

DOI:10.3969/j.issn.1673-4785.201402019

网络出版地址: <http://www.cnki.net/kcms/detail/23.1538.TP.20150326.1016.004.html>

动态调整策略改进的和声搜索算法

拓守恒, 雍龙泉, 邓方安

(陕西理工学院 数计学院, 陕西 汉中 723001)

摘要: 为了得到高维复杂问题的全局高精度最优解, 提出一种动态调整策略, 并用该策略改进和声搜索算法。算法选取和声记忆库中最差和声向量作为优化调整目标, 随着迭代的进行, 逐步降低决策变量的调整概率, 该方法能够使得算法在全局探索能力和局部高精度开发能力之间实现平衡, 有效提高了新和声更新最差和声的成功率。通过 6 个高维 Benchmark 测试函数的仿真结果表明, 提出的动态调整策略能够有效提高和声搜索算法求解高维复杂优化问题的能力。

关键词: 自适应调整策略; 高维优化问题; 和声搜索算法

中图分类号: TP391 **文献标志码:** A **文章编号:** 1673-4785(2015)02-0307-09

中文引用格式: 拓守恒, 雍龙泉, 邓方安. 动态调整策略改进的和声搜索算法[J]. 智能系统学报, 2015, 10(2): 307-315.

英文引用格式: TUO Shouheng, YONG Longquan, DENG Fang'an. Dynamic adjustment strategy for improving the harmony search algorithm[J]. CAAI Transactions on Intelligent Systems, 2015, 10(2): 307-315.

Dynamic adjustment strategy for improving the harmony search algorithm

TUO Shouheng, YONG Longquan, DENG Fang'an

(School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong 723001, China)

Abstract: A dynamic adjustment strategy is used to improve the harmony search algorithm for solving high-dimensional multimodal global optimization problems. It chooses the worst harmony vector from the harmony memory (HM) as an optimization objective vector. With the process of iteration, the adjustment probability of decision variables is reduced step by step. It can achieve the balance effectively between the global exploration powers and local exploitation competence, and can increase the success rate of evolution. Finally, the experimental results of 16 high-dimension benchmark functions demonstrated that the proposed method can enhance the performance and robustness of the harmony search algorithm obviously in solving large scale multimodal optimization problems.

Keywords: adaptive adjustment strategy; high-dimensional optimization problems; harmony search algorithm

近年来, 随着社会的发展和大数据时代的到来, 人们对科技产品的能耗和性能要求越来越高, 使得产品的设计遇到了极大的挑战。许多产品的设计需要考虑很多的设计要求, 并要使其产品整体设计能够达到最优, 这些问题都可转化为大规模复杂优化问题 (optimization problems, OP)。对于 OP 问题, 近

年来, 研究者将关注的焦点集中在模拟自然的启发式 (meta-heuristics) 优化方法^[1-9]等。

和声搜索算法是 Geem 等^[1]在 2001 年通过模拟音乐家创作音乐和调节和声的过程, 提出的一种新的启发式优化算法。音乐家在音乐创作过程中, 需要不断调整各个音符, 使其成为优美和声。由于和声算法搜索能力强, 并且结构简单, 很容易在各种软件和硬件中实现, 引起很多科学研究者和工程设计人员的关注, 近年来, 已经广泛应用于实践中, 例如, 管网优化设计^[10]、结构优化设计^[11]、交通运输

收稿日期: 2014-02-21. 网络出版日期: 2015-03-26.

基金项目: 国家自然科学基金资助项目 (11401357), 陕西省教育厅科研资助项目 (14JK1141); 汉中市科技局科研资助项目 (2013hzzx-39); 陕西理工学院科研资助项目 (SLGKY 13-27).

通信作者: 拓守恒. tuo_sh@126.com.

路径优化^[12]、电力系统经济负荷分配问题^[13]和 PID 控制器优化^[14]等领域。然而,研究发现,在有限的时间内,和声搜索算法具有很强的全局探索能力,但是,在实数优化问题中,求解精度较低^[15-17]。为此,很多改进的和声搜索算法被提出,潘全科等^[15]采用动态子种群策略提出了局部最好和声搜索算法,利用自适应动态策略提出一种全局最优和声搜索算法^[16]。M. Mahdavi 等设计出一种参数动态调整策略,有效改进了 HS 算法的性能 (IHS)^[18]; M. G. H. Omran 提出全局最优和声搜索算法 (GHS)^[19]; Zou^[20]采用一种很简单的差分学习策略,有效屏蔽了参数 HMCR (harmony memory considering rate) 和 PAR (pitch-adjusting rate),降低了算法的复杂性^[21-22]。P. Yadav 给出一种智能调整和声搜索算法 (ITHS)^[17]; S. Das 通过理论分析与证明,给出了一种新的和声步长 (pitch bandwidth, BW) 调整算法 (EHS)^[23];本文作者在文献[24]和[25]中分别提出了“混沌和声搜索算法”与“基于教与学策略的和声搜索算法”;另外,在一些具体应用中,对和声搜索算法进行了有效改进^[26-34]。尽管上述改进算法从某些方面对和声搜索算法进行了改进,但并没有从算法的运行代价考虑,比如 EHS 算法,虽然搜索能力有了明显的改进,但是,由于每次迭代都需要计算和声记忆库 (harmony memory, HM) 的方差,其计算量甚至超过了和声搜索算法本身的计算量,使得算法的运行代价是标准和声算法 HS 的好几倍。特别是在求解高维复杂优化问题时,目前的和声算法运行速度普遍较慢。为此,本文通过引入一种动态和声调整策略,使其能够有效提高和声算法的性能,并且,能使算法运行代价降低,提高其搜索速度。

1 标准和声搜索算法

考虑如下优化问题:

$$\min f(\mathbf{x}), \mathbf{x} = [x_1 \ x_2 \ \cdots \ x_D]$$

$$\text{s.t. } x_i \in [xL_i, xU_i], i = 1, 2, \cdots, D$$

式中: $f(\mathbf{x})$ 是目标函数, \mathbf{x} 是由 D 个决策变量 x_i ($i = 1, 2, \cdots, D$) 构成的解向量, $[xL_i, xU_i]$ 表示决策变量 x_i 的可行搜索区域。将该优化问题对应于和声算法, \mathbf{x} 表示一个和声向量, $f(\mathbf{x})$ 表示和声 \mathbf{x} 的旋律优美程度 (在该问题中, $f(\mathbf{x})$ 越小, 表示其和声旋律越优美)。

1.1 标准和声搜索算法

标准和声搜索算法的基本步骤如下:

1) 设置参数值 (D , HMS , T_{\max} , $HMCR$, PAR ,

BW)。各参数含义如下:

D 为问题的维数, HMS 为和声记忆库大小, T_{\max} 为算法迭代次数; $HMCR$ 为和声记忆库选择概率, PAR 为局部微调概率, BW 为局部微调步长值。

2) 随机初始化和声记忆库 HM

$$x_i^j = xL_i + \text{rand} \cdot (xU_i - xL_i)$$

$$i = 1, 2, \cdots, D; j = 1, 2, \cdots, HMS$$

式中: rand 是 (0, 1) 中的随机数。

$$\mathbf{HM} = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{HMS} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & \vdots & & \vdots \\ x_1^{HMS} & x_2^{HMS} & \cdots & x_D^{HMS} \end{bmatrix}$$

3) 利用 3 种和声调节规则创作新和声

通过如下 3 个规则产生新和声 $\mathbf{x}^{\text{new}} = [x_1^{\text{new}} \ x_2^{\text{new}} \ \cdots \ x_D^{\text{new}}]$ 。

①和声记忆库选择。新和声向量 \mathbf{x}^{new} 的决策变量 x_i^{new} ($i = 1, 2, \cdots, D$) 以概率 $HMCR$ 从和声记忆库的第 i 维 $[x_1^i \ x_2^i \ \cdots \ x_i^{HMS}]^T$ 中随机选取。

②局部微调。局部微调是将①中产生的 x_i^{new} ($i = 1, 2, \cdots, D$) 以概率 PAR 进行再次微调。微调方法如下:

$$x_i^{\text{new}} = x_i^{\text{new}} \pm \text{rand} \cdot BW(i)$$

③搜索空间内随机产生。新和声向量 \mathbf{x}^{new} 的决策变量 x_i^{new} ($i = 1, 2, \cdots, D$) 以概率 $1-HMCR$ 在搜索空间内随机产生。产生方法如下:

$$x_i^{\text{new}} = xL_i + \text{rand} \cdot (xU_i - xL_i)$$

4) 更新操作

如果新和声向量 \mathbf{x}^{new} 优于 HM 中最差的和声 $\mathbf{x}^{\text{worst}}$, 则用 \mathbf{x}^{new} 将其替换, 否则, 转至 (3) 重新产生新和声。重复 3)、4), 直到满足终止条件。

2 动态降维和声调整策略

2.1 2 种和声调整策略分析与比较

目前的和声搜索算法和一些改进算法是在整个种群的基础上通过组合策略 (规则①) 产生新的候选解, 这样实现了组合算子的多样性, 因此具有较好的全局搜索性能。但是, 在进化后期, 即使发现了全局最优解所在的区域, 由于其较低的更新成功率 (更新成功率是指每次产生的新解好于和声记忆库中最差解的概率), 使得算法往往很难获得高精度的最优解。

对于一个高维优化问题, 必然要从大范围的扰动开始逐步到小范围 (部分维度) 的微调, 最终使其在所有维上都能够达到最优, 然而, 和声优化算法在

进行优化时,总是通过组合算子产生一个全新的个体 \mathbf{x}^{new} ,然后与和声记忆库 \mathbf{HM} 中最差个体 $\mathbf{x}^{\text{worst}}$ 比较其优劣性,决定是替换还是丢弃。假设某一 D 维优化问题的最优解为 $[x_1^* \ x_2^* \ \cdots \ x_D^*]$,利用和声搜索算法对其进行优化,开始时,随机初始化后的和声记忆库 \mathbf{HM} 如式(1),经过一段时间优化后, \mathbf{HM} 变为式(2),

$$\mathbf{HM} = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{\text{HMS}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & \vdots & & \vdots \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \cdots & x_D^{\text{HMS}} \end{bmatrix} \quad (1)$$

$$\mathbf{HM} = \begin{bmatrix} x^{1'} \\ x^{2'} \\ \vdots \\ x^{\text{HMS}'} \end{bmatrix} = \begin{bmatrix} y_1 & x_2^* & \cdots & x_D^* \\ x_1^* & y_2 & \cdots & x_D^* \\ \vdots & \vdots & & \vdots \\ x_1^* & x_2^* & \cdots & y_D \end{bmatrix} \quad (2)$$

此时,假设 \mathbf{HM} 的每个和声中都仅有一个决策变量没有达到最优。由于和声算法中规则①的调整概率 HMCr 一般都大于 0.9,也就是说规则①在和声搜索算法中是非常重要的,这时,如果仅仅采用和声搜索算法中的规则①进行优化。采用下列 2 种方法分别产生一个新和声 \mathbf{x}^{new} ,分析 2 种方法的更新成功率。

1) 如果完全利用规则①产生一个新和声 $\mathbf{x}^{\text{new}} = [x_1^* \ x_2^* \ \cdots \ x_D^*]$ 。那么, $x_i^{\text{new}} (i = 1, 2, \cdots, D)$ 取到 $x_i^* (i = 1, 2, \cdots, D)$ 的概率为 $\frac{\text{HMS} - 1}{\text{HMS}}$,则 $\mathbf{x}^{\text{new}} = [x_1^* \ x_2^* \ \cdots \ x_D^*]$ 的概率为 $\left(\frac{\text{HMS} - 1}{\text{HMS}}\right)^D$ 。

2) 选取 \mathbf{HM} 中一个和声作为调整目标,然后,随机的将其中某一决策变量利用规则①重新生成。假设选取 x^1 作为优化调整目标 \mathbf{x}^{new} ,则 $\mathbf{x}^{\text{new}} = [y_1 \ x_2^* \ x_3^* \ \cdots \ x_D^*]$,随机选取一个决策变量进行调整,选取到 y_1 的概率为 $1/D$,并且利用规则①能够将其调整为 x_1^* 的概率为 $(\text{HMS} - 1)/\text{HMS}$,因此,优化调整后, $\mathbf{x}^{\text{new}} = [x_1^* \ x_2^* \ \cdots \ x_D^*]$ 的概率为 $(1/D) \cdot ((\text{HMS} - 1)/\text{HMS})$ 。

假设 $\text{HMS} = 10, D = 10$,则利用方法 1 得到最优解 $\mathbf{x}^{\text{new}} = [x_1^* \ x_2^* \ \cdots \ x_D^*]$ 的概率为 $\left(\frac{\text{HMS} - 1}{\text{HMS}}\right)^D = 0.348\ 678$,利用方法 2 得到最优解的概率为 $(1/D) \cdot ((\text{HMS} - 1)/\text{HMS}) = 0.09$,显然此时,方法 1 更容易得到最优解。但是,当 $\text{HMS} = 10$,维度 $D = 100$ 时,方法 1 得到最优解的概率为

$\left(\frac{\text{HMS} - 1}{\text{HMS}}\right)^D = 2.656 \times 10^{-5}$,利用方法 2 得到最优解的概率为 $(1/D) \cdot ((\text{HMS} - 1)/\text{HMS}) = 0.009$,显然,方法 2 具有更高的成功率。图 1 给出 2 种方法的成功率变化曲线。

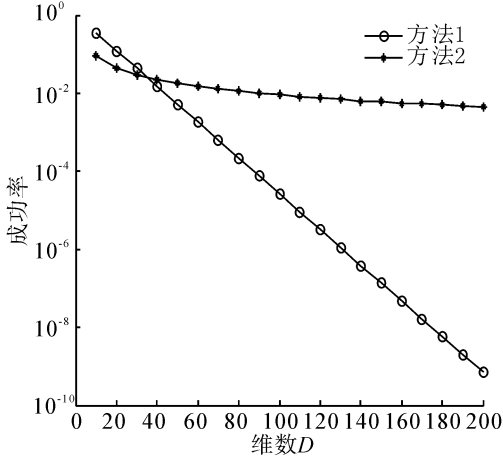


图 1 2 种方法在维数不同时的更新成功率曲线图

Fig.1 The update-success rate curve of two methods on different dimensionalities

图 1 可以看出,在 HMS 相同的情况下,随着维数 D 的增加,方法 1 的成功率急剧下降,而方法 2 下降幅度很小。在问题的维数较低时,方法 1 的更新成功率高于方法 2,但当维数 $D > 40$ 时,方法 2 的更新成功率高于方法 1。

由上例可知,对于一个高维优化问题,利用方法 1 难以驱动算法获得高精度的最优解。如果借鉴方法 2 的思想进行优化,就有可能取得较好的优化效果。因此,本文提出一种基于动态减少调整维数的和声优化策略。该策略首先选取和声记忆库中最差和声向量作为优化目标,然后,通过对最差和声向量的不断调整,使其达到最优解。在优化刚开始时,对最差和声向量进行较多维数的扰动,使得算法具有较强的空间探索能力,随着优化的进行,逐步降低扰动概率,仅仅在较少维上进行优化调整,使得优化调整具有更高的成功率,从而获得高精度的全局最优解。

2.2 基于动态降维调整的和声搜索算法

本文提出的基于动态降维调整策略的和声搜索算法流程请见图 2。本文算法中,调整概率 $\text{TP} = \text{TP}_{\text{max}} - (\text{TP}_{\text{max}} - \text{TP}_{\text{min}}) \cdot (t/\text{T}_{\text{max}})^2$ 随着迭代次数的增加逐步减小(如图 3),其中, TP_{max} 和 TP_{min} 分别为最大调整概率值和最小调整概率值。

在算法优化开始时,以较大的扰动调整概率 TP_{max} 进行优化,随着优化进行,调整概率 TP 逐渐减小,开始进行局部微调。但是,为了防止优化调整概率太小,可能导致所有维都得不到调整,因此,需要从

1~D 中随机选取一维 $J = \text{ceil}(\text{rand} \times D)$,使其必须得到调整,避免了算法“空转”。这样调整的好处是,迭代初期,需要较强的全局扰动能力,此时,可以在优化目标向量 \mathbf{x}^{new} 上加大扰动力度,增强种群多样性,使其具有较强的全局探索能力,随着优化的进行,到了后期,多数个体可能已经聚集在了全局最优解附近,此时,开始加强局部最优解的探索,为了保证较高的更新成功率,对优化目标向量 \mathbf{x}^{new} ,选择较少的维数进行优化调整,从而,增强算法的求解精度。

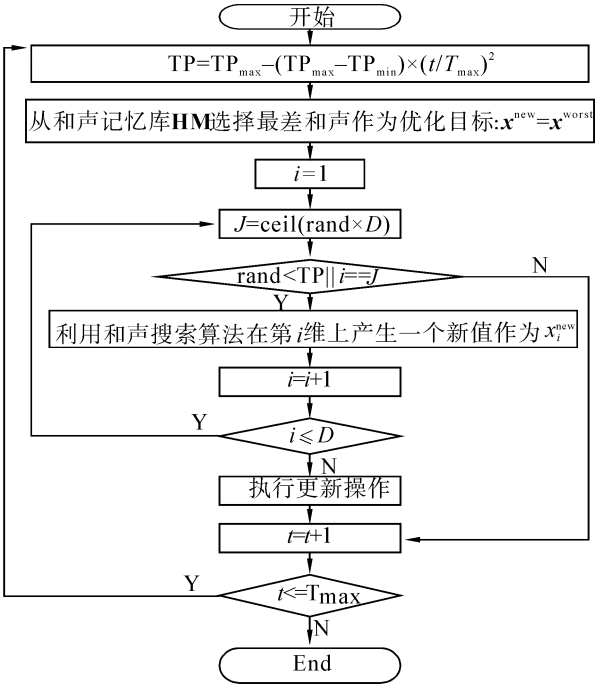


图 2 基于动态降维调整的和声搜索算法流程图

Fig.2 The flow chart of HS algorithm based on dynamic dimensionality reduction strategy

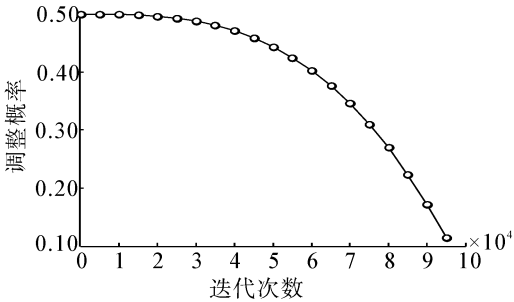


图 3 调整概率 TP 变化曲线

Fig.3 Adjustment curve of TP

3 仿真实验

为了评估本文算法提出的动态降维调整策略的性能,选取了 6 个复杂的 Benchmark 测试函数进行仿真测试^[35-38](见表 1),16 个函数除了 F_7 和 F_8 是单峰函数之外,其余函数都是复杂的高维多峰值函数。

分别将 HS、IHS、ITHS、EHS 和 GHS 利用本文算法思想进行改进,将其改进后分别称为 HS₂、IHS₂、ITHS₂、EHS₂ 和 GHS₂,并将其与改进前的算法进行比较。检验改进后的算法是否比改进前的算法能够获得更高精度的解,同时,检验其运行成本(运行时间)是否降低。

3.1 实验环境和算法参数设置

本文实验采用戴尔工作站 T7500 Inter(R) Xeon(R) CPU E560@ 2.1 GHz, 8.0 GB 内存,Windows Server2003 Server 操作系统,所有测试程序采用 Matlab R2009a 编写。10 种算法参数设置如表 2。

表 1 6 个 Benchmark 函数($F_1 \sim F_6$)

Table 1 Six Benchmark functions

函数名称	Benchmark 函数表达式	搜索范围	参数值
F_1 Ackley Function	$F_1(x) = -20e^{\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)} + 20 - e$	$[-15, 30]^D$	$x^* = (0, 0, \dots, 0)$ $F_1(x^*) = 0$
F_2 Griewank Function	$F_2 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D (\cos(\frac{x_i}{\sqrt{i}})) + 1$	$[-600, 600]^D$	$x^* = (0, 0, \dots, 0)$ $F_2(x^*) = 0$
F_3 Levy Function	$F_3(x) = \sin^2(\pi y_1) + \sum_{i=1}^{D-1} [(y_i - 1)^2 (1 + 10 \sin^2(\pi y_i + 1))] + (y_D - 1)2(1 + 10 \sin^2(2\pi y_D))$ $y_i = 1 + (x_i - 1)/4, i = 1, 2, \dots, D$	$[-10, 10]^D$	$x^* = (1, 1, \dots, 1)$ $F_3(x^*) = 0$
F_4 Michalewics Function	$F_4(x) = - \sum_{i=1}^D \sin(x_i) (\sin(ix_i^2/\pi))^{2m}; m = 10$	$[-10, \pi]^D$	$n = 5$ $F_4(x^*) = -4.687\ 658$
F_5 Rastrigin Function	$F_5(x) = 10D + \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i))$	$[-5.12, 5.12]^D$	$x^* = (0, 0, \dots, 0)$ $F_5(x^*) = 0$
F_6 Schwefel 2.26 Function	$F_6(x) = 418.982\ 9D + \sum_{i=1}^D x_i^2 \sin(\sqrt{ x_i })$	$[-512, 512]^D$	$x^* = (420.968\ 7, 420.968\ 7, \dots, 420.968\ 7)$ $F_6(x^*) = 0$

表 2 参数设置
Table 2 Parameters setting

算法	HMS HMCRCR			PAR	BW				TP			
HS	10	0.99			0.33	$(\mathbf{x}U-\mathbf{x}L)/2\ 000$				/		
HS ₂	10	0.99			0.33	$(\mathbf{x}U-\mathbf{x}L)/2\ 000$				TP _{max} = 0.6, TP _{min} = 5/ D		
IHS	10	0.99	PARmax = 0.99, PARmin = 0.1			BWmax = $(\mathbf{x}U-\mathbf{x}L)/20$ BWmin = $(\mathbf{x}U-\mathbf{x}L)/(1e+8)$				/		
IHS ₂	10	0.99	PARmax = 0.99, PARmin = 0.1			BWmax = $(\mathbf{x}U-\mathbf{x}L)/20$ BWmin = $(\mathbf{x}U-\mathbf{x}L)/(1e+8)$				TP _{max} = 0.6, TP _{min} = 5/ D		
GHS	10	0.90	PARmax = 0.99, PARmin = 0.01			/				/		
GHS ₂	10	0.90	PARmax = 0.99, PARmin = 0.01			/				TP _{max} = 0.6, TP _{min} = 5/ D		
EHS	50	0.99			0.33	/				/		
EHS ₂	50	0.99			0.33	/				TP _{max} = 0.6, TP _{min} = 5/ D		
ITHS	10	0.99	PARmax = 1, PARmin = 0			/				/		
ITHS ₂	10	0.99	PARmax = 1, PARmin = 0			/				TP _{min} = 0.6, TP _{min} = 5/ D		

表 3 在 D = 500 时, 5 种和声算法及其利用本文算法改进的 5 种和声算法对 6 个测试函数的运行结果比较
Table 3 Results comparison between five HS algorithms and five improved HS algorithms

算法	F ₁				F ₂				F ₃			
	Best	Mean	Std	Runtime	Best	Mean	Std	Runtime	Best	Mean	Std	Runtime
HS	2.34×10 ⁰	2.50×10 ⁰	6.67×10 ⁻²	2.52×10 ²	6.32×10 ⁻¹	6.75×10 ⁻¹	2.39×10 ⁻²	3.06×10 ²	8.04×10 ⁻²	9.70×10 ⁻²	1.80×10 ⁻²	1.61×10 ³
HS ₂	1.39×10 ⁻¹	1.61×10 ⁻¹	1.14×10 ⁻²	1.89×10 ²	4.20×10 ⁻¹	5.31×10 ⁻¹	8.25×10 ⁻²	2.48×10 ²	1.09×10 ⁻⁴	1.24×10 ⁻⁴	7.07E-06	1.53×10 ³
GHS	1.57×10 ¹	1.64×10 ¹	2.75×10 ⁻¹	2.67×10 ²	1.88×10 ³	2.11×10 ³	1.49×10 ²	3.45×10 ²	7.11×10 ²	7.96×10 ²	4.40×10 ¹	9.03×10 ²
GHS ₂	1.24×10 ⁻²	4.29×10 ⁻²	2.86×10 ⁻²	2.42×10 ²	9.85×10 ⁻³	2.83×10 ⁻¹	2.72×10 ⁻¹	3.06×10 ²	1.30×10 ⁻³	8.25×10 ⁻³	9.63×10 ⁻³	8.86×10 ²
IHS	6.57×10 ⁻¹	1.10×10 ⁰	1.74×10 ⁻¹	3.79×10 ²	5.92×10 ⁻²	7.60×10 ⁻²	8.85×10 ⁻³	4.64×10 ²	5.27×10 ⁻²	1.05×10 ⁻¹	4.09×10 ⁻²	1.24×10 ³
IHS ₂	8.71×10 ⁻⁷	8.81×10 ⁻⁷	6.81×10 ⁻⁹	3.16×10 ²	2.69×10 ⁻¹¹	3.70×10 ⁻⁴	1.65×10 ⁻³	4.00×10 ²	1.17×10 ⁻¹²	1.19×10 ⁻¹²	2.75×10 ⁻¹⁴	9.71×10 ²
ITHS	7.90×10 ⁻³	2.95×10 ⁻²	1.18×10 ⁻²	3.17×10 ²	6.41×10 ⁻³	1.92×10 ⁻¹	9.97×10 ⁻²	3.87×10 ²	1.35×10 ⁻³	4.75×10 ⁻³	2.16×10 ⁻³	1.70×10 ³
ITHS ₂	1.50×10 ⁻¹³	3.49×10 ⁻¹³	1.75×10 ⁻¹³	2.16×10 ²	6.66×10 ⁻¹⁶	7.72×10 ⁻¹⁶	5.67E-17	2.62×10 ²	1.50×10 ⁻³²	1.50×10 ⁻³²	0.00×10 ⁰	1.62×10 ³
EHS	4.55×10 ⁰	4.70×10 ⁰	8.66×10 ⁻²	5.02×10 ²	2.12×10 ¹	2.47×10 ¹	1.58×10 ⁰	5.75×10 ²	1.11×10 ²	1.26×10 ²	7.58×10 ⁰	2.04×10 ³
EHS ₂	2.71×10 ⁻¹³	2.21×10 ⁻⁵	6.85×10 ⁻⁵	4.32×10 ²	7.77×10 ⁻¹⁶	1.26×10 ⁻⁵	5.38×10 ⁻⁵	5.17×10 ²	3.54E-15	1.25×10 ⁻⁷	3.96×10 ⁻⁷	1.24×10 ³
算法	F ₄				F ₅				F ₆			
	Best	Mean	Std	Runtime	Best	Mean	Std	Runtime	Best	Mean	Std	Runtime
HS	-2.08×10 ²	-2.01×10 ²	4.92×10 ⁰	6.48×10 ²	1.10×10 ²	1.26×10 ²	1.07×10 ¹	3.21×10 ²	4.57×10 ¹	4.95×10 ¹	2.42×10 ⁰	2.98×10 ²
HS ₂	-4.28×10 ²	-4.26×10 ²	2.03×10 ⁰	5.25×10 ²	1.13×10 ⁻²	1.28×10 ⁻²	8.39×10 ⁻⁴	2.03×10 ²	6.99×10 ⁻²	8.04×10 ⁻²	6.73×10 ⁻³	2.15×10 ²
GHS	-2.26×10 ²	-2.22×10 ²	2.90×10 ⁰	5.22×10 ²	1.61×10 ³	1.90×10 ³	1.19×10 ²	3.50×10 ²	3.35×10 ⁴	3.81×10 ⁴	2.16×10 ³	3.47×10 ²
GHS ₂	-2.65×10 ²	-2.62×10 ²	1.55×10 ⁰	4.98×10 ²	8.81×10 ⁻⁴	1.10×10 ⁻¹	1.69×10 ⁻¹	3.14×10 ²	1.54×10 ⁻²	1.26×10 ⁰	1.51×10 ⁰	3.23×10 ²
IHS	-4.55×10 ²	-4.53×10 ²	2.43×10 ⁰	6.36×10 ²	1.02×10 ²	1.17×10 ²	5.45×10 ⁰	4.99×10 ²	6.66×10 ⁰	4.63×10 ¹	3.27×10 ¹	4.56×10 ²
IHS ₂	-4.98×10 ²	-4.97×10 ²	1.33×10 ⁻¹	5.83×10 ²	1.71×10 ⁻²	1.20×10 ⁻¹	1.05×10 ⁻¹	4.16×10 ²	1.37×10 ⁻⁹	1.47×10 ⁻⁹	6.97×10 ⁻¹¹	3.83×10 ²
ITHS	-4.51×10 ²	-4.42×10 ²	3.69×10 ⁰	5.96×10 ²	4.35×10 ⁻³	5.94×10 ⁻²	5.25×10 ⁻²	4.38×10 ²	6.79×10 ²	1.39×10 ³	5.00×10 ²	5.66×10 ²
ITHS ₂	-4.98×10 ²	-4.98×10 ²	3.11×10 ⁻¹	4.89×10 ²	1.18×10 ⁻¹¹	4.27×10 ⁻¹¹	3.71×10 ⁻¹¹	2.97×10 ²	1.16×10 ⁻⁹	1.39×10 ⁻⁹	2.85×10 ⁻¹⁰	3.98×10 ²
EHS	-1.63×10 ²	-1.58×10 ²	1.99×10 ⁰	8.64×10 ²	4.43×10 ³	4.52×10 ³	4.70×10 ¹	6.68×10 ²	1.27×10 ⁵	1.32×10 ⁵	1.78×10 ³	6.48×10 ²
EHS ₂	-4.43×10 ²	-4.41×10 ²	1.38×10 ⁰	8.10×10 ²	3.84×10 ²	4.52×10 ²	2.89×10 ¹	5.90×10 ²	2.36×10 ⁴	2.61×10 ⁴	1.33×10 ³	5.38×10 ²

3.2 实验结果与分析

为了保证算法测试的公平性,改进前的算法和改进后的算法取相同的初始和声记忆库 HM,每个算法中随机数设置 rand('twister', 5 489),每个算法程序独立运行 20 次,记录每次运行的过程,统计出 20

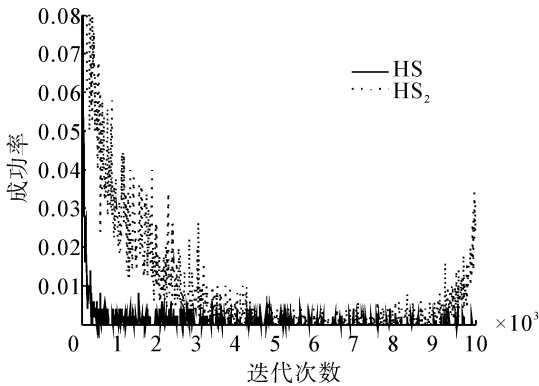
次运行的最优目标函数适应值的平均值 Mean,最优值 Best, 20 次最优解的标准差(Std)和平均运行时间(run time)。设置维数 D = 500,运行结果见表 3。表中将 HS、IHS、ITHS、EHS 和 GHS 分别与 HS₂、IHS₂、ITHS₂、EHS₂ 和 GHS₂ 进行比较,较好结果的用

粗体标出。

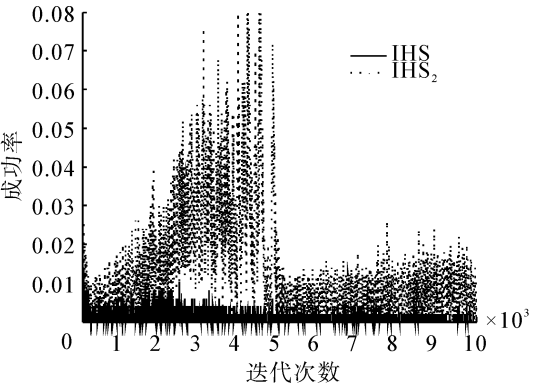
从表 3 可看出,对于高维多峰值优化函数(例如, F_3 : Levy, F_5 : Rastrigin, F_6 : Schwefel2.26),本文算法相比改进前的算法,更具有优势,算法的运行代价(运行时间)也相对较少。说明本文算法在求解高维多极值复杂优化问题时具有更好的性能优势。

3.3 算法更新成功率分析

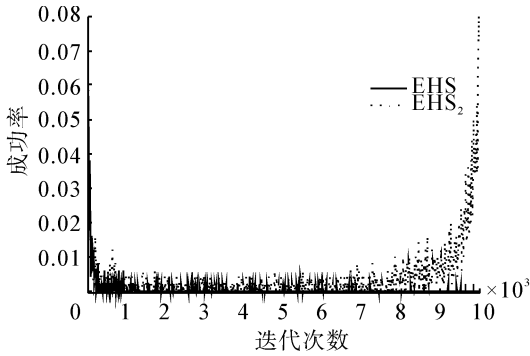
为了进一步分析本文提出的动态降维调整策略的性能,通过分析算法的更新成功率 SuccessRate 来判别算法的优劣性。更新成功率(SuccessRate)是指利用和声搜索算法产生的新解 \mathbf{X}^{new} 优于和声记忆库中最差解 $\mathbf{X}^{\text{worst}}$ 的概率,计算方法是记录最近 100 次更新操作成功的次数(SuccesTimes),让其除以总的比较次数 100 ($\text{SuccessRate} = \frac{\text{SuccessTimes}}{100}$)。实验中,对多峰值函数 Rastrigin 进行测试,设置函数的维数 $D = 1\,000$;将改进前的算法 HS、IHS、EHS 和 ITHS 算法分别与利用本文策略改进后的算法 HS_2 、 IHS_2 、 EHS_2 和 ITHS_2 进行比较。具体成功率曲线如图 4。



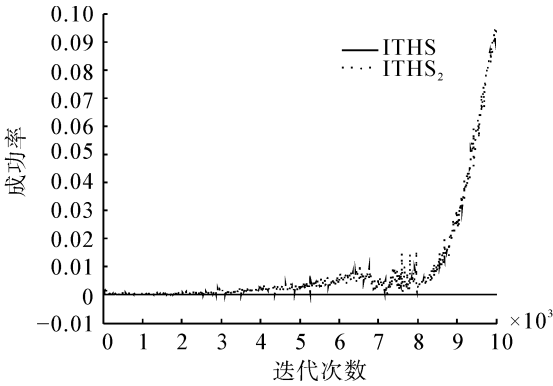
(a) 算法 HS 与 HS_2 成功率比较



(b) 算法 IHS 与 IHS_2 成功率比较



(c) 算法 EHS 与 EHS_2 成功率比较



(d) 算法 ITHS 与 ITHS_2 成功率比较

图 4 函数 Rastrigin 在 $D = 1\,000$ 时,改进前与改进后的算法成功率比较

Fig.4 Successrate comparison between ITHS and improved ITHS on $D = 1\,000$

由图 4 可以看出,本文策略改进后的算法成功率都明显高于改进前的算法,并且,改进后的算法成功率曲线成凹形曲线变化。这是由于初始时,种群是随机产生,个体的适应值较差,容易探索到比当前更好的解。随着迭代的进行,成功率逐渐降低。对改进前的算法来说,由于一个新解完全是靠组合算子和微调策略产生,成功率会越来越低(根据第 3 节的分析),而本文采用动态降维调整策略逐步减小最差解 $\mathbf{x}^{\text{worst}}$ 中决策变量的调整概率,从而增加了更新成功率。从图 4 中可以看出,在后期,算法的成功率不降反升,证明了本文改进策略的有效性。这是由于在后期,只对最差解向量 $\mathbf{x}^{\text{worst}}$ 中很少的几个决策变量进行调整,获得成功的机会远高于在所有维上的更新调整。

3.4 算法种群多样性分析

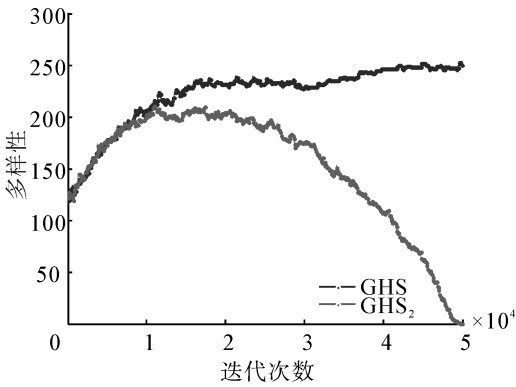
种群的多样性是指种群中个体间的差异性,个体差异越大,种群多样性越高,反之,差异性越小,种群多样性越低。本文采用如下公式计算种群的多样性。

$$\text{Diversity} = \frac{1}{D} \sum_{i=1}^D \sqrt{\frac{1}{\text{HMS}} \sum_{j=1}^{\text{HMS}} (x_i^j - \bar{x}_i)^2}$$
 , \bar{x}_i 是和

声记忆库 **HM** 中第 i 维的平均值。

对于群智能优化算法来说,种群的多样性直接决定算法的搜索能力,当具有较高的种群多样性时,算法的全局探索能力较强,适合探索新的搜索区域,但是,如果一直保持较高的种群多样性,种群很难向全局最优解靠近,往往难以获得高精度的全局最优解。所以,在搜索初期,需要种群具有较高的种群多样性,后期,为了获得高精度的全局最优解,种群需要向最优解聚集,多样性逐步降低。

和声搜索算法具有较强的全局探索能力,但求解精度较低^[19-20],主要是在进化后期,算法的局部求解能力较差。本文通过对多峰值函数 Schwefel2.26 进行测试(设置函数的维数 $D = 1\,000$),比较改进后的与改进前算法中种群多样性的变化(如图 5)。



(d)算法 GHS 与 GHS₂ 的种群多样性比较

图 5 函数 Schwefel2.26 的多样性曲线($D = 1\,000$)

Fig.5 diversity curve of function Schwefel2.26 ($D = 1\,000$)

图 5 可以看出,改进后算法种群多样性变化明显,在搜索初期,多样性较高,有助于进行全局探索,随着搜索的进行,当种群逐渐聚集到全局最优解附近区域时,开始进行局部高精度求解,种群的多样性迅速降低。

4 结束语

本文提出用一种新颖的维度动态调整策略改进和声搜索算法,使其通过对和声记忆库中最差和声向量进行调整,在优化初期,采用大范围、广维度调整策略保证了种群的多样性,增强了算法的全局探索能力;随着优化的进行,逐步降低调整维数,慢慢变为在部分维上进行调整,从而增强算法的局部开发能力,提高其求解精度。这样,和声搜索算法有效地在全局探索和局部开发之间实现了平衡。通过对 6 个高维复杂多极值测试函数进行实验,发现本文算法在求解精度和运算成本上都有了明显改进,并且,随着维数的增加,本文算法的优势更加显著,说明本文改进策略可用于大规模高维复杂问题的求解。

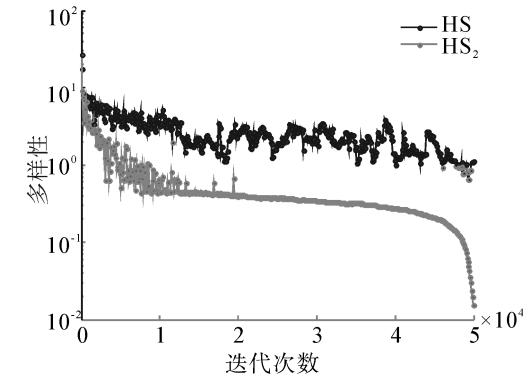
参考文献:

[1] GOLDBERG D E. Genetic algorithms in search optimization and machine learning[M]. Boston: Addison-Wesley, 1989: 25-30.

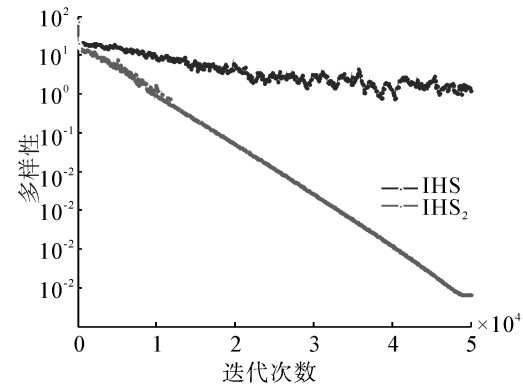
[2] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan, 1995: 23-30.

[3] STORN R, PRICE K V. Minimizing the real functions of the ICEC 1996 contest by differential evolution[C]//Proc IEEE Int Conf Evol Comput. Nagoya, Japan, 1996: 842-844.

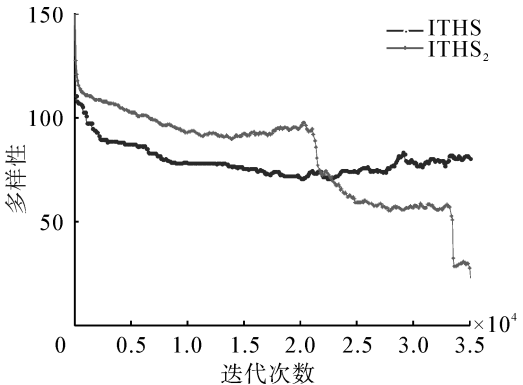
[4] DORIGO M, MANIEZZO V, COLORNI A. The ant system;



(a)算法 HS 与 HS₂ 的种群多样性比较



(b)算法 IHS 与 IHS₂ 的种群多样性比较



(c)算法 ITHS 与 ITHS₂ 的种群多样性比较

- optimization by a colony of cooperating agents[J]. IEEE Trans Syst Man Cybern, 1996, 26(1): 29-41.
- [5] KARABOGA D, BASTURK B. On the performance of artificial bee colony (ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697.
- [6] GEEM Z W, KIM J H, LOGANATHAN G V. A new heuristic optimization algorithm: harmony search[J]. Simulation, 2001, 76: 60-70.
- [7] SIMON D. Biogeography-based optimization[J]. IEEE Transactions on Evolutionary Computation, 2008, 12: 702-713.
- [8] RAO R V, SAVSANI V J, VAKHARIA D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. Computer-Aided Design, 2011, 43: 303-315.
- [9] 拓守恒, 雍龙泉, 邓方安. “教与学”优化算法研究综述[J]. 计算机应用研究, 2013(7): 1933-1938.
- TUO Shouheng, YONG Longquan, DENG Fang'an. Survey of teaching-learning-based optimization algorithms[J]. Application Research of Computers, 2013(7): 1933-1938.
- [10] GEEM Z W, KIM J H, LOGANATHAN G V. Harmony search optimization: application to pipe network design[J]. International Journal of Modelling & Simulation, 2002, 22(2): 125-133.
- [11] 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.
- [12] GEEM Z W, LEE K S, PARK Y. Application of harmony search to vehicle routing[J]. American Journal of Applied Sciences, 2005, 2(12): 1552.
- [13] VASEBI A, FESANGHARY M, BATHAEE S M T. Combined heat and power economic dispatch by harmony search algorithm[J]. International Journal of Electrical Power & Energy Systems, 2007, 29(10): 713-719.
- [14] WANG H, YUAN X, WANG Y, et al. Harmony search algorithm-based fuzzy-PID controller for electronic throttle valve[J]. Neural Computing and Applications, 2013, 22(2): 329-336.
- [15] PAN Q K, SUGANTHAN P N, LIANG J J, et al. A local-best harmony search algorithm with dynamic subpopulations[J]. Engineering Optimization, 2010, 42(2): 101-117.
- [16] 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.
- [17] YADAV P, KUMAR R, PANDA S K, et al. An intelligent tuned harmony search algorithm for optimization[J]. Information Sciences, 2012, 196: 47-72.
- [18] 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.
- [19] OMRAN M G H, MAHDAVI M. Global-best harmony search[J]. Applied Mathematics and Computation, 2008, 198(2): 643-656.
- [20] ZOU Dexuan, GAO Liqun, WU Jianhua, et al. Novel global harmony search algorithm for unconstrained problems[J]. Neurocomputing, 2010, 73: 3308-3318.
- [21] ZOU Dexuan, GAO Liqun, WU Jianhua, et al. A novel global harmony search algorithm for reliability problems[J]. Computers & Industrial Engineering, 2010, 58(2): 307-316.
- [22] ZOU Dexuan, GAO Liqun, LI S, et al. Solving 0-1 knapsack problem by a novel global harmony search algorithm[J]. Applied Soft Computing, 2011, 11: 1556-1564.
- [23] 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.
- [24] TUO Shouheng, YONG Longquan. An improved harmony search algorithm with chaos[J]. Journal of Computational Information Systems, 2012, 8(10): 4269-4276.
- [25] TUO Shouheng, YONG Longquan, ZHOU Tao. An improved harmony search based on teaching-learning strategy for unconstrained optimization problems[J]. Mathematical Problems in Engineering, 2013, 19: 69-76.
- [26] SARVARI H, ZAMANIFAR K. Improvement of harmony search algorithm by using statistical analysis[J]. Artificial Intelligence Review, 2012, 37(3): 181-215.
- [27] ASKARZADEH A, REZAZADEH A. A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell[J]. International Journal of Hydrogen Energy, 2011, 36(8): 5047-5053.
- [28] GHOSH S, KUNDU D, SURESH K, et al. Design of optimal digital IIR filters by using a bandwidth adaptive harmony search algorithm[C]//2009 World Congress on Nature & Biologically Inspired Computing. Coimbatore, India, 2009: 480-485.
- [29] HOANG D C, YADAV P, KUMAR R, et al. A robust harmony search algorithm based clustering protocol for wireless sensor networks[C]//2010 IEEE International Conference on Communications Workshops (ICC). [S.l.], 2010: 1-5.
- [30] JABERIPOUR M, KHORRAM E. Solving the sum-of-ratios problems by a harmony search algorithm[J]. Journal of Computational and Applied Mathematics, 2010, 234(3): 733-742.
- [31] POURSHA M, KHOSHNOUDIAN F, MOGHADAM A S.

Harmony search based algorithms for the optimum cost design of reinforced concrete cantilever retaining walls[J]. International Journal of Civil Engineering, 2011, 9(1): 1-8.

[32] KHAZALI A H, KALANTAR M. Optimal reactive power dispatch based on harmony search algorithm [J]. International Journal of Electrical Power & Energy Systems, 2011, 33(3): 684-692.

[33] KHAZALI A H, PARIZAD A, KALANTAR M. Optimal voltage/reactive control by an improve harmony search algorithm[J]. Int Rev Electr Eng, 2010, 5: 217-224.

[34] KHORRAM E, JABERIPOUR M. Harmony search algorithm for solving combined heat and power economic dispatch problems[J]. Energy Conversion and Management, 2011, 52(2): 1550-1554.

[35] FUKUSHIMA M. Test functions for unconstrained global optimization [OL/EB]. [2014-01-12]. <http://www.optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar-files/TestGO-files/Page364.htm>.

[36] TANG K, YAO X, SUGANTHAN P N, et al. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization[OL/EB]. [2013-10-21]. <http://www.ntu.edu.sg/home/EPNSugan/>, 2008.

[37] TANG K, LI X, SUGANTHAN PN, et al. Benchmark functions for the CEC'2010 special session and competition on large scale global optimization[R]. Nature Inspired

Computation and Applications Laboratory, USTC, China & Nanyang Technological University, 2009.

[38] HERRERA F, LOZANO M, MOLINA D. Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other meta-heuristics for large scale continuous optimization problems [OL/EB]. [2013-10-21]. <http://sci2s.ugr.es/eamhco/CFP.php>.

作者简介:



拓守恒,男,1978 年生,副教授,博士研究生,CCF 会员,主要研究方向为智能优化算法、生物信息分析与处理,发表学术论文多篇。



雍龙泉,男,1980 年生,副教授,博士,主要研究方向为优化理论与算法设计、智能优化算法等。



邓方安,男,1963 生,教授,博士,主要研究方向为代数系统、粗糙集理论和优化理论。