000 次时即可以获得较好的寻优结果. 总体而言, 引入新的位置更新策略后, AGHS 算法的寻优精度和收敛速度得到了明显提升.

5 结 论

本文针对 NGHS 算法因位置更新操作导致过早收敛、容易陷入局部最优的问题, 提出了自适应全局和声搜索 (AGHS) 算法. AGHS 算法具有以下特点: 1) 排除了 NGHS 算法的变异操作; 2) 通过引入向量范数定义和声记忆库多样性, 给出了新的位置更新策略; 3) 在所有 HS 和 HS 改进算法中需要主观设置参数最少. 利用优化算法 10 个标准测试函数对 AGHS 算法与改进 HS 算法 SGHS、NGHS、IGHS_V、IGHS_X、PSO 和 GA 算法进行寻优效果比较, 仿真结果表明, AGHS 算法的寻优性能得到了有效的提升, 验证了算法的有效性和稳定性.

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] Lee K S, Geem Z W. A new structural optimization method based on the harmony search algorithm[J]. *Computers & Structures*, 2004, 82(9): 781-798.

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

[4] Zou D, Gao L, Li S, et al. An effective global harmony search algorithm for reliability problems[J]. *Expert Systems with Applications*, 2011, 38(4): 4642-4648.

[5] Wang L, Yang R, Pardalos P M, et al. An adaptive fuzzy controller based on harmony search and its application to power plant control[J]. *Int J of Electrical Power & Energy Systems*, 2013, 53: 272-278.

[6] Zou D, Gao L, 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.

[7] 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.

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

[9] Kattan A, Abdullah R. A dynamic self-adaptive harmony search algorithm for continuous optimization problems[J]. *Applied Mathematics and Computation*, 2013, 219(16): 8542-8567.

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

[11] Zhao S Z, Suganthan P N, Pan Q K, et al. Dynamic multi-swarm particle swarm optimizer with harmony search[J]. *Expert Systems with Applications*, 2011, 38(4): 3735-3742.

[12] Wu B, Qian C, Ni W, et al. Hybrid harmony search and artificial bee colony algorithm for global optimization problems[J]. *Computers & Mathematics with Applications*, 2012, 64(8): 2621-2634.

[13] 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.

[14] Zou D, Gao L, Wu J, et al. Novel global harmony search algorithm for unconstrained problems[J]. *Neurocomputing*, 2010, 73(16): 3308-3318.

[15] Valian E, Tavakoli S, Mohanna S. An intelligent global harmony search approach to continuous optimization problems[J]. *Applied Mathematics and Computation*, 2014, 232(3): 670-684.

[16] Xiang W, An M, Li Y, et al. An improved global-best harmony search algorithm for faster optimization[J]. *Expert Systems with Applications*, 2014, 41(13): 5788-5803.

[17] Kennedy J, Eberhart R C. Particle swarm optimization[C]. *IEEE Int Conf on Neural Networks*. Piscataway: IEEE, 1995: 1942-1948.

[18] Holland J H. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*[M]. U Michigan Press, 1975: 41-100.