

# 控制与决策

Control and Decision

## 基于混沌“微变异”自适应遗传算法

潘伟, 丁立超, 黄枫, 孙洋

引用本文:

潘伟, 丁立超, 黄枫, 等. 基于混沌“微变异”自适应遗传算法[J]. *控制与决策*, 2021, 36(8): 2042–2048.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.0319>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 嵌入Circle映射和逐维小孔成像反向学习的鲸鱼优化算法

Whale optimization algorithm for embedded Circle mapping and one-dimensional oppositional learning based small hole imaging  
*控制与决策*. 2021, 36(5): 1173–1180 <https://doi.org/10.13195/j.kzyjc.2019.1362>

### 基于混合差分遗传算法的Bouc–Wen迟滞模型辨识策略

Bouc–Wen hysteresis model identification strategy based on hybrid differential genetic algorithm  
*控制与决策*. 2021, 36(2): 371–378 <https://doi.org/10.13195/j.kzyjc.2019.0663>

### 基于预防维护的单机调度问题

Single-machine scheduling problem with preventative maintenance activities  
*控制与决策*. 2021, 36(2): 395–402 <https://doi.org/10.13195/j.kzyjc.2019.0626>

### 基于改进多目标优化算法的分布式数据中心负载调度

Multi-objective optimization of energy and performance management in distributed data centers  
*控制与决策*. 2021, 36(1): 159–165 <https://doi.org/10.13195/j.kzyjc.2019.0702>

### 基于机床超低待机状态的流水车间能耗调度

Energy consumption scheduling in flow shop based on ultra-low idle state of numerical control machine tools  
*控制与决策*. 2021, 36(1): 143–151 <https://doi.org/10.13195/j.kzyjc.2019.0433>

# 基于混沌“微变异”自适应遗传算法

潘伟<sup>1</sup>, 丁立超<sup>1†</sup>, 黄枫<sup>1</sup>, 孙洋<sup>2</sup>

(1. 陆军炮兵防空兵学院士官学校, 沈阳 110867; 2. 东北大学信息科学与工程学院, 沈阳 110004)

**摘要:** 遗传算法可以较好地解决复杂的组合优化问题,但也存在两方面不足:一是搜索效率比其他优化算法低;二是容易过早收敛,陷入局部最优. 对此,提出一种混沌“微变异”遗传算法. 利用混沌优化算法具有随机性和遍历性的特点,解决遗传算法容易陷入局部最优解的早熟问题,使得新算法同时具有较强的局部搜索能力和完成全局寻找最优解的能力. 同时,对遗传算法的选择算子增加了混沌扰动,对交叉算子和变异算子进行自适应调整,对适应度函数进行改进,使遗传算法整体性能得到提高. 最后,通过经典函数验证表明,混沌“微变异”遗传算法比一般的混沌遗传算法和经典遗传算法的进化速度更快,搜索精度更高.

**关键词:** 混沌; 微变异; 遗传算法; 过早收敛; 搜索乏力

中图分类号: TP18

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.0319

开放科学(资源服务)标识码(OSID):



引用格式: 潘伟,丁立超,黄枫,等. 基于混沌“微变异”自适应遗传算法[J]. 控制与决策, 2021, 36(8): 2042-2048.

## Adaptive genetic algorithm based on chaos “micro variation”

PAN Wei<sup>1</sup>, DING Li-chao<sup>1†</sup>, HUANG Feng<sup>1</sup>, SUN Yang<sup>2</sup>

(1. Noncommissioned Officer Academy, PLA Army Academy of Artillery and Air Defense, Shenyang 110867, China; 2. College of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

**Abstract:** Complex combinatorial optimization problems can be solved using the genetic algorithm, but there are also two shortcomings, one is lower search efficiency than other optimization algorithms, the other is easy to premature convergence and fall into local optimum. Therefore, this paper proposes the chaos “micro variation” genetic algorithm. Due to the characteristics of randomness and ergodicity, the chaos optimization algorithm solves the premature problem that genetic algorithm is easy to fall into the local optimal solution, which makes the proposed algorithm have strong local search ability and the ability to complete the global search for the optimal solution. At the same time, chaos disturbance is added into the selection operator of the genetic algorithm, the crossover operator and mutation operator are adjusted adaptively, and the fitness function is improved, such that the overall performance of the genetic algorithm is improved. Finally, the chaotic “micro mutation” genetic algorithm has faster evolution speed and higher search accuracy than the general chaotic genetic algorithm and classical genetic algorithm through the classical function verification.

**Keywords:** chaos; micro variation; genetic algorithm; premature convergence; search fatigue

## 0 引言

遗传算法模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程,是一种通过模拟自然进化过程搜索最优解的优化方法. 20世纪70年代,美国Michigan大学Holland教授最先提出,与传统优化算法比较,遗传算法具有全局自适应的随机搜索能力,是一种智能优化算法. 遗传算法通过数学的方式,将问题的求解过程转换成类似生物进化的过程,利用适应度函数作为评判标准,通过对染色体基因的遗传操

作完成进化过程. 对于复杂的组合优化问题有着很好的应用效果,具有较强的鲁棒性、随机性和良好的可扩展性,已广泛地应用于科学的各个领域<sup>[1-4]</sup>.

随着人们对遗传算法的认识和使用的加深,遗传算法在理论和应用技术上的不足也日趋明显. 在面对规模较大或者是复杂的非线性问题时,遗传算法无法保持群体多样性,这就可能导致搜索初期陷入局部最优解,搜索后期进化缓慢,搜索乏力,这也使得传统的遗传算法存在一定的局限性.

收稿日期: 2021-02-25; 修回日期: 2021-03-29.

基金项目: 国家自然科学基金项目(61902057).

责任编委: 陈家伟.

†通讯作者. E-mail: onlydingding@qq.com.

混沌的概念最早是由美国的著名气象学家洛伦兹(Lorenz)提出的.混沌是指在具有确定性的系统中所表现出来的一种貌似毫不规则或者类似随机的物理现象,这种物理现象普遍存在于复杂运动的形式之中.混沌的运动轨迹虽然在开始阶段只有微小差别,但在运动了一段时间之后的相互关系会逐渐消失,直至无关.混沌运动具有随机性和遍历性:随机性体现了混沌系统内部的不可预测性,对初始值变化强烈的敏感性决定了它的这一性质,也说明混沌是局部不稳定的;遍历性是混沌系统在一定范围内按其自身规律不重复地遍历所有状态,即在有限时间内可以到达每一个状态点<sup>[5-7]</sup>.

由于混沌优化算法的随机性可以克服遗传算法容易陷入局部最优解的问题,遍历性可以改善遗传算法的全局搜索能力,将混沌优化算法与遗传算法相结合可以优势互补,提高遗传算法的局部搜索能力,避免出现局部最优解,从而有效避免遗传算法“早熟”问题.

## 1 混沌遗传算法研究现状

关于混沌遗传算法已经有很多学者进行了研究.一般的方式是将混沌优化与遗传算法分别独立求解,也就是在遗传算法结束后再利用混沌优化或者添加混沌扰动的方式进行二次寻优.这种方法易于操作,本文不做赘述.下面就目前对混沌遗传算法的主流改进方法进行介绍.

将混沌扰动作为一个分量加入到遗传算法的各步操作中,形成多种混沌遗传算法.针对群体初始化,文献[1]对Logistic映射性能进行了分析,得出Logistic映射适合作为产生遗传算法初始群体的算法.文献[5]对比了Logistic映射、Cat映射和Tent映射3种混沌映射,分别对3种映射的遍历性、初值敏感性和Lyapunov指数等方面进行了综合分析,发现Cat映射的混沌特性更加优越,更适合为遗传算法产生初始值.

文献[8]把混沌扰动加入到遗传操作的交叉算子和变异算子中,提出了混沌交叉算子和混沌变异算子.混沌交叉算子是指首先设置交叉算子的初始值;然后进行Logistic映射,迭代一定次数后,得到个体 $P$ .当群体中最优个体参与交叉时,交叉算子将直接选择混沌映射后生成的个体 $P$ .通过混沌交叉算子可以减缓收敛速度,防止最优个体使算法迅速收敛而产生局部最优解.混沌变异算子是指首先进行个体变异操作,将变异得到的个体映射到(0,1)范围之内;然后利用Logistic映射,迭代一定次数,将得到

的数值再重新映射回个体,形成新的混沌变异后的个体.变异操作本身具有局部搜索能力,混沌变异操作由于具有遍历性而使变异操作的局部搜索能力更加强大.

文献[9]提出遗传操作结束产生子代个体之后,在群体中添加混沌扰动,以期通过混沌扰动产生的个体中有更优秀的个体出现,加快寻优进化速度,减少寻优进化代数,提高遗传算法的搜索效率.

借鉴以上将混沌与遗传算法相结合的方法,本文提出一种混沌“微变异”自适应遗传算法.分别在两个环节进行混沌扰动处理:一是在选择操作之后,当相同个体数量增加时,对相同的个体随机添加混沌扰动,这样可以大量减少待进入交叉操作的个体中相同个体的个数;二是在遗传操作结束后,对适应度函数值较低的个体进行混沌扰动,这样可以加快最优个体的产生速度,提高算法的搜索效率.另外,本文在原有工作基础上,对适应度函数、交叉算子和变异算子都进行了改进,使之可以更加适应有约束条件下最优化的需要,弥补遗传算法的原有缺陷.

## 2 “微变异”自适应混沌遗传算法

### 2.1 编码方式

在编码方式上本文采用实数编码<sup>[10]</sup>.传统的二进制编码,染色体的长度与解向量是一一对应的,一个基因对应一个解向量的分量.实数编码在基因型空间和表现型空间中是一致的,可以避免编码解码带来的计算误差和时间浪费.另外,实数编码在不牺牲计算精度的前提下扩大了编码范围,对于具有约束条件的优化问题最为有效.

### 2.2 自适应适应度函数

遗传算法来源于达尔文的生物进化论,本身对自然界和外部环境要求较低,一切最优化的评价标准和判断方法只需要计算适应度函数就可以得出.因此,适应度函数的选择恰当与否,决定了算法在优化时效率上的高低.对于经典遗传算法的适应度函数而言:在搜索初期,对于突出个体无法进行,导致突出个体把整个搜索过程带入局部最优而无法突破;在搜索后期,适应度函数高度一致,交叉和变异无力跳出局部范围,导致算法陷入局部最优.这主要是由于适应度函数直接被选中作为目标函数,虽然它们能够直接表现出对优化变量的适应性,但带来上述缺陷也是必然的.

本文提出改进适应度函数,改进方式如下:

$$F = f(X) \cdot e^{(1+\lambda \cdot |f(X)|)}. \quad (1)$$

其中:  $F$ 为改进后的适应度函数值,  $f(X)$ 为目标函数值,  $\lambda$ 为特定参数.

### 2.3 选择+“微变异”

本文在轮盘赌选择的基础上,提出“轮盘赌+微变异”.“微变异”是指,将复制之后的个体进行小概率的变异,变异方式为添加混沌扰动.这样做在一定程度上减少了由于选择操作带来的“早熟”,同时也降低了相同个体之间交叉的概率.“微变异”由下式确定:

$$\begin{cases} X'_n = (1 - \alpha \cdot \beta) \cdot X_n + (-1)^n \cdot \alpha \cdot \beta \cdot X_n / \|X_n\|, \\ X_n = (x_1, x_2, \dots, x_i), n = 1, 2, \dots, \text{size}, \\ \beta = (\beta_1, \beta_2, \dots, \beta_i, \dots, \beta_L). \end{cases} \quad (2)$$

其中:  $X'_n$ 为新的个体;  $X_n$ 为当前个体;  $\beta$ 为混沌扰动算子;  $\alpha$ 为人工退化的影响因子,取值范围(0, 0.01); size为群体规模;  $L$ 为个体长度.

式(2)首先计算混沌扰动因子 $\beta$ ,再由人工退化因子 $\alpha$ 缩小扰动大小,确定扰动半径 $\alpha \cdot \beta$ ,  $(-1)^n$ 能使扰动均匀遍历以基因为中心、区间长度为 $2\alpha \cdot \beta$ 中的点.

### 2.4 自适应交叉

本文基于文献[11]提出一种改进的自适应交叉算子,该算子满足搜索算法的两个趋势:一是随着进化进程的推进,算法需要快速收敛,全局搜索的需求降低,此时的交叉算子应逐渐减小;二是对于适应度函数值高于当代平均适应函数值的个体(假设优化问题求最大值),因该个体优秀更需要被保留,此时的交叉算子也应该逐渐减小.因此,改进的自适应交叉算子既与进化的代数有关,又与相应代中的适应度函数值有关;既要增加个体的多样性,又要利于搜索快速寻找到最优解.本文改进的自适应交叉算子如下:

$$P_c = \begin{cases} P_{c\min} + \rho \cdot \frac{f_{\max} - f}{f_{\max} - f_{\min}} \cdot \frac{1}{1 + e^t}, f \geq \bar{f}; \\ P_{c\max}, f < \bar{f}. \end{cases} \quad (3)$$

其中:  $f = \max(f_n, f_{n+1})$ ;交叉算子最大值和最小值分别为  $P_{c\max} = 0.8, P_{c\min} = 0.5; \rho = P_{c\max} - P_{c\min}$ .

交叉方式为

$$\begin{cases} X'_n = \gamma \cdot X_n + (1 - \gamma) \cdot X_{n+1}, \\ X'_{n+1} = \gamma \cdot X_{n+1} + (1 - \gamma) \cdot X_n, \end{cases} \quad (4)$$

其中 $\gamma$ 为系统产生的(0, 1)之间随机数.

### 2.5 自适应变异

本文提出的改进自适应变异算子的思路与上节改进自适应交叉算子一致,也是既考虑了进化代数的影响,又考虑了个体适应度函数值的大小<sup>[12-14]</sup>.与进化代数有关是因为到了搜索后期,主要是防止陷入局部最优解,此时的变异算子应随着代数的增加而逐渐加大,以便搜索算法能跳出局部最优.与个体适应度函数值有关是指当大于平均适应度函数值时,加大变异算子,这样可以增加个体的多样性,解决了算法后期搜索乏力的问题.基于经典实数遗传算法中的变异算子,本文提出了改进自适应变异算子,即

$$P_m = \begin{cases} P_{m\max}, f \geq \bar{f}; \\ P_{m\min} + \eta \cdot \frac{f_{\min} - f}{f_{\max} - f_{\min}} \cdot \frac{1}{1 + e^{-t}}, f < \bar{f}. \end{cases} \quad (5)$$

其中:变异算子最大值最小值分别为  $P_{m\max} = 0.05, P_{m\min} = 0.001; \eta = P_{m\max} - P_{m\min}$ .

参数变异的方式为

$$X'_n = (1 - \beta) \cdot X_n + (-1)^n \cdot \beta \cdot X_n / \|X_n\|, \quad (6)$$

其中 $\beta$ 为混沌系统产生的(0, 1)之间随机数.

### 2.6 混沌扰动

经过遗传操作之后,对种群中适应度函数高的个体进行保留,对种群中适应度函数低的个体进一步进行混沌优化处理.这样做可以增加种群基因多样性,使种群不易陷入局部最优解.同时,添加混沌优化可使个体更加接近最优解,减少进化代数.本文采取的混沌优化迭代公式为Tent映射,即

$$t_{n+1} = \begin{cases} \nu \cdot t_n, t_n \leq 0.5; \\ \nu \cdot (1 - t_n), t_n > 0.5. \end{cases} \quad (7)$$

式(7)中的 $\nu$ 在(1, 2)之间时,Tent具有混沌效应,而且混沌效果随着 $\nu$ 增加而增强,故本文将 $\nu$ 设置为  $2 - 10^{-5}$ (对Tent迭代公式计算10000次,记录每个点所在位置).

混沌扰动的添加方式如下式所示:

$$X'_n = (X_{\max} - X_{\min}) \cdot (t - 0.5) + (X_{\max} + X_{\min}) \cdot 0.5, \quad (8)$$

其中 $X_{\max}, X_{\min}$ 分别为个体的上限和下限.

## 3 算法的收敛性

混沌“微变异”自适应遗传算法的流程可以简述如下:

1) 随机产生初始种群 $P(0)$ ,令 $t = 0, P(t)$ 中适应

度函数值最高的个体记为  $x_0(t)$ , 称为超级个体;

2) 计算  $P(t)$  中所有个体的适应度函数值;

3) 交叉操作: 用自适应交叉算子进行交叉, 使得  $P(t)$  变成  $P_1(t)$ ;

4) 变异操作: 用自适应变异算子对种群中的每一个个体进行变异操作, 使得  $P_1(t)$  变成  $P_2(t)$ ;

5) 选择操作: 对任意的  $P_2(t)$ , 进行选择与“微变异”后变成  $P(t+1)$ , 其中每个个体被选择的概率大于零;

6) 更新超级个体: 在新的种群中, 通过  $x_0(t) \cup P(t+1)$  选出新的超级个体, 记为  $x_0(t+1)$ , 称该过程为更新超级个体过程;

7) 判断是否满足结束条件, 若满足则算法结束, 不满足则重新进入3)。

个体空间设为  $E$ , 基数记为  $|E| = N$ . 有限种群规模 ( $n \geq 2$ ), 编码方式不限, 设种群中最佳个体为  $Z_i^*$  ( $i = 1, 2, \dots, n$ ), 状态空间中最佳个体为  $f^*$ , 状态空间为  $|E^n| = |E|^n = N^n$ . 算法中, 经过选择+“微变异”后概率变为  $S$ , 自适应交叉后概率变为  $C$ , 自适应变异概率变为  $M$ . 因此, 由遗传操作引起的概率变化整体为  $P = CMS^{[15]}$ .

**引理1**  $S$  为列容许的随机矩阵.

**证明** 对状态编号为  $1, 2, \dots, N^n$ , 由于  $\forall \alpha \in E, \text{prob}\{a \xrightarrow{S} b\} > 0$ , 其中  $\text{prob}\{a \xrightarrow{S} b\}$  表示  $a$  经过选择+“微变异”后仍可以被选上的概率大小, 可以得出  $\forall i \in E^n$ , 记  $i = \{x(1) \dots x(n)\}$ ,  $S = (s_{ij})_{N^n \times N^n}$ ,  $s_{ij}$  表示经过选择+“微变异”从  $i$  转移到  $j$  的概率, 则有

$$s_{ii} = \prod_{k=1}^n \text{prob}\{x(k) \xrightarrow{S} x(k)\}, \quad (9)$$

因此  $S$  为列容许的.  $\square$

**引理2**  $M$  为正随机矩阵.

**证明**  $\forall \alpha, b \rightarrow E$ , 存在  $\text{prob}\{a \xrightarrow{M} b\} > 0$ . 状态个数  $N^n$  有限, 编号记为  $1, 2, \dots, N^n$ , 则对于任意状态  $i = \{x(1) \dots x(n)\}$ , 状态  $j = \{y(1) \dots y(n)\}$ , 若记  $M = (m_{ij})_{N^n \times N^n}$ , 表示经自适应变异从  $i$  转移到  $j$  的概率, 则有

$$m_{ii} = \prod_{k=1}^n \text{prob}\{x(k) \xrightarrow{M} x(k)\} > 0, \quad (10)$$

因此  $M$  为一正随机矩阵.  $\square$

**定理1**  $C, M, S$  是随机矩阵, 当  $M$  是正定的且  $S$  是列可容时,  $P = CMS$  是正定的.

**证明** 设  $A = CM, B = AS$ , 由于  $C$  是随机矩阵,  $C$  的每一个行至少有一个正元素. 对于任意的

$i, j \in (1, \dots, n)$ , 有  $\alpha_{ij} = \sum_{k=1}^n c_{ik} \cdot m_{kj} > 0$ , 即  $A > 0$ .

同理, 由于  $S$  是列可容的, 对于任意  $i, j \in (1, \dots, n)$ , 有  $b_{ij} = \sum_{k=1}^n \alpha_{ik} \cdot s_{kj} > 0$ , 即  $B > 0$ .  $\square$

为便于描述作如下假设: 问题有且只有一个最优解, 超级个体保留后的群体  $x_0(t+1) \cup P(t+1)$ , 此时的群体被记为一种新的状态, 于是每个状态都由  $n+1$  组成, 因此状态空间的基数为  $N^{n+1}$ . 通过分析, 状态可按照如下规则进行排序:

1) 随着适应度函数值的增高, 超级个体的编码将变小.

2) 超级个体相同时, 群体中的最佳个体适应度函数值越高, 状态顺序编码在群体中越小.

3) 超级个体不同时, 群体空间的所有个体排序顺序编号均相同. 若状态  $i$  的超级个体用  $x_0(i)$  表示, 其余个体用  $x_1(i) \dots x_n(i)$  表示, 则状态  $i$  可以表示为  $x_0(i)x_1(i) \dots x_n(i)$ , 按这种规则给状态排序, 可知:

① 每一个超级个体都存在于  $N^n$  状态中;

② 超级个体成为全局最优个体时, 其状态必然排在靠前的  $N^n$  个状态中.

实际上, 对于个体空间  $E$ , 其中的任意个体都能被定义为超级个体, 因此, 存在的超级个体有  $N$  个之多. 在众多超级个体中, 每一个都有可能成为任意一个群体的超级个体, 所以每个超级个体对应  $N^n$  个状态, 把这  $N^n$  状态看成一个状态组, 于是存在  $N$  个状态组, 记为  $A_1, A_2, \dots, A_N$ . 另外, 因为超级个体不参加选择、交叉和变异的遗传操作, 并且任意超级个体所对应的  $N^n$  个状态包含着群体空间中所有可能的  $N^n$  个状态, 所以, 选择、交叉和变异的遗传操作每个环节后的状态依然存在于这个状态组中; 在精英保留策略之前, 遗传算法中各个操作的结果依然在该状态组中; 在精英保留策略之后, 选择、交叉和变异的转移矩阵分别为

$$\tilde{S} = \begin{bmatrix} S & & & \\ & S & & \\ & & \ddots & \\ & & & S \end{bmatrix}, \tilde{C} = \begin{bmatrix} C & & & \\ & C & & \\ & & \ddots & \\ & & & C \end{bmatrix}, \quad (11)$$
$$\tilde{M} = \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{bmatrix}.$$

因此, 经过遗传操作后的矩阵为

$$\tilde{C}\tilde{M}\tilde{S} =$$

$$\begin{bmatrix} CMS & & & \\ & CMS & & \\ & & \ddots & \\ & & & CMS \end{bmatrix} = \begin{bmatrix} P & & & \\ & P & & \\ & & \ddots & \\ & & & P \end{bmatrix}, \quad (12)$$

其中  $P = CMS$ .

在各种遗传操作之后,需要更新超级个体,也就是用  $b(i) = \operatorname{argmax}\{f(x_k(i)) | k = 1, 2, \dots, n\}$  来标记状态  $i$  所对应的群体中不算超级个体之外的最好个体,其中  $f(x_k(i))$  表示个体适应度函数值,表示为  $i = x_0(i)x_1(i) \dots x_n(i), j = b(i)x_1(i) \dots x_n(i)$ . 如果用  $U = (u_{ij})_{N^{n+1} \times N^{n+1}}$  表示因需要更新超级个体而产生的状态转移矩阵,则对于  $i = x_0(i)x_1(i) \dots x_n(i)$ , 当  $f(x_0(i)) < f(b(i))$  时,有  $u_{ij} = 1, u_{ir} = 0 (r \neq j)$ ; 否则,有  $u_{ii} = 1$  (即  $x_0(i)$  不会被  $b(i)$  更新),  $u_{ij} = 0 (j \neq i)$ . 因此,  $U$  中的每行中有且只有一个元素为1,其余为零.

更新后超级个体的状态转移矩阵  $U$  可由矩阵  $U_{rk} (1 \leq r, k \leq N)$  表示,有

$$U = \begin{bmatrix} U_{11} & U_{12} & \dots & U_{1N} \\ U_{21} & U_{22} & \dots & U_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ U_{N1} & U_{N2} & \dots & U_{NN} \end{bmatrix}. \quad (13)$$

下面对矩阵  $U_{rk} (1 \leq r, k \leq N)$  进行分析. 首先对于状态  $i = x_0(i)x_1(i) \dots x_n(i)$ , 设超级个体  $x_0(i)$  在个体空间  $E$  中按由优到劣排到第  $r$  位, 记为  $a_r$ , 则状态  $i$  应排在状态组  $A_r$  中. 设  $q(i)$  为状态  $i$  在状态组  $A_r$  中的排序编号, 则有  $i = (r-1)N^n + q(i)$ . 在每个状态组  $A_r$  中,除了超级个体之外,最好个体为  $a_k (1 \leq k \leq N)$  的状态的总数为  $n_k$ . 对于不同的状态组,  $n_k$  是相同的, 并有  $\sum_{k=1}^N n_k = N^n$ .

**引理3** 在任意确定的状态组  $A_r (r = 1, 2, \dots, N)$  中,除超级个体  $a_r$  外最好的个体记为  $a_k (1 \leq k \leq N), a_k$  的状态的总数记为  $n_k, n_k = (N - k + 1)^n - (N - k)^n$ .

**证明** 在状态组  $A_r$  中,考虑除去超级个体  $a_r$ , 剩余  $n$  个个体的状态共有  $(N - k + 1)^n$  个,其中不包含个体  $a_1, a_2, \dots, a_{k-1}$  的状态. 进而在这  $(N - k + 1)^n$  个状态中,最好的个体不是  $a_k$  的状态,其不包含个体  $a_1, a_2, \dots, a_{k-1}$  的状态共有  $(N - k)^n$  个,因此引理得证;除超级个体  $a_r$  外最好的个体为  $a_k, a_k$  的状态总数为  $(N - k + 1)^n - (N - k)^n$ .  $\square$

**定理2** 混沌“微变异”自适应遗传算法能收敛到全局最优解.

**证明** 子矩阵  $PU_{11} = P > 0$ ,集中包含了全局适应度函数值最高个体的状态变换概率. 由于  $P$  是随机矩阵,并且公式中  $R \neq 0$ ,由引理1可以推断出所有的非全局最优解收敛概率都等于零;换言之,全部包含全局最优解的收敛概率一定等于1,表达式为

$$\lim_{t \rightarrow \infty} P(Z_i^*(t) = f^*) = 1. \quad (14)$$

定理得证.  $\square$

### 4 仿真分析

文献[16]对混沌遗传算法与经典遗传算法进行了对比分析,通过3个传统优化函数的仿真计算,对比了混沌遗传算法、经典遗传算法和粒子群算法的收敛速度和收敛精度,验证了混沌遗传算法在收敛速度和收敛精度上好于经典遗传算法和粒子群算法. 3个函数分别为

函数1

$$f = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5i \cdot x_i\right)^2 + \left(\sum_{i=1}^n 0.5i \cdot x_i\right)^4, \quad n = 30, x_i \in (-5, 10); \quad (15)$$

函数2

$$f = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi \cdot x_i) + 10], \quad n = 30, x_i \in (-5.12, 5.12); \quad (16)$$

函数3

$$f = 20 + e^{-20e^{\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right)}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi \cdot x_i)}, \quad n = 10, x_i \in (-32, 32). \quad (17)$$

其中:函数1为单峰值问题,全局只有一个极点,该函数用来测试算法的收敛速度和准确性;函数2为多峰值问题,全局具有多个极小点,该函数用来测试算法的局部搜索能力;函数3也为多峰值问题,全局具有多个极小点,也用来测试算法的局部搜索能力.

为了便于比较研究,利用本文提出的混沌“微变异”自适应遗传算法对函数1、函数2和函数3进行仿真计算,将得到的寻优平均代数与文献[16]中数据进行对比,如表1所示.

表1 本文算法的平均代数与文献[16]数据对比

算法	函数1	函数2	函数3
粒子群算法	1229	1254	1928
经典遗传算法	410	340.4	382
混沌遗传算法	334	318	304
本文算法	45	40	42

为进一步验证算法的有效性,引入文献[11]中的两个函数,定义为函数4和函数5,即

函数4

$$f(x_1, x_2) = (x_1^2 + x_2^2)^{0.25} \cdot \{\sin^2[50(x_1^2 + x_2^2)^{0.1}] + 1.0\},$$

$$x_1, x_2 \in (-100, 100); \quad (18)$$

函数5

$$f(x_1, x_2) = \left\{ \sum_{i=1}^5 i \cos[(i+1) \cdot x_1 + i] \right\} \cdot \left\{ \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \right\},$$

$$x_1, x_2 \in (-10, 10). \quad (19)$$

再次利用文献[16]中的混沌遗传算法和本文提出的混沌“微变异”自适应遗传算法对函数4和函数5进行仿真分析,可以得到图1和图2.

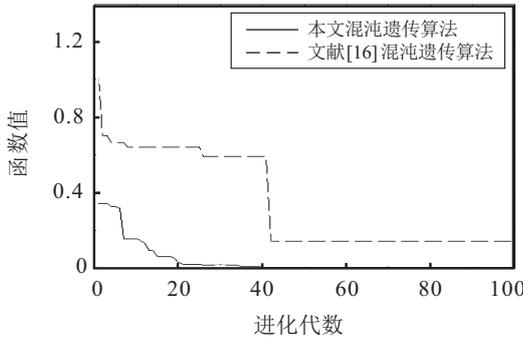


图1 函数4的仿真对比曲线

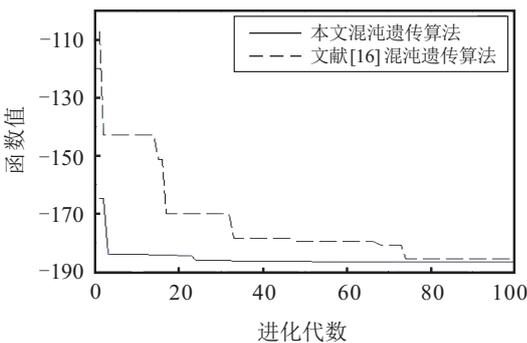


图2 函数5的仿真对比曲线

通过上述5个函数的对比分析,可以得出如下结果:

1)通过表1的数据对比可知,混沌“微变异”自适应遗传算法的迭代次数都不足50次,远远小于在相关文献中提出的经典遗传算法、粒子群算法和混沌遗传算法,表明了本文所提出的遗传算法收敛速度更快.

2)通过图2、图3的对比分析可知,文献[16]的混沌遗传算法在参数收敛的计算准确性上不如本文的

混沌“微变异”自适应遗传算法,说明本算法提高了遗传算法在参数收敛时的计算准确性.

3)通过图2、图3的对比分析可知:文献[16]的混沌遗传算法接近最优解时,陷入了局部最优解;而本文的混沌“微变异”自适应遗传算法仍然可以继续搜索寻优,表明本文遗传算法具有更强的全局寻优能力,不易陷入局部最优解.

## 5 结论

本文基于混沌优化的随机性和遍历性,结合遗传算法的并行性和可扩展性,提出了混沌“微变异”自适应遗传算法.该算法在选择算子中增加了“微变异”操作,对交叉算子和变异算子进行了自适应调整,从而保证了遗传操作产生新个体的有效性.在遗传操作结束后,又对群体中的最优个体进行了保留,对适应度函数值低的个体添加了混沌扰动.通过5个函数优化,并与已有文献的结果进行了比较分析,仿真结果表明,混沌“微变异”自适应遗传算法不仅加快了寻优速度,而且提高了寻优精度,克服了遗传算法在处理复杂的组合优化问题时易陷入局部最优的问题.

## 参考文献(References)

- [1] Zhan Z L, Wang Q. An improved chaos immune genetic algorithm[C]. International Conference on Mechatronic Science, Electric Engineering and Computer. Jilin, 2011: 1147-1150.
- [2] Zhang Y, Zhao L, Pan W. Radar fault diagnosis based on chaos genetic reduction algorithm[C]. The 27th Chinese Control and Decision Conference (CCDC). Qingdao, 2015: 1376-1381.
- [3] Huang W L, Zhang T. An improved genetic algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows[C]. The 35th Chinese Control Conference (CCC). Chengdu, 2016: 9639-9643.
- [4] 张先炼, 王国杰. 混合遗传算法综述[J]. 电子世界, 2015(14): 120-121.  
(Zhang X L, Wang G J. A review of hybrid genetic algorithms[J]. Electronics World, 2015(14): 120-121.)
- [5] Li D J, Li Y Y, Li J X, et al. Gesture recognition based on BP neural network improved by chaotic genetic algorithm[J]. International Journal of Automation and Computing, 2018, 15(3): 267-276.
- [6] Wang S X, Wang Z Q, Li Z X. Fuzzy modeling of thermal process based on chaos genetic algorithm[C]. The 8th World Congress on Intelligent Control and Automation (WCICA). Jinan, 2010: 5851-5855.

[7] Guesmi R, Ben Farah M A, Kachouri A, et al. A novel design of chaos based S-boxes using genetic algorithm techniques[C]. 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA). Doha, 2014: 678-684.

[8] 周强锋. 基于混沌遗传算法的宽零陷波束赋形方法[J]. 雷达科学与技术, 2020, 18(3): 262-266. (Zhou Q F. Wide null steering based on chaos genetic algorithm[J]. Radar Science and Technology, 2020, 18(3): 262-266.)

[9] Ghazvini M, Mirzadi M, Parvar N. A modified method for image encryption based on chaotic map and genetic algorithm[J]. Multimedia Tools and Applications, 2020, 79(37/38): 26927-26950.

[10] 宋莹莹, 王福林, 兰佳伟. 基于跳跃基因算子的改进实数遗传算法[J]. 控制与决策, 2020, 35(9):2277-2284. (Song Y Y, Wang F L, Lan J W. Improved real-coded genetic algorithm based on jumping gene operator[J]. Control and Decision, 2020, 35(9): 2277-2284.)

[11] Xu X Q, Lei L. The research of advances in adaptive genetic algorithm[C]. 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). Xi'an, 2011: 1-6.

[12] 王春阳, 赵玉庆, 谢金兴, 等. 遗传算法变异算子的改进[J]. 山东农业大学学报: 自然科学版, 2019, 50(5): 898-901. (Wang C Y, Zhao Y Q, Xie J X, et al. An improved genetic algorithm variation operator[J]. Journal of Shandong Agricultural University: Natural Science Edition, 2019, 50(5): 898-901.)

[13] 李书全, 孙雪, 孙德辉, 等. 遗传算法中的交叉算子的述评[J]. 计算机工程与应用, 2012, 48(1): 36-39. (Li S Q, Sun X, Sun D H, et al. Summary of crossover operator of genetic algorithm[J]. Computer Engineering and Applications, 2012, 48(1): 36-39.)

[14] 张宏伟, 张向锋, 文传博. 一种含全交叉算子的改进遗传算法[J]. 上海电机学院学报, 2017, 20(4): 196-200. (Zhang H W, Zhang X F, Wen C B. An improved genetic algorithm with total crossover operator[J]. Journal of Shanghai Dianji University, 2017, 20(4): 196-200.)

[15] 孙洋. 改进混沌遗传算法研究及其在炮兵火力分配上的应用[D]. 沈阳: 东北大学, 2016: 36-40. (Sun Y. Improved chaos genetic algorithm and its application in artillery firepower distribution[D]. Shenyang: Northeastern University, 2016: 36-40.)

[16] Mohammad J, Roghiyeh H. Chaos genetic algorithm instead genetic algorithm[J]. The International Arab Journal of Information Technology, 2015, 12(2): 1-6.

作者简介

潘伟(1975—), 男, 副教授, 博士, 从事非线性系统的鲁棒控制、遗传算法应用等研究, E-mail: pan.w@126.com;

丁立超(1990—), 女, 讲师, 硕士, 从事计算机视觉识别、遗传算法应用的研究, E-mail: onlydingding@qq.com;

黄枫(1992—), 男, 助教, 硕士, 从事非线性系统的鲁棒控制、遗传算法应用的研究, E-mail: 867894005@qq.com;

孙洋(1991—), 男, 工程师, 硕士, 从事混沌遗传算法的研究, E-mail: 495851348@qq.com.

(责任编辑: 李君玲)

下 期 要 目

电动汽车时代的电网-交通网协同优化综述 ..... 郭 戈, 等

大规模固定翼无人机集群编队控制方法 ..... 王祥科, 等

参数不确定离散时间系统的有限时间输出反馈预见控制器设计 ..... 李 丽, 等

基于R2指标和目标空间分解的高维多目标粒子群优化算法 ..... 李 飞, 等

敏感度函数未知下的非均匀直线覆盖控制算法设计与PLEXE仿真 ..... 左 磊, 等

具有执行器故障的四旋翼无人机自适应预定性能控制 ..... 赵广磊, 等

空间机器人双臂捕获卫星操作的事件采样输出反馈神经网络避撞柔顺控制 ..... 曾晨东, 等

平台品牌赋能情境下考虑信息不对称的供应链渠道冲突 ..... 肖 迪, 等

面向建材装备集团制造的分布式多项目资源调度 ..... 张 锦, 等

基于混合模型驱动的红外与可见光图像融合 ..... 沈 瑜, 等