

Immune Multi-Direction Binary Particle Swarm Optimization Algorithm*

Ziyuan Qi^{1,2}, Jinqiu Zhang¹, Jie Yue¹, Chao Ma³

¹Brigade of Equipment Trial and Training, Academy of Armored Forces Engineering, Beijing

²Department of Artillery Engineering, Ordnance Engineering College, Shijiazhuang

³Troop No.75134 of PLA, Chongzuo

Email: jxxyqzy@126.com

Received: Oct. 8th, 2013; revised: Oct. 29th, 2013; accepted: Nov. 7th, 2013

Copyright © 2013 Ziyuan Qi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: A novel algorithm of BPSO is presented, which is named immune multi-direction binary particle swarm optimization algorithm (IMBPSO). Operators including immune memory and clone selection of immune algorithm are introduced into BPSO in order to ensure the algorithm to find the best solution quickly and the diversity of colony. Furthermore, by modifying the formula of renewal of speed, the particle is translated from single direction into multi-direction. So it overcomes the disadvantages of BPSO algorithm, including falling into local best easily, low convergence speed as well as low quality in evolution evening. By testing and estimating with some standard functions, IMBPSO algorithm's ability in finding the best solution is proved.

Keywords: Binary Particle Swarm Optimization Algorithm (BPSO); Immune Operator; Clone Selection

免疫多向二进制粒子群优化算法*

齐子元^{1,2}, 张进秋¹, 岳杰¹, 马朝³

¹装甲兵工程学院装备试用与培训大队, 北京

²军械工程学院火炮工程系, 石家庄

³中国人民解放军 75134 部队, 崇左

Email: jxxyqzy@126.com

收稿日期: 2013 年 10 月 8 日; 修回日期: 2013 年 10 月 29 日; 录用日期: 2013 年 11 月 7 日

摘要: 提出了一种新的 BPSO 改进算法——免疫多向二进制粒子群优化算法(IMBPSO), 算法中引入了免疫算法的克隆选择算子和免疫系统的免疫记忆特性, 保证了算法寻优的快速性以及群体的多样性。此外, 通过修改速度更新公式, 把粒子群的搜索方向从单向变为多向, 从而克服了 BPSO 算法中易陷入局部最优, 进化后期收敛速度慢和精度不高等缺点。通过几个标准函数进行测试与评价, 证明了 IMBPSO 算法具有良好的寻优性能。

关键词: 二进制粒子群优化算法; 免疫算子; 克隆选择

1. 引言

粒子群优化算法(Particle Swarm Optimization, PSO)是由 Eberhart 博士和 Kennedy 博士在 1995 年提出的一种新型全局优化进化算法^[1], 该算法源于对鸟

类捕食行为的模拟。PSO 同遗传算法类似, 是一种基于群体(population)的优化工具。系统初始化为一组随机解, 通过迭代搜寻最优值。但不同于遗传算法, PSO 没有采用复杂的交叉、变异算子和优胜劣汰的自然选择过程, 而是基于信息共享机制, 通过粒子的自我学习和向最佳个体学习的方法来实现对解空间的快速

*基金项目: 军械工程学院科学研究基金资助项目(YJXM11010)。

搜索。和其它优化算法相比, PSO 具有简单、快速、调节参数少、易于实现等优点。Kennedy 和 Eberhart 认为 PSO 算法应该是介于遗传算法和进化规划间的一种算法, 在编码方式上也较其它的算法简单。因此, PSO 一经提出, 立刻引起了演化计算等领域的学者们的广泛关注, 并在短短的几年时间内出现了大量的研究成果, 形成了一个研究热点。

为解决离散问题, Kennedy 和 Eberhart 等人^[2]于 1997 年提出了一种二进制粒子群算法——BPSO, 用以求解离散优化问题。然而不管 PSO 算法还是 BPSO, 都存在易陷入局部极值点、进化后期收敛速度慢和精度不高等缺点。本文针对粒子群算法这些缺点, 提出了一种改进 BPSO 算法——免疫多向二进制粒子群优化算法(Immune Multi-direction Binary Particle Swarm Optimization, IMBPSO), 它以二进制粒子群算法的进化框架为基础, 主要从两个方面作以改进: 1) 在算法中引入免疫算法的克隆选择算子和免疫系统的免疫记忆特性。克隆选择算子的引入可以提高算法寻优的快速性, 而记忆细胞的演化是用来保证群体的多样性的。2) 通过修改速度更新公式, 把粒子群的搜索方向从单向变为多向, 以提高搜索速度和精度。

2. 标准粒子群优化算法(PSO)

1998 年 Yuhui Shi^[3]提出了带带惯性权重的改进粒子群算法。目前, 有关 PSO 算法的研究大多以带惯性权重的 PSO 算法为基础进行扩展和修改, 为此将带惯性权重的 PSO 算法称之为标准 PSO 算法(PSO)。SPSO 算法将每个粒子看作是在 n 维搜索空间中的一个没有重量和体积的微粒, 每个微粒作为特征空间的一个点, 也就对应着一个特征子集。粒子群在特征空间中以一定的速度“飞行”, 该飞行速度由粒子的飞行经验和群体的飞行经验进行动态调整, 改变着位置、进行相互交流, 对单个粒子的最优位置和整个群体的最优位置进行记忆。目的就是找到最优的特征子集, 这个特征子集具有最短的长度, 而使用这些特征进行分类却能得到最好的分类精度^[4]。

对于任何一个粒子 i 都包含如下 3 个向量:

$$X_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{in}(t))$$

向量记录了粒子在搜索空间第 t 次叠代时的位置:

$$V_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{in}(t))$$

向量包含了粒子在不受干扰的情况下位置的改变, 即粒子第 t 次叠代时的飞行速度:

$$P_i(t) = (p_{i1}(t), p_{i2}(t), \dots, p_{in}(t))$$

向量记录了粒子所找到的当前最优解的位置, 即个体极值(记为 P_{best})。

除了个体极值外, 还有一个全体极值, 即粒子群到目前为止找到的最优解, 称为全局极值(记为 G_{best}), 表示为: $P_g(t) = (p_{g1}(t), p_{g2}(t), \dots, p_{gn}(t))$ 。在第 $t+1$ 次叠代时, 粒子 i 按照下面规则更新位置和速度:

$$v_{ik}(t+1) = \omega v_{ik}(t) + c_1 rand_1(p_{ik}(t) - x_{ik}(t)) + c_2 rand_2(p_{gk}(t) - x_{ik}(t)) \quad (1)$$

$$x_{ik}(t+1) = x_{ik}(t) + v_{ik}(t+1) \quad (2)$$

式中, ω 为惯性权重, c_1, c_2 为学习因子, $rand_1$ 和 $rand_2$ 为两个均匀分布在(0,1)之间的随机数; 粒子个数 $I = 1, 2, \dots, N$; 搜索空间的维数 $k = 1, 2, \dots, n$, t 为迭代次数。

粒子通过不断学习更新, 最终飞至解空间中最优解所在的位置, 整个搜索过程结束, 最后输出的 G_{best} 就是算法找到的全局最优解。迭代终止条件根据具体问题一般选为最大迭代次数或粒子群迄今为止搜索到的最优位置满足预定最小适应阈值。

3. 二进制粒子群优化算法(BPSO 算法)

基本的 PSO 算法定义于连续的函数空间, 为了解决工程实际中的组合优化问题, 相对地, 离散二进制优化算法具有很多优势, 首先对于纯组合优化问题的表达形式要求优化算法离散的, 其次二进制算法可以表达浮点数, 因此也同样适用于连续空间的问题求解。BPSO 算法中, 每个粒子的位置使用二进制编码, 粒子的速度定义为粒子位置改变的概率。

在 BPSO 中, 将每一维 x_{ik} 和 $p_{best_{ik}}$ 限制为 1 或者 0, 而速度 v_{ik} 不作这种限制。用速度来更新位置时, 如果 v_{ik} 高一些, 粒子的位置 x_{ik} 更有可能选 1, v_{ik} 低一点则 x_{ik} 选 0, 阈值在[0,1]之间, 而有这种特点的函数就是模糊函数 Sigmoid 函数^[5]:

$$Sig(v_{ik}) = \frac{1}{1 + \exp(-v_{ik})} \quad (3)$$

BPSO 的速度 v_{ik} 的计算公式与 SPSO 相同, 如式

(1). 粒子位置更新公式为:

$$x_{ik}(t+1) = \begin{cases} 1, & \text{if } rand() < sig(v_{ik}(t+1)) \\ 0, & \text{其他} \end{cases}$$

其中, rand()为服从均匀分布的随机数。这样 x_{ik} 被限制在集合 $\{0,1\}$ 中。

4. BPSO 算法的改进

4.1. 免疫算法的引入

抗体多样性和免疫记忆是免疫系统的重要特性, 在进化过程中, 高亲和力低浓度的抗体受到促进, 而低亲和力高浓度的抗体受到抑制, 以此来保证抗体的多样性。免疫记忆是免疫系统将与入侵抗原反应部分的抗体作为记忆细胞保留下来。对于同类抗原的再次入侵, 相应的记忆细胞被激活而产生大量的抗体。克隆选择是在给定的选择率 α , $0 < \alpha < 1$, 在抗体群中选择部分抗体的确定性映射, 克隆选择仅选择群体中亲和力较高的抗体参与繁殖及突变, 而亲和力低的抗体仍存在于免疫系统中, 并逐渐被驱除。

根据高浓度低亲和力的微粒(抗体)受到抑制, 低浓度高亲和力的微粒(抗体)受到促进这一免疫原理, 用基于浓度的选择概率来保证群体的多样性^[6]。第 i 个微粒(抗体)的浓度和选择概率公式如下:

$$D(X_i) = \frac{1}{\sum_{j=1}^{N_0} |f(X_i) - f(X_j)|} \quad i = 1, 2, \dots, N_0 \quad (5)$$

$$p(x_i) = \frac{\frac{1}{D(X_i)}}{\sum_{i=1}^{N_0} \frac{1}{D(x_i)}} = \frac{\sum_{j=1}^{N_0} |f(x_i) - f(x_j)|}{\sum_{i=1}^{N_0} \sum_{j=1}^{N_0} |f(x_i) - f(x_j)|} \quad (6)$$

$$i = 1, 2, \dots, N_0$$

其中 $N_0 = round(\alpha, N)$, α 满足 $round(\alpha, N) + M + N_0 = N$, M 为随机产生的新粒子(抗体), N 为群体规模。

免疫算法的引入主要是为了保证粒子群中的粒子的多样性, 因此粒子群中的初始化粒子不是随机产生, 而是由免疫算法产生的抗体作为初始化粒子。满足约束条件的最优解即是抗原; 候选解即是抗体, 抗体和抗原之间的亲和力反映了候选解和最优解的接

近程度, 也即反映候选解对约束条件和目标函数的满足程度; 抗体和抗体之间的亲和力反映了不同候选解之间的异同, 也即反映了抗体的多样性。在进化算法中, 保持抗体的多样性可以防止算法陷入局部最优解。依据抗体和抗原之间的亲和力来选择有效抗体能更好地体现“优胜劣汰”的原则, 特别是当待选抗体相差不大时, “优胜劣汰”的效果更明显, 搜索效率会更高^[3]。

4.2. 速度更新的改进

BPSO 算法中每个粒子仅根据自身的个体极值和全局极值这两个信息量来更新自己的速度和位置, 并没有考虑其他粒子的信息, 粒子群在解空间的搜索是单向的, 因而精度不高。把粒子群的搜索方向从单向变为多向是提高速度的一种有效方法。为此, 将速度更新公式(1)式修改成(7)式:

$$v_{ik}(t+1) = \omega v_{ik}(t) + \sum_{i=1}^m c_1 rand_1(p_{ik}(t) - x_{ik}(t)) + c_2 rand_2(p_{gk}(t) - x_{ik}(t)) \quad (7)$$

与 BPSO 算法相比, IMBPSO 算法中的粒子根据最优的 m 个个体极值、全局极值和其他的一些粒子的信息来修正每个粒子下一次迭代的行动策略更新自己的速度和位置, 从而考虑了更多粒子在迭代寻优过程中包含的信息, 整个粒子群在解空间的搜索是多方向性的, 搜索过程更均匀, 能有效提高算法的精度和全局收敛能力^[7]。

4.3. 算法的实现流程

如图 1。

IMBPSO 算法的实现步骤:

Step1: 初始化粒子群的位置和速度;

Step2: 计算每个粒子的适应度值;

Step3: 对于每个粒子, 将其适应值与所经历过的最好位置的适应度值进行比较, 若较好。则将其作为当前的粒子最好位置;

Step4: 对每个粒子, 将其适应值与全局所经历过的最好位置的适应度值进行比较。若较好。则将其作为当前的全局最好位置;

Step5: 根据式(4)和(7)更新粒子的速度和位置;

Step6: 产生记忆粒子(抗体)。将每一次迭代的 G_{best}

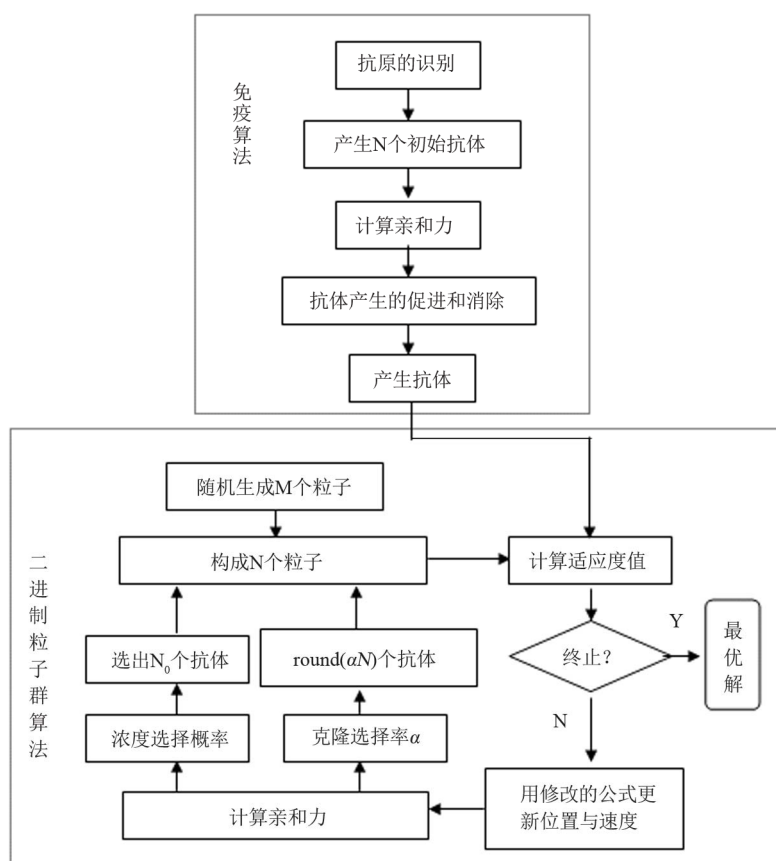


Figure 1. Realization flow of IMBPSO
图 1. IMBPSO 算法的实现流程

作为记忆粒子放入记忆库中;

Step7: 当 G_{best} 连续进化若干代未有变化, 如果达到终止条件, 则结束; 如果未达到终止条件, 则停止进化, 并按如下的方法更新粒子。否则转 Step2;

Step8: 更新粒子(抗体)。新粒子(抗体)由以下三种方式来分别产生:

- a) 随机产生 M 个新粒子(抗体)。
- b) 从记忆库中根据基于粒子(抗体)浓度的概率选择公式(5)和(6), 选出 N_0 个粒子(抗体)。
- c) 克隆选择。对当前的更新代按选择率 α 选择 $round(\alpha N)$ 个较低适应度值(较高亲和力)的粒子(抗体)。其中 $round(\alpha N)$ 满足 $round(\alpha N) + M + N_0 = N$ ($round()$ 函数为就近取整数)。

形成新一代粒子(抗体)群。然后, 返回 Step2。

5. 算法的性能评价

为测试免疫多向微粒群算法(IMBPSO)的性能, 选择了四个著名的 benchmark 测试函数对 IMBPSO 和

BPSO 算法进行对比测试。四个测试函数如下:

F1: Sphere 函数, 它在 $x_i = 0$ 时达到最小值 0。

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad -100 \leq x_i \leq 100$$

F2: Ackley 函数, 在 $x_i = 0$ 时达到最小值 0。

$$f_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=0}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$$

$$-30 \leq x_i \leq 30$$

F3: Rastrigin 函数, 有很多正弦凸起的局部极小点。在 $x_i = 0$ 处取到全局最小值 0。

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$-5.12 \leq x_i \leq 5.12$$

F4: Rosenbrock 函数。是一个经典的优化函数,

Table 1. Test results of IMBPSO and BPSO
表 1. IMBPSO 与 BPSO 的测试结果

标准函数	算法	达到精度时停止搜索			
		收敛率	最好解	平均最优解	平均收敛代数
F1	BPSO	50/50	0	3.3675e-06	246
	IMBPSO	50/50	0	0	39
F2	BPSO	50/50	2.804e-08	4.5409e-06	395
	IMBPSO	50/50	0	0	25
F3	BPSO	9/50	1.4924	2.4843	2937
	IMBPSO	45/50	0.1932	0.4874	846
F4	BPSO	18/50	0.2846	0.7502	3145
	IMBPSO	47/50	0	0.0247	510

在 $x_i = 1$ 处取到全局最小值 0。

$$f_2(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i)^2 + (x_i - 1)^2 \right)$$

$$-30 \leq x_i \leq 30$$

测试条件: IBM 笔记本 T1400 1.83 GHz, 512 M 内存, Windows XP Professional, Matlab7.01 下编程运行。

实验中种群规模为 40, 维数为 10, F1 和 F2 的最大进化代数是 1000 代, F3 和 F4 的最大进化代数是 5000 代, F1 和 F2 的误差设置为 10^{-8} , F3 和 F4 的误差为 10^{-5} , 各函数的 V_{\max} 取变量范围的上限。学习因子 $C1 = C2 = 2.0$, 惯性权重 ω 从 0.9 到 0.4 线性减少。克隆选择率 $\alpha = 0.3$, 记忆粒子选择率为 0.7。每组参数均重复实验 10 次, 并给出相应参数下 50 次实验的收敛率、最好解、平均最优解、平均收敛代数。结果如表 1 所示。

从上面的实验结果可以看出, IMBPSO 比 BPSO 有着更好的搜索性能。对于较易收敛的 F1 和 F2, IMBPSO 在与 BPSO 有着相同的收敛率和相近的平均最优解的情况下, 收敛速度明显要好于 BPSO。而对于较难收敛的 F3 和 F4, IMPSO 不管从收敛率还是收敛速度和平均最优解上都明显优于 BPSO。

6. 结论

BPSO 由于群体的多样性在进化后期变差, 而导

致收敛后期速度变慢, 易陷于局部最优和收敛的精度降低。IMBPSO 算法借鉴免疫学习中较好的多样性, 将每一进化代的 G_{best} 作为记忆细胞保存下来, 并按基于浓度的选择概率来更替部分微粒, 使进化群体保持了很好的多样性, 有效地避免了算法陷于局部最优。同时对更替当前代中适应值较好的微粒进行克隆选择操作, 从而加快了算法的收敛速度。测试结果也显示, MPSO 算法具有精度高、收敛速度快的优点, 为解决最优问题提供了一条新的途径。

参考文献 (References)

- [1] Kennedy, J. and Eberhart, R.C. (1995) Particle swarm optimization. *IEEE International Conference on Neural Networks*, Perth, 1942-1948.
- [2] Kennedy, J. and Eberhart, R.C. (1997) A discrete binary version of the particle swarm algorithm. *Proceedings of the World Multi-conference on Systemic, Cybernetics and Informatics*, IEEE Service Center, Piscataway, 4104-4109.
- [3] Shi, Y. and Eberhart, R. (1998) A modified particle swarm optimizer. *IEEE World Congress on Computational Intelligence*, 69-73.
- [4] 郑洪英 (2007) 基于进化算法的入侵检测技术研究. 博士学位论文, 重庆大学, 63-69.
- [5] 王新峰, 邱静, 刘冠军 (2005) 基于离散粒子群优化算法的直升机减速器齿轮故障特征选择. *航空动力学报*, 6, 969-972.
- [6] 胡春霞 (2007) 免疫微粒群算法的研究. 硕士学位论文, 太原科技大学, 16-21.
- [7] 曾慧娟, 潘文斌, 朱健全 (2008) 基于改进粒子群优化算法的水质模型参数识别. *环境污染与防治*, 3, 1-7.