

文章编号: 1001-0920(2008)09-1073-04

## 基于克隆选择和粒子群思想的动态多群体优化算法

王巧灵<sup>1</sup>, 高晓智<sup>2</sup>, 王常虹<sup>1</sup>, 刘福荣<sup>1</sup>

(1. 哈尔滨工业大学 空间控制与惯性技术研究中心, 哈尔滨 150001;

2. 赫尔辛基工业大学 智能电力电子实验室, 埃斯波 02150, 芬兰)

**摘要:** 针对粒子群算法和克隆选择原理的特点, 提出了基于克隆选择和粒子群思想的动态多群体优化算法. 该算法将整个群体分为若干子群体, 在子群体内部应用基本的粒子群算法, 以子群体作为抗体设计了克隆、变异、选择和受体编辑算子. 变异算子使子群体动态变化实现子群体间相互交换信息, 具有良好的全局搜索能力. 实验结果表明, 该算法具有寻优能力强、搜索精度高的优点, 可用于工程问题中具有各种特性的复杂函数优化.

**关键词:** 克隆选择; 粒子群; 优化算法; 多维函数优化; 多群体

**中图分类号:** TP301 **文献标识码:** A

## Dynamic multi-swarm optimization based on clonal selection and particle swarm

WANG Qiao-ling<sup>1</sup>, GAO Xiaozhi<sup>2</sup>, WANG Chang-hong<sup>1</sup>, LIU Fu-rong<sup>1</sup>

(1. Space Control and Inertial Technology Research Center, Harbin Institute of Technology, Harbin 150001, China;

2. Institute of Intelligent Power Electronics, Helsinki University of Technology, Espoo 02150, Finland.

Correspondent: WANG Qiao-ling: qiaoling\_830305 @163.com)

**Abstract:** Based on the promising fusion of the clonal selection and particle swarm principles, a dynamic multi-swarm optimization algorithm is proposed. In the approach, the whole swarm is divided into dynamic subpopulations, which are considered as the evolving antibodies. These subpopulations are further optimized by using the particle swarm method to increase the necessary antibody diversity. Moreover, they can exchange useful optimization information among themselves during the iteration procedure. The cloning, hypermutation, selection and receptor editing operators are also employed in the proposed hybrid scheme. Simulations demonstrate that the optimization algorithm can overcome the premature and slow convergence drawbacks of the standard particle swarm and clonal selection methods, and it is very effective in dealing with the challenging nonlinear function optimization problems.

**Key words:** Clonal selection; Particle swarm; Optimization algorithm; Multi-dimension function optimization; Multi-swarm

### 1 引言

许多实际工程的优化问题, 最后都可归结为复杂函数的优化. De Castro<sup>[1]</sup> 基于克隆选择和亲和力成熟过程提出一种克隆选择算法, 解决了函数优化和简单的模式识别问题. 基于克隆选择学说的这种算法只考虑了个体自身的进化, 在群体中个体之间没有学习功能, 没有有效地利用每一次迭代得到的信息. 粒子群算法是 Eberhart<sup>[2,3]</sup> 等首先提出并改进的一种基于群体智能的优化算法. 由于其设置参数少、计算简单、收敛速度快等优点, 很快得到广泛

的应用<sup>[4]</sup>, 如函数优化、组合优化、半导体器件中和、神经网络优化、Job-shop 优化、机器人路径规划等.

文献[5]利用多个邻域值(不只是全局最优值)来更新粒子的速度, 但易使粒子在最优值附近徘徊而收敛精度不高. [6]采用多个子群体施行局部粒子群优化, 但是收敛精度必须由最后的全局粒子群算法来保证, 仍然存在基本粒子群算法的缺点. [7]采用混沌变异改进群体的多样性, 得到了良好的结果. Hu 和 Eberhart<sup>[8]</sup> 采用调整邻域之间的关系, 很好地解决了动态的多目标优化问题.

收稿日期: 2007-06-05; 修回日期: 2007-09-10.

基金项目: 国防预研项目(9140A17030207HT0150); 芬兰科学院研究项目(Grant 214144).

作者简介: 王巧灵(1983—), 女, 内蒙古四子王人, 博士生, 从事人工免疫系统、进化计算等研究; 王常虹(1961—), 男, 辽宁鞍山人, 教授, 博士生导师, 从事网络控制、智能控制等研究.

本文基于克隆选择和粒子群的优缺点,用粒子群算法的快速性来补偿克隆选择训练时间长的缺点,同时缩小群体规模,减少了冗余;用克隆选择的多样性来补偿粒子群算法多样性差的缺点,提高了搜索精度.为区别于目前的克隆选择算法和粒子群算法,本文以子群体作为抗体,以含有全局最优值的子群体作为抗原,整个群体的适应值作为抗体的亲和力.子群体的粒子通过变异操作动态变化.

## 2 克隆选择和粒子群思想

### 2.1 克隆选择

克隆选择学说最早由 Burnet 提出.该学说认为当动物体遇到外来病毒(抗原)的侵入时,只有与这种抗原匹配的抗体被激发,继而增殖分化,得到成熟的抗体.成熟的抗体一部分分泌浆细胞清除抗原,一部分形成记忆细胞.当同样或类似的病毒再次侵入时,记忆细胞被激活,能迅速高效地清除病毒.在抗体增殖分化的过程中,抗体同时经历着高频变异.实际的工程应用中,人们针对克隆选择的机制,设计了克隆、变异、受体编辑、抑制删除等算子来解决问题,但目前还没有统一的结构框架<sup>[9]</sup>.

### 2.2 基本粒子群算法

粒子群算法由 Kennedy 和 Eberhart 提出,其基本思想是模拟鸟类寻找栖息地的群体智能行为.假设群体中有  $m$  个粒子  $X_i (i = 1, 2, \dots, m)$ , 每个粒子是  $D$  维空间中的一个个体,每个粒子的位置表示为  $X_i = (x_1, x_2, \dots, x_D)$ . 每个粒子都在  $D$  维空间中运动,其速度为  $V_i = (v_1, v_2, \dots, v_D)$ . 不同的粒子在空间中处于不动的位置,相对于目标函数有不同的适应值.群体中最好的位置即适应值最好的粒子用  $P_g$  表示,第  $i$  个粒子经历过的最好位置用  $P_i$  表示.基本粒子群算法的步骤如下:

Step1: 对粒子群的随机位置和速度、群体的最好位置、每个粒子的最好位置进行初始化.

Step2: 计算每个粒子的适应值.

Step3: 更新群体的最好位置和每个粒子经历过的最好位置.

Step4: 按下式更新群体的位置和速度:

$$\begin{cases} v_i(t+1) = \\ v_i(t) + c_1 * \text{rand1} * (p_i(t) - x_i(t)) + \\ c_2 * \text{rand2} * (p_g(t) - x_i(t)), \\ x_i(t+1) = x_i(t) + v_i(t+1). \end{cases} \quad (1)$$

其中:  $c_1$  表示惯性系数;  $c_1$  和  $c_2$  为加速常数,通常在 0 ~ 2 之间;  $\text{rand1} \sim U(0, 1)$  和  $\text{rand2} \sim U(0, 1)$  为两个独立的随机函数.

Step5: 判断是否满足终止条件,满足则结束;不满足则返回 Step2.

## 3 动态多群体优化算法设计

粒子群算法的群体智能思想有效地利用了每次迭代的信息,使迭代在整个群体中实现了良好的正反馈;基于克隆选择思想设计的算子具有良好的多样性.本文以这两点为基础,将整个群体分为若干个子群体,对每个子群体施行粒子群算法,对整个群体施行克隆选择算子.

### 3.1 克隆选择相关概念的设计

本算法中,将整个群体分为  $m$  个子群体,每个群体中含有  $n$  个粒子.每个子群体定义为一个抗体,抗体的亲和力用子群体中全局最优粒子的适应值和子群体所有粒子的平均适应值表示;抗原定义为含有全局最优值的子群体;每个粒子的位置和速度定义为抗体中包含的染色体;抗体之间的距离定义为抗体的亲和力之差的绝对值,两个亲和力之差为零的抗体认为是相同的抗体;染色体之间的差别定义为代表粒子位置和速度的差别,在本文的函数优化中,定义为粒子位置和速度的欧氏距离,如果粒子的位置和速度的欧氏距离同时为零,则认为染色体相同.

### 3.2 算法具体步骤

Step1: 随机产生初始群体,初始化粒子的速度和位置信息,置迭代计数器  $\text{gen} = 1$ , 最大的迭代次数为  $\text{max-gen}$ .

Step2: 在子群体内施行基本的粒子群优化  $R$  次,粒子的位置更新按式(1)进行,其中  $P_g$  为子群体内的全局最优值,  $\text{gen} = \text{gen} + R$ .

Step3: 按下式计算子群体即抗体的亲和力,并按亲和力排序:

$$\text{affinity}_i = (\text{average-affinity}_i + \text{gbest}_i) / 2, \quad i = 1, 2, \dots, m. \quad (2)$$

其中:  $\text{average-affinity}$  是子群体中粒子的平均适应值,  $\text{gbest}$  是子群体的全局最好适应值.

Step4: 克隆算子:按抗体亲和力的高低,对亲和力高的抗体进行克隆.对于亲和力高的前 50% 的抗体,每个抗体进行一次细胞分裂,抗体个数增加一倍.为了实现群体的多样性,对其中一个抗体的每个染色体施加一个很小的随机值,另一个抗体则保持不变.

Step5: 变异算子:对所有克隆后的子群体随机地重组.

Step6: 计算重组后的抗体的亲和力,并按亲和力对抗体排序.

Step7: 选择算子:按抗体的亲和力和相同抗体的数量,从变异后的抗体中选择  $m$  个抗体.

Step8: 受体编辑算子:对每个抗体中任意两个

相同的染色体,重新初始化速度.

Step9: 如果满足终止条件,则结束;否则,转 Step2.

该算法的特点是:1) 在子群体内部施行基本粒子群算法,每个子群体中的粒子都飞向各自的目标,扩大了搜索空间,避免了在整个群体中所有个体逐渐陷入局部最优值. 2) 克隆操作作用于适应值高的子群体,增加了位置好的粒子进入下一次迭代的机会,加速了群体的收敛速度;同时,在克隆过程中染色体添加一个很小的随机值,也增加了个体的多样性,使抗体跳出了局部最优值. 3) 抗体的变异使得每个子群体发生了动态变化,在子群体之间有了信息交换. 如果没有抗体的变异,子群体之间没有信息交换,则子群体很容易收敛到局部最优值. 4) 选择算子选出亲和力较高的抗体,使整个群体向着最优值方向搜索. 5) 受体编辑使每个子群体的多样性得到改善. 由于子群体的群体规模很小,群体中的粒子即染色体相似会导致粒子的速度为零,从而导致整个子群体停滞.

### 4 仿真结果

#### 4.1 测试函数

从文献 [10] 中选取 Sphere Function ( $F_1$ ), Rosenbrock's Function ( $F_2$ ), Ackley's Function ( $F_3$ ), Griewank's Function ( $F_4$ ), Quartic Function ( $F_5$ ), Weierstrass Function ( $F_6$ ), Rastrigin's Function ( $F_7$ ). 用这 7 个函数的最小化问题来测试算法的性能,每个函数都计算 50 维的优化结果.

其中: $F_1$  只有一个峰值,很容易收敛到全局最优解; $F_2$  虽然只有一个峰值,但在峰值附近呈香蕉状弯曲; $F_3$  的全局最优值处于一个很窄的深谷; $F_4$  是不可分的,自变量之间存在一定的联系; $F_5$  是一个带有随机干扰的函数; $F_6$  有大量的局部最优值; $F_7$  的绝大部分点是连续不可微的. 这 7 个函数各有特点,对算法的各种性能基本上都可测试得到.

#### 4.2 参数确定

本文算法中参数设置为:群体规模为 30,迭代

次数为 2000,惯性系数为 0.9 ~ 0.4,粒子群的速度限制取位置限制的 20%, $D = 50$ .

除了基本粒子群算法的参数外,还有子群体施行内部粒子群算法的迭代次数  $R$  和子群体的规模  $n$  需要确定. 首先固定  $R = 10$ ,确定  $n$  值. 对于不同的  $n$  值,分别重复试验 50 次得到函数最优值的平均结果,从中发现当  $n = 3$  时,每个函数的优化结果较好. 然后固定  $n = 3$ ,确定  $R$  值. 对于不同的  $R$  值,分别重复 50 次得到函数最优值的平均结果,当  $R = 5$  时函数的优化结果最好. 当  $n = 30, R = 2000$  时,算法不执行克隆选择部分,等同于基本的粒子群算法. 当  $n = 1, R = 1$  时,算法子群体中只有一个个体,近似于克隆选择算法. 所以,当  $n = 3, R = 5$  时,算法的性能最好.

#### 4.3 与其他算法比较

选择文献 [1, 3, 6] 中的方法进行比较. 参数选择如下:文献 [1] 中选择整个群体克隆,即  $n = N, N_c = 5, d$  为群体规模的 10%. 文献 [3] 中  $w$  为 0.9 ~ 0.4,速度限制与位置限制相同. 文献 [6] 中子群体数为 10,子群体规模为 3,群体更新周期为 5. 测试函数中  $D = 50$ ,每种方法运行 50 次,最大迭代次数均为 2000. 得到最优值的平均结果见表 1.

由表 1 可见,除了  $F_1$  之外,与其他算法相比,本文算法的收敛性更好,精度更高. 文献 [3] 方法对  $F_1$  的收敛精度很高,因为这个函数对群体多样性的要求不高,收敛速度快,所以同样的迭代次数下结果较好. 但是对于更复杂的其他函数,特别是  $F_2$  和  $F_7$ ,文献 [3] 方法的结果不是很好,因为这些函数对群体多样性和搜索方向要求较高.

表 2 是 4 种算法运行 1 次的平均时间. 运行环境为 P43.0, 512MB 内存, Windows XP, Matlab7.1. 每种算法对每个函数都运行 50 次. 从表中结果可以看出,本文在运行时间上比文献 [3] 短很多.

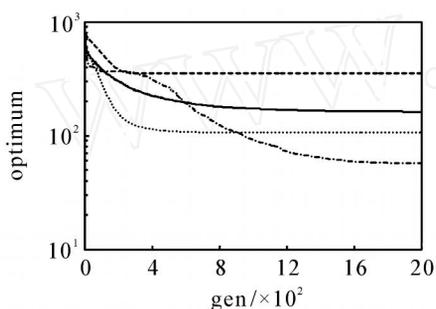
为了进一步说明本文算法良好的寻优能力,图 1 给出了几种算法应用于函数  $F_7$  的收敛曲线. 其中:本文算法用点划线表示,文献 [1] 算法用虚线表示,

表 1 多种算法最优值的平均结果

函数	本文方法	文献[6]	CLONAL	PSO
$F_1$	5.0978e - 004	8.7325e - 003	6.9731e - 002	8.6245e - 015
$F_2$	3.7105e + 001	5.0155e + 001	3.0956e + 002	5.2350e + 001
$F_3$	1.4971e - 000	2.1255e + 000	1.9233e + 001	2.4414e + 000
$F_4$	1.1356e - 001	7.5061e - 001	8.4141e + 002	2.9542e - 002
$F_5$	9.3785e - 003	4.1972e - 002	4.2613e + 002	6.4921e - 002
$F_6$	5.1869e - 000	9.9627e + 000	2.3025e + 001	5.4137e + 000
$F_7$	5.8022e + 001	1.6137e + 002	3.5310e + 002	1.0644e + 002

表2 多种算法的运行时间

函数	本文方法	文献[6]	CLONAL	PSO
$F_1$	27.0753	26.3281	197.884	24.8951
$F_2$	27.5541	28.8335	190.965	23.2892
$F_3$	28.4959	27.1124	211.919	25.9035
$F_4$	29.0157	27.8698	195.464	29.1596
$F_5$	28.8083	28.0804	208.087	24.8969
$F_6$	286.498	284.196	1528.51	290.914
$F_7$	27.7356	26.8133	211.160	24.1646

图1 几种方法优化函数  $F_7$  的收敛性

文献[3]算法用点线表示,文献[6]算法用实线表示.可以看出,本文算法比其他算法具有更强的寻优能力.从对7个函数优化的综合性能来看,本文算法适用于不同特点的复杂函数的优化,其综合性能优于其他算法.特别是对  $F_7$  的优化结果,本文算法具有一定的鲁棒性.

## 5 结 论

本文基于克隆选择原理和粒子群的群体智能思想,提出一种动态多群体的优化方法.该算法应用于子群体内部的粒子群算法扩大了群体的搜索空间,加快了搜索速度;利用克隆选择原理设计了克隆、变异、选择和受体编辑算子,保证了群体的多样性.特别是通过变异算子实现了子群体之间的信息交换,使子群体不断地跳出局部极值,最后搜索到全局最优值.通过对7个各有特点的复杂函数的优化,证明本文算法具有很强的寻优能力.下一步工作将集中于对本文算法的理论分析,从理论上证明算法的全局收敛性.

## 参考文献(References)

- [1] De Castro L N. Learning and optimization using the clonal selection principle [J]. IEEE Trans on Evolutionary Computation, 2002, 6(3): 239-251.
- [2] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc of the IEEE Int Conf on Neural Networks. Perth, 1995: 1942-1948.
- [3] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]. Proc of the IEEE Congress on Evolutionary Computation. Piscataway, 1998: 69-73.
- [4] 谢晓锋, 张文俊, 杨之廉. 粒子群算法综述[J]. 控制与决策, 2003, 18(2): 129-134.  
(Xie X F, Zhang W J, Yang Z L. Overview of particle swarm optimization[J]. Control and Decision, 2003, 18(2): 129-134.)
- [5] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210.
- [6] Liang J J, Suganthan P N. Dynamic multi-swarm particle swarm optimizer [C]. Proc of the 2005 IEEE Swarm Intelligence Symposium. Pasadena, 2005: 127-132.
- [7] 贾东立, 张家树. 基于混沌变异的小生境粒子群算法[J]. 控制与决策, 2007, 22(1): 117-120.  
(Jia D L, Zhang J S. Niche particle swarm optimization combined with chaotic mutation [J]. Control and Decision, 2007, 22(1): 117-120.)
- [8] Hu X, Eberhart R C. Multiobjective optimization using dynamic neighborhood particle swarm optimization [C]. Proc of IEEE Congress on Evolutionary Computation. Hawaii, 2002: 1677-1681.
- [9] Wang X, Gao X Z, Ovaska S J. Artificial immune optimization methods and applications: A survey [C]. Proc of the IEEE Int Conf on Systems, Man and Cybernetics. The Hague, 2004: 3415-3420.
- [10] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.  
(Wang L. Intelligent optimization algorithms with application [M]. Beijing: Tsinghua University Press, 2001.)