**果蝇算法步骤**

1. 初始化

初始化种群大小、最大迭代次数、果蝇群体位置范围

初始化果蝇群体每一个个体的位置

C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml1852\wps1.jpg

1. 嗅觉搜索过程

对每个果蝇个体，赋予随机飞行方向和距离

C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml1852\wps2.jpg

计算每个个体到原点的距离：

C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml1852\wps3.jpg

然后计算这个个体的维度浓度判定值：

C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml1852\wps4.jpg

最终计算这个个体的适应度函数值：

C:\Users\ADMINI~1\AppData\Local\Temp\ksohtml1852\wps5.jpg

1. 选择当前群体中适应度函数值最大的个体BestIndividual，更新最优适应度值和位置，
2. 视觉搜索过程

对群体中的所有果蝇，都飞向这个BestIndividual的位置（直接坐标复制就好）



记迭代次数+1

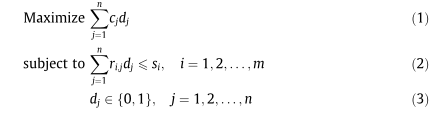
1. 终止判断

然后重复步骤1-5，直到达到最大迭代次数。

果蝇算法解决多维背包问题

**<1>问题描述：**

1. 有n项物品，每项物品有一定价值，某项物品的价值用表示。
2. 有1个背包，对于这个背包装物品有多个维度容量的限制（比如重量、空间等），某个维度的容量由表示。
3. 不同物品在不同维度所占用的容量都不同，占用用一个矩阵r表示,表示物品j在背包i维度限制中所占的容量。
4. 要求的是如何分配才能装上最多价值的物品。



**<2>解决思路**

2.1 <解方案是怎么表示的？方案是如何生成的？>

1. 用一个长度为N的0-1串代表解方案（果蝇个体），其中第i个值代表第i项物品是否被选中。
2. 在每一轮迭代中都有一个和0-1串一起的长度也是N的概率向量，代表这个物品被选中的概率。这个向量的初始值是[0.5,0.5….,0.5,]。在后面每一轮迭代中，个体方案的生成都是根据这个概率向量来的。

2.2 <嗅觉搜索阶段>

在这个嗅觉搜索阶段，对于每一个果蝇个体，最终都要生成S个邻居。算法步骤：

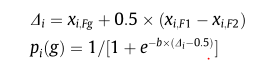
对于每个个体，执行一下算法：

1. 随机选择L个bit位。
2. 对这些位元素进行变换（0->1, 1->0）

进行上述步骤后，对于每一个个体果蝇，都能在这S个邻居中找到局部最优个体，如果局部最优个体的适应度值比自身高，那么这个果蝇就用这S个邻居中的最优个体代替。

2.3 <全局视觉搜索阶段>

概率向量更新公式：



其中Fg是全局最优个体，F1和F2代表两个随机选择的个体。I代表概率向量中的第i个维度。

2.4<修复操作>

**一阶算法如下**

输入：解个体

输出：优化后（保证符合约束条件）的解个体。

步骤：

1. 从所有维度计算当前解的空间占用（是一个长为限制维度个数N的向量）。计算每项物品（或者说每个变量）的伪效用值向量。
2. 对于每一个物品（按照伪效用值向量升序），进行如下判断和计算：

2.1 如果这个物品被选中了，并且存在某个维度的空间占用超过了限制，那么drop掉这个物品，然后返还这个物品在所有维度的空间占用。

3、对于每一个物品（按照伪效用值向量降序），进行如下判断和计算：

3.1如果这个物品没有被选中，并且对于这个物品，加上这个物品后，在所有维度的空间占用都小于限制，那么这个物品就被add,然后添加更新空间占用向量。

**二阶算法如下：**

输入：解个体

输出：优化后（适应度值提升后）的解个体。

步骤：

1. 计算单元价值矩阵Q(一个m\*n的矩阵，Q[i][j] =，代表物品j在维度i的单位价值)。
2. 计算序矩阵F（一个m\*n的矩阵），这个序矩阵每一列都代表一项物品在各维度的单位价值的序（备注，这个序是指降序，比如F [i][j]代表物品j在维度i的价值的排序，如第1大，第2大…等）。
3. 进行drop和add操作：
4. 从所有维度计算当前解的空间占用（是一个长为限制维度个数N的向量）。
5. 对每一个维度，进行如下操作：

2.1、按照F在这个维度中各物品的序（从小到大）对物品进行遍历，如果该物品被选中了并且这个物品中有任何一个维度超出了限制，那么便舍弃这个物品，并且返还其在所有维度的空间占用。（DROP）

1. 对每一个维度，进行如下操作：

3.1、按照F在这个维度中各物品的序（从大到小）对物品进行遍历，如果该物品没被选中并且这个物品所有维度的空间占用加上后都在限制之内，那么便添加上这个物品，并且负载上其在所有维度的空间占用。（ADD）

**<3>整体算法步骤和框图**

