

# 多子群混合和声搜索算法

夏红刚, 欧阳海滨, 高立群

(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

**摘 要:** 为提高和声搜索算法的优化性能, 提出一种多子群混合和声搜索(MHHS)算法. 该算法基于每个和声到最好和声的距离进行排序, 并依据排序结果分层, 每一层作为一个独立的子群. 不同的子群融合不同的差分调整策略, 以拓宽搜索范围; 同时建立通信机制, 使各子群以一定规格进行信息交流, 促进子群的协同进化. 实验仿真表明, 本文算法在寻优精度、收敛性和鲁棒性方面均优于文献中报道的 HS, EHS, NGHS, MP, PSO, CLPSO, DE, ODE 和 IABC 算法.

**关 键 词:** 和声搜索算法; 子群; 差分调整; 通信机制

中图分类号: TP 301

文献标志码: A

文章编号: 1005-3026(2015)02-0171-06

## Multiple-sub-groups Hybrid Harmony Search Algorithm

XIA Hong-gang, OUYANG Hai-bin, GAO Li-qun

(School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: OUYANG Hai-bin, E-mail: oyhb1987@163.com)

**Abstract:** To improve the optimization performance of harmony search algorithm, a multiple-sub-groups hybrid harmony search (MHHS) algorithm is proposed. The algorithm sorts the entire harmony individuals according to the space distance between each harmony and the best harmony, and then builds multiple layers based on the ranked results, where each layer is as a unique sub-group. Different sub-groups integrate various differential adjustment strategies to broaden search ranges. Meanwhile, the communication mechanism is built to facilitate the information exchange among the multiple-sub-groups and to promote multiple-sub-groups coevolution. Simulation results show that the proposed algorithm is better than the existing algorithms such as HS, EHS, NGHS, MP, PSO, CLPSO, DE, ODE and IABC in terms of optimization accuracy, convergence and robustness.

**Key words:** harmony search algorithm; sub-group; differential adjustment; communication mechanism

和声搜索算法(HS)是一种简单有效的启发式智能算法,由 Geem 等<sup>[1]</sup>于 2001 年提出. HS 算法模拟即兴创作过程,具有机制简单、随机性强、全局优化效果好的特点. 通过近十几年的研究, HS 算法及其改进算法已被成功地应用到许多实际优化问题中. 至今,学者们对 HS 算法的研究主要通过参数的有效调整,优化机制策略的改进以及与其他智能算法的融合这三种途径. 一些改进的和声搜索算法逐渐被提出,例如改进和声搜索算法(IHS)<sup>[2]</sup>,探索和声搜索算法(EHS)<sup>[3]</sup>,新颖

全局和声搜索算法(NGHS)<sup>[4]</sup>,全局动态和声搜索算法(GDHS)<sup>[5]</sup>等. 虽然这些算法提高了 HS 算法的优化性能,但仍存在着搜索盲目、易陷入局部最优、多样性差的缺陷.

子种群的思想在粒子群算法和差分进化算法中都得到了很好的应用,其突出的特点是能够有效增强种群的多样性,提高算法逃逸局部最优的能力. 为了克服 HS 算法的不足,进一步增强 HS 算法的寻优能力,本文提出了一种多子群混合和声搜索(MHHS)算法. 该算法引入子种群的思想,

利用和声之间的距离划分子种群,并结合相应的进化策略,改善算法的优化性能.数值仿真验证了本文算法的有效性.

1 相关知识

和声搜索算法<sup>[1]</sup>的基本步骤如下.

步骤 1 算法参数初始化,给定和声记忆库大小(HMS)、最大迭代次数( $G$ )、和声记忆库的考虑概率(HMCR)、基音调整概率(PAR)、基音调整步长(BW).

步骤 2 给定范围 $[x_i^L, x_i^U]$ ,  $x_i^L$ 和  $x_i^U$ 分别为第  $i$  维变量的下限和上限.根据式(1)随机初始化,产生 HMS 个和声向量存入和声记忆库 HM.

$$x_i^m = x_i^L + \text{rand} \times (x_i^U - x_i^L),$$
$$m = 1, 2, \cdots, \text{HMS}.$$

(1)

步骤 3 基于 HMCR, PAR 和 BW 进行即兴创作,产生一个新的和声向量.在进行基音调整时,当随机数大于 0.5 时,取“+”号,小于 0.5 时,取“-”号.其具体伪代码如图 1 所示.

```
For i = 1 to N    % 即兴创作过程开始
If rand < HMCR  % 和声记忆库的考虑操作
   $r \in \{1, 2, \cdots, \text{HMS}\} \mid x_i^{\text{new}} = x_i^r$ 
  If rand < PAR  % 基音调整操作
     $x_i^{\text{new}} = x_i^r \pm \text{rand} \times \text{BW}$ 
  End If
Else  % 随机变异操作
   $x_i^{\text{new}} = x_i^L + \text{rand} \times (x_i^U - x_i^L)$ 
End If
End For  % 即兴创作结束
```

图 1 HS 算法即兴创作过程的伪代码  
Fig. 1 Pseudo codes of HS improvisation

步骤 4 更新和声记忆库.判断新和声  $x^{\text{new}}$  是否优于当前 HM 内最差和声  $x^{\text{worst}}$ ,若是,则用新和声代替当前 HM 内最差和声.

步骤 5 核查终止准则.如果当前迭代次数  $g$  等于最大迭代次数  $G$ ,则终止运行 HS 算法,否则重复执行步骤 3 和步骤 4.

2 多子群混合和声搜索算法

为进一步增强和声搜索算法的优化性能,本节提出一种多子群混合和声搜索算法.下面将具

体介绍多子群划分、差分调整和通信机制.

2.1 多子群划分

对于多子群划分,传统方法主要有三种:一是按照种群数目进行等量随机划分,例如种群数目为 100,划分为 4 个子种群,每个子种群随机选择 25 个不同的个体,这种方法随机性强,有一定的盲目性,容易导致各个子种群的差别性低,划分子群或者不划分子群效果相差不大;二是按照个体的适应值排序进行划分,即把所有个体按照适应值的大小排序,然后从排序的第一个开始每 25 个划分为一个子种群,虽然这样能够有效显示各个子群间的差异,但是每个子群内部的差异性变差,容易导致算法全局搜索能力下降;三是依照整个种群的具体信息设定平衡标准来进行划分,例如将整个种群划分为优于平均适应值和劣于平均适应值的两个子种群.这种方式在迭代前期有利于算法的搜索,使算法的局部搜索和全局搜索相结合,但是到了迭代后期,平均适应值变化范围非常狭小,个体之间已很难用平均适应值加以区分,这样会导致划分的两个子种群非常相似,使算法难以跳出局部最优.本文提出一种多子群划分策略.具体操作如下.

步骤 1 计算每一个和声到最好和声的空间距离,表达式如下:

$$d_{i, \text{best}} = \|x_i - x_{\text{best}}\|.$$

(2)

步骤 2 根据所计算的距离  $d_{i, \text{best}}$ ,从小到大进行排序.

步骤 3 根据距离的排序进行分层,为简单起见,分为顶层、中间层、底层,基于文献[1-2, 5]的研究,和声记忆库大小最好取 5,因此,在本文算法中保证每一层的和声数目等于 5,如果存在某一层和声数目小于 5,则随机初始化和声,并将其加入这一层直至数目为 5.

2.2 差分调整

在 HS 算法中,原始的基音调整操作受步长 BW 大小的影响:BW 大时有利于广泛搜索,BW 小时有利于精细搜索;可 BW 往往难确定,导致算法难以有效平衡广泛搜索和精细搜索.因此,本文借助个体的差异性,利用差分调整取代基音调整,并针对不同子群的具体信息,设计不同的差分调整策略.

对于处于顶层的子群,每个和声离当前最优和声的空间距离比较小,基于算法聚集和发散思想的考虑,所有和声应逐渐向当前最优和声聚集,同时也进行局部的发散搜索,因此,提出一种最优

差分调整操作,具体表达式如下:

$$x_{i,j} = \alpha \cdot x_{i,j} + \beta \cdot (x_{\text{best},j} - x_{i,j}) + \varepsilon \cdot (1 - \text{rand}); \quad (3)$$

$$\alpha = \alpha_0 \cdot [1 / (1 + \exp(- (d_{i,\text{best}})^\mu))],$$

$$\beta = \beta_0 \cdot [1 / (1 + \exp(- d_{i,\text{best}} \cdot \mu))],$$

$$\mu = g/2.$$

式中:  $x_{i,j}$  表示第  $i$  个和声的第  $j$  维变量,  $x_{\text{best},j}$  表示最优和声的第  $j$  维变量;  $\alpha$  为权重,  $\alpha_0$  为初始权重;  $\beta$  为差分缩放值,  $\beta_0$  为差分缩放初始值;  $\mu$  为距离调整系数,  $g$  为当前迭代次数;  $\varepsilon \cdot (1 - \text{rand})$  为扰动项,  $\varepsilon$  为扰动精度,  $\text{rand}$  为 0 和 1 之间的均匀随机数, 扰动项的目的是进行有效的局部发散搜索。

处于中间层的子群是整个种群空间分布的核心区域,既需要扩大搜索范围,又需要有效的邻域搜索,因此,设计一种多选择差分调整策略,如下式所示:

$$x_{i,j} = \begin{cases} x_{i,j} + \text{rand} \cdot (x_{r_2,j} - x_{r_1,j}), & f(x_{r_2}) \leq f(x_{r_1}); \\ x_{i,j} + \text{rand} \cdot (x_{r_1,j} - x_{r_2,j}), & f(x_{r_2}) > f(x_{r_1}). \end{cases} \quad (4)$$

其中,  $r_1, r_2$  为随机选择的和声标号且  $r_1 \neq r_2 \neq i$ ,  $f(x)$  表示计算  $x$  的目标函数值。

对处于底层的子群进行广泛的随机搜索,因此,采用一种随机差分调整策略,如下式所示:

$$x_{i,j} = x_{r_1,j} + \text{rand} \cdot (x_{r_2,j} - x_{r_3,j}). \quad (5)$$

其中,  $r_1, r_2, r_3$  为随机选择的标号且  $r_1 \neq r_2 \neq r_3 \neq i$ 。

### 2.3 通信机制

在本文算法中,3 个子群处在不同的层次,顶层子群处于群体优化方向的指导阶段,中间层子群处于群体发展阶段,底层子群处于群体培育阶段,为了协调各个子群的进化,提出一种简单通信机制。

通信过程如下:

步骤 1 顶层、中间层和底层子群分别产生 3 个和声向量  $\mathbf{x}_{t,i}$ ,  $\mathbf{x}_{m,i}$ , 和  $\mathbf{x}_{b,i}$ 。每个子群的最差和声向量为  $\mathbf{x}_{t,w}$ ,  $\mathbf{x}_{m,w}$ ,  $\mathbf{x}_{b,w}$ , 每个子群中距离当前最优和声向量最远的和声向量分别为  $\mathbf{x}_{t,f}$ ,  $\mathbf{x}_{m,f}$ ,  $\mathbf{x}_{b,f}$ , 对应的距离为  $d_{t,f}$ ,  $d_{m,f}$ ,  $d_{b,f}$ , 每个子群距离当前最优和声向量最近的和声向量为  $\mathbf{x}_{t,n}$ ,  $\mathbf{x}_{m,n}$ ,  $\mathbf{x}_{b,n}$ , 对应的距离为  $d_{t,n}$ ,  $d_{m,n}$ ,  $d_{b,n}$ 。

步骤 2 独立的子群进行选择操作:

If  $f(\mathbf{x}_{t,i}) < f(\mathbf{x}_{t,w})$ ,  $\mathbf{x}_{t,w} = \mathbf{x}_{t,i}$ , End If

同理  $\mathbf{x}_{m,i}$  和  $\mathbf{x}_{b,i}$  分别与  $\mathbf{x}_{m,w}$  和  $\mathbf{x}_{b,w}$  比较。

步骤 3 分别计算 3 个和声向量  $\mathbf{x}_{t,i}$ ,  $\mathbf{x}_{m,i}$  和  $\mathbf{x}_{b,i}$  到当前最优和声向量的距离, 得到  $d_{t,d}$ ,  $d_{m,d}$  和  $d_{b,d}$ 。

步骤 4 以距离为标准进行通信。

步骤 4.1 顶层子群的新和声向量与中间层的距离最优和声向量比较:

If  $d_{t,d} < d_{m,n}$ ,  $\mathbf{x}_{m,n} = \mathbf{x}_{t,i}$ , End If

步骤 4.2 中间层的新和声向量与顶层的距离最差和声向量比较:

If  $d_{m,d} < d_{t,f}$ ,  $\mathbf{x}_{t,f} = \mathbf{x}_{m,i}$ , End If

步骤 4.3 中间层的新和声向量与底层的距离最优和声向量比较:

If  $d_{m,d} < d_{b,n}$ ,  $\mathbf{x}_{b,n} = \mathbf{x}_{m,i}$ , End If

步骤 4.4 底层的新和声向量与中间层的距离最差和声向量比较:

If  $d_{b,d} < d_{m,f}$ ,  $\mathbf{x}_{m,f} = \mathbf{x}_{b,i}$ , End If

步骤 5 通信结束。

### 2.4 算法步骤

多子群混合和声搜索算法的具体步骤如下。

步骤 1 初始化算法参数。给定和声记忆库大小 HMS, 最大迭代次数  $G$ , 和声记忆库考虑概率 HMCR, 基音调整概率 PAR,  $\alpha_0$  为初始权重,  $\beta_0$  为差分缩放初始值, 距离调整系数  $\mu$ 。

步骤 2 初始化和声记忆库 HM。

步骤 3 进行子群的划分。计算每个和声到当前最优和声的距离  $d_{i,\text{best}}$  并进行排序划分出顶层子群、中层子群和底层子群。对于不同的子群采用不同的标记号, 顶、中、底层子群采用数字 1, 0, -1 标记。

步骤 4 针对所划分的子群分别进行即兴创作。

步骤 5 基于通信机制的要求, 各个子群进行信息交流, 并更新子群。

步骤 6 验证终止条件。如果迭代次数达到最大值  $G$ , 则终止算法运行, 否则重新执行步骤 4 和步骤 5。

## 3 实验仿真与结果

### 3.1 函数优化实验准备

为了验证本文算法的有效性, 与 EHS 算法<sup>[3]</sup>, NGHS 算法<sup>[4]</sup>, MPSO 算法<sup>[6]</sup>, CLPSO 算法<sup>[7]</sup>, DE 算法<sup>[8]</sup>, ODE 算法<sup>[9]</sup>, IABC 算法<sup>[10]</sup> 及 HS 算法<sup>[1]</sup> 进行了比较。所有算法的参数均参照文献中给定的最优参数设置, 在本文算法中, HMS = 15, 每个子群的和声记忆库大小为 5, HMCR = 0.99, PAR = 0.45, 差分缩放初始值  $\beta_0 = 0.5$ , 距离调整系数  $\mu = g/2$ , 对 5 个经典测试函数进行优化。5 个经典函数的表达式如下:

$f_1$ : Schwefel's problem 2.2

$$\min f_1 = \sum_{i=1}^N |x_i| + \prod_{i=1}^N |x_i|.$$

其中  $-100 \leq x_i \leq 100$ , 在  $x_i = 0$  时达到极小值 0.

$f_2$ :rotated hyper-ellipsoid function

$$\min f_2 = \sum_{i=1}^N (\sum_{j=1}^i x_j)^2.$$

其中  $-100 \leq x_i \leq 100$ , 在  $x_i = 0$  时达到极小值 0.

$f_3$ :Griewank function

$$\min f_3 = \frac{1}{4\,000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos(\frac{x_i}{\sqrt{i}}) + 1.$$

其中  $-600 \leq x_i \leq 600$ , 在  $x_i = 0$  时达到极小值 0.

$f_4$ :Ackley's function

$$\min f_4 = 20 + e - 20 \exp(-0.2 \sqrt{\frac{\sum_{i=1}^N x_i^2}{N}}) - \exp(\frac{\sum_{i=1}^N \cos(2\pi x_i)}{N}).$$

其中  $-32 \leq x_i \leq 32$ , 在  $x_i = 0$  时达到极小值 0.

$f_5$ :Zakharov function

$$\min f_5 = \sum_{i=1}^N x_i^2 + (\sum_{i=1}^N 0.5ix_i)^2 + (\sum_{i=1}^N 0.5ix_i)^4.$$

其中  $-100 \leq x_i \leq 100$ , 在  $x_i = 0$  时达到极小值 0.

3.2 仿真结果与分析

3.2.1 参数对 MHHS 算法的影响

以平均最优值作为指标,利用 5 个经典函数来测试 PAR 和  $\beta_0$  对本文算法优化性能的影响,维数  $N = 30$ ,HMS = 5,最大迭代次数  $G = 50\,000$ ,独立运行算法程序 30 次,分别得到优化结果如表 1~2 所示.

表 1 PAR 变化对 MHHS 算法的影响  
Table 1 Effect of PAR on MHHS

函数	PAR				
	0.1	0.25	0.5	0.75	0.9
$f_1$	5.38e-46	8.34e-60	0.00e-00	2.30e-35	1.47e-25
$f_2$	3.38e-19	4.23e-61	0.00e-00	3.35e-25	1.19e-10
$f_3$	5.35e-16	1.28e-36	1.89e-45	2.33e-15	7.28e-02
$f_4$	3.17e-14	8.34e-14	3.49e-15	1.05e-05	1.26e-04
$f_5$	5.33e-36	9.46e-56	2.59e-80	7.29e-45	1.74e-02

由表 1 可知,不同的 PAR 值,所优化的结果偏差比较大,针对不同的函数,需要适当地调整 PAR 值.整体上,PAR 的值在范围[0.25, 0.5]之间,优化的效果较好,因此,针对具体的优化问题,

合理的 PAR 需要具体的调试.

表 2  $\beta_0$  变化对 MHHS 算法的影响  
Table 2 Effect of  $\beta_0$  on MHHS

函数	$\beta_0$				
	0.1	0.25	0.5	0.75	0.9
$f_1$	7.03e-018	2.54e-193	0.00e-000	3.80e-015	9.89e-014
$f_2$	1.63e-013	1.01e-203	0.00e-000	1.69e-016	5.89e-016
$f_3$	3.06e-012	1.20e-045	6.39e-046	5.89e-004	2.03e-015
$f_4$	4.25e-012	3.84e-014	4.04e-015	6.00e-001	6.75e-002
$f_5$	1.33e-013	7.23e-075	4.13e-087	1.87e-002	2.30e-003

由表 2 可知, $\beta_0$  的值影响 MHHS 算法最终解的质量,虽然每一个确定的  $\beta_0$  适合于所有的优化问题,但在整体上, $\beta_0$  的值在范围[0.25, 0.5]之间,优化的效果较好.

3.2.2 各种算法的优化结果比较与分析

本文所进行的数值测试,均采用 MATLAB 7.04 实验仿真软件, Pentium (R) 4, CPU 为 2.93 GHz 的电脑运行算法程序.对  $f_1 \sim f_5$  进行 30 次独立优化仿真,所得的结果如表 3 所示,其中函数维数分别为 10,30,50.当优化结果的数量级小于  $-330$  时,计算机显示结果为 0.

由表 3 可以看出,对于  $f_1$  函数,虽然 EHS 算法、NGHS 算法、CLPSO 算法、ODE 算法和 IABC 算法能搜索到靠近全局最优值的值,并且 MPSO 算法和 IABC 算法在维数为 10 时能够搜索到最优值,但本文算在维数为 10 和 30 时能找到全局最优值,并比其他 8 种和声搜索算法所优化的结果要好. MPSO 算法和 ODE 算法求解出  $f_2$  函数的平均最优值和方差都达到了较高的数量级, EHS 算法、CLPSO 算法和 IABC 算法也取得了较好的优化效果,但本文算法在整体上所搜索到结果相对这些算法来说有进一步的提高.对于  $f_3$  函数,除了 HS 算法优化的精度稍微低一些外,其他 8 种算法优化的精度都较高,所找到的最优值靠近全局最优值.本文算法优化的结果虽没有较大的提高,但仍显示出了良好的优化性能.对于函数  $f_4$ ,EHS 算法、NGHS 算法、CLPSO 算法、ODE 算法、IABC 算法和本文算法所取得的结果相差不多,表明本文算法具有很好的优化竞争能力.函数  $f_5$  是一个典型的高维复杂优化测试问题,求解过程比较困难.本文算法搜索到靠近全局最优值的值,相对于其他算法的优化结果有很大的进步.通过这 5 个函数的实验仿真,从整体上看,本文算法优于其他算法,其原因在于本文算法针对不同的和声采用了不同的优化措施,有效平衡算法的搜索和探测过程,提高算法逃逸局部最优的能力.

表 3 9 种算法所获得的  $f_1 \sim f_5$  ( $N=10,30,50$ ) 的优化结果  
Table 3 Optimization results of nine algorithms for  $f_1 \sim f_5$  ( $N=10,30,50$ )

函数	维数	性能指标	算法								
			HS	EHS	NGHS	MPSO	CLPSO	DE	ODE	IABC	MHHS
$f_1$	10	mean	6.92e-007	4.36e-033	5.98e-016	0.00e-000	8.76e-033	1.56e-011	1.65e-016	0.00e-000	0.00e-000
		SD	1.10e-006	1.88e-029	1.00e-015	0.00e-000	3.78e-035	7.55e-011	6.25e-017	0.00e-000	0.00e-000
	30	mean	8.53e-004	5.11e-013	2.04e-008	8.89e-033	7.35e-014	5.05e-009	2.13e-012	2.75e-025	0.00e-000
		SD	2.55e-003	5.16e-013	2.33e-008	4.64e-033	3.22e-014	4.89e-009	4.52e-012	8.47e-025	0.00e-000
	50	mean	2.65e+001	3.71e-006	5.36e-003	3.72e-023	7.23e-004	3.48e-004	9.23e-007	5.26e-015	8.15e-044
		SD	5.68e-001	5.00e-007	5.60e-003	3.40e-023	1.43e-004	1.09e-004	7.20e-008	7.54e-015	9.06e-043
$f_2$	10	mean	1.87e-004	1.65e-016	1.00e-005	3.00e-256	4.59e-023	3.97e-035	0.00e-000	9.24e-033	0.00e-000
		SD	3.2e-005	6.25e-017	2.20e-005	0.00e+000	2.65e-033	5.01e-033	0.00e-000	1.94e-029	0.00e-000
	30	mean	1.71e-001	2.13e-012	7.28e-002	1.13e-219	3.54e-014	3.13e-006	1.36e-013	3.26e-013	0.00e-000
		SD	7.28e-002	4.52e-012	1.14e-001	0.00e+000	4.86e-015	2.64e-006	3.03e-012	1.07e-012	0.00e-000
	50	mean	3.40e+002	9.23e-003	4.96e+001	3.90e-015	3.43e-002	2.03e+000	1.17e-006	1.15e-002	7.43e-017
		SD	2.66e+002	7.20e-004	5.91e+001	7.85e-015	5.45e-002	1.96e+000	2.27e-006	1.17e-002	2.32e-016
$f_3$	10	mean	1.87e-004	2.44e-033	0.00e-000	0.00e-000	2.35e-032	5.76e-048	0.00e-000	0.00e-000	0.00e-000
		SD	3.24e-005	7.80e-035	0.00e-000	0.00e-000	1.59e-032	9.63e-046	0.00e-000	0.00e-000	0.00e-000
	30	mean	1.71e-003	3.97e-032	8.04e-034	2.78e-044	3.90e-035	1.34e-012	5.76e-038	2.78e-034	3.90e-045
		SD	7.28e-002	5.01e-033	3.04e-031	4.98e-042	7.85e-032	4.64e-012	1.36e-038	4.98e-036	7.85e-044
	50	mean	3.40e-002	8.04e-014	2.88e-018	3.48e-014	1.26e-033	4.64e-006	2.35e-012	2.35e-022	2.78e-034
		SD	2.66e-002	3.04e-015	1.23e-018	7.97e-012	1.30e-033	7.68e-007	1.57e-012	1.60e-022	4.98e-032
$f_4$	10	mean	2.37e-004	1.39e-016	7.11e-016	1.27e-015	4.32e-015	8.62e-003	1.56e-015	2.34e-016	3.87e-016
		SD	1.24e-005	4.42e-015	4.11e-015	0.00e-000	6.21e-015	1.52e-002	1.75e-015	3.40e-015	0.00e-000
	30	mean	2.31e-001	3.89e-016	8.04e-013	2.56e-015	7.90e-013	2.09e-002	1.78e-015	6.32e-016	7.11e-016
		SD	4.23e-002	8.32e-015	3.04e-012	2.75e-015	2.38e-015	2.16e-002	0.00e-000	1.80e-015	3.15e-015
	50	mean	3.40e+002	8.52e-015	3.97e-015	5.05e-009	2.48e-015	1.02e-001	2.01e-015	1.79e-015	3.55e-015
		SD	2.66e+002	8.21e-015	5.01e-013	4.89e-009	4.48e-015	1.75e-001	1.77e-015	1.87e-015	2.28e-015
$f_5$	10	mean	2.08e+004	6.85e-024	2.42e-016	1.09e-026	2.74e-011	1.65e-003	2.35e-032	5.76e-058	6.76e-108
		SD	2.17e+003	6.79e-022	4.98e-016	2.14e-026	8.11e-012	1.38e-002	7.66e-029	1.30e-057	6.30e-107
	30	mean	2.13e+005	1.92e-004	9.19e-001	5.38e-018	9.32e-002	1.38e-001	2.51e-028	4.47e-048	2.47e-098
		SD	2.83e+004	6.85e-004	1.93e-002	1.12e-018	1.50e-003	1.99e-001	1.34e-032	2.88e-048	3.48e-097
	50	mean	3.43e+002	4.91e-001	1.90e+001	4.74e-016	1.50e-003	2.99e-000	3.08e-018	9.28e-018	2.89e-038
		SD	2.51e+001	5.29e-002	1.32e+001	4.26e-017	3.34e-004	4.99e-001	7.26e-017	5.78e-017	6.88e-037

4 结 语

本文提出一种多子群混合和声搜索(MHHS)算法,该算法对 HS 算法进行了三点改进:一是根据每个和声到最好和声的距离进行子种群的划分,以更好地增加算法的搜索范围;二是基于子种群的划分,融合了不同的差分调整策略,提高算法的寻优能力,增加和声记忆库的多样性;三是建立通信机制,使各子群以一定规格进行信息交流,提高算法的搜索空间. 仿真结果表明,本文算法在寻优精度、收敛性和鲁棒性方面整体上优于文献中报道的算法,具有良好的优化潜力.

参考文献:

[ 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 ] 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.

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