

DOI: 10.7652/xjtuxb201302006

自适应折叠混沌优化方法

傅文渊^{1,2}, 凌朝东^{1,2}

(1. 华侨大学信息科学与工程学院, 361002, 福建厦门; 2. 厦门市专用电路系统重点实验室, 361008, 福建厦门)

摘要: 针对传统混沌优化方法中优化结果对搜索初始值要求极高以及搜索效率较低的问题, 提出一种自适应折叠混沌优化方法。该方法首先提出一种新型无限折叠混沌映射, 并证明了该映射无有理数不动点; 根据映射关系式建立混沌模型求解 Lyapunov 指数, 并基于该混沌模型对搜索初值采用大幅度改变和小幅度改变两种方式考察映射对初值的依赖程度。采用所提映射取代传统的有限折叠映射作为混沌产生器, 提高了混沌的动态优化性能, 通过在优化过程中不断缩小优化变量的搜索空间来提高搜索效率和精度。实验结果表明, 该方法的优化结果不依赖于初始值位置, 具有搜索效率高的特点。与 Logistic 映射和 Tent 映射优化方法相比, 平均搜索效率分别提高了 71.6% 和 62.6%。

关键词: 混沌优化; 混沌映射; 折叠; 自适应

中图分类号: TP18 **文献标志码:** A **文章编号:** 0253-987X(2013)02-0033-06

An Adaptive Iterative Chaos Optimization Method

FU Wenyuan, LING Chaodong

(1. College of Information Science and Engineering, Huaqiao University, Xiamen, Fujian 361002, China; 2. Xiamen Key Laboratory of Application-Specific Integrated Circuit System, Xiamen, Fujian 361008, China)

Abstract: A new adaptive iterative chaos optimization method is proposed to improve the problems that the optimal results generated from the existing chaotic optimization methods rely on initial points and that search efficiency of these methods is lower. It is proved that the chaotic map has no rational number fixed point, then the mapping relational formula is used to establish a chaotic model that is used to solve the Lyapunov exponent, and the sensitivity of chaotic maps to initial values is investigated under large variation and small variation on initial starting points. The chaotic map is then used to establish chaotic generator to replace the finite-collapse map, and to improve the dynamic performance of chaotic optimization. The method improves the search efficiency by continuously reducing the searching space of variables and enhancing search precision. Numerical results show that the optimal results generated by the proposed method do not depend on the initial value, and the search efficiency is high. Comparisons with the Logistic mapping and the Tent mapping optimization method show that the average search efficiency of the proposed method improves about 71.6% and 62.6%, respectively.

Keywords: chaos optimization; chaotic map; iterative; adaptive

混沌是一种复杂的非线性系统, 将混沌思想应用于全局最优问题, 核心是把混沌变量进行优化搜

收稿日期: 2012-05-08。 作者简介: 傅文渊(1983—), 男, 讲师。 基金项目: 国家自然科学基金资助项目(60772164); 福建省科技计划资助项目(2011H6018); 华侨大学科研基金资助项目(11HZR03)。

网络出版时间: 2012-10-31 网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20121031.1056.001.html>

索,利用混沌的遍历性、不可预测性以及不确定性完成全局最优值的搜索。由于混沌优化比模拟退火算法搜索速度更快,比遗传算法具有更好的全局寻优能力,因此已逐步成为国内外研究的热点问题^[1-2]。然而,传统的研究方法存在对搜索初始值要求高、搜索效率不高、遍历区间过大等问题。为此国内外学者提出许多改进方法,如文献[3]将混沌变量与优化变量的区间联系,以克服搜索区间过小的缺点,设置内外2个循环,内循环进行混沌搜索,外循环进行区间缩小,但该方法需要设置一个较优值次数的变量,如果该变量设置不当会使算法过早收敛而无法得到全局最优解;文献[4]将混沌与模式搜索算法结合,由混沌的遍历性来生成搜索方向和步长,同时采取二级模式移动策略进行全局寻优,但此方法在搜索区间较大时,搜索速度较慢,搜索效率不高;文献[5]通过遗传算法与混沌相结合,提出一种基于猫映射的混沌遗传算法,克服了 Tent 映射容易陷入局部循环,提高了全局最优解的搜索效率,但该方法实现起来较为复杂,在算法运行阶段容易陷入不动点。

传统的混沌优化方法对搜索初始值要求高,必须选择合适,否则很难得到全局最优解,同时优化问题的遍历区间过大、优化参数过多以及混沌控制参数选取不合理也会对搜索结果产生较大的影响。基于此,本文提出一种新型的无限折叠映射和基于此映射的自适应全局优化方法,用该映射取代传统的有限折叠映射作为混沌产生器,提高了混沌的动态优化性能,通过在优化过程中不断地缩小优化变量的搜索空间来提高搜索效率和搜索精度。理论分析和仿真实验表明,本文算法的优化结果不依赖于初始值,可以显著减少函数的运行次数,提高求解全局最优问题的效率。

1 新型一维无限折叠混沌映射

1.1 Fuch 映射不动点特性

为便于分析,将本文提出的一维无限折叠混沌映射记为 Fuch 映射,其表达式为 $x_{n+1} = \cos(1/x_n^2)$,其中 $x_n \neq 0, n \in \mathbf{Z}^+$ 。

考察此映射的不动点,即求解超越方程 $x = \cos(1/x^2)$,由于 $-1 \leq \cos(1/x^2) \leq 1$,因此 $-1 \leq x \leq 1$,假设 $\cos(1/x^2)$ 是代数函数,则存在二元多项式 $M(x, \cos(1/x^2))$ 满足以下等式,且 $M_1(x) \neq 0$

$$M\left(x, \cos\left(\frac{1}{x^2}\right)\right) = \sum_{i=1}^{n+1} M_i(x) \cos^{n-i+1}\left(\frac{1}{x^2}\right) \equiv 0 \quad (1)$$

当 $x = \pm(2/k\pi)^{1/2}$ 时, k 为正整数,则有 $M_n(\pm(2/k\pi)^{1/2}) = 0$,由代数基本定理得到 $M_n(x) \equiv 0$ 。当 $x \neq \pm(2/k\pi)^{1/2}$ 时, $\sum_{i=1}^n M_i(x) \cos^{n-i}(1/x^2) \equiv 0$,根据多项式的连续性以及 $\cos(1/x^2)$ 的连续性,可以得出 $M_{n-1}(\pm(2/k\pi)^{1/2}) = 0$,因此 $M_{n-1}(x) = 0$,反复对式(1)进行同样的推演,得出 $M_1(x) = 0$,这与假设矛盾,因此 $\cos(1/x^2)$ 是超越函数。令 $t = 1/x^2$,则 $\cos(t) = \pm t^{-1/2}$,根据泰勒级数展开得到

$$\sum_{k=1}^{\infty} \frac{t^{2k-2} (-1)^{k-1}}{(2k-2)!} = \pm (t)^{-1/2} \quad (2)$$

由式(2)得 k 趋于无穷大,因此 t 必定是超越数, x 也为超越数。映射的不动点为超越数,若该映射的初始数值不为 0,则由该映射产生的混沌不收敛于有理数不动点。这表明, Fuch 映射在初始值不为 0 的情形下均能产生混沌。对于传统的 Logistic 映射和 Tent 映射,由于其自身映射存在有理数不动点,如果在算法初始阶段数据落在不动点周期内,则产生的序列趋于稳定状态,算法失效。因此,为避免传统映射产生的序列落入不动点周期内,通常需要限制初始状态的取值范围和设计相应的预防算法避免平衡态的产生,这将增加设计难度和降低算法执行效率。

1.2 Fuch 映射 Lyapunov 指数

Lyapunov 指数(λ)是衡量混沌程度的重要特征参量,描述 2 个初始状态所产生的轨道分离程度,同时表征混沌系统随时间变化而改变的程度。其数值以 0 作为分界线:若小于 0,动力系统的相体积减小,运动趋于稳定;若大于 0,动力系统的相对体积增大,运动趋于混沌;而 Lyapunov 指数等于 0,则表征动力系统处于稳定和混沌边界。

提出的混沌映射关系式为 $x_{n+1} = f(x_n) = \cos(1/x_n^2)$,由于混沌系统轨道分离的临界点为 0,因此 df/dx 表征状态产生的轨道分离程度。设置初始状态为 x_0 ,每次混沌迭代过程中轨道分离指数为 λ ,迭代一次距离为 k ,则混沌运行 n 次后,轨道分离的距离为 $f_n(x_0 + k) - f(x_0) = ke^\lambda$,代入混沌映射关系式,并令 k 趋近于无穷小, n 趋近于无穷大,则得到

$$e^\lambda = \left| \frac{f_n(x_0 + k) - f(x_0)}{k} \right| = \lim_{n \rightarrow \infty} \left| \frac{d^n(x)}{dx} \right| \quad (3)$$

化简式(3)得到

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{df}{dx} \right|_{x=x_i} =$$

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{2}{x_n^3} \sin \left(\frac{1}{x_n^2} \right) \right| =$$
$$\lim_{n \rightarrow \infty} \frac{1}{n} \ln \left(\prod_{i=0}^{n-1} \left| \frac{2}{x_n^3} \sin \left(\frac{1}{x_n^2} \right) \right| \right) =$$
$$\lim_{n \rightarrow \infty} \ln \left(\prod_{i=0}^{n-1} \left| \frac{2e^{1/n}}{x_n^3} \sin \left(\frac{1}{x_n^2} \right) \right| \right) \quad (4)$$

为便于比较,表 1 分别列举了文献[6]、Tent 映射、Logistic 映射和 Chebyshev 映射的 Lyapunov 指数。

表 1 5 种混沌映射的 Lyapunov 指数比较

指数	Fuch 映射	Tent 映射	文献[6] 映射	Logistic 映射	Chebyshev 映射
λ	2.165	0.693 1	1.669 5	0.694 1	1.386 7

从表 1 所示的 Lyapunov 指数对比可以看出,本文提出的 Fuch 映射具有更强的混沌特性。给定微小的初始值变化,混沌映射产生的输出是完全不同的,并且系统输出毫无规律。考察遍历整个系统解空间的混沌迭代次数时,本文提出的混沌映射与文献[6]以及传统有限折叠混沌映射相比具有更小的迭代次数,能更好地实现混沌寻优。

1.3 Fuch 映射初值分析

为了考察提出的 Fuch 映射对初值的依赖程度,本节采用 2 种不同方式来讨论该映射的初值特性。首先,对混沌映射的初值进行大幅度的改变,考察混沌初值改变时,系统输出状态变化程度,同时也表征基于混沌映射的优化算法对初值的依赖程度。其次,对混沌映射的初值进行微小幅度改变,考察混沌对初值微小变化的敏感程度。

图 1 和图 2 分别为 Logistic 映射和 Tent 映射在 2 个不同初值情形下迭代 2×10^4 次得到的数据分布图,混沌映射的 2 个初值相差 101 倍,满足数据大幅度变化的要求。

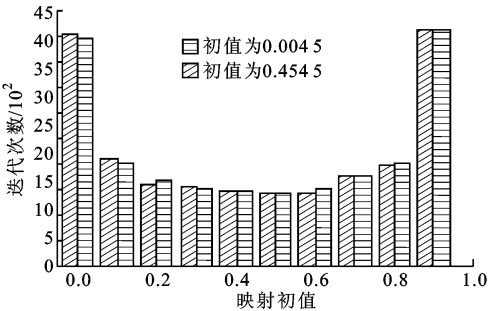


图 1 Logistic 映射初值变化的数据分布图

由图 1 可知,Logistic 映射 2 个初始状态对应

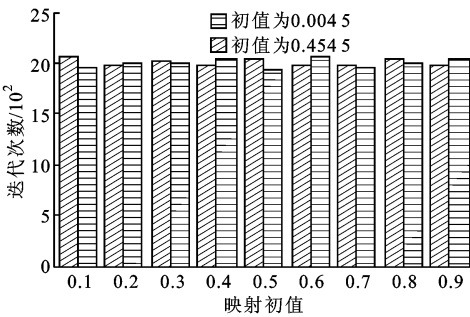


图 2 Tent 映射初值变化的数据分布图

的混沌映射迭代区间最大变化为 193 次。当系统最优解落在区间中间时,则需要迭代多次才能搜索到全局最优解。Tent 映射的分布较为均匀,但在区间两端数据占有的几率较大,2 个初始状态对应的混沌映射迭代区间最大变化为 137 次,均集中于边界处。当系统最优解落在区间两端时,则需要迭代多次才能搜索到全局最优解,这极大降低了混沌优化效率。

图 3 为提出的 Fuch 映射在图 1 和图 2 相同条件的初值下迭代 2×10^4 次得到的数据分布图。从图中可以看出,该映射相比于 Logistic 映射和 Tent 映射在解空间内具有更加均衡的遍历性,2 个初始状态对应的混沌映射迭代区间最大变化仅为 52 次,远小于 Logistic 映射和 Tent 映射的变化次数,并且在各个区间内变化的平均次数为 73 次。此映射对初始状态的位置设置不敏感,能够降低搜索初值的选择难度。

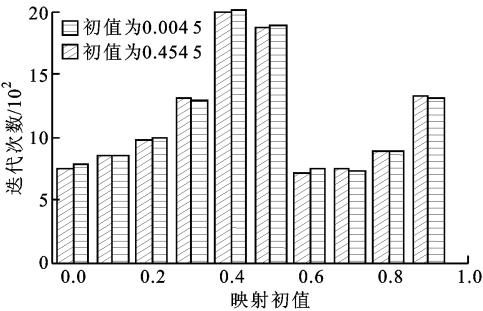


图 3 Fuch 映射初值变化的数据分布图

对混沌系统的初值进行微小变化,将导致系统产生巨大的变化。考察混沌系统的初值敏感性,采用位变化率来衡量^[7]。位变化率的定义:将初值进行微小变化(变化幅度不超过 10^{-6})后生成的混沌序列与原始混沌序列相比,其数值发生改变的位个数和原始混沌序列位个数的比值。

从表 2 得到,本文提出的 Fuch 映射在初值变化幅度为 10^{-6} 时,混沌系统的最大位变化率超过 53%。与之相比的 Tent 映射的位变化率最大不超过 34.5%, Logistic 映射的位变化率最大不超过 37.5%。同时,该映射最小位变化率均大于 Logistic 映射和 Tent 映射,因此本文提出的映射具有更加优良的初值敏感性能。

表 2 各种混沌映射的初值敏感性分析

初值	变化幅度	Logistic 映射 变化率/%	Tent 映射 变化率/%	Fuch 映射 变化率/%
0.13	10^{-6}	25.00	28.13	46.88
0.15	10^{-6}	25.00	34.37	37.50
0.18	10^{-6}	28.13	21.88	43.50
0.20	10^{-6}	25.00	28.13	34.38
0.23	10^{-6}	37.5	34.38	28.13
0.27	10^{-6}	31.25	34.37	34.38
0.30	10^{-6}	21.88	25.00	53.13

2 混沌优化方法

具有简单边界约束条件的最优化问题为 $G = \min f(x_i), i=1, 2, 3, \dots, n, x_i \in [p_i, q_i]$, 其中 f 为待优化函数, x_i 为优化自变量, n 代表变量个数, p_i 和 q_i 分别为待优化变量的上限和下限。用 β_i 表示微小扰动因子, $\dot{a}$ 表示第 1 次混沌搜索中逼近最优解的控制因子, $\bar{e}$ 为自适应因子。混沌优化方法对于 G 的优化的基本步骤如下。

(1) 算法初始化。设置初始数值 $x_i(0), i=1, 2, 3, \dots, n$, 其中每个初值各不相同, 同时设置算法终止参数 $s=s_0$, 令优化函数的极小值和极大值分别为 $f_{\min}=f[x_i(0)], f_{\max}=f[x_i(0)]$ 。

(2) 利用 Fuch 折叠映射建立优化函数区间和折叠映射区间的对应关系, 生成相应的混沌扰动变量 x_s , 具体对应关系为 $x_s \rightarrow (p_i + q_i)/2 + x_i(q_i - p_i)/2$ 。

(3) 将混沌扰动变量 x_s 进行 k 次搜索, 若小于预设的极小值, 则将搜索变量更新为 $x_{\min} + \dot{a}(x_{\min} - 0.01)(q_i - p_i)$, 否则将函数值变更为预设极大值。重复步骤 2 和步骤 3, 设此时迭代次数到达 k 次, 转入步骤 4。

(4) 对 $x_{\min}$ 进行 k_2 次微小扰动。在 $x_{\min}$ 的邻域内进行混沌扰动, $x'_i = x_{\min} + \beta_i(k_1 + 1)$, 如果 $f(x'_i) < f_{\min}$, 则 $x_{\min} = x'_i, f_{\min} = f(x'_i)$, 否则, 更新预设极大值, 当迭代次数超过 k_2 时, 转入步骤 5。

(5) 对扰动后的数据进行二次优化, 以 $x_{\min}$ 为基

准, 将 q_i 和 p_i 的取值变更为下列递归表达式, $p_{i+1} = x_{\min} - \bar{e}(q_i - p_i), q_{i+1} = x_{\min} + \bar{e}(q_i - p_i)$ 。为确保 p_{i+1} 和 q_{i+1} 的取值范围不超过 p_i, q_i , 做如下处理: 如果 $p_{i+1} < p_i$, 则 $p_{i+1} = p_i$; 如果 $q_{i+1} > q_i$, 则 $q_{i+1} = q_i$ 。

(6) 以 $x_{\min}$ 作为新的混沌变量, 重复步骤 2, 产生混沌数值映射于 $[p_{i+1}, q_{i+1}]$ 内, 具体的映射关系满足, $x_s \rightarrow (p_i + q_i)/2 + x_{\min}(q_i - p_i)/2$, 重复步骤 3 ~ 步骤 5, 直到一定次数后 $f_{\min}$ 保持不变为止。然后继续下列步骤。

(7) 减小 $\bar{e}$ 的数值, 重复步骤 5 ~ 步骤 6。

(8) 重复步骤 7, 达到算法的迭代终止次数 s_0 , 结束算法。此时的 $x_{\min}$ 为算法得到的最优变量, $f_{\min}$ 为全局最优解。

混沌系统轨道分离的临界点为 0, 在 Lyapunov 指数小于 0 的区域, 轨道分离随时间增大 2 个初始状态汇合为一点, 成为稳定的不动点和周期点, 此时系统的混沌状态将退化为稳定状态。本文提出的 Fuch 映射相比于传统的 Logistic 映射、Tent 映射等具有更大的 Lyapunov 指数, 在指数大于 0 的区域, 随时间增大, 算法的初始状态产生的轨道分离程度增大, 在所有可能的取值状态空间中轨道分离的平均指数率增大。可以得出, 在时间相同的情形下, 本文提出的映射具有更好的遍历特性。采用本文映射作为混沌产生器, 在执行过程中为达到某一状态, 本文方法能迅速搜寻到该状态, 从而有效缩短搜索时间和提高搜索速度。

由于无法得到确切概率密度函数的映射, 因此从概率密度分布图来分析初值变化对算法的影响。考察本文方法在初值相差较大时的敏感度, 从图 2 的概率密度分布图得出, 初始状态对应的混沌映射迭代区间最大变化次数极小, 在解空间内具有更加均衡的遍历性进而改变优化算法初值敏感程度, 降低初值的选择难度, 同时提高算法的适用性。

本文优化算法中, β_i 的取值范围在 $(0, 0.5)$; $\dot{a}$ 一般取值为 $0.02 \sim 0.30$, $\dot{a}$ 越小, 则 $k_1 + 1$ 逼近最优解概率越大, 因而能显著提高搜索全局最优解的速度; $\bar{e} = 1 - (1 - 1/s_0)^r$, 其中 r 为正整数, 由 $\bar{e}$ 的定义可知, $\bar{e}$ 随 s_0 的增加而减小, $[p_{i+1}, q_{i+1}]$ 不断减小, 使得寻优过程不断逼近全局最优解。

3 算例验证

为了验证本文优化方法的效率和适应性, 采用文献[8]、文献[9]和文献[11]提出的 4 个经典测试

函数进行测试。同时考虑算法的适用性,利用计算机分别对低维函数和高维函数进行模型实验。

$$F_1=100(x_1^2-x_2)^2+(1-x_1)^2-2.048 < x_i < 2.048$$

$$F_2=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)][30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)], \quad -2 < x_i < 2$$

$$F_3=0.01\sum_{i=1}^4(x_i^4-16x_i^2+5x_i), \quad -10 < x_i < 10$$

函数 F_1 是一种特殊的鞍形函数,搜索接近全局最优点处函数急剧变化,极难寻找到全局极小值。 F_2 是多峰函数,有且仅有一个全局极小点。 F_3 为高维函数,该函数整体形状呈现 U 形,顶部陡峭,而在底部则非常平坦。对 F_1 、 F_2 和 F_3 来说,优化过程中极易陷入局部极值,因此算法需具备极强的全局寻优能力。为了比较本文提出的方法与文献[10]算法的性能,采用文献[10]相同的仿真设置,对 F_1 迭代 500 次,对 F_2 迭代 1 000 次,对 F_3 迭代 5 000 次进行仿真测试。传统的混沌优化对搜索初始值要求高,必须选择合适的初始值否则很难搜索到全局

最优解。文献[9]的优化思想是将最速下降法与混沌结合,而文献[12]则把模式搜索法与混沌结合得到混合优化算法。表 3 为本文方法与文献[10-12]方法的仿真结果比较,列举了不同初始搜索数值对应的优化方法寻优结果。

优化的结果表明,本文提出的优化方法能完整地搜索到函数的全局最优解。优化函数 F_1 的初始值为(2,0)时,4 种优化方法中,文献[12]、文献[11]和本文方法均能够寻找得到全局最优解,而在初始值为(-1,-2)、(10,20)和(-1,100)时,只有本文方法能搜索到全局最优解。当搜索初值超过优化函数遍历区间时,文献[10-12]的优化结果不甚理想,优化结果与函数初始值选择密切相关。一方面如果初始值选择不恰当,例如初始值大于或者小于遍历区间内的数值,则函数寻优将很难得到全局最优解。另一方面,对高维数函数优化时,例如由 F_3 的优化结果知,高维函数极易落入局部极值点,而一旦落入局部最优点则很难跳出,无法得到全局最优解。本文的混沌自适应算法较好的解决了这两方面问题。

表 4 为各种混沌映射对应的函数优化结果,将

表 3 各种混沌优化算法比较

优化函数	初始点	F(x)				
		理论最优	文献[9]	文献[11]	文献[10]	本文方法
F ₁	(-1,-2)	0	0.647 4	0.931 1	0.003 4	0
	(2,0)	0	0.164 9	0	0	0
	(10,20)	0	18.169 2	16	11.30	0
	(-1,100)	0	565.913 5	110.25	52	0
F ₂	(-1,-2)	3.000 0	26	3	3	3.000 0
	(2,0)	3.000 0	84	84.003	3.001	3.000 0
	(10,20)	3.000 0	2.843 2×10 ⁷	84.001	42	3.000 0
	(-1,100)	3.000 0	4.530 2×10 ¹³	12.789 3	22.25	3.000 0
F ₃	(1,1,1,1)	-3.133 3	-2.003 0	-1.985 4	-2.274 1	-3.1332 9
	(-1,-1,-1,-1)	-3.133 3	-3.133 0	-2.466 6	-3.133 3	-3.1332 9
	(3,-2,10,-10)	-3.133 3	-1.453 3	-2.567 8	-2.520 0	-3.1332 9
	(11,11,11,1)	-3.133 3	-2.012 1	-2.002 4	-2.852 2	-3.1332 9

表 4 各种混沌映射对应的优化结果

优化函数	F(x)优化结果			平均步数		
	基于 Logistic 映射	基于 Tent 映射	Fuch 映射	基于 Logistic 映射	基于 Tent 映射	Fuch 映射
F ₁	0.002 5	0.002 0	1.169×10 ⁻⁸	878	477	1 048
F ₂	3.006 2	3.001 0	3.000 0	1 320	1 315	877
F ₃	-3.133 3	-3.133 3	-3.133 3	8 763	6 544	1 192

文中 1.1 节产生的混沌映射和第 2 节的自适应优化方法配合,得到优化结果以及算法运行的平均步数。为了比较本文方法,还列举了基于 Logistic 映射与基于 Tent 映射的优化结果。由表 4 得出,本文方法能很好地搜索到函数全局最优值,同时函数运行的平均步数是 3 种混沌映射中最少的,这表明了本文方法的高效性。与 Logistic 映射和 Tent 映射优化方法相比,本文方法的平均搜索效率分别提高了 71.6% 和 62.6%。

4 结 论

自适应折叠混沌优化方法利用提出的无限折叠 Fuch 映射,通过在优化过程中不断的缩小优化变量搜索空间以提高搜索效率。用该映射取代传统的混沌映射作为混沌产生器,解决了目前混沌优化算法中优化结果对搜索初始值要求极高的问题。数值仿真实验表明,本文方法不依赖于初始值,可以显著减少函数的运行次数,提高求解全局最优问题的效率。本文方法对于求解实际优化问题具有很强的实用性和实效性,是一种高效的智能启发式全局优化算法。

参考文献:

- [1] 谭良, 吴波, 刘震, 等. 一种基于混沌和小波变换的大容量音频信息隐藏算法 [J]. 电子学报, 2010, 38(8): 1812-1824.
TAN Liang, WU Bo, LIU Zhen, et al. An audio information hiding algorithm with high-capacity which based on chaotic and wavelet transform [J]. Chinese Journal of Electronics, 2010, 38(8): 1812-1824.
- [2] YANG Yimin, WANG Yaonan, YUAN Xiaofang, et al. Hybrid chaos optimization algorithm with artificial emotion [J]. Applied Mathematics and Computation, 2012, 11(2): 6585-6611.
- [3] 邹恩, 陈建国, 李祥飞. 一种改进的变尺度混沌优化算法及其仿真研究 [J]. 系统仿真学报, 2006, 18(9): 2426-2432.
ZOU En, CHEN Jianguo, LI Xiangfei. Improving mutative scale chaos optimization algorithm and simulation study [J]. Journal of System Simulation, 2006, 18(9): 2426-2432.
- [4] 袁小芳, 王耀南, 吴亮红. 基于混沌变量的模式搜索法及其应用 [J]. 湖南大学学报: 自然科学版, 2007, 34(9): 30-33.
YUAN Xiaofang, WANG Yaonan, WU Lianghong. Pattern search algorithm using chaos and its applica-
tion [J]. Journal of Hunan University: Natural Sciences, 2007, 34(9): 30-33.
- [5] 王芳, 戴永寿, 王少水. 改进的混沌遗传算法 [J]. 计算机工程与应用, 2010, 46(6): 29-32.
WANG Fang, DAI Yongshou, WANG Shaoshui. Modified chaos-genetic algorithm [J]. Computer Engineering and Application, 2010, 46(6): 29-32.
- [6] HE Di, HE Chen, JIANG Lingge, et al. Chaotic characteristics of a one dimensional iterative map with infinite collapses [J]. IEEE Trans on Circuits and Systems: I, 2001, 48(7): 900-906.
- [7] 杨海东, 鄂加强. 自适应变尺度混沌免疫优化算法及其应用 [J]. 控制理论与应用, 2009, 26(10): 1069-1074.
YANG Haidong, E Jiaqiang. Adaptive chaotic immune optimization algorithm with mutative scale and its application [J]. Control Theory & Application, 2009, 26(10): 1069-1074.
- [8] 修春波, 刘向东, 张宇河. 双混沌机制优化方法及其应用 [J]. 控制与决策, 2003, 18(6): 724-726.
XIU Chunbo, LIU Xiangdong, ZHANG Yuhe. Optimization algorithm using two kinds of chaos and its application [J]. Control and Decision, 2003, 18(6): 724-726.
- [9] 杨迪熊, 李刚, 程耿东. 非线性函数的混沌优化方法比较研究 [J]. 计算力学学报, 2004, 21(3): 257-261.
YANG Dixiong, LI Gang, CHENG Gengdong. Comparative study on chaos optimization algorithm for non-linear function [J]. Chinese Journal of Computational Mechanics, 2004, 21(3): 257-261.
- [10] 李文, 梁昔明. 基于混沌优化和最速下降法的一种混合算法 [J]. 计算计算与自动化, 2003, 22(2): 12-14.
LI Wen, LIANG Ximing. A hybrid algorithm based on chaos optimization and steepest descent algorithm [J]. Computing Technology and Automation, 2003, 22(2): 12-14.
- [11] 单梁, 强浩, 李军, 等. 基于 Tent 映射的混沌优化算法 [J]. 控制与决策, 2005, 20(2): 179-182.
SHAN Liang, QIANG Hao, LI Jun, et al. Chaotic optimization algorithm based on Tent map [J]. Control and Decision, 2005, 20(2): 179-182.
- [12] FENG Jun, YING Wenye, CHEN Binkang. Pattern search algorithm an its application research in three dimensional component layout [J]. Journal of Wuhan University of Technology: Transportation Science & Engineering, 2003, 27(2): 280-284.

(编辑 刘杨)