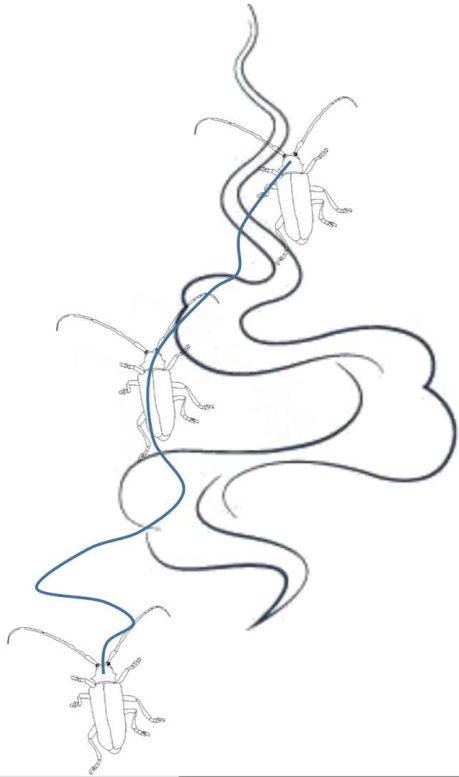
**进化算法**

# 天牛须搜索算法

## 基本天牛须搜索

图1(a)所示的长角天牛是天牛科的一种，其特征是触角非常长，通常和天牛的身体一样长，甚至更长。天牛家族成员众多，共有26,000多个物种。它们中的大多数都有长长的触角。通常由多种嗅觉感受器细胞组成的触角结构，在特定物种中往往是独特的，而这类传感系统的功能仍在研讨中。然而，这种大型触角的两个基本功能是与猎物的气味结合，并获得潜在的合适伴侣的性信息素，其中大触角可以扩大探测区域。此外，大触角也可以作为一种保护预警机制。



（a） （b）

图1 长角天牛及其使用长触角的搜索行为。（a）长角天牛。（b）长角天牛利用长触角进行搜索，其中黑色线表示气味的传播，蓝色线表示天牛的轨迹。

我们知道，天牛在捕食或寻找配偶时，会摆动身体一侧的每根触角来接收气味。也就是说，天牛用两根触角随机探索附近的区域。此外，当一侧的触角探测到更高浓度的气味时，天牛会转向同一方向，否则就会转向另一侧。如图1(b)所示，这两个因素结合在一起，使得大多数天牛能够捕食或者寻找配偶，这启发我们设计一种元启发式优化算法。基于这两个方面，Li Shuai和Jiang Xiangyuan[1]提出了天牛须搜索算法（Beetle Antennae Search Algorithm，BAS），其伪代码如下：

Input：建立目标函数f(**x**t)，其中变量**x**t=[x1,x2,...,xi]T，初始化参数**x**0，d0，δ0。

Output：**x**bst，fbst。

while (t<Tmax) or (stop criterion) do

根据(1)创建单位方向向量；

根据(2)使用两类触角在变量空间中搜索；

根据(3)更新状态变量**x**t；

if f(**x**t)> fbst then

f(**x**t)= f(**x**t)，**x**bst=**x**t；

分别使用下降函数(4)和(5)更新感知参数d和步长δ；

return **x**bst，fbst。

通过将双触角天牛的搜索行为与待优化的目标函数相关联，可以构造出新的优化算法。接下来，将概述BAS算法的形式化表达，然后详细讨论其实现。

为了便于说明，使用**x**t表示t时刻天牛的位置，则位置**x**处的气味浓度为f(**x**)，也称为适应度函数，其最大值的位置对应于气味的源点。

为了简单描述BAS算法，现在使用以下两条受到触角天牛搜索行为（包括搜素行为和探测行为）启发的规则。注意，天牛为探索未知的环境是随机进行搜索的。

第一，为了对天牛的搜索行为进行建模，提出了如下描述天牛搜索的随机方向：

 （1）

其中rnd(.)表示随机函数，k代表位置的维度。此外，还分别提出了左右两侧的搜索行为，以模拟天牛触角的活动：

 （2）

其中**x**r表示位于右侧的搜索区域的位置，**x**l表示位于左侧的搜索区域的位置，d是触角的感知长度，代表探索能力，初始时该值应该足够大以覆盖合理的搜素区域，从而保证跳出局部最优点，然后随着时间的流逝而逐渐衰减。

第二，为了表达检测行为，进一步生成如下迭代模型，通过考虑搜索行为来关联气味检测：

 （3）

其中δ是搜索步长，用于调整收敛速度，是时间t的递减函数，而不是递增函数或常量。δ应该初始化为与搜索相同的值。sgn(.)表示符号函数，即输入大于0返回1，等于0返回0，小于0返回-1。

对于搜索参数，即触角长度d和步长δ，使用如下如下规则进行更新：

 （4）

 （5）

## 天牛须搜索变体

### Beetle Antennae Search without Parameter Tuning (BAS-WPT)

在原始的BAS中，参数d和δ的设定会严重影响算法的性能，因此作者Jiang Xiangyuan和Li Shuai[2]试图对其进行改进以提高有效性和鲁棒性。

图2描述了BAS-WPT的迭代优化过程，可以清楚地看出其实一种变尺度算法。

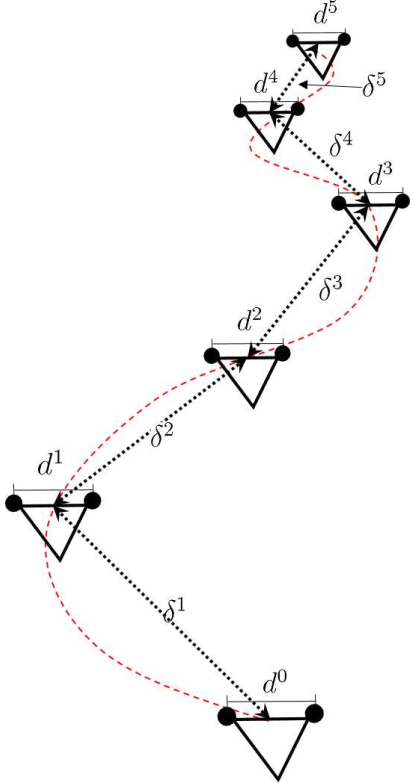


图2

为了简便起见，采用了归一化方法自适应地调整BAS参数，假设**x**的第i个元素为xi，其取值范围为xi,l到xi,h，那么**x**满足**x**∈[**x**l,**x**h]，其中**x**l为下界，**x**h为上界。每次迭代时适应度函数的输入数据可以通过以下表达：

 （6）

最后通过BAS-WPT算法，使用归一化的变量**x**bst获得对应位置bst处的全局最优值f()bst。

为了进一步简化参数调优，构建了如下所示的搜索距离d和步长δ之间的关系;

 （7）

其中c1和c2为由用户设定的常量。

作者还利用惩罚函数方法将BAS-WPT扩展到了约束优化问题。

约束问题的一般形式如下：

 （8）

为了能够解决约束优化问题，提出了惩罚函数方法来处理不等式约束。在惩罚函数中，通过在原适应度函数上加入惩罚项，对违反不等式约束的非可行解进行惩罚，从而降低选择非可行解的概率。其使用的惩罚函数表达如下：

 （9）

其中F(**x**)为改进的适应度函数，f(**x**)为原始适应度函数，λ为惩罚参数，通常定义为一个足够大的值（如1010），约束违反hj(**x**)定义为：

 （10）

当任一不等式约束gj(**x**)满足同时伴随着很大的λ值，那么式（9）中的第二项将主导适应度函数，使得F(**x**)趋于无穷大，反之，如果所有的hj(**x**)均满足，那么F(**x**)=f(**x**)。

BAS-WPT算法伪代码如下：

Input：初始化输入数据**x**0，对其每一个元素进行标准归一化：xi0=[rnd(.)-(ximin)]/(ximax-ximin)，初始化参数c1，c2和δ0。

Output：**x**bst，fbst。

while (t<Tmax) or (停止准则) do

根据(2)使用两类触角在变量空间中搜索；

根据(3)更新状态变量**x**t；

根据(7)生成归一化向量；

根据(9)和(10)构造改进的适应度函数；

if F(**x**t)满足最优条件 then

Fbst=F(**x**t)，**x**bst=**x**t；

使用(7)更新参数，使用**x**bst计算利用式(6)计算最优位置bst；

return **x**bst，fbst。

### 天牛群算法（Beetle Swarm Antennae Search，BSAS）

**参考文献**

1. Jiang, X. and S. Li, *BAS: Beetle Antennae Search Algorithm for Optimization Problems.* International Journal of Robotics and Control, 2017. **1**.

2. Jiang, X. and S. Li *Beetle Antennae Search without Parameter Tuning (BAS-WPT) for Multi-objective Optimization*. arXiv e-prints, 2017.