

求解多维函数优化问题的混沌和声算法

闫 涛, 沙 锋

(河南城建学院 信息中心, 河南 平顶山 467036)

摘 要: 为了提升和声算法的局部搜索能力, 提出了一种新颖的混沌和声算法. 新算法首先设计了一种佳点集初始化操作, 使得初始和声库更加均匀的分布在定义域空间; 其次引入了一种自适应候选和声产生策略, 增强了算法的收敛速度; 再次设计了一种混沌局部搜索机制, 提升算法的局部搜索能力. 针对五个标准测试函数的仿真实验结果表明, 新算法无论是收敛速度还是求解质量都显著优于基本和声算法.

关键词: 优化; 混沌; 和声算法

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2013)01-0118-05

Chaos Harmony Search Algorithm for Optimization

YAN Tao, SHA Feng

(Information Center, Henan University of Urban Construction, Pingdingshan 467036, China)

Abstract: An novel chaos harmony search algorithms (CHS) are proposed to overcome the shortcoming of lacking the local searching ability. Firstly, in order to distribute more evenly in the domain space, the initialized population is generated based on the good point set. Secondly, an adaptive mechanism, in which candidate harmony is generated, is introduced to accelerate the convergence speed. Thirdly, a local search mechanism based on the chaos sequence is designed to improve the local search capability of CHS. The simulation results for five benchmarks show that compared with the basic harmony search algorithm, the new algorithms performance better.

Key words: optimization; chaos; harmony search algorithm

1 引言

和声算法(Harmony Search, HS)是 2001 年提出的一种新型进化算法, 它模拟了乐器演奏行为, 主要用于求解连续函数优化问题^[1]. 与遗传算法相比, 和声算法的全局寻优能力较强^[2], 因此吸引了众多研究者的注意, 并被广泛的应用于各类优化问题. 和声算法虽然全局探索能力较强, 但是也存在收敛速度较慢, 局部开采能力较差的缺陷^[3-4].

为了提升和声算法的收敛速度和寻优能力, 许多学者都提出了相应的改进算法. 目前, 常见的改进思路主要分为两类: 第一类是提出新的候选和声产生策略, 例如全局最优和声算法(global-best Harmony search, GHS)^[5]; 第二类是针对和声算法中

微调概率 PAR 和微调幅度 BW 两个参数, 改进其设置方式, 例如改进和声算法(Improved harmony search, IHS)^[2]. 然而, 以上两类改进算法针对 Rosebrock 函数都表现不佳.

为了进一步提升和声算法的搜索能力, 本文提出了一种新颖的混沌和声算法(Chaos Harmony Search, CHS). 新算法在基本和声算法的基础上进行了三点改进: (1)在初始化时设计了一种佳点集初始化操作, 使得初始和声库更均匀地分布在定义域内; (2)引入了一种根据和声库自身信息生成新候选和声向量的自适应策略, 加强了算法的收敛速度; (3)为了增强算法的局部搜索能力, 设计了一种混沌局部搜索机制. 本文利用仿真实验证明了新算法的有效性.

2 基本和声算法

基本和声算法的思路如下:初始化时,随机生成 HMS 个和声向量组成初始和声库 HM ; 随后,根据 HM 中所有的和声向量,利用记忆考虑、微调扰动和随机选择三种规则,产生一个新的候选和声向量 X_{new} ; 最终,在 X_{new} 与和声库中最差的和声向量 HM_{worst} 之间进行选择操作,更新和声库. 以上的过程不断重复直到最大迭代次数 $maxIter$ 为止.

以下是 HS 算法关键步骤的详细解释. 其中假定:解决最小化问题;和声库的大小为 HMS ;生成新候选和声向量时,记忆考虑的概率、微调概率和微调幅度分别为 $HMCR$ 、 PAR 和 BW .

2.1 初始化

该步骤利用随机初始化公式(1)生成 HMS 个 D 维和声向量 $\{X_i = (x_i^1, x_i^2, \dots, x_i^D) \mid i=1, \dots, HMS\}$, 进而组成初始和声库 HM .

$$x_i^j = lb^j + rand(0, 1) \times (ub^j - lb^j) \quad (1)$$

式中, ub^j 和 lb^j 分别是定义域上界向量 UB 和下界向量 LB 的第 j 维分量.

2.2 生成候选和声向量

该步骤是根据记忆考虑、微调扰动和随机选择3个规则,生成新的候选和声向量的过程. 其详细思路是:首先,分别以概率 $HMCR$ 和 $1-HMCR$ 在和声库内和和声库外进行搜索. 当在和声库外搜索时,按照公式(1)进行随机选择操作. 当在和声库内搜索时,以概率 PAR 按照公式(3)或(4)进行微调扰动操作,或以概率 $1-PAR$ 按照公式(2)进行记忆考虑操作. 最终,搜索得到的各个分量构成新的候选和声向量 X_{new} . 其详细过程如图1所示.

```

if rand(0,1) < HMCR then
    i = randn(1, HMS);
     $x_{new}^i = hm^i$ ; (2)
    if rand(0,1) < PAR then
        if rand(0,1) < 0.5 then
             $x_{new}^i = x_{new}^i + rand(0,1) * BW$ ; (3)
        Else
             $x_{new}^i = x_{new}^i - rand(0,1) * BW$ ; (4)
        endif
    endif
else
     $x_{new}^i = lb_i + (ub_i - lb_i) * rand(0,1)$ ;
endif

```

图1 生成候选和声向量的伪码

图1中, $rand(0, 1)$ 表示生成 $(0, 1)$ 之间均匀分布的随机数; $randn(1, HMS)$ 表示生成 $[1, HMS]$ 之间均匀分布的随机整数.

2.3 更新和声库

该步骤比较了新和声向量 X_{new} 与和声库中最差的和声向量 HM_{worst} 的目标函数, 如果 $f(X_{new}) < f(HM_{worst})$, 则用 X_{new} 替换和声库中的 HM_{worst} ; 否则, 和声库保持不变.

3 混沌和声算法

3.1 基于佳点集的初始化操作

文献[4]指出如果初始化种群能够更均匀的分布在定义域空间, 那么进化算法会拥有更快的收敛速度. 基于此, 本部分设计了基于佳点集的初始化种群操作.

佳点集是数论中的一个概念, 它具有以下性质: 当计算函数 f 在 D 维欧式空间的单位立方体上 G^D 的积分时, 利用 n 个佳点函数值构成的任意加权可以使得误差最小. 基于以上性质, 文献[6]在理论和实验上都证明了利用佳点集组成初始种群比随机生成的初始种群在定义域空间内分布更均匀, 进而使得算法收敛速度更快, 求解效果更优, 故 CHS 算法利用它进行初始化操作, 它包含两个基本步骤:

第一: 利用随机初始化公式(5)生成 HMS 个 D 维和声向量 $\{Y_i = (y_i^1, y_i^2, \dots, y_i^D) \mid i=1, \dots, HMS\}$, 进而组成初始和声库 $HM1$.

$$y_i^j = dec(r_j \times i) \quad (5)$$

$$r_j = dec(2\cos(2\pi j/p))$$

式中: $j = \{1, 2, \dots, D\}$; p 是满足 $(p-3)/2 \geq 0$ 的最小素数; $dec(a)$ 函数表示求 a 的小数部分.

第二: 利用公式(6), 将和声库 $HM1$ 中的和声向量 $\{Y_i = (y_i^1, y_i^2, \dots, y_i^D) \mid i=1, \dots, HMS\}$ 从 $[0, 1]^D$ 映射到定义域 $[LB, UB]$, 最终生成新的和声库 $HM = \{X_i = (x_i^1, x_i^2, \dots, x_i^D) \mid i=1, \dots, HMS\}$.

$$x_i^j = lb^j + y_i^j \times (ub^j - lb^j) \quad (6)$$

ub^j 和 lb^j 分别是定义域上界向量 UB 和下界向量 LB 的第 j 维分量.

3.2 生成候选和声向量

为了提升算法的收敛速度和局部搜索能力, 新算法的生成候选和声向量操作进行了两项改进: 第一, 微调扰动时引入了自适应学习机制; 第二, 每隔 LP 代, 利用混沌局部搜索算子进一步加强和声库质量.

3.2.1 自适应的微调扰动操作

和声算法的求解能力依赖于微调幅度参数 BW 的设置,新算法期望改变这种情况,故在微调扰动操作时,CHS 算法不再是调用公式(3)或(4)生成候选和声向量 $\mathbf{X}_{\text{new}}$;而是利用和声库自身的上下边界信息,根据公式(7)或(8)进行局部搜索,生成候选和声 $\mathbf{X}_{\text{new}}$.

$$x_{\text{new}}^j = x_{\text{new}}^j + \text{rand}(0,1) * (hm_{\text{max}}^j - x_{\text{new}}^j) \quad (7)$$

$$x_{\text{new}}^j = x_{\text{new}}^j - \text{rand}(0,1) * (x_{\text{new}}^j - hm_{\text{min}}^j) \quad (8)$$

其中:

$hm_{\text{max}}^j = \max(hm_1^j, hm_2^j, \dots, hm_{\text{HMS}}^j) \quad j=1, \dots, D$;
 $hm_{\text{min}}^j = \min(hm_1^j, hm_2^j, \dots, hm_{\text{HMS}}^j) \quad j=1, \dots, D$;
 分别表示和声库 $\mathbf{HM}$ 中所有和声向量第 j 维的最大值和最小值.

这种方式存在两个优势:(1)不需要设置微调幅度参数 BW ;(2)具有自适应性,初始阶段,和声库的方差较大,由式(7)和(8)可知微调扰动操作的搜索范围较大,相当于全局性搜索;随着算法进行,和声库的方差变小,微调扰动的幅度也变小,相当于局部搜索.

3.2.2 混沌局部搜索

在文献[4-6]中,相应的改进和声算法求解 Rosenbrock 函数时,都表现不佳.本文认为这是相应改进算法局部搜索能力较差造成的.

为了克服这个缺陷,本文设计了一种混沌局部搜索算法.新算法的基本思想是:当执行 LP 次生成候选和声向量操作时,从和声库最优的前 20% 个体中随机选出一个和声向量 $\mathbf{X}_{\text{md}}$,进一步利用混沌局部搜索算子增强其质量.该算子的五个具体步骤如下:

第一步:从和声库 $\{X_i \mid i=1, \dots, \text{HMS}\}$ 最优的前 20% 个体中随机选出一个和声向量 $\mathbf{X}_{\text{md}}$.

第二步:利用公式(9)生成 Logistic 混沌变量 ch_i

$$ch_{i+1} = 4 \times ch_i \times (1 - ch_i) \quad i = \{1, \dots, K\} \quad (9)$$

其中, $ch_i \in (0,1)$ 且 $ch_i \neq 0.25, 0.5$ 和 0.75 ;同时 ch_1 是 $(0,1)$ 之间均匀分布的随机数; K 表示混沌序列的长度.

第三步:利用公式(10)将混沌变量 ch_i 映射成为定义域 $[LB, UB]$ 内的混沌向量 $\mathbf{CH}_i$.

$$\mathbf{CH}_i = \mathbf{LB} + ch_i \times (\mathbf{UB} - \mathbf{LB}) \quad i = \{1, \dots, K\} \quad (10)$$

其中 $\mathbf{LB}$ 表示定义域的下限向量, $\mathbf{UB}$ 表示定义域的上限向量.

第四步:利用公式(11)将混沌向量 $\mathbf{CH}_i$ 与和声向量 $\mathbf{X}_{\text{md}}$ 线性组合,生成候选和声向量 $\mathbf{X}_{\text{new}}$.

$$\mathbf{X}_{\text{new}} = (1 - \lambda) \mathbf{X}_{\text{md}} + \lambda \times \mathbf{CH}_i \quad i = \{1, \dots, K\} \quad (11)$$

其中 λ 是收缩因子.

$$\lambda = \frac{\text{maxIter} - \text{iter} + 1}{\text{maxIter}} \quad (12)$$

其中 maxIter 表示算法的最大迭代次数, iter 表示算法的当前迭代次数.

由公式(12)可知,收缩因子 λ 随着进化代数增加逐渐变小,又由公式(11)可知, λ 越小则混沌搜索的范围越小,综上所述可知随着进化过程,局部搜索范围越来越小.

第五步:如果候选和声向量 $\mathbf{X}_{\text{new}}$ 的目标函数优于 $\mathbf{X}_{\text{md}}$,那么 $\mathbf{X}_{\text{new}}$ 进入和声库替代 $\mathbf{X}_{\text{md}}$,局部搜索结束;否则,如果混沌序列长度达到 K ,那么局部搜索也结束;否则跳到第二步继续执行.

与一般的混沌局部搜索策略不同,新的混沌局部搜索算子不是针对和声库中最优秀的和声向量进行的,而是针对和声库中目标函数较优的前 20% 个体展开的.故增大了混沌搜索的范围,进一步提升算法的寻优能力.

3.2.3 生成候选和声向量的流程

基于自适应微调扰动操作和混沌局部搜索算子,生成候选和声向量 $\mathbf{X}_{\text{new}}$ 的新的流程如图 2 所示.

```

for j=1 to D
  if rand(0,1)<HMCR then
    i=randn(1,HMS);
     $x_{\text{new}}^j = hm_i^j$ ;
    if rand(0,1)<PAR then
      if rand(0,1)<0.5 then
         $x_{\text{new}}^j = x_{\text{new}}^j + \text{rand}(0,1) * (hm_{\text{max}}^j - x_{\text{new}}^j)$ ;
      else
         $x_{\text{new}}^j = x_{\text{new}}^j - \text{rand}(0,1) * (x_{\text{new}}^j - hm_{\text{min}}^j)$ ;
      endif
    endif
  else
     $x_{\text{new}}^j = lb_j + (ub_j - lb_j) * \text{rand}(0,1)$ ;
  endif
endfor
if mod(iter, LP)=0 then
  利用混沌局部搜索算子进行局部搜索
end
  
```

图 2 新的生成候选和声向量的伪码

图 2 中, $\text{rand}(0,1)$ 表示生成 $(0,1)$ 之间均匀分

布的随机数; $\text{randn}(1, \text{HMS})$ 表示生成 $[1, \text{HMS}]$ 之间均匀分布的随机整数。

3.3 参数设置

在基本 HS 算法中,参数记忆考虑概率 HMCR 、微调概率 PAR 、微调幅度 BW 和和声库大小 HMS 对于算法的性能影响最大,它们被设置成为固定数值,HS 算法的创始人 Geem 推荐的数值分别为 0.9、0.3、0.01 和 5。

在 CHS 算法中,为了提升算法的局部搜索能力,我们提出了新的设置原则。

(1) 参数 BW 不需要设置。这是由于微调波动操作时,不再是根据微调幅度进行调整,故参数 BW 已经不存在。详见 3.2.1 部分。

(2) 记忆考虑概率 HMCR 设置为 0.99。这是因为参数 HMCR 用于控制探索和开采的平衡。例如,当 HMCR 为 0 时,和声算法变成了一个纯粹的随机搜索算法,此时算法利用定义域内随机生成的向量作为新的候选和声向量;当 HMCR 为 1 时,情形正好与之相反,此种情形下,新生成的和声向量不再接受来自和声库之外的信息,而是完全利用和声库内的信息。我们基于实验的证据设置其为 0.99。

(3) 微调概率 PAR 按照公式(13)进行设置。如果候选和声向量中某个变量由和声库生成,那么 PAR 确定该位置是否需要进一步进行的局部调整。 PAR 越大,则需要需要进行局部调整的位置越多; PAR 越小,需要进行局部调整的位置越少。一般而言,算法开始阶段,为了加大搜索范围,一般需要设置较大的 PAR ;而算法结束阶段,由于和声库内的和声向量质量都比较高,故设置较小的 PAR ,用于保持原和声库的信息。综上所述,本文将 PAR 按照公式(13),设置成为一个随着进化代数 iter 增加,由 $\text{PAR}_{\max}$ 到 $\text{PAR}_{\min}$ 不断减小的数值。

$$\text{PAR} = \text{PAR}_{\max} - \frac{(\text{PAR}_{\max} - \text{PAR}_{\min}) \times \text{iter}}{\text{maxIter}} \quad (13)$$

其中 $\text{PAR}_{\max}$ 表示微调概率 PAR 的上界; $\text{PAR}_{\min}$ 表示微调概率 PAR 的下界; maxIter 表示最大进化代数。

(4) 和声库尺寸 HMS 设置成 50。由公式(7)和(8)可知,和声库尺寸 HMS 越大,则算法的搜索空间越大,同时收敛速度也较慢; HMS 越小,情形正好相反。事实上,基于实验的证据显示, HMS 太大或者太小都不利于算法,基于实验的证据推荐 $\text{HMS}=50$ 。

(5) 混沌局部算子中,存在两个参数,一个是搜索频率 LP ,一个是搜索长度 K 。当 LP 过小和 K 过长时,虽然会增强算法的局部搜索能力,但是也会造成过多的适应度函数评价次数;当 LP 过大和 K 过小时,情形正好相反。基于实验的证据,本文设置 $\text{LP}=200, K=\text{HMS}/5=10$ 。

3.4 时间复杂度分析

若以适应度函数评价次数做为时间复杂度分析的标准,则 HS 算法的时间复杂度为 $O(\text{HMS} + \text{maxIter})$,其中 HMS 表示和声库中和声的数目, maxIter 表示算法中产生候选和声向量步骤的运行次数。又由 HMS 通常远远小于 maxIter ,故时间复杂度可以简化为 $O(\text{maxIter})$ 。

与之相比,CHS 算法在 HS 算法基础上,增加了混沌局部搜索操作,故时间复杂度为 $O(\text{HMS} + \text{maxIter} + \text{maxIter}/\text{LP} \times K)$,根据 3.3 部分的参数设置约减后,其时间复杂度也为 $O(\text{maxIter})$ 。综上所述,改进算法 CHS 并没有增加算法的时间复杂度。

4 实验结果

本部分针对五个基本的单模和多模函数,分别利用 CHS 算法和基本 HS 算法进行测试,比较两种算法的求解效果和收敛速度。所有试验都是在硬件环境为 Intel Pentium Dual CPU 1.73GHz,内存 3GB;软件环境为 windows sp3 操作系统,matlab7.7.0 版本的机器上运行的。所有实验都独立运行 30 次。

通过实验得出以下结论:

第一,CHS 算法在初始阶段的收敛速度明显快于基本 HS 算法,且最终的最优解明显优于基本和声算法。它主要是由佳点集初始化和混沌局部搜索算子决定的。佳点集初始化的和声库比随机初始化的和声库在整个定义域内分布更均匀。此外,由于初始阶段 HM 库中和声向量可提升空间较大,故混沌局部搜索算子进一步的提升了和声向量的质量。

第二,CHS 算法的收敛过程基本可以分为两个阶段,每个阶段又都存在一个快速下降期和一个平台期。在第一个阶段开始时,佳点集初始化操作和混沌局部搜索算子使得收敛曲线下降较快,然而,第一阶段的后期,局部搜索已经不能进一步提升最优解,故收敛曲线出现了一个平滑的阶段。在第二阶段开始时,由于整个和声库中和声向量之间的差异在不断减小,故自适应候选和声向量产生机制的公式(7)

和(8)起到了决定的作用,从而使得收敛曲线继续快速下降;这个过程一直持续到算法收敛为止,此时收敛曲线变得平滑。

5 结束语

本文提出了一种混沌和声算法求解连续函数优化问题。为了提升求解效果,新算法在基本和声算法基础上进行了三方面改进:第一,引入了佳点集初始化操作,增强了初始种群的均匀性;第二,在生成新候选和声向量时,引入了一种自适应的学习机制;第三,设计了一种混沌局部搜索算子,用于提升较优和声向量的质量。五个标准测试函数的仿真实验结果证明:新算法无论是求解质量还是收敛速度都显著优于基本和声算法。

参考文献:

- [1] Geem Z W, Kim J H. A new heuristic optimization algorithm: harmony search[J]. Simulation, 2001, 76 (2):60-68.
- [2] 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.
- [3] 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.
- [4] Wang C M, Huang Y F. Self-adaptive harmony search algorithm for optimization[J]. Expert Systems with Applications, 2010, 37(4):2826-2837.
- [5] Omran M G H, Mahdavi M. Global-best harmony search[J]. Applied Mathematics and Computation, 2008, 198(2):643-656.
- [6] 张铃,张钊. 佳点集遗传算法[J]. 计算机学报, 2001, 24(009):917-922.
- [7] LiMingxing, Heng Ping, Dong Peiwu. Research on artificial neural network method for credit application [J]. Research Information Ltd(RIL) UK, 2004(2): 127-131.
- [8] 张中辉,操家庆,梁意文. 基于联动机制的入侵防御系统[J]. 计算机时代,2006(7):28-30.
- [9] 宁海滨. 基于校园网的入侵检测系统设计与实现[D]. 北京:北京大学,2008.
- [10] 刘文星. 网络攻击频率混沌时间序列预测[D]. 长沙:国防科学技术大学,2008.
- [11] 潘炜,李伟华. 网络安全联动模型的设计与应用[J]. 计算机科学,2006(5):113-116.

作者简介:

闫涛男,(1970—),本科,副教授.研究方向为数据挖掘。

作者简介:

孙利男,(1972—),硕士,副教授.研究方向为计算机网络与安全。

徐亮男,(1978—),硕士,讲师.研究方向为数字图像处理与可视化技术研究。

吴辞文男,(1975—),博士研究生,讲师.研究方向为计算机网络与应用。

(上接第 117 页)