生物智能算法之群智能算法
人工鱼群算法&萤火虫算法

王鹏

2017年4月24号

目录

第一部分　人工鱼群算法

1　人工鱼群算法介绍

人工鱼群算法是李晓磊等人于2002年在动物群体智能行为研究的基础上提出的一种新型方盛优化算法，该算法根据水域中鱼生存数目最多的地方就是本水域中富含营养物质最多的地方这一特点来模拟鱼群的觅食行为而实现寻优。算法主要利用鱼的三大基本行为：觅食、聚群和追尾行为，采用自上而下的寻优模式从构造个体的底层行为开始，通过鱼群中各个体的局部寻优，达到全局最优值在群体中凸显出来的目的。

1.1　生态学基础

在一片水域中，鱼存在的数目最多的地方就是本水域富含营养物质最多的地方，依据这一特点来模仿鱼群的觅食、聚群、追尾等行为，从而实现全局最优，这就是鱼群算法的基本思想。鱼类活动中，觅食行为、群聚行为、追尾行为和随机行为与寻优命题的解决有较为密切的关系，如何利用简单有效的方式来构造和实现这些行为将是算法实现的主要为题。

1.2　人工鱼的结构模型

人工鱼是真实鱼抽象化、虚拟化的一个实体，其中封装了自身数据和一系列行为，可以接受环境的刺激信息，做出相应的活动。其所在的环境由问题的解空间和其他人工鱼的状态，它在下一时刻的行为取决于自身的状态和环境的状态，并且它还通过自身的活动来影响环境，进而影响其他人工鱼的活动。

人工鱼对外的感知是依靠视觉来实现的，人工鱼的模型中使用如下方法实现人工鱼的虚拟视觉：

$$X_V = X + Visuanl * Rand()$$

$$X_{next} = X + \frac{X_V - X}{||X_V - X||} * Step * Rand()$$

其中$Rand()$为随机函数，产生0到1之间的随机数，$Step$为步长。

1.3 人工鱼中封装的变量和函数

变量部分 人工鱼的总数N、人工鱼个体的状态$X = (x_1, x_2, ..., x_n)$（其中$x_i(i = 1, 2, ..., n)$为寻优的变量）、人工鱼移动的最大步长$Step$、人工鱼的视野Visual、尝试次数$Try - number$、拥挤度因子δ、人工鱼个体i, j之间的距离$d_{ij} = |x_i - x_j|$。

函数部分 人工鱼当前所在位置食物浓度表示为$Y = f(x)$（Y为目标函数值）、人工鱼各种行为函数（觅食行为$Prey()$、聚群行为$Swarm()$、追尾行为$Follow()$、随机行为$Move()$以及行为评价函数$Evaluate()$）。

1.4 人工鱼的四种基本行为算法描述

人工鱼群算法主要利用鱼的三大基本行为：觅食、聚群和追尾行为，再加上一种缺省行为：随机行为，采用自上而下的寻优模式从构造个体的底层行为开始，通过鱼群中各个体的局部寻优，达到全局最优值在群体中凸显出来的目的。

1.4.1 觅食行为$Prey()$

这是鱼趋向食物的一种活动，一般认为它是通过视觉或味觉来感知水中的食物量或食物浓度来选择行动的方向。设置人工鱼当前状态，并在其感知范围内随机选择另一个状态，如果得到的状态的目标函数大于当前的状态，则向新选择得到的状态靠近一步，反之，重新选取新状态，判断是否满足条件，选择次数达到一定数量后，如果仍然不满足条件，则执行随机行为。

算法描述 人工鱼X_i在其视野内随机选择一个状态X_j

$$X_j = X_i + Visuanl * Rand()$$

分别计算X_i与X_j的目标函数值Y_i与Y_j，如果发现Y_j比Y_i好，则X_i向X_j的方向移动一部：

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{||X_j - X_i^t||} * Step * Rand()$$

否则，X_i继续在其视野内选择状态X_j，判断是否满足前进条件，反复尝试$Tray - number$次后，仍没有满足前进条件，则执行随机行为。

1.4.2 聚群行为$Swarm()$

大量或少量的鱼聚集成群，进行集体觅食和躲避敌害，这是它们在进化过程中形成的一种生存方式。人工鱼探索当前邻居内的伙伴数量，并计算伙伴的中心位置，然后把新得到的中心位置的目标函数与当前位置的目标函数相比较，如果中心位置的目标函数优于当前位置的目标函数并且不是很拥挤，则当前位置向中心位置移动一步，否则执行觅食行为。鱼聚群时会遵守两条规则：一是尽量向邻近伙伴的中心移动，二是避免过分拥挤。

算法描述 人工鱼X_i搜索当前视野内（$d_{ij} < Vaisual$）的伙伴数目n_f和中心位置X_c，若$Y_c/n_f > \delta Y_i$，则表明伙伴中心位置状态较优且不太拥挤，则X_i朝伙伴的中心位置移动一步：

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{||X_c - X_i^t||} * Step * Rand()$$

否则进行觅食行为。

1.4.3 追尾行为$Follow()$

当某一条鱼或几条鱼发现食物时，它们附近的鱼会尾随而来，导致更远处的鱼也会尾随过来。人工鱼探索周围邻居鱼的最优位置，当最优位置的目标函数值大于当前位置的目标函数值并且不是很拥挤，则当前位置向最优邻居鱼移动一步，否则执行觅食行为。

算法描述 人工鱼X_i搜索当前视野内（$d_{ij} < Vaisual$）的伙伴中函数Y_j最优伙伴X_j，如果$Y_j/n_f > \delta Y_i$，表明最优伙伴的周围不太拥挤，则X_i朝词伙伴移动一步：

$$X_i^{t+1} = X_i^j + \frac{X_j - X_i^t}{||X_j - X_i^t||} * Step * Rand()$$

否则执行觅食行为。

1.4.4 随机行为$Move()$

它是觅食行为的一个缺省行为，指人工鱼在视野内随机移动。当发现食物时，会向食物逐渐增多的方向快速的移动。

算法描述 人工鱼X_i随机移动一步，到达一个新的状态：

$$X_i^{t+1} = X_i^t + Visuanl * Rand()$$

2 人工鱼群算法过程

公告牌 是记录最优人工鱼个体状态的地方。每条人工鱼在执行完一次迭代后将自身当前状态与公告牌中记录的状态进行比较，如果优于公告牌中的状态则用自身状态更新公告牌中的状态，否则公告牌的状态不变。当整个算法的迭代结束后，公告牌的值就是最优解。

行为评价 是用来反映鱼自主行为的一种方式，在解决优化问题时选用两种方式评价：一种是选择最优行为执行；另一种是选择较优方向。对于解决极大值问题，可以使用试探法，即模拟执行群聚、追尾等行为，然后评价行动后的值选择最优的来执行，缺省的行为为觅食行为。

迭代终止条件 通常的方法是判断连续多次所得值得均方差小鱼允许的误差；或判断聚集于某个区域的人工鱼的数目达到某个比例；或连续多次所得的均值不超过已寻找的极值；或限制最大迭代次数。若满足终止条件，则输出公告牌的最优记录；否则继续迭代。

人工鱼群算法的步骤

1. 初始化设置，包括种群规模N、没条人工鱼的初始位置、人工鱼的视野$Visual$、步长$step$、拥挤度因子δ、重复次数$Try - number$；
2. 计算初始鱼群各个体的适应值，取最优人工鱼状态及其值赋予给公告牌；
3. 对每个个体进行评价，对其要执行的行为进行选择，包括觅食、聚群、追尾和随机行为；
4. 执行人工鱼的行为，更新自己，生成新鱼群；
5. 评价所有个体。若某个体优于公告牌，则将公告牌更新为该个体；
6. 当公告牌上最优解达到满意误差界内或者达到迭代次数上限时算法结束，否则转步骤3。

3 人工鱼群算法分析

3.1 寻优原理

人工鱼群算法在寻优的过程中，可能会集结在几个局部最优解的周围，使人工鱼跳出局部最优解，实现全局寻优的因素主要有：

- 觅食行为中重复次数较少时，为人工鱼提供了随机移动的机会，从而可能跳出局部最优解；
- 随机步长使得人工鱼在前往局部最优解的途中，有可能转向全局最优解；
- 拥挤度因子 δ 限制了聚群的规模，使得人工鱼能够更广泛的寻优；
- 聚群行为能够促使少出陷于局部最优解的人工鱼趋向全局最优解的人工鱼方向聚集，从而逃出局部最优解；
- 追尾行为加快了人工鱼向更优状态游动。

3.2 参数设置

人工鱼群算法有5个基本参数：群规模N、人工鱼的视野$Visual$、步长$step$、拥挤度因子δ、重复次数$Try-number$。

群规模N 人工鱼的数目越多，跳出局部最优解的能力越强，同时，收敛的速度也越快。当然，付出的代价就是算法每次迭代的计算量也越大，因此，在使用过程中，满足稳定收敛的前提下，应当尽可能的减少个图数目。

视野$Visual$ 由于视野对算法中个行为都有较大影响，因此，它的变化对收敛性能影响也比较复杂。当视野范围较小时，人工鱼的觅食行为和随机行为比较突出；视野范围较大时，人工鱼的追尾行为和聚群行为将变得比较突出，相应的算法的复杂度也会有所上升。总的来说：视野越大，越容易使人工鱼发现全局最优解并收敛。

步长$step$ 对于固定步长，随着步长的增加，收敛的速度得到了一定的加速，但在超过一定的范围后，有使得收敛速度减缓，步长过大时会出现震荡现象而大大影响收敛速度。采用随机步长的方式在一定程度上防止了震

荡现象的发生，并使得该参数的敏感度大大降低了，但最快的收敛速度还是最优固定步长的收敛速度，所以，对于特定的优化问题，我们可以考虑采用合适的固定步长或者变尺度方法来提高收敛速度。

拥挤度因子δ 在求极大值问题中，$\delta = 1/(\alpha n_{max}), \alpha \in (0, 1]$；在求极小值问题中，$\delta = \alpha n_{max}, \alpha \in (0, 1]$。其中$\alpha$为极值接近水平，$n_{max}$为期望在该邻域内聚集的最大人工鱼数目。拥挤度因子与n_f相结合，通过人工鱼是否执行追尾和聚群行为对优化结果产生影响。以极大值为例（极小值的情况正好与极大值相反），δ越大，表明允许的拥挤程度越小，人工鱼摆脱局部最优解的能力越强；但是收敛速度会有所减缓，这主要因为人工鱼在逼近最优解的同时，会因避免过分拥挤而随机走开或者受其他人工鱼的排斥作用，不能精确逼近极值点。可见，虽然δ的引入避免了人工鱼过度拥挤而陷入局部最优解，但是另一方面，该参数会使得位于极值点附件的人工鱼之间存在相互排斥的影响，而难以想极值点精确逼近。所以，对于某些局部极值不是很严重的具体问题，可以忽略拥挤的因素，从而在简化算法的同时也加快算法的收敛速度和提高结果的精确程度。

尝试次数$Try - number$ 尝试次数越多，人工鱼的觅食行为能力越强，收敛的效率也越高。在局部极值突出的情况下，应该适当的减少以增加人工鱼随机游动的概率，克服局部最优解。

3.3 算法优缺点

算法优点

- 只需比较目标函数值，对目标函数的性质要求不高；
- 对初值的要求不高，随机产生或设置为固定值均可，鲁棒性强；
- 对参数设定的要求不高，容许范围大；
- 具备较好的全局寻优能力，能快速跳出局部最优点；
- 对于一些精读要求不高的场合，可以用它快速得到一个可行解。

算法缺点

- 参数选择凭借经验；

- 收敛速度较慢，收敛速度没有保证；
- 求解精读较低。

3.4 算法改进

由于人工鱼群算法仍有诸如收敛速度慢、算法后期搜索精度不高以及参数选定全凭经验等问题需要改进，近些年来，对于鱼群算法的改进研究十分活跃，主要涉及自身改进以及与其他算法融合两个大方向：

人工鱼群算法自身改进 鱼群算法的自身改进主要包括参数改进、行为改进及领域结构改进等方面，提出了诸如全局版鱼群算法、动态调参的人工鱼群算法以及自适应调参的人工鱼群算法等。

与其他算法融合 各种智能算法都有其各自的优缺点，通过混合算法的方法，可以提高算法的性能，实现算法之间的优势互补。当前的混合算法的研究主要包括与遗传算法的结合、与粒子群算法的结合、与蚁群算法的结合、余混沌算法的结合等等。

第二部分 萤火虫算法

2005年，印度学者K.N.Krishnanand和D.Ghose在IEEE群体智能会议上提出了一种新的群智能优化算法，人工萤火虫群优化（Glowworm Swarm Optimization, GSO）算法。2009年，剑桥学者Xin-She Yang根据自然界中萤火虫的发光行为提出萤火虫算法（Firefly Algorithm, FA）。自这两种萤火虫算法提出以来，各国学者对这两种算法进行了研究、改进和应用。经过几年的发展，在连续空间的寻优过程和一些生产调度方面萤火虫算法具有良好的应用前景。

4 人工萤火虫群优化算法GSO简介

在GSO算法中，每一只人工萤火虫散步在解空间中，这些萤火虫带着荧光，并且拥有各自的视线范围，称为决策域（local-decision range）。它们的亮度与自己所在位置上的目标值有关。越亮的萤火虫表示它所在的位

置越好，即有较优的目标函数值。萤火虫会在决策域范围内寻找邻居集合，在集合当中，越亮的邻居拥有越高的吸引力吸引此萤火虫往这个方向移动，每一次的飞行方向会随着挑选的邻居不同而改变。此外，决策域范围的大小会受到邻居数量的影响，当邻居密度越低，萤火虫的决策半径会加大以寻找更多的邻居；当邻居密度越高，它的决策半径会缩小。最后，大部分萤火虫会聚集在多个位置上，即达到极值点。简单的说，GSO算法主要包含四个阶段：萤火虫的部署（初始化）、荧光素更新、位置更新阶段和决策半径更新阶段。

5 人工萤火虫群优化算法GSO数学描述

初始化 在可行域中随机放置n个萤火虫，并赋予每个萤火虫的荧光素为l_0，动态决策域为r_0。初始化步长s，领域阈值n_t，荧光素消失率ρ，荧光素更新率γ，动态决策域更新率β，萤火虫感知域r_s，迭代次数M。

更新萤火虫i的荧光素

$$l_i(t) = (1-\rho)l_i(t-1) + \gamma J(x_i(t))$$

其中$J(x_i(t))$表示萤火虫i在t时刻所在位置的目标函数值；$l_i(t)$表示萤火虫i在t时刻荧光素值。

寻找萤火虫i的邻居

$$N_i(t) = \{j : ||x_j(t) - x_i(t)|| < r_d^i(t); l_i(t) < l_j(t)\}$$

其中$N_i(t)$表示萤火虫i在t时刻邻居的集合；$r_d^i(t)$表示萤火虫i在t时刻动态决策域。

确定萤火虫i动作移动方向

$$j = max(p_i)$$

其中$p_i = (p_{i1}, p_{i1}, ..., p_{iN_i(t)})$，其中$p_{ij}(t) = \frac{l_j(t)-l_i(t)}{\sum_{k\in N_i(t)}(l_k(t)-l_i(t))}$为转移概率。

更新萤火虫i的位置

$$X_i(t+1) = X_i(t) + s(\frac{X_j(t) - X_i(t)}{||X_j(t) - X_i(t)||})$$

更新动态决策域

$$r_d^i(t+1) = min\{r_s, max\{0, r_d^i(t), \beta(n_t - |N_i(t)|)\}\}$$

6 人工萤火虫群优化算法GSO基本流程

1. 初始化各个参数；
2. 随机初始化第$i(i = 1, 2, ..., n)$个萤火虫在目标函数搜索范围内的位置；
3. 计算萤火虫i在t时刻的荧光素值$l_i(t)$；
4. 每只萤火虫在其动态决策域半径$r_d^i(t+1)$，选择荧光素值比自己高的个体组成其领域集$N_i(t)$，其中$0 < r_d^i(t) \leq r_s$。r_s为萤火虫个体的感知半径。
5. 计算萤火虫i移向邻域集内个体j的概率$p_{ij}(t)$;
6. 利用轮盘赌的方式算则个体j，然后移动，更新位置；
7. 更新萤火虫动态决策域半径的值；
8. 是否到达最大迭代次数或者要求精度，如果达到这转下一步骤，否则转向步骤4；
9. 输出全局极值点和最优个体值。

7 萤火虫算法FA简介

把空间各点看成萤火虫，利用发光强的萤火虫会吸引发光弱的萤火虫的特点。在发光弱的萤火虫向发光强的萤火虫移动的过程中，完成位置的迭代，从而找出最优位置，即完成了寻优过程。

萤火虫算法有以下条件：

- 假设所有萤火虫都是同一性别且相互吸引;
- 吸引度只与发光强度和距离有关，发光强的萤火虫会吸引周围发光弱的萤火虫，但是随着距离的增大吸引度逐渐减小，发光强的萤火虫会做随机运动;

- 发光强弱由目标函数决定，在制定区域内与指定函数成比例关系。

搜索过程和萤火虫的两个重要参数有关:萤火虫的发光亮度和相互吸引度，发光亮的萤火虫会吸引发光弱的萤火虫向它移动，发光越亮代表其位置越好，最亮萤火虫即代表函数的最优解。发光越亮的萤火虫对周围萤火虫的吸引度越高，若发光亮度一样，则萤火虫做随机运动，这两个重要参数都与距离成反比，距离越大吸引度越小。

8 萤火虫算法FA数学描述

萤火虫的相对荧光亮度

$$I = I_0 e^{-\gamma r_{ij}}$$

其中，I_0表示最亮萤火虫的亮度，即自身（$r = 0$处）荧光亮度，与目标函数值相关，目标函数组越优，自身亮度越高；γ表示光吸收系数，因为荧光会随着距离的增加和传播媒介的吸收逐渐减弱，所以设置光强吸收系数以体现此特性，可设置为常数；r_{ij}表示萤火虫i与j之间的距离。

相互吸引度β

$$\beta(r) = \beta_0 e^{-\gamma r_{ij}^2}$$

其中，β_0表示最大吸引度，即光源处（$r = 0$处）的吸引度。

最优目标迭代

$$x_i(t+1) = x_i(t) + \beta(x_j(t) - x_i(t)) + \alpha(rand - 1/2)$$

其中，X_i与X_j表示i、j两个萤火虫的空间位置，α为步长因子，$rand$为$[0,1]$上服从均匀分布的随机因子。

9 萤火虫算法FA基本流程

1. 初始化算法基本参数。设置萤火虫数目n，最大吸引度β_0，光强吸收系数γ，步长因子α，最大迭代次数$MaxGeneration$或搜索精度ε；

2. 随机初始化萤火虫的位置，计算萤火虫的目标函数值作为各自最大荧光亮度I_0；

3. 计算群体中萤火虫的相对亮度I和吸引度β，根据相对亮度决定萤火虫的移动方向；
4. 更新萤火虫的空间位置，对处在最佳位置的萤火虫进行随机移动；
5. 根据更新后萤火虫的位置，重新计算萤火虫的亮度；
6. 当满足搜索精度或达到最大搜索次数，则转下一步；否则，搜索次数增加1，转第3步，进行下一次搜索。
7. 输出全局极值点和最优个体值。

10 两种萤火虫算法分析及改进

10.1 算法分析

GSO和FA十分相似，只是具体实现方面有一定差异。

共性

- 算法中都加入了随机干扰项，可以跳出局部最优解
- 萤火虫算法中每个萤火虫是一个独立的个体，它们具有自己的感知能力和搜索半径，所以在寻找局部最优解和全局最优解方面都有很好的能力;
- 算法过程可以并发进行，节省寻优时间。

算法缺陷

- 算法参数都是事先设定的，而且要求较高，往往会导致算法收敛早熟，或因参数设置不当而导致算法无法收敛。
- 在寻找局部最优和全局最优之间的平衡能力较弱，容易陷入局部最优。
- 萤火虫位置迭代的过程中的收敛速度较慢。

10.2 算法改进实例

10.2.1 混沌萤火虫算法

混沌策略是一种新颖的优化技术，指确定性系统中的一种貌似随机地行为或者性态，其具有普遍性、随机性和规律性，非常适合与萤火虫算法结合，以改善算法的相关缺陷。

混沌算法步骤

1. 令$t=0$，将萤火虫位置信息X_j^t，$j=1,2,...,n$按下式映射为0到1之间的混沌参量cx_j^t.

$$cx_j^t = \frac{x_j^t - x_{min,j}}{x_{max,j} - x_{min,j}}, j = 1,2,...,n$$

   映射公式中：变量$x_{max,j}$和变量$x_{min,j}$分别为搜索空间中第j维的上界和下界。

2. 根据cx_j^t，计算得到下一步迭代的混沌参量cx_j^{t+1}.

3. 将混沌参量cx_j^{t+1}转换为位置信息x_J^{t+1}

$$x_j^{t+1} = x_{min,j} + cx_j^{t+1}(x_{max,j} - x_{min,j}), j = 1,2,...,n$$

4. 根据位置信息x_j^{t+1}，$j=1,2,...,n$，对所得新解进行性能评价。

5. 若所得新解优于初始解$X^{(0)} = [x_i^0,...,x_n^0]$或者混沌搜索已到预先设定的精度或迭代次数，则新解作为算法的最终结果，否则令t=t+1并返回步骤2。

改进混沌萤火虫算法流程

1. 系统初始化，生成萤火虫初始种群的规模、位置信息，设置光强吸收系数、最大吸引因子和步长因子等参数；

2. 计算群体中每个萤火虫的荧光亮度；

3. 更新权重公式计算惯性权重；

4. 根据位置更新公式更新萤火虫位置，最亮的萤火虫个体随机移动；

5. 计算位置更新后群体中每个萤火虫的荧光亮度；
6. 计算位置更新后群体中适应度最高的10%的个体，进行混沌局部搜索；
7. 计算位置更新后群体中每个萤火虫的荧光亮度；
8. 判断是否满足终止条件，如果满足终止条件则跳出循环并输出全局最优解，如果不满足条件则继续；
9. 按下列两式收缩搜索区域：

$$x_{min,j} = max\{x_{max,j}, x_{maxlight,j} - r(x_{max,j} - x_{min,j})\}, 0 < r < 1$$

$$x_{max,j} = min\{x_{max,j}, x_{maxlight,j} + r(x_{max,j} - x_{min.j})\}, 0 < r < 1$$

其中，$x_{maxlight,j}$表示当前最亮萤火虫个体的第j为变量的值。

10. 在收缩后的新搜索区域内随机产生群体中剩余80%的萤火虫，然后返回步骤3.

10.2.2 变步长萤火虫算法

在算法开始时，将初始步长设定为相对较大值，而后随着迭代次数以及萤火虫之间距离增加设定一个判定条件：当个体距离小于某一固定步长时，使步长减小。则萤火虫算法将在开始时具有较好的全局寻优能力，迅速定位在接近全局最优解的区域，而后期也具有良好的局部搜索能力，能精确得到全局最优解。

萤火虫i被亮度更大的萤火虫j吸引向其移动而更新自己的位置，位置更新公式如下：

$$x_i(t+1) = x_i(t) + \beta * (l_{ij})(x_j(t) - x_i(t)) + \alpha * (rand - 1/2)$$

式中，α为步长因子，一般取$[0,1]$上的常数；$rand$为$[0,1]$上服从均匀分布的随机因子。

萤火虫算法寻优是通过萤火虫之间的相互吸引来实现的，而随着迭代次数的增加，萤火虫群会在最优值附件聚集。此时萤火虫个体与最优值之间的距离已经非常小，在个体向最优值趋近的过程中，很可能会出现萤火虫移动的距离大于个体与最优值间距的情况，而导致个体更新自己位置时

跳过了最优值，出现震荡，将会导致最优值发现率降低，影响算法的收敛精度和速度。

为了尽量避免由上述原因造成的收敛较慢情况，在算法开始时，将初始步长设定为相对较大值，而后随着迭代次数以及萤火虫之间距离增加设定一个判定条件：当个体距离小于某一固定步长时，使步长减小。则萤火虫算法将在开始时具有较好的全局寻优能力，迅速定位在接近全局最优解的区域，而后期也具有良好的局部搜索能力，能精确得到全局最优解。

初始步长α设定为0.25，当笛卡尔距离$l > 0.25$时，位置更新公式变为：

$$x_i(t+1) = x_i(t) + \beta * (l_{ij})(x_j(t) - x_i(t)) + k * l_{ij} * (rand - 1/2)$$

式中，k为根据实际情况可调整的正相关系数，即用kl_{ij}代替固定步长α。