

# 混合遗传算法及在多处理机调度问题中的应用<sup>①</sup>

张长伟, 李小青

(浙江万里学院, 宁波 315101)

**摘要:** 针对遗传算法存在收敛速度较慢, 易陷入局部极值的缺点, 通过算法混合, 提出一种基于混沌及差分演化的混合遗传算法。该算法利用混沌运动的遍历性和内在随机性择优产生初始群体, 借鉴差分进化算法中的繁殖策略, 使染色体在解空间中更有效的搜索最优解。最后将该混合遗传算法应用于多处理机调度问题中, 实验表明, 该混合算法具有较高的优化效率, 能寻找到更好的优化结果。

**关键词:** 遗传算法; 混沌; 差分演化; 多处理机调度

## Mixed Genetic Algorithms and its Application to Multiprocessor Scheduling Problem

ZHANG Chang-Wei, LI Xiao-Qing

(Zhejiang Wanlin University, Ningbo 315101, China)

**Abstract:** The genetic algorithm convergence slower, easily into the local extremum faults. Through the algorithm is proposed based on hybrid differential evolution of chaos and hybrid genetic algorithm. The algorithm utilizes chaotic motions of the ergodicity and intrinsic stochastic according to create initial group, the evolutionary algorithm for difference strategy, make the chromosomes in solution space more effective search optimal solution. Finally, this hybrid genetic algorithm applied to the multi-processor scheduling problem, the experimental results show that the hybrid algorithm is high efficiency for the optimization, can better optimization results.

**Keywords:** genetic algorithm; chaos; differential evolution; multiprocessor scheduling

### 1 前言

遗传算法(genetic algorithm, GA)是模拟生物进化过程中优胜劣汰规则与群体内部染色体信息交换机制的一类处理复杂优化问题的新方法。GA操作简单, 实用性强, 具有良好的全局搜索能力, 但对于局部空间的搜索问题不是很有效, 容易产生早熟收敛, 个体的多样性减少得很快<sup>[1,2]</sup>。而演化算法(differential evolution algorithm, DEA)是一类求解连续全局优化问题的演化算法, 其基本思想在于运用当前种群个体的差来重组得到中间种群, 然后运用直接的父子混合个体适应值竞争来获得新一代种, 该算法实施随机、并行、直接的全局搜索, 简单易用, 稳健性好。

本文将混沌优化、差分演化以及遗传算法3种优化算法有机结合起来, 提出一种基于混沌差分演化的混合遗传算法。该算法利用混沌运动的遍历性和内在

随机性择优产生初始群体, 借鉴差分进化算法中的繁殖策略, 使染色体在解空间中更有效的搜索最优解。最后将该混合遗传算法应用于多处理机调度问题中。

### 2 融合混沌和差分演化算法的混合遗传算法

#### 2.1 混沌初始种群

由于混沌运动具有遍历性、随机性等特点, 混沌运动能在一定范围内按照其自身的规律不重复的遍历所有的状态。由混沌序列搜索产生的初始种群, 数字取值会均匀的分布在解的空间中, 减少了可能出现的数据冗余, 提高了搜索效率。本文采用 Logistic 方程产生初始种群, 该方程定义为:

$$\chi_{n+1} = \mu\chi_n(1 - \chi_n) \quad (1)$$

式中控制参数  $\mu = 4$ , 抗体数  $n = 0, 1, 2, \dots$ ,

<sup>①</sup> 收稿时间:2010-09-17;收到修改稿时间:2010-10-18

$\chi_n$  代表第  $n$  个抗体,  $\chi_{n+1}$  代表第  $n+1$  个抗体。将  $n$  个混沌变量  $\chi_c$  作逆映射得到相应优化空间中的变量  $m\chi_i^r$

$$m\chi_i^k = a_i^k + x_i^k (b_i^r - a_i^r) \quad (2)$$

式中  $k$  为混沌变量的迭代序号,  $r$  为细搜索标志。如果  $f(m\chi_i^k) < f^*$ , 则  $f^* = f(m\chi_i^k)$ ,  $x_i^* = x_i^k$ , 利用下式进行优化搜索:

$$x_i^{k+1} = 4x_i^k(1 - x_i^k), k = k + 1 \quad (3)$$

### 2.2 构造适应度函数

适应度函数体现了生物进化论“适者生存”理论中的“适者”概念, 是评价个体优劣的指标, 适应度函数的选择应该能够较为准确地反映解的优劣性。本文用惩罚函数法处理约束条件<sup>[3]</sup>, 构造如下适应度函数  $F(\zeta)$ :

$$F(\zeta) = \begin{cases} f(\zeta) \\ f(\zeta) + a \left\{ \sum_{k=1}^s [\max\{0, g_k(\zeta)\}] + \sum_{k=1}^s |h_k(\zeta)| \right\} \end{cases} \quad (4)$$

式中,  $h_k(\zeta) = 0$  表示求解优化问题中约束条件的等式约束,  $g_k(\zeta) \leq 0$  表示求解优化问题中约束条件的不等式约束,  $a$  为罚因子。

### 2.3 自适应交叉概率和变异概率

遗传算法的参数中交叉概率  $p_c$  和变异概率  $p_m$  的选择是影响遗传算法性能的关键。为了使得在进化初期有较强的全局搜索能力和较弱的局部搜索能力, 而随着进化的进行, 全局搜索能力逐渐减弱, 局部搜索能力逐渐增强, 防止“早熟”现象出现, 需要遗传算子必须根据种群的进化情况, 随时调整进化策略<sup>[4]</sup>。本混合遗传算法中, 交叉概率和变异概率按如下公式进行自适应调整:

$$p_c = \begin{cases} p_{c0}, & N \leq N_f \\ a - (a - p_{c0}) \frac{N_f}{N}, & N > N_f \end{cases} \quad (5)$$

$$p_m = \begin{cases} p_{m0}, & N \leq N_f \\ \beta - (\beta - p_{m0}) \frac{N_f}{N}, & N > N_f \end{cases} \quad (6)$$

式中,  $p_{c0}$  为交叉概率初始值,  $p_{m0}$  为变异概率初始值,  $a$  和  $\beta$  为小于 1 的常数,  $N$  为具有相同搜索结果的进化代数。

### 2.4 算法流程

Step 1: 初始化: 确定出某个问题解的表现类型和潜在的解空间、目标函数的类型及其数学描述形式和量化方法及编码方式; 设置交叉概率  $P_c$ 、变异概率  $P_m$  和最大迭代数  $Gen$ 。

Step 2: 混沌初始化得到初始群体。

Step 3: 计算适应度: 计算群体中每个个体  $\chi_i$  的适应度  $f(\chi_i)$ ,  $i=1, 2, \dots, M$ 。并判断是否符合进化终止条件, 若符合输出最佳结果并结束计算, 否则转向 Step 4。

Step 4: 用排序法选择父代染色体用来繁殖<sup>[5,6]</sup>。

Step 5: 交叉操作产生第 2 个子代群体: 根据选择概率  $p_s(i) = F(i) / \sum_{i=1}^n F(i)$  随机选择一对父代个体

$y(j, i_1)$  和  $y(j, i_2)$  作为双亲, 共产生  $n$  个子代个体。

Step 6: 变异操作产生第 3 个子代群体: 以  $p_m(i) = 1 - p_s(i)$  的概率选择个体  $y(j, i)$ , 产生子代个体  $y_3(j, i)$ 。

Step 7: 差分算子: 对每个染色体  $x_{i,G}, i=1, 2, \dots, NP$ , 产生  $v_{i,G+1} = x_{r1,G} + F(x_{r2,G} - x_{r3,G})$  变异向量, 以  $x_{i,G}, i=1, 2, \dots, NP$ , 采用递减策略的缩放因子

$$F = F_{\max} - (F_{\max} - F_{\min}) \left( \frac{g}{G - \max} \right)^2,$$

按  $y(j, i) = y(j, i_1) + F \times [y(j, i_2) - y(j, i_3)]$  生成差分个体  $y(j, i)$ 。

Step 8: 循环: 遗传及差分演化算法产生的子代染色体合并, 按适应度排序, 取前 50% 染色体, 判断是否满足终止条件。

## 3 混合遗传算法在多处理机调度问题应用

### 3.1 多处理机调度问题描述及数学模型

设有台相同的处理机  $p_1, p_2, \dots, p_n$ , 处理  $m$  个独立的作业  $J_1, J_2, \dots, J_m$ , 以互不相关的方式工作, 任何作业可以在任何处理机上运行, 未完成的作业不允许中断, 作业也不能拆分为更小的子作业。多处理机调度问题 (Multiprocessor Scheduling Problem, MSP) 就是给出一种作业调度方案, 使个作业在尽可能短的时间内由这台相同的处理机完成。

已知作业  $J_i$  需要的处理机时间为  $t_i (i=1, 2, \dots, m)$ , 若作业  $i$  分配到处理机  $j$  上处理, 令

$x_{ij} = 1$ , 否则令  $x_{ij} = 0$ 。  $\sum_{i=1}^m x_{ij}t_i$  表示处理机  $j$  完工时间,  $\sum_{j=1}^n x_{ij} = 1$  表示作业  $i$  只能分配到一个处理机上, 因此可用如下的数学模型表示多处理机调度问题:

$$\min \max_{1 \leq j \leq n} \sum_{i=1}^m x_{ij}t_i$$

$$s.t. \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, 2, \dots, m) \quad (7)$$

$x_{ij} = 0, 1$   
把(7)式变换如下:

$$\min v$$

$$s.t. \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, 2, \dots, m)$$

$$\sum_{i=1}^m x_{ij}t_i \leq v \quad (j = 1, 2, \dots, n) \quad (8)$$

$$x_{ij} = 0, 1$$

### 3.2 仿真实验

引用参考文献[7]的一个实例: 某汽车发动机厂金工车间要加工 12 个工件, 每个工件都有车、刨、磨 3 个工序, 现有 3 台车床, 2 台刨床, 4 台磨床, 每台机床的加工能力不同, 加工数据如表 1 所示:

表 1 各机器加工数据

工件	工序 1			工序 2				工序 3		
	机器 1	机器 2	机器 3	机器 4	机器 5	机器 6	机器 7	机器 8	机器 9	
1	2	2	3	4	5	2	3	2	3	
2	4	5	4	3	4	3	4	5	4	
3	6	5	4	4	2	3	4	2	5	
4	4	3	4	6	5	3	6	5	8	
5	4	5	3	3	1	3	4	6	5	
6	6	5	4	2	3	4	3	9	5	
7	5	2	4	4	6	3	4	3	5	
8	3	5	4	7	5	3	3	6	4	
9	2	5	4	1	2	7	8	6	5	
10	3	6	4	3	4	4	8	6	7	
11	5	2	4	3	5	6	7	6	5	
12	6	5	4	5	4	3	4	7	5	

采用基于混度与差分演化思想的混合遗传算法进行优化。设初始群体数 popsize= 500, 算法最大迭代数 Gen=200, 采用 3 参数级联编码, 编码长度 96 位。仿真实验得到的最优解是 24。对应的机器加工情况和加工时间如下:

M1: (2,1)-(12,1)-(8,1):13

M2:(11,1)-(4,1)-(7,1)-(6,1)-(1,1)-(3,1):19

M3:(9,1)-(10,1)-(5,1):11

M4:(11,2)-(2,2)-(9,2)-(7,2)-(10,2)-(6,2)-(1,2):19

M5:(4,2)-(12,2)-(5,2)-(8,2)-(3,2):22

M6:(11,3)-(2,3)-(7,3)-(5,3)-(1,3):24

M7:(4,3)-(12,3)-(6,3):23

M8:(9,3)-(10,3)-(3,3):24

M9:(8,3):24

对照文献[7]用遗传算法优化的结果, 如表 2 所示, 混合遗传算法效果要优于一般遗传算法, 且能很快地收敛到最优解, 如图 1 所示。

最优解	运行次数	1	2	3	4	5	6	7	8	9	10
遗传算法		29	30	29	29	29	31	29	29	29	30
混合遗传算法		26	24	24	24	26	24	24	24	25	24

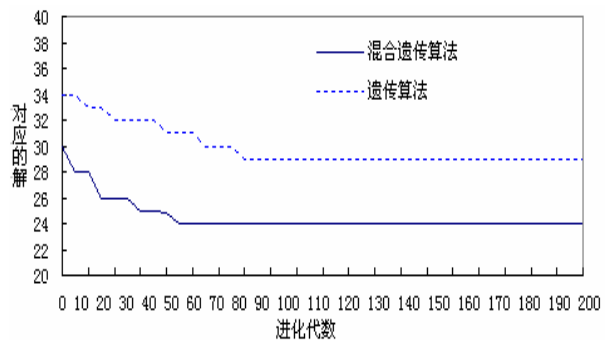


图1 遗传算法与混合遗传算法性能示意图

### 4 结语

针对遗传算法存在收敛速度较慢, 易陷入局部极值问题, 将混沌及差分演化思想融入到遗传算法中, 提出一种混合遗传算法。算法利用混沌运动的遍历性和内在随机性择优产生初始群体, 借鉴差分进化算法中的繁殖策略, 使染色体在解空间中更有效的搜索最优解。仿真实验表明, 混合遗传算法能有效避免早熟收敛问题, 具有较高的优化效率。

### 参考文献

1 Piszcz A, Soule T. Genetic Programming: optimal population sizes for varying complexity problems. Proc. of the 8th Annual Genetic and Pvolutionary Computation Conference. Seattle, WA, USA, 2006: 953-954.

(下转第 215 页)

表 5 ORL 库上本文算法与其他方法对比试验

	训练时间 (s)	识别时间 (s)	平均识别率 (%)
PCA	196.26	0.24	90.23
LDA	49.57	0.09	90.74
LPP	51.24	0.11	91.17
本文算法	25.63	0.10	92.41

从实验结果可以得出,本文算法在平均识别率上强于其他算法,在识别时间上,虽然略不如 LDA 算法,但已非常接近。

## 5 结论

针对 2DPCA 的人脸识别算法维数过高的问题,提出了一种 2DPCA 改进算法,经过两次 2DPCA 特征提取,能够有效的降维。在 Yale 与 ORL 标准人脸数据库上实验结果证明,与传统 2DPCA 算法相比,改进后的 2DPCA 算法在保证较好的识别率的同时,并具有较快的识别速度与较少的数据存储空间,与其他经典算法相比,此算法在平均识别率上具有一定的优势。但在实验中发现,对不同训练样本,如 Yale 与 ORL 数据库,识别率不尽相同,如何探索更高并且更稳定的识别方法有待进一步研究。

## 参考文献

- 1 Turk M, Pentland A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991,3(1):71-86.
- 2 Shufelt JA. Performance evaluation and analysis of monocular building extraction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1997,19(4):311-326.
- 3 Osuna E, Freund R, Girosi F. Training Support Vector Machines an Application to Face Detection. *Proc. Computer Vision and Pattern Recognition*, 1997:130-136.
- 4 Lin SH, Kung S, Lin LJ. Face recognition detection by probabilistic decision based neuralnetwork, *Neural Networks*, 1997,8(1):114-132.
- 5 Lu J, Plataniotis K, Venetsanopoulos A. Face recognition using LDA-based algorithms. *IEEE Trans. on Neural Networks*, 2003,14(1):195-200.
- 6 Yang J, Zhang D, Frangi AF, Yang JY. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2004,26(1):131-137.
- 7 Keun-Chang K, Pedrycz W. Face recognition using an enhanced in-dependent component analysis approach. *IEEE Trans. on Neural Networks*, 2007,18(2):530-541.
- 8 Matej Arta\_c et al. incremental PCA for on-line visual learning and recognition. *Pattern Recognition*, 2002, 781-784.
- 9 Xiaoming Liu, Tsuhan Chen, Susan M. Thomson.eigenspace updating for Non-Stationary Process and its Application to Face recognition. *Pattern Recognition*, 2002,(9).
- 10 Yang J, Yang JY. From image vector to matrix: A straightforward image projection technique-IMPCA vs. PCA. *Pattern Recognition*, 2002,35(9):1997-1999.
- 1 Lopez-Cruz I. L. Efficient Evolutionary Algorithms for Optimal Control [Ph.D. Thesis]. 2002.
- 2 刘武,等.天然气管网优化调度方法研究.西南石油大学学报,2009,31(3):146-149.
- 3 黄江波,一种基于自适应遗传算法统一潮流控制器.重庆理工大学学报,2010,24(3):81-84.
- 4 周江林,差分加速遗传算法及其在市政工程中的应用[硕士学位论文].合肥:合肥工业大学,2007.
- 5 Li D, Wang L. A study on the optimal population size of genetic algorithm. *Proc. of the 4th World Congress on Intelligent Control and Automation*. Shanghai, China, 2002: 3019-3021.
- 6 王万良,吴启迪.生产调度智能算法及其应用.北京:科学出版社,2007.27-30.

(上接第 146 页)