

文章编号: 1001-0920(2009)04-0514-06

一类自适应混沌文化算法

郭一楠, 王辉, 程健

(中国矿业大学 信息与电气工程学院, 江苏 徐州 221008)

摘要: 混沌变异进化算法忽略了混沌规律性, 未充分利用知识来提高算法的局部收敛能力. 为此, 借鉴文化算法的双层进化结构, 在文化算法的进化引导函数中引入自适应混沌变异策略, 提出一种自适应混沌文化算法. 利用进化过程隐含知识控制变异尺度, 使知识引导个体能跳出局部较优解, 在保证种群多样性的同时, 实现进化后期的精细搜索. 仿真结果表明, 该算法可以有效提高进化收敛速度, 具有较好的计算稳定性.

关键词: Logistic 序列; 自适应; 混沌变异; 文化算法

中图分类号: TP301 **文献标识码:** A

A novel adaptive chaotic cultural algorithm

GUO Yi-nan, WANG Hui, CHENG Jian

(College of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221008, China. Correspondent: GUO Yi-nan, E-mail: nanfly@126.com)

Abstract: In evolutionary algorithms with chaotic mutation, implicit knowledge and orderliness of chaos are not fully utilized to improve local convergence. An adaptive chaotic cultural algorithm is proposed by adopting dual structure in cultural algorithm and adaptive chaotic mutation in evolution induction functions. Implicit knowledge extracted from evolution process is used to control mutation scale, which inducts individuals escaping from local best solutions. This strategy can ensure the diversity of population and exploitation in the latter evolution. Simulation results indicate that the algorithm can effectively improve the speed of convergence and has better computation stability.

Key words: Logistic sequence; Adaptive; Chaotic mutation; Cultural algorithms

1 引言

一个好的优化算法既要利用积累信息搜索当前空间, 又要兼顾对未知空间的探索, 以实现在加快算法收敛速度的同时, 克服早熟收敛. 文化算法正是从进化过程知识利用的角度, 为上述目标提供了实现基础. 它是由 Reynolds^[1]提出的一种源于文化进化的双重进化模型, 由实现个体进化的种群空间和实现知识更新的信度空间构成. 文化算法通过信度空间实现进化信息的有效提取和管理, 并利用进化信息指导种群空间的进化过程, 从而提高进化效率. 该算法为进化计算中领域知识提取及利用提供了一个通用框架, 目前已与进化规划^[1]、量子计算^[2]、粒子群算法^[3]、微分进化算法^[4]、交互式遗传算法^[5]等智能优化方法成功结合, 并用于解决实值函数优化^[6]、产品设计^[5]、生产调度^[7]和工业优化控制^[8]等问题.

目前, 该算法在知识类型描述、信度空间结构和接口函数等方面已有丰富的研究成果. 知识通常从两方面作用于种群: 1) 搜索范围. 标准知识和拓扑知识利用搜索范围限制新个体, 保证解的可行性; 2) 变异算子. 变异算子体现进化过程的探索能力, 恰当的变异算子可以保持种群多样性, 避免早熟收敛. 通常, 各类知识通过影响变异算子中的参考个体或变异尺度来引导进化过程, 因此, 变异算子对文化算法的性能具有显著影响.

混沌是非线性动力系统的一种特有运动形式, 具有随机性、遍历性及规律性等一系列特质. 将其引入进化计算, 利用其遍历性可以有效避免搜索过程陷入局部最优. 但相关研究成果多数是将变异算子中的随机序列替代成混沌序列^[9], 以表明混沌变异是实数编码进化算法变异算子中的有效实现. 它忽

收稿日期: 2008-03-10; 修回日期: 2008-05-09.

基金项目: 国家自然科学基金项目(60805025); 国家 863 计划项目(2007AA12Z162); 中国博士后科学基金项目(2005037225).

作者简介: 郭一楠(1975—), 女, 太原人, 副教授, 博士, 从事智能优化、网络控制系统等研究; 王辉(1982—), 女, 山东兖州人, 硕士生, 从事文化计算的研究.

略了混沌规律性的特点,未充分利用知识来提高算法的局部收敛能力。

针对上述问题,本文提出一种自适应混沌文化算法。在文化算法的进化引导函数中引入自适应混沌变异策略,从而利用进化过程隐含知识控制变异尺度,使知识引导个体能跳出局部较优解,在保证种群多样性的同时,实现进化后期的精细搜索。

2 自适应混沌文化算法结构与描述

算法采用文化算法的双层进化结构,由底层种群空间和上层信度空间构成,如图 1 所示。

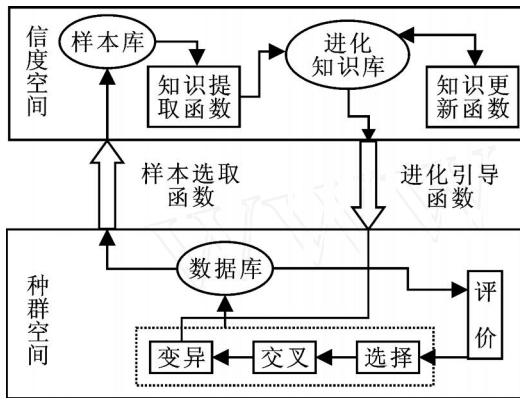


图 1 自适应混沌文化算法结构

种群空间实现个体评价和选择-交叉-变异等进化操作,并将存储于数据库中的已评价个体及其适应值提供给信度空间。信度空间通过样本选取函数从种群空间已评价种群中选取样本个体,通过知识提取函数从样本个体提取进化过程中的隐含信息,存储于进化知识库,并通过进化引导函数作用于种群空间,实现对进化操作的引导。

基于双层进化结构,将自适应混沌变异策略引入进化引导函数,具体算法实现步骤如下:

Step1: 种群和各类知识初始化。初始化种群描述为

$$X(0) = \begin{bmatrix} x_{11}(0) & x_{12}(0) & \dots & x_{1m}(0) \\ x_{21}(0) & x_{22}(0) & \dots & x_{2m}(0) \\ \dots & \dots & \ddots & \dots \\ x_{n1}(0) & x_{n2}(0) & \dots & x_{nm}(0) \end{bmatrix}. \quad (1)$$

其中: $x_{ij}(0) \in [0, 1]$, m 为变量维数, n 为种群规模。

Step2: 解码。为便于计算个体适应值,将个体通过坐标变换解码成可行解

$$\tilde{x}_{ij}(t) = l_j(0) + (u_j(0) - l_j(0)) x_{ij}(t), \quad i \in [1, n], j \in [1, m], \quad (2)$$

其中 $l_j(0)$ 和 $u_j(0)$ 分别为标准知识中变量 j 的初始上限和下限。

Step3: 计算可行解 $\tilde{x}_{ij}(t)$ 的适应度 $f(\tilde{x}_{ij}(t))$, 并依据 $f(\tilde{x}_{ij}(t))$ 降序排列种群个体。

Step4: 按接受比例提取样本个体,并更新信度空间各类知识。

Step5: 利用信度空间知识,确定自适应混沌变异尺度;根据各类知识的影响比例,对个体进行变异操作形成子群体 $X^{\text{sub}}(t)$ 。

Step6: 选择 $X^{\text{sub}}(t)$ 中的最优个体,替代原个体构成下一代种群。

Step7: 判断停止准则是否满足,满足则停止;否则,则返回 Step2。

可见,算法的核心在于信度空间各类知识的提取、更新与利用策略。

3 信度空间的知识更新与利用

本算法中,信度空间采用标准知识和拓扑知识两类知识描述。

3.1 标准知识

标准知识 $K_1(t) = [L_{K_1}(t), U_{K_1}(t)]$ 描述决策空间中可行解区域 $S_{K_1}(t)$,其中 $L_{K_1}(t)$ 和 $U_{K_1}(t)$ 分别为区域的下、上限。

初始状态的标准知识就是问题定义的变量范围。只有当种群中全部个体都位于新的可行解区域时,标准知识才发生更新,如下所示:

$$\begin{aligned} K_1(t) &= [L_{K_1}(t), U_{K_1}(t)] \quad K_1(t-1), \\ \text{s.t. } \forall x_i(t) &\in S_{K_1}(t-1) - S_{K_1}(t), \\ S_{K_1}(t) &\subset S_{K_1}(t-1). \end{aligned} \quad (3)$$

根据可行解区域的变化情况,基于混沌序列来自适应调节变异尺度。借鉴文献[10]中的“尺度收缩”变异算子,提出新型变异算子为

$$\tilde{x}_{ij}(t+1) = \begin{cases} l_j(t) + (u_j(t) - l_j(t)) a_{ij}^h(t), & \tilde{x}_i(t) \in S_{K_1}(t); \\ \tilde{x}_{ij}(t) + \frac{w_i}{n} \exp\left(-\frac{t}{T}\right) (u_j(t) - l_j(t)) a_{ij}^h(t), & \tilde{x}_i(t) \notin S_{K_1}(t). \end{cases} \quad (4)$$

其中: $l_j(t)$ 和 $u_j(t)$ 分别为第 t 代变量 j 的可行解区域下限和上限, $a_{ij}^h(t)$ 为混沌序列值。本文采用 Logistic 混沌序列,即

$$\mu_{l+1} = \mu_l(1 - \mu_l), \quad l = 0, 1, \dots, \quad (5)$$

其中 $\mu \in [3, 4]$ 为混沌因子。当 μ 确定后,由任意初值 $\mu_0 \in [0, 1]$,可迭代出一个确定的时间序列 μ_1, μ_2, \dots 且 μ 越大,Logistic 序列的遍历性越好。

本文将 μ 的取值与可行解区域变化相结合,提出一种自适应混沌因子,描述为

$$\mu(t) = 3.5 + 0.5 \times \frac{u_j(t) - l_j(t)}{u_j(0) - l_j(0)}. \quad (6)$$

根据标准知识定义可知, $\mu(t)$ 的取值范围为 $[3.5,$

4].

为确保初始种群的非重复性,本文进一步利用 Logistic 序列 ($\mu = 4$) 在 $[0, 1]$ 范围内的遍历性,将其引入种群初始化过程,即在 $[0, 1]$ 范围内随机生成 m 个初值 $x_j(0), j \in [1, m]$. 基于这些初值产生 m 个序列长度为 n 的不同混沌序列,构成 $X(0) \in R^{n \times m}$.

3.2 拓扑知识

拓扑知识 K_2 描述了进化过程中个体的适应度分布状况,表示为 $K_{2,1}, K_{2,2}, \dots, K_{2,q}, \dots, q = 1, 2, \dots, m_{K_2}$, 其中 $K_{2,q}(t) = [L_{K_{2,q}}(t), U_{K_{2,q}}(t), \tilde{x}_{K_{2,q}}^{\text{best}}(t)]$ 分别代表划分后子区域 $S_{K_{2,q}}$ 的下限、上限和该区域中的最优个体,满足 $L_{K_{2,q}} \leq L_{K_1}, U_{K_{2,q}} \leq U_{K_1}$.

拓扑知识更新体现为两方面:

1) 标准知识更新引发拓扑知识的重新划分,即

$$K_{2,1}(t) = [L_{K_1}(t), U_{K_1}(t), \tilde{x}^{\text{best}}(t)], \quad \text{if } S_{K_1}(t) \neq S_{K_1}(t-1). \quad (7)$$

2) 拓扑知识的细化.

定义 1 信息增益 I_j 记录各变量维上的适应度梯度,记作

$$I_j = \frac{f(\tilde{x}_j^{\text{best}}(t)) - f(\tilde{x}_j^{\text{best}}(t-1))}{\tilde{x}_j^{\text{best}}(t) - \tilde{x}_j^{\text{best}}(t-1)}. \quad (8)$$

拓扑知识采用二叉树结构,根节点为 S_{K_1} . 当式 (8) 细化条件满足时,在相应知识空间进行细化,有

$$S_{\text{split}}(t) = S_{K_{2,q},j}, j = \arg \max_{j=1,2,\dots,m} I_j, \quad \text{s. t. } \tilde{x}_j^{\text{best}}(t) \in S_{K_{2,q}}, \tilde{x}_j^{\text{best}}(t-1) \in S_{K_{2,q}}. \quad (9)$$

显然,细化只可能发生在叶子节点,并且选取具有最大信息增益的变量维作为细化方向. 相应变量维的二叉树分割点取值为

$$\tilde{x}_{\text{split},j}(t) = 0.5(\tilde{x}_j^{\text{best}}(t) + \tilde{x}_j^{\text{best}}(t-1)). \quad (10)$$

根据随机选中子区域,基于 Logistic 序列来自适应调节变异尺度. 变异算子如下:

$$\tilde{x}_{ij}(t+1) = \tilde{x}_{ij}(t) + \frac{w_i}{n} \exp\left(\frac{-t}{T}\right) (u_{K_{2,q},j}(t) - l_{K_{2,q},j}(t)) a_{ij}^h(t), \quad \text{s. t. } \tilde{f}(\tilde{x}_{K_{2,q}}^{\text{best}}(t)) > r, r \in (0, 1). \quad (11)$$

其中: $u_{K_{2,q},j}(t)$ 和 $l_{K_{2,q},j}(t)$ 为 t 代子区域 $S_{K_{2,q}}$ 中变量 j 的上限和下限, $\tilde{f}(\tilde{x}_{K_{2,q}}^{\text{best}}(t))$ 为子区域最优解的相对适应值. 则 μ 依照式 (6) 由随机选中子区域跨度确定.

分析表明,新型自适应混沌变异算子具有以下特性:

1) w_i 为个体 i 在种群中的排名,适应度越高的个体 w_i 越小,其变步步长越小.

2) t 为进化代数,随着进化深入变步步长减小.

3) 可行解区域跨度越小,变异尺度越小;反之,变异尺度越大.

可见,知识引导的自适应变异可以保证进化过程中种群的多样性,有利于进化后期的搜索.

3.3 接口函数

接口函数是指为上层知识和下层进化过程提供作用通道的样本选取函数和进化引导函数.

样本选取函数采用比例选择策略,即按接受比例 k_a 依适应值降序选取样本个体

$$n_a = k_a \% n. \quad (12)$$

进化引导函数的核心在于确定各类知识对种群的影响比例和作用形式. 文中知识作用于自适应混沌变异算子,作用方式见式 (4) ~ (6). 各类知识对种群的影响比例则依据知识影响成功率动态确定如下:

$$k_l(t) = \begin{cases} 1/N_K, & t = 0; \\ + \frac{n_{K_l}(t-1)}{n}, & t > 0; \end{cases} \quad \text{s. t. } N_K + \dots = 1. \quad (13)$$

其中: N_K 为信度空间中的知识类型; n_{K_l} 表示受各类知识影响所生成的个体数目,其所占种群比例即为知识影响成功率. 本文只考虑两类知识作用,因此 $N_K = 2$. 根据约束条件确定 k_l 取值范围为 $[0, 0.5]$; 选取 $\mu = 0.1$, 以保证每类知识的影响比例至少占种群的 10%.

4 仿真结果与分析

为深入分析本文提出的自适应混沌文化算法性能,针对函数优化问题,采用标准测试函数对于算法中的关键参数对进化性能的影响进行深入分析,并将其与传统文化算法、混沌变异进化算法进行性能比较.

采用的标准测试函数及其最优解分布如表 1 所示. 仿真分析中算法主要参数取值如表 2 所示.

4.1 自适应混沌文化算法性能分析

信度空间中的各类知识是从不断更新的样本库中提取获得的进化过程隐含信息. 在这个过程中,优势区域逐渐明确并对进化过程加以引导,从而改善算法性能.

4.1.1 信度空间拓扑知识的演化

拓扑知识记录历史较优解,并以此为依据实现可行解区域的细分. 不失一般性,以标准测试函数 f_7 的寻优过程(收敛代数 = 37) 为例,记录其二叉树分割维选取及相应取值,如表 3 所示.

基于上述拓扑知识分割,获得拓扑知识演化过

表 1 测试函数

测试函数 ($N = 2$)	变量范围	最优解 x^*	最优值 $f(x^*)$
$\max f_1(x, y) = 1 + x \sin(4x) - y \sin(4y) + \frac{\sin(6\sqrt{x^2 + y^2})}{6\sqrt{x^2 + y^2} + 10^{-15}}$	$x, y \in [-1, 1]$	$(+0.64, +0.64)$ $(-0.64, -0.64)$ $(+0.64, -0.64)$ $(-0.64, +0.64)$	2.118
$\min f_2(x, y) = 20 + x^2 - 10 \cos(2x) + y^2 - 10 \cos(2y)$	$x, y \in [-5, 12, 5, 12]$	(0, 0)	0
$\min f_3(x, y) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{ 1 + 0.001 \times (x^2 + y^2)^2 }$	$x, y \in [-10, 10]$	(0, 0)	0
$\min f_4(x_i) = \sum_{i=1}^N x_i^2$	$x_i \in [-100, 100]$	(0, 0)	0
$\max f_5(x_i) = - \sum_{i=1}^{N-1} [(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	$x_i \in [-10, 10]$	(1, 1)	0
$\max f_6(x_i) = - \left[\sum_{i=1}^N x_i + \sum_{i=1}^N x_i \right]$	$x_i \in [-10, 10]$	(0, 0)	0
$\max f_7(x_i) = - \sum_{i=1}^N \frac{x_i^2}{4000} + \sum_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) - 1$	$x_i \in [-600, 600]$	(0, 0)	0

表 2 主要参数取值

交叉概率	变异概率	种群规模	运行次数	解精度
P_c	P_m	n	L	
0.9	0.08	50	30	10^{-3}
接受比例	样本库规模	进化终止代数	混沌序列长度	
k_a	n_s	T	Lm	
0.2	30	100	100	

表 3 拓扑知识划分维及其数值

次序	1	2	3	4	5
维数	x_2	x_2	x_1	x_1	x_1
取值	-19.038	-10.93	-6.2766	-3.0606	1.5272
次序	6	7	8	9	10
维数	x_1	x_1	x_2	x_1	x_1
取值	-0.0316	0.00793	-4.3351	0.00364	0.002656

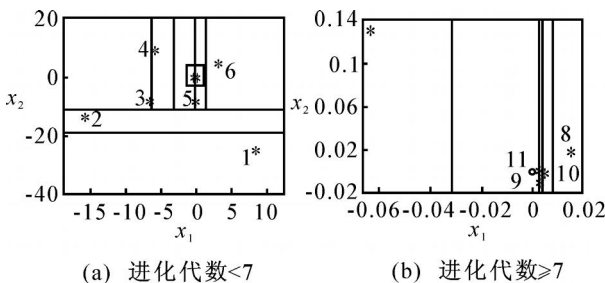


图 2 拓扑知识划分图

程的图形描述,如图 2 所示.图中: * 表示细化区域的较优解, ° 表示函数 f_7 的最优解.

上述拓扑知识的演化过程反映了进化过程中对优势区域的探索和细化,从而引导进化过程在更优势的区域实现高效搜索.

4.1.2 对算法性能的影响

各类知识对种群的影响比例取决于知识影响成

功率和 λ 不同的 λ 值,导致知识作用于种群的基础比例发生变化,从而对算法性能产生影响.为此,针对标准测试函数,选取 4 组不同 λ 值进行仿真运行,统计结果如表 4 所示.表 4 中, M_1 表示平均最优目标值, M_2 表示最优值均方差, M_3 表示获得最优解的运行次数, M_4 表示获得最优解的平均进化代数.

可见,选取较小值 (0 或 0.1) 时,算法收敛更快、稳定性更好、性能更优.分析表明,取值越小,知识影响成功率在确定知识影响比例中所起的作用越大,即 $k_i(t)$ 几乎完全根据各类知识对种群进化的历史贡献来确定,从而在不同进化阶段,可以根据进化需求,实现更加有效的进化搜索.

当选取 0.5 时,根据约束条件 $\sum k_i(t) = 0.5$,即不管每类知识对种群进化的历史贡献大小,始终只影响一半种群.显然,这会降低优良知识的引导效率,使知识引导进化的优势不能得到最好体现.

值不同,进化过程中各类知识的作用比例也有所区别.以 f_7 为例,进化过程中各类知识作用的比例曲线如图 3 所示.

可见,拓扑知识在进化中后期的作用比例显著升高,表明种群在细化后的优势区域深度搜索;并且随着 λ 值升高,两类知识的作用趋势虽然相同,但受两类知识影响的种群比例明显接近.

4.1.3 混沌序列长度对算法性能的影响

为深入分析混沌序列长度 Lm 对算法性能的影响,针对标准测试函数 f_1 ,分别选取 6 组 Lm ,统计其 30 次运行结果,如表 5 所示.

显然,混沌序列长度越长,序列中个体在可行区

表4 不同取值的算法性能比较

函数	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
= 0	M_1	2.11871	5.54×10^{-3}	3.80×10^{-4}	6.53×10^{-4}	-7.66×10^{-6}	-1.82×10^{-3}	-3.94×10^{-3}
	M_2	2.15×10^{-4}	1.28×10^{-2}	1.39×10^{-3}	1.33×10^{-3}	1.01×10^{-5}	1.85×10^{-3}	3.35×10^{-3}
	M_3	30	30	30	30	30	30	17
	M_4	10	24	13	25	10	32	35
= 1	M_1	2.11875	2.03×10^{-3}	7.01×10^{-5}	5.89×10^{-4}	-1.26×10^{-5}	-1.95×10^{-3}	-4.76×10^{-3}
	M_2	1.17×10^{-5}	4.52×10^{-3}	1.80×10^{-4}	1.58×10^{-3}	1.55×10^{-5}	3.02×10^{-3}	5.37×10^{-3}
	M_3	30	30	30	30	30	30	16
	M_4	10	26	12	27	10	33	37
= 0.3	M_1	2.11875	2.05×10^{-3}	1.04×10^{-4}	1.44×10^{-3}	-6.09×10^{-5}	-3.49×10^{-3}	-5.13×10^{-3}
	M_2	1.77×10^{-5}	4.67×10^{-2}	2.54×10^{-4}	3.50×10^{-2}	8.27×10^{-5}	6.19×10^{-3}	5.74×10^{-3}
	M_3	30	30	30	30	30	30	18
	M_4	10	26	13	29	10	37	39
= 0.5	M_1	2.11875	7.89×10^{-3}	3.98×10^{-4}	1.48×10^{-3}	-6.80×10^{-5}	-4.49×10^{-3}	-6.76×10^{-3}
	M_2	2.36×10^{-5}	1.88×10^{-2}	1.80×10^{-3}	5.13×10^{-3}	1.25×10^{-5}	8.75×10^{-3}	7.84×10^{-3}
	M_3	30	30	30	30	30	30	14
	M_4	11	30	13	32	11	39	43

表5 不同 L_m 取值的算法性能比较

函数	f_1	f_2	f_3	f_4	f_5	f_6	f_7	
$L_m = 20$	M_1	2.11877	4.04×10^{-2}	8.58×10^{-4}	6.39×10^{-2}	-2.02×10^{-5}	-6.97×10^{-3}	-1.22×10^{-2}
	M_2	1.97×10^{-4}	6.92×10^{-2}	1.33×10^{-3}	0.191	2.28×10^{-5}	9.85×10^{-3}	1.39×10^{-2}
	M_3	30	30	30	30	30	30	9
	M_4	17	45	21	48	17	51	55
$L_m = 60$	M_1	2.11873	2.35×10^2	3.19×10^{-4}	4.39×10^{-3}	-1.60×10^{-5}	-2.89×10^{-3}	-5.26×10^{-3}
	M_2	4.92×10^{-5}	6.83×10^{-2}	8.74×10^{-4}	1.12×10^{-2}	3.3×10^{-5}	3.33×10^{-3}	7.32×10^{-3}
	M_3	30	30	30	30	30	30	18
	M_4	12	32	15	34	12	40	45
$L_m = 100$	M_1	2.11875	2.05×10^{-3}	1.04×10^{-4}	1.44×10^{-3}	-6.09×10^{-6}	-1.95×10^{-3}	-4.76×10^{-3}
	M_2	1.77×10^{-5}	4.67×10^{-2}	2.54×10^{-4}	3.50×10^{-2}	8.27×10^{-6}	3.02×10^{-3}	5.37×10^{-3}
	M_3	30	30	30	30	30	30	18
	M_4	10	26	13	29	10	37	39
$L_m = 140$	M_1	2.11875	1.12×10^{-3}	1.43×10^{-5}	7.30×10^{-4}	-3.46×10^{-6}	-1.60×10^{-3}	-4.08×10^{-3}
	M_2	1.57×10^{-5}	3.40×10^{-3}	2.93×10^{-5}	1.20×10^{-3}	4.55×10^{-6}	2.27×10^{-3}	3.60×10^{-3}
	M_3	30	30	30	30	30	30	18
	M_4	10	23	12	26	9	33	39

表6 不同算法性能比较

函数	传统文化算法				混沌进化算法				自适应混沌文化算法			
	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4	M_1	M_2	M_3	M_4
f_1	-	-	-	-	2.11876	3.32×10^{-5}	29	71	2.11875	1.17×10^{-5}	30	10
f_2	4.88×10^{-6}	0	1	34	1.77×10^{-5}	1.05×10^{-5}	2	99	2.03×10^{-3}	4.52×10^{-3}	30	26
f_3	3.02×10^{-2}	4.03×10^{-2}	19	18	4.94×10^{-2}	4.10×10^{-2}	12	9	7.01×10^{-5}	1.80×10^{-4}	30	12
f_4	2.11×10^{-5}	3.11×10^{-5}	29	62	2.55×10^{-5}	2.78×10^{-5}	30	36	5.89×10^{-4}	1.58×10^{-3}	30	27
f_5	-2.01×10^{-5}	-2.09×10^{-5}	30	44	-2.48×10^{-5}	1.69×10^{-5}	30	12	-1.26×10^{-5}	1.55×10^{-5}	30	10
f_6	-	-	-	-	-	-	-	-	1.95×10^{-3}	3.02×10^{-3}	30	33
f_7	-0.143	0.155	3	46	-2.33	2.08	1	10	-4.76×10^{-3}	5.37×10^{-3}	18	37

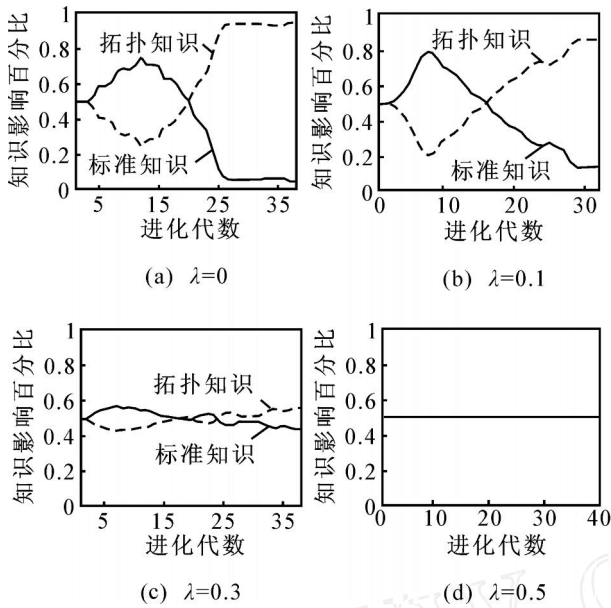


图 3 进化过程中各类知识作用比例曲线

域的遍历性越好,算法具有更好的稳定性.但考虑到序列长度增加所带来的计算复杂度升高,本文折衷选取混沌序列长度为 100.

4.2 与其他算法性能对比分析

针对表 1 中 7 个测试函数,分别采用传统文化算法^[1]、混沌变异进化算法^[9]和自适应混沌文化算法,参照表 2 所示的算法参数进行仿真实验,运行结果的统计分析如表 6 所示.相比而言,自适应混沌文化算法在达到相同或较高求解精度时,其平均收敛速度较快,解的稳定性较好.

可见,本文采用的自适应混沌变异算子与进化过程隐含知识有效结合,可以在进化初期获得较大的变异尺度,实现在较大空间搜索;而在进化后期逐渐缩小变异尺度,使求解的精度获得提高.因此,自适应混沌文化算法在整体上具有较高的求解精度和求解稳定性,能有效抑制早熟收敛,跳出局部较优解,具有较快的收敛速度.

5 结 论

本文借鉴文化算法的双层进化结构,在文化算法的影响函数中引入自适应混沌变异策略,提出一种自适应混沌文化算法.它利用进化过程隐含知识控制变异尺度,使知识引导个体能跳出局部较优解,在保证种群多样性的同时,实现进化后期的精细搜索.仿真结果表明,该算法可以有效提高进化收敛速度.进一步提高文化算法的性能将是今后深入研究的课题.

参考文献(References)

- [1] Bin Peng. Knowledge and population swarms in cultural algorithms for dynamic environments [D]. Detroit: Wayne State University, 2005.
- [2] Cruz A V, Pacheco M C, Marley Vellasco, et al. Cultural operators for a quantum-inspired evolutionary algorithm applied to numerical optimization problems [J]. Lecture Notes in Computer Science, 2005, 3562: 1-10.
- [3] Coelho L S, Mariani V C. An efficient particle swarm optimization approach based on cultural algorithm applied to mechanical design [C]. IEEE Conf of Evolutionary Computation. Piscataway, 2006: 1099-1104.
- [4] Ricardo L Becerra, Carlos A Coello. Cultured differential evolution for constrained optimization [J]. Computer Methods in Applied Mechanics and Engineering, 2006, 195(1): 4303- 4322.
- [5] 郭一楠, 巩敦卫. 双层进化交互式遗传算法的知识提取与利用[J]. 控制与决策, 2007, 22(12): 1329-1335. (Guo Y N, Gong D W. Extraction and utilization about knowledge in hierarchical interactive genetic algorithms [J]. Control and Decision, 2007, 22(12): 1329- 1335.)
- [6] Becerra R L, Coello C A C. Solving hard multiobjective optimization problems using ϵ -constraint with cultured differential evolution[C]. The 9th Int Conf on Parallel Problem Solving from Nature. Iceland, 2006, 4193: 543-552.
- [7] Ho N B, Tay J C. Gemace: An effective cultural algorithm for solving the flexible job-shop problem[C]. Proc 2004 Congress on Evolutionary Computation. Piscataway, 2004: 1759-1766.
- [8] 黄海燕, 顾幸生, 刘漫单. 求解约束优化问题的文化算法研究[J]. 自动化学报, 2007, 33(10): 1115-1120. (Huang H Y, Gu X S, Liu M D. Research on cultural algorithm for solving nonlinear constrained optimization [J]. Acta Automatica Sinica, 2007, 33(10): 1115-1120).
- [9] 骆晨钟, 邵惠鹤. 采用混沌变异的进化算法[J]. 控制与决策, 2000, 15(5): 557-560. (Luo C Z, Shao H H. Evolutionary algorithms with chaotic mutations[J]. Control and Decision, 2000, 15(5): 557-560).
- [10] 杜海峰, 公茂果, 刘若辰, 等. 自适应混沌克隆进化规划算法[J]. 中国科学 E, 2005, 35(8): 817-829. (Du H F, Gong M G, Liu R C, et al. Adaptive chaos clonal evolutionary programming[J]. Science in China Ser E, 2005, 35(8): 817-829).