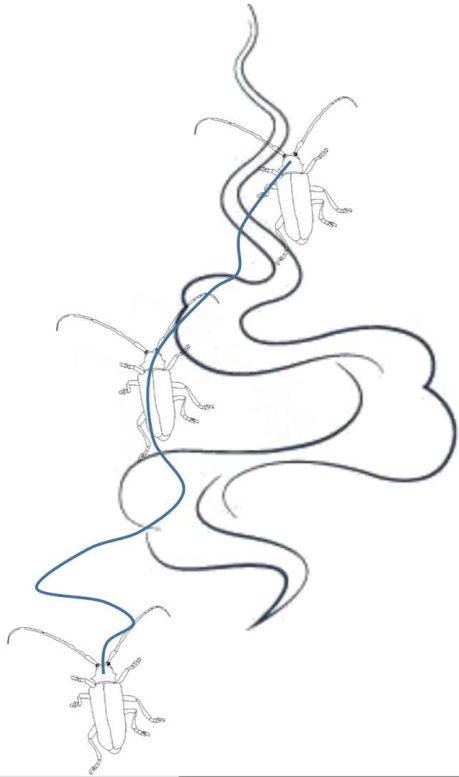
# 天牛须搜索算法

## 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。

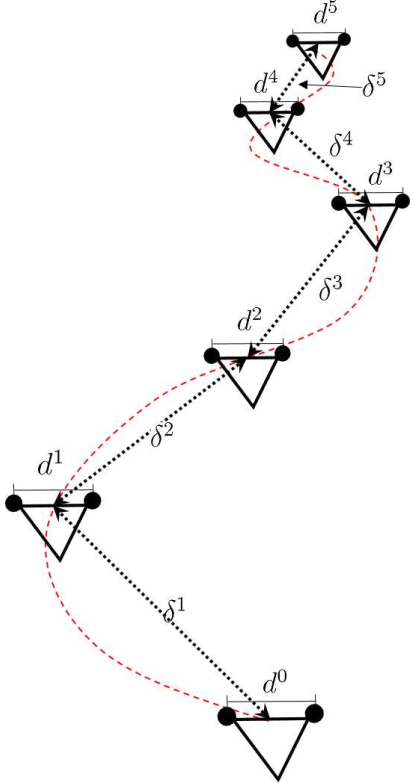


图2 BAS-WPT5步内的迭代优化过程。红色虚线表示适应度函数的轨迹，三角形表示天牛，其两边的实心圆是天牛的触角，d是触角间的距离，黑色虚线表示搜索的步长δ。

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）

一般来说，BAS算法有两个缺点。首先，天牛在每次迭代中都向一个单一的随机方向移动，但不能保证天牛的移动会使目标函数值更好。相应地，我们可以通过k只天牛向k个方向移动来改善BAS算法的搜索行为，从而提高天牛找到更好位置的可能性。其次，天牛的每一次移动都会导致位置和步长的更新，事实上，如果目标函数较好，应该更新天牛的位置，而不更新步长，反之亦然。因此，作者Wang Jiangyu和Chen Huanxin[3]设计了一个基于反馈的位置和步长更新策略，如下所示。

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，k，pδ。

Output：**x**bst，fbst。

while (t<Tmax) or (停止准则) do

创建k个单位方向向量[1,2,...k]；

根据(2)让k只天牛使用两类触角在变量空间中搜索；

if min(f(**x**it))<fbst，其中i∈[1,2,...k] then

fbst=f(**x**it)；

**x**bst=argmin f(**x**it)；

**x**t+1=**x**bst。

else

if rnd(1)>pδ then

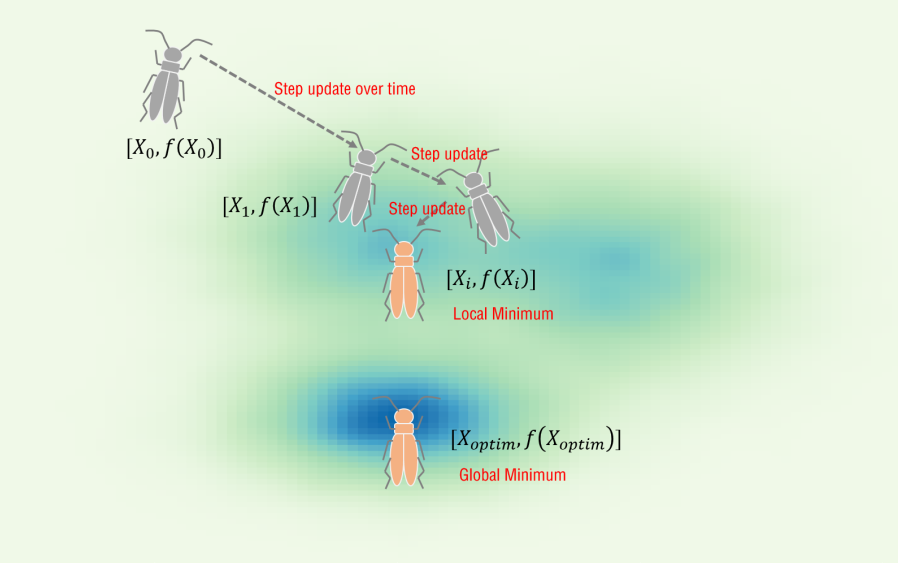
分别使用下降函数(4)和(5)更新感知参数d和步长δ；

else

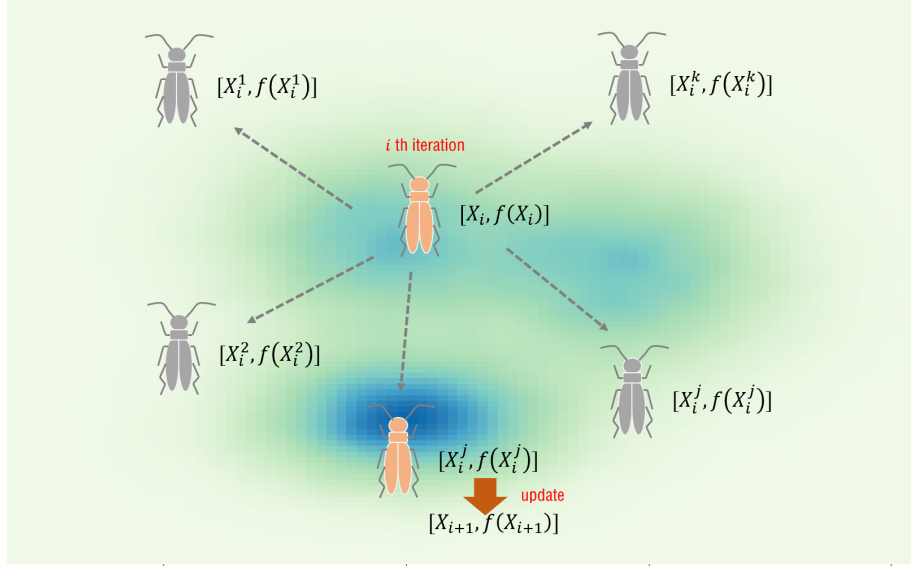
参数δ和d保持不变。

return **x**bst，fbst。

此外，还引入了概率常量pδ来度量随机方向的影响，换句话说，就是有一较小概率pδ，k只天牛会错过当前步长下目标函数的更优位置（参数集），因此作者创建了一个属于[0,1]的随机数并与pδ比较，由此，在大多数情况下，如果k只天牛不能找到更小的目标值，BSAS算法就认为在当前步长下没有更优的位置，需要对步长进行更新。在少数情况下，算法认为目前仍有一些位置可以满足天牛将目标最小化的需求，但由于天牛数量k有限，这些天牛无法找到最优位置，因此如果随机数小于pδ，那么步长δ和感知长度d仍然保持不变。然而，只有当天牛群找到了优化目标函数的更优位置时，才会对天牛的位置进行更新，如图3所示。



（a）BAS中的步长更新。收敛结果不是全局最优，因为步长随时间降低，而不是目标函数。



（b）BSAS中的位置更新过程。在每次迭代中，采用k只天牛寻找更优的参数集，当其中一只天牛找到优化目标函数的更优参数集时才会更新天牛位置。

图3 BAS和BSAS的区别

**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.

3. Wang, J. and H. Chen *BSAS: Beetle Swarm Antennae Search Algorithm for Optimization Problems*. arXiv e-prints, 2018.