

细菌觅食算法报告

刘子旋，学号3150104437，计算机学院大三

2018 年 5 月 10 日

目录

1 觅食理论基础	3
2 觅食过程简介	3
2.1 搜索	3
2.2 群体觅食和智能觅食	3
3 大肠杆菌的特征	4
4 大肠杆菌游动觅食的最优化	5
4.1 数学描述	5
4.2 趋化步骤	6
4.3 复制步骤	6
4.4 消除-分散步骤	7
4.5 终止	7
5 算法运行情况	7
6 觅食算法与遗传算法的联系	8
7 更高的搜索效率——莱维飞行	8
8 我的体会	9
9 参考资料	9

1 觅食理论基础

自然选择会将不佳的觅食策略淘汰掉，或者将其改造成好的觅食策略。觅食理论将觅食活动模型化为一个最优化过程，认为觅食的动物倾向于最大化单位时间获得的能量。

$$\frac{E}{T}$$

这个假设的合理性在于单位时间获得的能量对生物的生存至关重要。最大化这个函数可以使得生物个体获得最多的营养物质以生存和进行其他活动。这个最优化问题的约束是觅食者所处的环境、觅食者感知能力等。环境决定了营养物质的分布、运动的阻碍，觅食者的感知能力决定了信息的完整程度。营养物质通常是一块块的分布。觅食的决策包括是否进入此块营养物质，继续在此块营养物质中获取食物还是离开此块营养物质转移到下一块营养物质。

2 觅食过程简介

2.1 搜索

生物个体觅食的过程可以分为搜索定位食物、追逐攻击食物、处理消化食物。下面我们只考虑搜索的过程。

搜索的模式主要有两种——巡航式和埋伏式。跳跃式搜索策略介于两者之间。在跳跃式的策略中，可以通过调整巡航的速率和埋伏的时间来适应环境。

2.2 群体觅食和智能觅食

群体合作觅食会带来一些好处：

1. 更多的动物一起寻找食物，效率更高
2. 合作捕食大型的猎物
3. 遇上天敌时可以受到同伴的保护

有时候可以把一群捕猎的生物看成是一个整体。通过组团和交流可以产生集体智慧。例如蚂蚁利用信息素进行交流可以解决很多组合最优化问题。蜜蜂通过舞蹈告知同伴花蜜的信息。

3 大肠杆菌的特征

大肠杆菌 (the *E.coli* bacterium) 指一种被深入研究的细菌。在觅食方面，它有一个控制系统使它能够趋向营养物质，避开有害物质，并尝试离开中性环境。相关研究表明，大肠杆菌的搜索策略是一种跳跃式的搜索。

大肠杆菌通过鞭毛运动。鞭毛的不同旋转方向决定了它的运动形式：逆时针转动则游动，顺时针转动则打转。游动的时候不会改变运动方向，打转的过程中决定新的前进方向。

观察发现，大肠杆菌觅食行为的特征是攀爬营养物质的梯度，即偏向于选择营养物质浓度增加的方向运动，并且在营养物质越丰富的地方游动的速率越快。

1. 如果大肠杆菌在一个中性的环境里面，它会进行一种随机游动——在游动和打转之间切换。
2. 如果大肠杆菌在营养物质浓度均匀的环境中的话，与中性的环境相比，大肠杆菌游动的时间和速率增加，打转的时间减少。（也就是说尽管在此时的位置有食物，大肠杆菌仍然会继续在周围搜索，它总是想追求更多的营养物质。我们将这种行为叫做**基线行为 (baseline behavior)**。)
3. 如果大肠杆菌在营养物质有梯度的环境中，它会花更多的时间来游动，并进一步减少打转的时间。只要其朝着营养物质增加的方向前进，它就会倾向于增加游动的时间。如果大肠杆菌移动的方向是浓度减小的方向，那么接下来它会回到基线行为，游动的时间减少、速率降低，打转的时间增加，尝试搜索一个使得浓度增大的方向。

更加细致的实验表明，大肠杆菌会根据之前4秒的营养物质浓度来做出判断。这些行为的生物学机制也已经得到研究。

还存在一种运动形式：消除-分散 (Elimination and Dispersal)。环境发生变化时，可能一个区域的所有细菌被杀死或者移动到新的环境。从大的视角来看，消除-分散是种群层面上的长距离的行为。

4 大肠杆菌游动觅食的最优化

4.1 数学描述

下面我们将大肠杆菌的觅食行为看作一个非梯度最优化（nongradient optimization）问题：假设我们想要最小化函数

$$J(\theta), \theta \in R^p$$

其中 θ 是细菌所处的位置， $J(\theta)$ 是细菌受到环境的吸引和排斥的合效果，其含义为

$$J(\theta) \begin{cases} < 0, \text{富含营养物质的环境} \\ = 0, \text{中性环境} \\ > 0, \text{有害环境} \end{cases}$$

并且我们没有其梯度 $\nabla J(\theta)$ 的测量或者解析描述。

我们知道大肠杆菌的趋向性是避开有害环境、离开中性环境、前往营养物质浓度高的地方，所以它实现的是一种有偏向性的随机行走（biased random walk）。

设细菌种群的大小为 S 。根据觅食理论，迭代的有3层：趋化、复制和消除-分散，分别用 j, k, l 标志其索引。用 $\theta^i(j, k, l)$ 表示种群中第 i 个细菌在第 l 次消除，第 k 次复制和第 j 次趋化时的位置， $J(i, j, k, l)$ 表示位置 $\theta^i(j, k, l)$ 的目标函数值（cost）， $P(j, k, l) = \{\theta^i(j, k, l) | i = 1, 2, \dots, S\}$ 表示种群。在实际的种群中， S 很大， $p \leq 3$ 。但是在觅食算法中 S 一般要小很多， p 可以大于3。

此外考虑细菌之间的通信，引入下面的细胞-细胞相互作用（cell-to-cell）项，即第 i 个细菌对位置在 θ 的细菌的作用

$$J_{cc}^i(\theta, \theta^i(j, k, l)), i = 1, 2, \dots, S$$

在群体觅食行动中，发现食物的细菌会发出信号让其他细菌聚集到食物处来。用 d_{attract} 表示吸引信号的“深度”，用 w_{attract} 表示吸引信号的“宽度”。因为很小的区域的营养物质只能供给一个细菌，所以细菌也会排斥附近的细菌，避免太过接近。我们用排斥的“高度” $h_{\text{repellant}} = d_{\text{attract}}$ 来衡量这种效果。用 $w_{\text{repellant}}$ 表示排斥的“宽度”。这些作为高斯项的参数来量化细胞-细胞相互作用的强度与两者距离的关系。

考虑以上的两种相互作用，合并的效果是

$$J_{cc}(\theta, P(j, k, l)) = \sum_i^S J_{cc}^i(\theta, \theta^i(j, k, l)) \quad (1)$$

$$= \sum_i^S [-d_{\text{attract}} \exp(-w_{\text{attract}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] + \quad (2)$$

$$\sum_i^S [h_{\text{repellant}} \exp(-w_{\text{repellant}} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \quad (3)$$

将这个效果叠加到单个细菌的cost上得到：

$$J'(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta, P)$$

4.2 趋化步骤

在趋化步骤中，用 $C(i)$ 表示第 i 个细菌的随机步长，用单位向量 $\phi(i)$ 表示一个随机方向。那么在趋化步骤中的位置变化为

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\phi(i) \quad (4)$$

如果 $J'(i, j+1, k, l) < J'(i, j, k, l)$ ，表明向这个方向前进能够使得cost减小，那么继续向这个方向前进，否则停止此步趋化。只要向这个方向游动能够减小cost，那么就一直朝这个方向前进，直到不再减小cost或者在这个方向上的步数达到 N_s 。这样就完成了一个趋化步骤。

4.3 复制步骤

经过 N_c 步的趋化之后，细菌完成了一次生命周期，将进行一次复制。选择一半的细菌进行复制，它们将一分为二。选择的标准便是他们生命周期中的趋化值之和。

$$J_{\text{health}}^i = \sum_{j=1}^{N_c+1} J'(i, j, k, l) \quad (5)$$

J_{health} 大的那半数细菌被认为得不到足够的营养将会死去，剩下的一半产生的新的细菌放置在同一位置。这种方法使得获得营养物质多的细菌存活下来繁衍后代，并且使得种群数量不变。

4.4 消除-分散步骤

当复制的次数达到 N_{re} 时，进行一次消除-分散。对种群中每个细菌以一定的概率除去并随机分到一个地方。

4.5 终止

当消除-分散的次数达到 N_{ed} 时，算法停止。

5 算法运行情况

为了说明算法的有效性，使用论文中提供的参数

```
pop_size = 50
step_size = 0.1 # Ci
elim_disp_steps = 1 # Ned
repro_steps = 4 # Nre
chem_steps = 70 # Nc
swim_length = 4 # Ns
p_eliminate = 0.25 # Ped
d_attr = 0.1
w_attr = 0.2
h_rep = d_attr
w_rep = 10
```

采用<http://www.cleveralgorithms.com/nature-inspired/swarm/bfoa.html>提供的Ruby代码对一个简单的最优化问题进行模拟进行计算。目标函数是

$$\min \sum_{i=1}^2 x_i^2, -5.0 < x_i < 5.0 \quad (6)$$

每经过一次细菌生命周期， J' 和 J 以及细菌的坐标的记录如下：

J'	J	$[x_1, x_2]$
-2.8789	0.00015078	[-0.010171, -0.0068793]
-3.4527	1.3425e-05	[0.0036215, -0.00055663]
-3.4527	1.3425e-05	[0.0036215, -0.00055663]
-3.4527	1.3425e-05	[0.0036215, -0.00055663]

这表明仅仅经过一轮细菌的生命周期，就已经收敛到了最优值。目标函数的最小值为0，最优解在(0.0,0.0)处。

要证明觅食算法在多极值、非梯度的最优化问题上的表现良好，还需进一步大量的实验或查找相关的实验文献。

6 觅食算法与遗传算法的联系

我们可以将遗传算法与觅食算法的过程进行以下类比：

遗传算法	觅食算法
适应度函数	营养物质浓度函数
选择个体繁殖	选择细菌复制
交叉	细菌分裂
变异	消除-分散

但是这两个算法的含义很不相同。适应度函数指的是个体生存的概率，营养物质浓度函数指的是该个体所处环境的营养物质的丰富程度或者有害物质的浓度和其他环境影响因素；交叉指的是繁殖产生不同的后代，而在觅食算法中没有考虑这一点——细菌分裂产生的两个后代被认为是具有相同的觅食能力；变异指的是基因发生变化导致表现型发生变化，而消除-分散指的是分布在一个地理区域。

但是从另一个角度看，遗传算法的所有特征都可以用来增强觅食算法，只要我们将觅食者演化的特征表示在它所处的环境中。例如遗传算法中的突变可以体现在觅食算法中，两个生成的细菌被放置在不同的位置。

7 更高的搜索效率——莱维飞行

细菌在随机搜索时，如果采用的是布朗运动，那么单位速度在时间 t 内的位移量约为 \sqrt{t} ，这种效率相对比较低。如果采用莱维飞行的话，因为莱维分布是重尾分布（Heavy tailed distribution）的，而布朗运动是一种典型的轻尾分布。这样的话，平均位移就能提升到 t^γ , ($\gamma > 1/2$)。

莱维过程 $X(t), t \geq 0$ 的定义是：

1. $X(0)$ 几乎处处为0
2. 具有独立增量性
3. 具有稳定增量性

4. 样本轨道右连续

莱维过程可以看作是“跳跃”的布朗过程，这些不连续的“跳跃”给予了莱维过程“重尾”的特性。

以下是关于莱维过程两个重要的定理：

（莱维-辛钦公式）莱维过程的傅里叶变换：

$$\phi_X(\theta)(t) := E[e^{i\theta X(t)}] = \exp(t(ai\theta - \frac{1}{2}\sigma^2\theta^2 + \int_{R \setminus \{0\}} (e^{i\theta x} - 1 - i\theta x I_{|x| < 1}) \Pi(dx))) \quad (7)$$

（莱维-伊藤分解）每个莱维过程都可以分解为 $\{S(t), t \geq 0\}$ 、 $\{Y(t), t \geq 0\}$ 和 $\{Z(t), t \geq 0\}$ 三个子过程，其中 $S(t)$ 是维纳过程（就是布朗运动，莱维过程的连续部分）， $Y(t)$ 是复合泊松过程（刻画了较极端的“跳跃”现象）； $Z(t)$ 是平方可积的离散鞅（刻画了较小的“跳跃”现象）。

8 我的体会

在这门课的学习中，接触到了很多生物学启发下的最优化算法，通过模仿自然的现象可以设计出一系列执行效率良好并且获得较优的解的算法，包括遗传算法、免疫算法、鱼群算法、觅食算法等，还有强化学习、深度学习等技术。这让我对组合优化算法有了新的、更深刻的理解。

在数学方面，我认识到尝试构建一个问题的几何直观和物理直观对于深刻理解该问题具有很大的帮助。培养这种能力需要多阅读相关的科普性质的文献，并在实际的研究中加以应用。对于一个算法或者数学公式，如果能在脑海中用几何的方式呈现出来，那么不容易忘记、也不容易记错。

在小组的合作方面，Github是一个很好的平台。通过Github进行协作，有序有效率、不会丢失版本，各人之贡献一目了然。用 \LaTeX 进行文档的撰写可以更好的搭配Github，在排版、公式编辑方面亦有帮助。

9 参考资料

[1] Biomimicry of bacterial foraging for distributed optimization and control, K.M. Passino

[2] <http://blog.sciencenet.cn/blog-3364297-1100074.html>, yxgyylj的个人博客