

# 基于混合果蝇算法的桩锚支护深基坑 临界滑面搜索

马泽宁, 沙成满, 路明浩  
(东北大学 资源与土木工程学院, 辽宁 沈阳 110819)

**摘要:** 进行基坑整体稳定性分析常采用极限平衡法, 但仍然需要依据经验试算一系列滑面, 将安全系数最小的滑面确定为最危险滑面. 针对此问题, 提出将果蝇优化(FOA)算法与禁忌搜索(TS)算法融合, 提出自适应步长的混合果蝇优化算法(HFOA), 以克服基本果蝇算法局部寻优精度不高且易陷入局部最优的缺点, 确保获得全局最优解, 并结合简化Bishop算法用于临界滑面的搜索. 在Matlab中编程实现该算法, 通过与6种启发式算法进行对比, 结果表明, HFOA适用于均质土悬臂支护基坑、成层土和含软弱夹层的桩锚支护基坑, 相较于遗传算法等6种算法具有更快的收敛速度、更高的收敛精度和可靠性, 为深基坑临界滑动面搜索提供了一种新的求解策略.

**关键词:** 深基坑; 整体稳定性; 果蝇优化算法; 禁忌搜索算法; 最小安全系数

中图分类号: TU 473.1 文献标志码: A 文章编号: 1005-3026(2024)01-0120-09

## Critical Slip Surface Search for Pile-Anchor Supported Deep Foundation Pits Based on Hybrid Fruit Fly Algorithm

MA Ze-ning, SHA Cheng-man, LU Ming-hao

(School of Resources & Civil Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: SHA Cheng-man, E-mail: shachengman@mail.neu.edu.cn)

**Abstract:** The limit equilibrium method is used for the overall stability analysis of the pit, requiring iterative calculation of various potential slip surfaces. The slip surface with the smallest safety factor is identified as the most dangerous slip surface. The fruit fly optimization algorithm (FOA) with the tabu search (TS) algorithm is fused, and the hybrid fruit fly optimization algorithm (HFOA) with adaptive step size is proposed to overcome the limitations of the fruit fly optimization algorithm, such as lower accuracy and susceptibility to local optimum. In order to ensure that the global optimal solution is obtained, the simplified Bishop algorithm for the search of critical slip surfaces is combined. By comparing with the six heuristic algorithms, it is shown that the HFOA is applicable to cantilever-supported pits with homogeneous soil, layered soil and pile-anchored supported pits with soft soil. It is more accurate and efficient compared with the six algorithms such as genetic algorithm, and provides a new solution strategy for identifying critical sliding surface in deep foundation pits.

**Key words:** deep foundation pit; integral stability; fruit fly optimization algorithm; tabu search algorithm; minimum safety factor

边坡稳定性分析是岩土工程的重要课题, 评价边坡稳定性常采用极限平衡法、极限分析法和数值计算方法. 极限平衡法公式简单, 工程应用方便, 因此我国采用极限平衡法作为工程规范,

用于求解基坑整体失稳临界滑动面的安全系数, 即求解目标函数(安全系数)的最小值. 基坑作为边坡的特殊形式, 极限平衡法也适用于基坑的整体稳定计算. 实际工程中大多数基坑为非均质

收稿日期: 2023-04-12

作者简介: 马泽宁(1999-), 男, 河北承德人, 东北大学硕士研究生; 沙成满(1964-), 男, 黑龙江哈尔滨人, 东北大学副教授, 硕士生导师.

土,且土体参数和土层分布十分复杂<sup>[1]</sup>,导致基坑整体稳定安全系数的目标函数通常为多维函数,且存在多个极值,工程实践中往往采用枚举法进行最小安全系数的搜索.枚举法根据工程具体情况,设定一个圆心坐标和半径范围,并根据精度要求在该范围进行逐一搜索,其最小值即为安全系数的真实值.在此基础上,为提高搜索效率,出现了二分法、坐标轮转法等方法.随着计算能力的提高,采用智能优化算法搜索临界滑动面已成为主流方法,如遗传算法<sup>[2]</sup>、粒子群算法<sup>[3]</sup>、蚁群算法<sup>[4]</sup>、禁忌算法<sup>[5]</sup>、鱼群算法<sup>[6]</sup>、模拟退火算法<sup>[7]</sup>等.与二分法、坐标轮转法等方法相比,以上算法的精度和收敛性能已经有了很大的提高,对临界滑动面搜索研究和实际工程应用起到了积极的作用.但随着基坑环境复杂程度的不断增加,这些算法表现出收敛速度慢、易早熟、鲁棒性差等缺点.

果蝇优化算法(fruit fly optimization algorithm, FOA)<sup>[8]</sup>是由潘文超在 2011 年提出的全局优化算法,在多个领域已得到应用,但尚未应用到深基坑桩锚支护结构的整体稳定性计算中.已有研究表明,FOA 相较于遗传算法、鱼群算法和蚁群算法调节参数少、计算量小、收敛速度快;相较于粒子群算法计算量小,计算精度高<sup>[9]</sup>;相较于模拟退火算法无需设置初始值即可搜索到最优解.但是该算法也存在一些不足之处:其局部搜索能力不强,导致后期搜索效率降低,容易陷入局部最优.相较之下,禁忌搜索(TS)算法局部寻优能力很强,但是非常依赖于初始值的设定,如果初始值设定不合理,则很难搜索到最优解.

本文将收敛速度快的 FOA 和局部搜索能力很强但是对初始值选取要求很高的 TS 算法相结合,首先借助果蝇算法快速收敛到全局最优解附近,再通过禁忌算法进行深度寻优,这样既解决

了果蝇算法局部寻优能力不强的问题,也满足禁忌算法对初始值选取的要求.将融合算法应用在基坑复杂土层条件下的临界滑动面搜索问题中,探究新算法的准确性和可靠性,并与 6 种不同的启发式优化方法进行对比,探究各算法的适用场景.

## 1 边坡临界滑面搜索理论模型

简化 Bishop 法<sup>[10]</sup>是目前常用的一种稳定性分析方法,其力学模型如图 1 所示.该模型假定每一个土条的条间力方向水平,切向力为 0.根据滑动面上各土条的静力平衡条件和整体的力矩平衡条件可得到以下安全系数计算公式:

$$FS = \frac{1}{\sum_i^n W_i \sin \alpha_i} \cdot \sum_i^n \frac{c_i b_i + (W_i - u_i b_i) \cdot \tan \varphi_i}{\cos \alpha_i + \frac{\tan \varphi_i \cdot \sin \alpha_i}{FS}} \quad (1)$$

式中:FS 为对应滑动面的安全系数; $n$  为滑动土体条分数; $\alpha_i, c_i, \varphi_i$  为第  $i$  条土条底面中点切向与水平向的夹角、黏聚力和内摩擦角; $b_i$  为滑动体第  $i$  条土条宽度; $W_i$  为滑动体第  $i$  条土条所受重力; $u_i$  为第  $i$  条土条底面的孔隙水压力.

当锚索对土体进行加固,只考虑滑动面上锚索的抗拔力,改进简化 Bishop 计算公式为<sup>[11]</sup>.

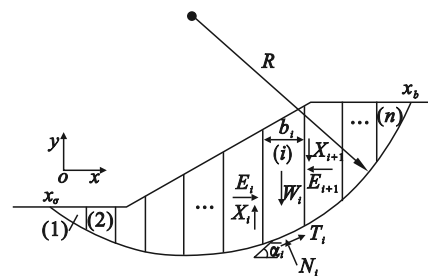


图 1 简化 Bishop 法滑体与条块示意图

Fig. 1 Schematic diagram of the simplified Bishop method for sliders and bars

$$FS = \frac{\sum_{i=1}^n \left\{ \frac{c_i b_i + (W_i + q b_i - u_i b_i) \tan \varphi_i + \zeta_i R_j \sin \delta_j \tan \varphi_i / S_x}{\cos \alpha_i + \frac{\tan \varphi_i \cdot \sin \alpha_i}{FS}} \right\}}{\sum_{i=1}^n \left\{ (W_i + q b_i) \sin \alpha_i - \zeta_i R_j \cos(\delta_j + \alpha_i) / S_x \right\}} \quad (2)$$

式中: $R_j$  为锚索锚固力; $S_x$  为锚索水平间距; $q$  为外荷载; $\delta_j$  为锚索倾角; $\zeta_i$  锚索穿过第  $i$  条块底面的判定参数,若  $\zeta_i = 1$  则说明该条块  $i$  有锚索穿过,  $\zeta_i = 0$  表示无索杆穿过.

式(2)两侧均有安全系数未知量,需迭代计算求解.若土体的形状、力学性质和支锚参数确

定,那么安全系数的目标函数是关于圆弧滑动面圆心坐标和半径的函数,如式(3)所示:

$$FS = f(x_0, y_0, R). \quad (3)$$

式中, $x_0, y_0, R$  分别为滑动面圆弧的圆心坐标和圆弧半径.

那么可转化成式(4)所示的优化问题.

$$\left. \begin{array}{l} \min f(x_0, y_0, R), \\ \text{s.t. } a(j) \leq x(j) \leq b(j), (j=1, 2, \dots) \end{array} \right\} \quad (4)$$

式中:  $x_0, y_0, R$  为优化变量;  $[a(j), b(j)]$  为第  $j$  个变量的取值区间.

## 2 混合果蝇优化算法原理

### 2.1 基本果蝇优化算法

果蝇优化算法是通过模拟果蝇觅食过程形成一种新型启发式优化算法, 步骤如下<sup>[8, 12-13]</sup>:

步骤 1 设置果蝇种群规模初始值  $\text{Sizepop}$ 、最大迭代次数  $\text{Maxgen}$ , 随机生成果蝇群体初始坐标  $X\_axis$  和  $Y\_axis$ .

步骤 2 果蝇个体利用嗅觉搜索随机搜索食物位置,  $\text{RandomValue}$  为飞行距离.

$$\left. \begin{array}{l} X_i = X\_axis + \text{RandomValue}, \\ Y_i = Y\_axis + \text{RandomValue}. \end{array} \right\} \quad (5)$$

步骤 3 由于食物位置未知, 先计算果蝇个体位置到原点的距离  $D_i$ , 再计算味道浓度判定值  $S_i$ , 即  $S_i = 1/D_i$ .

步骤 4 把  $S_i$  代入适应度函数  $\text{Fitness}$ , 求出各果蝇个体味道浓度值  $\text{Smell}(i)$ .

步骤 5 把味道浓度最佳的果蝇个体找出, 记录其最佳味道浓度值  $\text{bestSmell}$ , 以及其  $X, Y$  坐标值,  $\text{bestindex}$  为味道浓度最佳的果蝇的位置索引.

步骤 6 果蝇群体通过视觉飞向味道浓度最佳的果蝇个体.

$$\left. \begin{array}{l} \text{Smellbest} = \text{bestSmell}, \\ X\_axis = X(\text{bestindex}), \\ Y\_axis = Y(\text{bestindex}). \end{array} \right\} \quad (6)$$

步骤 7 进入迭代过程, 在最大迭代次数内重复执行步骤 2 ~ 步骤 6, 若本次迭代最佳味道浓度优于前一次迭代的味道浓度, 那么更新最佳味道浓度值及果蝇位置坐标, 否则结束算法.

### 2.2 禁忌搜索算法

禁忌搜索算法可分为以下 5 个步骤<sup>[13]</sup>:

步骤 1 给定初始解  $x_0$ , 即当前解  $x = x_0$ ;

步骤 2 生成当前解的邻域, 找出邻域内的最优解  $x^*$ ;

步骤 3 若邻域内最优解好于当前最优解 (即蔑视原则), 把邻域最优解作为当前解  $x = x^*$ , 同时进行赋值:  $x_{\text{best}} = x^*$ ,  $f(x_{\text{best}}) = f(x^*)$ , 转入步骤 5; 否则进入步骤 4;

步骤 4 若禁忌表中储存了  $x^*$ , 那么将邻域

内的  $x^*$  删除, 转入步骤 3, 否则将  $x^*$  赋值给当前解, 进入步骤 5;

步骤 5 更新禁忌表, 判断当前是否满足终止条件, 若满足终止条件, 则停止计算, 否则进入步骤 2.

### 2.3 混合果蝇优化算法

#### 2.3.1 Kent 混沌映射

混沌映射常在优化领域用于替代伪随机数生成器, 生成 0 到 1 之间的混沌数, 取得的效果比伪随机数要好. 为了保证果蝇初始位置的均匀性, 采用 Kent 混沌映射方程对果蝇群体进行初始化, Kent 混沌方程为<sup>[14]</sup>

$$z^{(i+1)} = \begin{cases} \frac{z^{(i)}}{\zeta}, & 0 \leq x \leq \zeta; \\ \frac{1-z^{(i)}}{1-\zeta}, & \zeta < x \leq 1. \end{cases} \quad (7)$$

式中:  $\zeta$  为常数, 取 0.4 时概率密度呈均匀分布<sup>[15]</sup>;  $z$  为混沌序列.

若优化变量的取值范围为  $[x_{\min}, x_{\max}]$ , 则生成的初始种群位置公式为

$$\text{xid}_0 = x_{\min} + z(i)(x_{\max} - x_{\min}). \quad (8)$$

#### 2.3.2 指数递减步长

搜索步长是决定 FOA 搜索能力的关键. 采用自适应步长可在迭代初期采用较大步长进行全局搜索, 加快迭代过程的收敛速度. 随迭代次数增加, 搜索区域已是最优解的潜力区域, 因此通过较小步长进行搜索. 本文采用指数递减策略<sup>[16]</sup>对果蝇算法进行优化.

$$w = w_{\min} \times \left( \frac{w_{\max}}{w_{\min}} \right)^{\frac{1}{1+c \cdot \frac{t}{t_{\max}}}}. \quad (9)$$

式中:  $w$  为搜索步长;  $w_{\max}, w_{\min}$  分别为  $w$  的最大值和最小值, 取  $w_{\max} = 1, w_{\min} = 0.3$ ;  $t$  为当前迭代次数;  $t_{\max}$  为最大迭代次数;  $c$  取值 0.5.

#### 2.3.3 FOA 后期的收敛指标

果蝇算法在迭代过程中, 全体果蝇每次都是从某一最佳味道浓度的位置出发进行搜索, 当这一果蝇群体趋近最优解时, 需要转入禁忌算法, 如果过早进入禁忌算法进行深度寻优, 果蝇算法尚未趋近最优解, 由于禁忌算法具有很高的初始值要求, 算法将陷入局部最优; 如果过晚进入禁忌算法, 那么计算效率低下. 群体适应度方差是一个表征算法局部收敛状态的指标, 可以将其设置为某一常数, 若群体适应度方差小于此值, 则转入禁忌算法; 否则, 继续迭代, 直到进入局部收敛状态为止. 群体适应度方差的计算公式为

$$\left. \begin{aligned} \sigma^2 &= \sum_{i=1}^{Sizepop} (Smell_i - Smell_{avg})^2, \\ Smell_{avg} &= \sum_{i=1}^{Sizepop} Smell_i / Sizepop. \end{aligned} \right\} \quad (10)$$

式中,  $Smell_{avg}$  为每次迭代果蝇群体的味道浓度平均值.

### 2.4 混合果蝇算法的运算流程

混合果蝇算法分为以下几个步骤:

步骤 1 采用 Kent 映射初始化果蝇位置  $X\_axis, Y\_axis$ , 种群规模  $Sizepop$ 、适应度方差阈值  $\delta$ 、最大迭代次数  $Maxgen$  和初始飞行距离  $L$ ;

步骤 2 自适应变化飞行距离,式(9)为飞行距离变换公式;

步骤 3 执行标准果蝇算法中的步骤 2~步骤 6;

步骤 4 计算本次果蝇种群觅食迭代的群体适应度方差  $\sigma^2$ ,式(10)为群体适应度方差的计算公式;

步骤 5 若本次迭代的群体适应度方差小于适应度方差阈值,即  $\sigma^2 < \delta$ ,表明果蝇算法已进入有潜力区域,发生了局部收敛,进入禁忌算法步骤 2~步骤 5.

图 2 为 HFOA 的流程图.

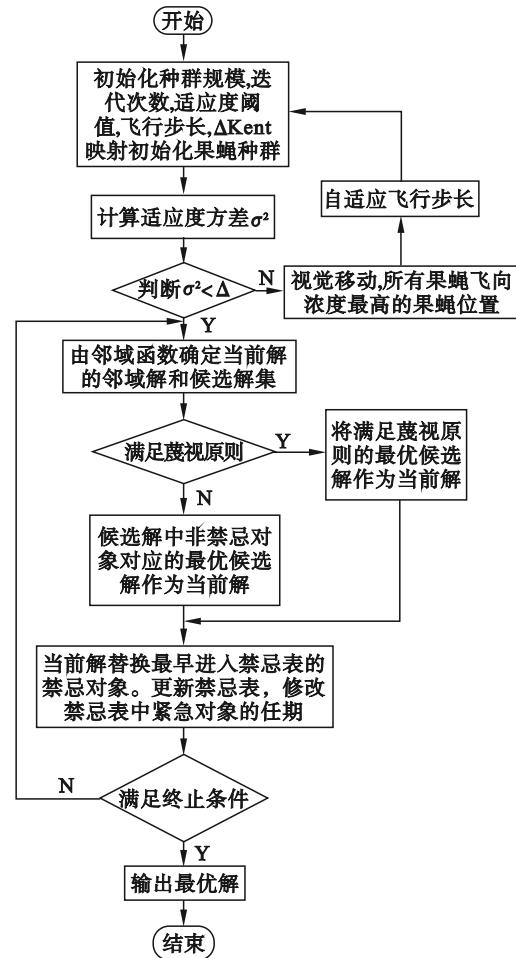


图 2 HFOA 流程图

Fig. 2 Flow chart of HFOA

## 3 算例分析

基于混合果蝇算法,结合安全系数计算的简化 Bishop 法,在 Matlab 中编写了相应的基坑临界滑面搜索程序,并对 3 个算例进行了验证分析.

### 3.1 均质土悬臂支护基坑

由于本文研究的是算法在基坑整体稳定中的应用,因此其他破坏模式暂不考虑.以某均质土悬臂支护基坑为例,表 1、表 2 为工程参数信息.

表 1 总体信息  
Table 1 General information

土层数	1	支挡层数	无
潜水情况	无	承压水	无
附加荷载类型	无	支挡类型	悬臂式
附加荷载大小	无	嵌固深度/m	5
桩内侧水深/m	无	桩顶到基坑底深度/m	13
桩外侧水深/m	无	桩顶到地面距离/m	0

表 2 土层信息  
Table 2 Soil information

序号	每层土深度 m	重度 kN·m <sup>-3</sup>	内聚力 kPa	内摩擦角 (°)
1	50	18.8	36.7	38.8

为了测试 HFOA 的性能,与遗传算法、粒子群优化、蚁群算法、人工鱼群算法、模拟退火算法、标准果蝇算法和枚举法进行了对比测试.各算法参数为对本算例进行测试后,安全系数达到最小且用时最短的组合作为本文对比算法参数,取值如下:

遗传算法:个体数为 50,采用精英保留策略,精英数为 10,突变概率为 0.4,交叉概率为 0.8,迭代次数 500;

粒子群优化算法:粒子数为 50,惯性因子为 0.5,学习因子为 2,迭代次数为 500;

蚁群算法:蚂蚁数为 50,信息素因子 0.8,区间缩小因子 0.8,迭代次数 200;

人工鱼群算法:鱼群数为 50,感知距离为 15,移动步长为 5,拥挤度因子为 10,迭代次数 200;

模拟退火算法:初始温度 1 000 °C,最大迭代次数为 500,调节降温系数为 0.998,重复降温次数 500;

标准果蝇算法:果蝇数为 50,迭代次数 500,飞行步长为 0.001,0.001,0.05;

混合果蝇算法:果蝇数为 50,迭代次数 500,飞行步长为 0.001,0.001,0.05,禁忌步长为 0.1,邻域范围 13,禁忌表长度 500;

枚举法:网格大小为 0.1,搜索区域范围:

$X \in [40, 100]$ ;  $Y \in [40, 100]$ ;  $R \in [10, 40]$ .

通过 Matlab 编程计算得到的各种算法的结

果如表 3 所示,为减小算法本身的随机性产生的误差,每种方法均运行 10 次,计算 10 次运行结果安全系数的平均值.以枚举法结果为真值,误差是由迭代结果和枚举法结果差值除以枚举法的结果得到的相对误差(下同),对其结果进行分析.

表 3 不同算法计算结果  
Table 3 Calculation results of various algorithms

算法	圆心坐标/m	半径/m	平均值	最大值	最小值	标准差	相对误差/%	绝对误差	单次计算时长/s
遗传算法	(53.332, 54.521)	22.774	3.165	3.202	3.101	$3.773 \times 10^{-3}$	2.329	0.072	18.8
粒子群算法	(52.908, 52.302)	20.051	3.145	3.266	3.081	$5.890 \times 10^{-3}$	1.681	0.052	7.9
蚁群算法	(53.791, 52.439)	20.788	3.117	3.186	3.092	$2.356 \times 10^{-3}$	0.776	0.024	679.5
鱼群算法	(53.701, 52.008)	20.395	3.124	3.144	3.089	$1.683 \times 10^{-3}$	1.002	0.031	125.6
模拟退火算法	(47.432, 57.014)	25.209	3.675	3.891	3.688	$1.459 \times 10^{-2}$	18.817	0.582	24.1
标准果蝇算法	(52.765, 51.508)	19.718	3.108	3.165	3.083	$3.609 \times 10^{-3}$	0.485	0.015	4.3
混合果蝇算法	(51.935, 50.694)	18.792	3.089	3.095	3.082	$6.771 \times 10^{-4}$	-0.129	-0.004	15.8
枚举法	(52.000, 50.900)	19.200	3.093	—	—	—	—	—	12359.2

由表 3 可知,其中粒子群算法、蚁群算法、人工鱼群算法、标准果蝇算法、混合果蝇算法计算得到的最小安全系数均接近枚举法得到的安全系数 3.093,说明以上几种算法均有可能搜索得到全局最优解;混合果蝇算法的平均安全系数为 3.089,与枚举法结果绝对误差为 -0.004,相对误差为 -0.129%,最接近枚举法计算得到的真实值,标准差为  $6.771 \times 10^{-4}$ ,低于其他几种算法,说明该算法稳定性最好,每次计算均可收敛到最优解附近;蚁群算法、人工鱼群算法,遗传算法和粒子群的平均安全系数也接近枚举法的结果,相对误差均在 2% 以内;标准果蝇算法计算时长为 4.3s,用时最少,其次是粒子群算法,用时 7.9s,混合果蝇算法计算时长为 15.8s,较标准果蝇算法计算时长略有增加,原因是通过禁忌思想增加了计算时长,提高了计算的精度和稳定性,但是计算速度仍然优于遗传算法、模拟退火算法、蚁群算法、鱼群算法.综合来看,混合果蝇算法计算的安全系数最小,稳定性最优,但计算时长略有增加,但远优于蚁群算法,人工鱼群算法和模拟退火算法.图 3 为各算法计算得到的临界滑动面示意图.

### 3.2 层状土桩锚支护基坑算例

以沈阳市某层状土桩锚支护基坑为例,各算法参数设置同均质土基坑,表 4~表 6 为工程实例信息.

通过表 7 不同算法的计算结果可以看出,除模拟退火算法外的各算法搜索得到最小安全系

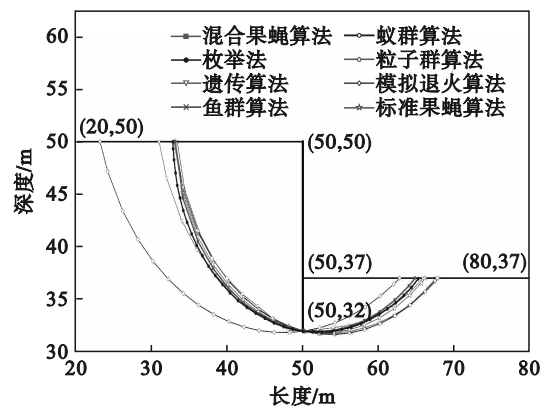


图 3 均质土层悬臂支护基坑剖面图及临界滑动面

Fig. 3 Section and critical sliding surface of cantilever supported pit with homogeneous soil layer

数均接近枚举法的安全系数 1.877,表明这几种算法有可能搜索到最优解,但是根据 10 次搜索的平均安全系数和误差的结果,除混合果蝇算法外,其他算法平均安全系数均偏离了最优解,粒子群算法和果蝇算法相对误差接近 2%,其他算法相对误差均在 2% 以上;混合果蝇算法的标准差最小,搜索得到的平均安全系数与枚举法基本一致,而其他算法标准差较大,比混合果蝇算法要大一个数量级,表明混合果蝇算法的稳定性要远优于其他算法.从计算时间来看,混合果蝇算法仅次于标准果蝇算法,用时 94.1s,计算效率优于其他各算法.综合来看,混合果蝇算法的适用性最好,粒子群算法和果蝇算法绝对误差和相对误差较小,计算速度较快,效果尚可,其他几种算

法效果较差.图4为各算法计算得到的临界滑动面示意图.

### 3.3 含软弱土层基坑算例

以某含软弱夹层的桩锚支护基坑为例,各算法参数设置同均质土基坑,总体信息同表4,支锚信息同表6,表8为土层信息.

由表9可知,遗传算法、粒子群算法、标准果蝇算法、混合果蝇算法的最小安全系数接近枚举法的结果1.285,表明这几种算法有可能搜索得到最优解,鱼群算法、蚁群算法和模拟退火算法

陷入局部最优,无法搜索到最优解;根据各算法的平均值和相对误差,混合果蝇算法最接近枚举法搜索得到的安全系数,绝对误差为0.005,相对误差为0.427%,而其他算法各均偏离了最优解,绝对误差较大且相对误差大于4%.结合标准差分析可知,遗传算法、粒子群算法、标准果蝇算法有概率搜索得到最优解但是非常不稳定,搜索效果较差;鱼群算法、蚁群算法、模拟退火算法很早就陷入局部最优,不仅稳定性较差,而且无法搜索到最优解,图5为各算法计算得到的临界滑动面

表4 总体信息  
Table 4 General information

土层数	6	支挡层数	2
潜水情况	静水	承压水	无
附加荷载类型	均布附加荷载	支挡类型	锚拉式
附加荷载大小/(kPa·m <sup>-1</sup> )	15	嵌固深度/m	5
桩内测水深/m	14	桩顶到基坑底深度/m	13
桩外侧水深/m	14	桩顶到地面距离/m	0

表5 层状土桩锚支护基坑土层信息  
Table 5 Soil information

序号	每层土深度 m	重度 kN·m <sup>-3</sup>	内聚力 kPa	内摩擦角 (°)	与锚固体摩擦阻力 kPa	水下黏聚力 kPa	水下内摩擦角 (°)	水下重度 kN·m <sup>-3</sup>
1	1.2	18.0	5.0	10.0	18.0	5.0	10.0	18.0
2	3.3	19.4	36.7	13.8	50.0	36.7	13.8	19.4
3	2.7	18.8	35.5	12.7	35.0	35.5	12.7	18.8
4	1.3	19.0	0	33.7	10.0	0	33.7	19.0
5	4.5	19.5	0	38.3	190.0	0	38.3	19.5
6	50.0	18.0	0	38.3	190.0	0	38.0	18.0

表6 支锚信息  
Table 6 Anchor information

支锚 序号	水平间距 m	竖向间距 m	入射角 (°)	总长 m	锚固段长度 m	预加力 kN	支锚刚度 mN·m <sup>-1</sup>	初始位移 m	锚固体直径 mm	材料抗力 kN
1	1.2	4.50	15.00	16.50	9.00	150.00	6.54	0.00	150.00	520.80
2	1.2	9.00	15.00	11.50	6.50	200.00	23.18	0.00	150.00	781.20

表7 层状土桩锚支护基坑算例不同算法计算结果  
Table 7 Calculation results of various algorithms

算法	圆心坐标/m	半径/m	平均值	最大值	最小值	标准差	相对误差/%	绝对误差	单次计算 时长/s
遗传算法	(53.181,50.848)	20.944	1.920	1.951	1.914	1.120×10 <sup>-2</sup>	2.280	0.043	208.3
粒子群算法	(54.003,51.015)	20.028	1.918	1.927	1.906	8.123×10 <sup>-3</sup>	1.854	0.041	114.2
蚁群算法	(55.156,54.401)	23.071	1.924	1.954	1.906	2.571×10 <sup>-2</sup>	2.493	0.047	861.2
人工鱼群算法	(52.779,50.588)	18.571	1.940	1.973	1.910	1.809×10 <sup>-2</sup>	3.345	0.063	231.3
模拟退火算法	(61.065,64.253)	34.503	2.094	2.257	1.982	1.154×10 <sup>-1</sup>	11.549	0.217	34.6
标准果蝇算法	(54.855,50.609)	19.138	1.926	1.962	1.885	2.24×10 <sup>-2</sup>	1.960	0.049	78.1
混合果蝇算法	(54.068,50.600)	19.051	1.875	1.897	1.872	6.506×10 <sup>-3</sup>	-0.117	-0.002	94.1
枚举法	(54.100,50.600)	19.000	1.877	—	—	—	—	—	14 052.4

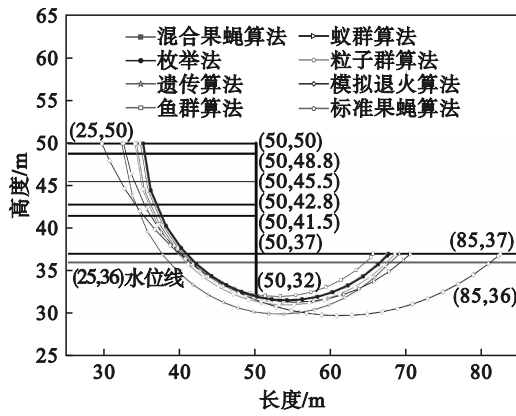


图 4 层状土桩锚支护基坑剖面图及临界滑动面

Fig. 4 Section and critical sliding surface of pile-anchor support pit with laminated soil

示意图.

图 6~图 7 为 3 种类型基坑的相对误差、绝对误差和标准差趋势图,可知鱼群算法、蚁群算法、标准果蝇算法和混合果蝇算法绝对误差在 0.03 左右、相对误差在 1% 以内,标准差较小,适用于均质土基坑. 对于桩锚支护的成层土基坑来说,除混合果蝇算法之外各算法计算结果的相对误差和标准差均有所增加,其中粒子群算法和果蝇优化算法的相对误差小于 2%,绝对误差在 0.04 左右,搜索能力一

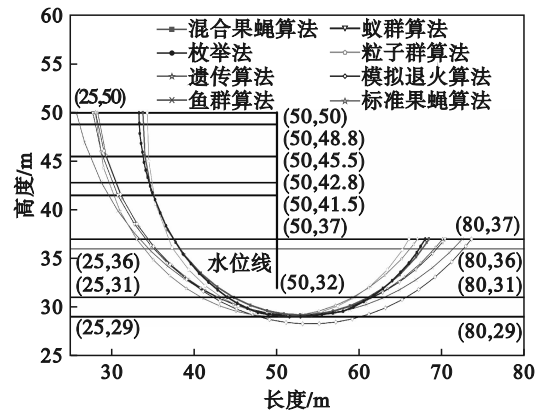


图 5 含软弱夹层桩锚支护基坑剖面图及临界滑动面

Fig. 5 Section and critical sliding surface of pile-anchor supported foundation pit with soft interlayer

般;遗传算法、鱼群算法、鱼群算法、模拟退火算法相对误差大于 2% 标准差较大,搜索能力较差且不稳定. 对于含软弱夹层的桩锚支护基坑来说,除混合果蝇算法之外各算法绝对误差均大于 0.05,相对误差明显增大,在 5% 以上,标准差显著上升,计算结果非常不稳定,常常陷入局部最优;而混合果蝇算法的相对误差和标准差接近 0,说明混合果蝇算法的计算精度更高,收敛可靠性更加稳定.

表 8 软弱土层信息  
Table 8 Soil information

序号	每层土深度 m	重度 kN·m <sup>-3</sup>	内聚力 kPa	内摩擦角 (°)	与锚固体摩擦阻力 kPa	水下黏聚力 kPa	水下内摩擦角 (°)	水下重度 kN·m <sup>-3</sup>
1	1.20	18.00	5.00	10.00	18.00	5.00	10.00	18.00
2	3.30	19.40	36.70	13.80	50.00	36.70	13.80	19.40
3	2.70	18.80	35.50	12.70	35.00	35.50	12.70	18.80
4	1.30	19.00	0	33.70	10.00	0	33.70	19.00
5	4.50	19.50	0	38.30	190.00	0	38.30	19.50
6	6.00	18.00	0	38.30	190.00	0	38.00	8.00
7	2.00	18.00	0	10.00	18.00	0	10.00	8.00
8	50.00	18.00	0	38.30	190.00	0	38.00	8.00

表 9 含软弱土层基坑算例不同算法计算结果  
Table 9 Calculation results of various algorithms

算法	圆心坐标/m	半径/m	平均值	最大值	最小值	标准差	相对误差/%	绝对误差	单次计算 时长/s
遗传算法	(52.181, 53.136)	29.023	1.361	1.482	1.287	6.146×10 <sup>-2</sup>	5.937	0.076	312.8
粒子群算法	(51.324, 46.403)	17.372	1.343	1.398	1.287	4.665×10 <sup>-2</sup>	4.502	0.058	156.2
蚁群算法	(52.199, 53.865)	24.733	1.371	1.435	1.312	2.053×10 <sup>-2</sup>	6.697	0.086	1035.2
人工鱼群算法	(52.881, 57.135)	28.023	1.387	1.434	1.320	3.245×10 <sup>-2</sup>	7.899	0.102	326.4
模拟退火算法	(53.928, 54.474)	26.220	1.673	1.959	1.434	1.555×10 <sup>-1</sup>	30.215	0.388	91.4
标准果蝇算法	(50.012, 50.960)	21.968	1.347	1.376	1.297	3.609×10 <sup>-2</sup>	4.855	0.062	78.2
混合果蝇算法	(52.900, 48.341)	19.167	1.290	1.293	1.288	1.591×10 <sup>-3</sup>	0.427	0.005	155.8
枚举法	(52.300, 48.000)	19.100	1.285	—	—	—	—	—	14945.1

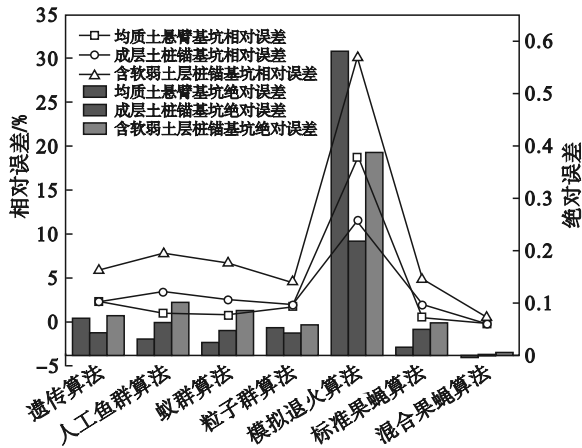


图 6 各算法在三种基坑中的相对误差和绝对误差  
Fig. 6 Errors and standard deviations of various algorithms in three types of pit

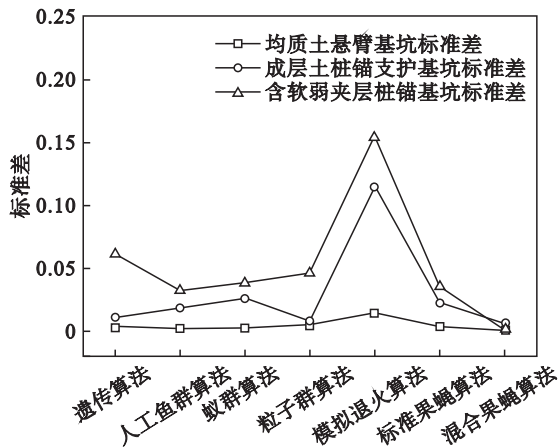


图 7 各算法在三种基坑中的标准差  
Fig. 7 Standard deviation of each algorithm in three types of pit

## 4 结 论

1) 本文将果蝇优化算法与禁忌算法结合,并引入自适应步长对解空间进行搜索,保留了果蝇算法收敛快速的优点,弥补了标准果蝇算法寻优精度不高的缺点,形成了自适应步长的混合果蝇算法.并与简化 Bishop 算法结合,通过对均质土悬臂支护基坑、成层土桩锚支护基坑和含软弱夹层的桩锚支护基坑三种类型基坑整体稳定安全系数的计算,证明本文算法可以快速准确地搜索到临界滑动面,对以上三种类型基坑整体稳定临界滑面搜索有较好的适用性.

2) 多数智能算法对于均质土悬臂支护基坑的稳定性计算有较好的适用性,果蝇优化算法、混合果蝇算法误差小、效率高,鱼群算法、蚁群算法误差处于中等水平,但是计算效率较低;遗传算法、粒子群算法误差较大,但是计算效率高;模

拟退火算法由于受限于初始滑面位置的影响,不适用于直接进行最危险滑面搜索.对于土质分布不均匀的成层土桩锚支护基坑最佳选择是混合果蝇算法,其次也可选择粒子群算法和果蝇算法,计算精度一般,但是计算效率较高;鱼群算法、蚁群算法和模拟退火算法不适用于成层土桩锚支护基坑整体稳定计算,计算效率低,误差大;对于含软弱夹层的桩锚支护基坑,混合果蝇算法可以很好解决,其他算法由于计算误差较大,稳定性差、计算效率低,不宜选用.

3) 混合果蝇算法较标准果蝇算法虽然计算时间略有增加,但是局部搜索能力有显著提高,且搜索稳定性更优,收敛精度更高,寻优结果更精确,为桩锚支护基坑临界滑动面搜索提供了新的选择.

## 参考文献:

[ 1 ] Li Z, Zhao G F, Deng X F, et al. Further development of distinct lattice spring model for stability and collapse analysis of deep foundation pit excavation [J]. *Computers and Geotechnics*, 2022, 144: 104619.

[ 2 ] 肖专文,张奇志,梁力,等.遗传进化算法在边坡稳定性分析中的应用[J].*岩土工程学报*, 1998, 20(1): 44-46. (Xiao Zhuan-wen, Zhang Qi-zhi, Liang Li, et al. Application of genetic evolutionary algorithm for slope stability analysis [J]. *Chinese Journal of Geotechnical Engineering*, 1998, 20 (1): 44-46.)

[ 3 ] Liu B W, Wang Z W, Zhong X Y. Particle swarm optimization algorithm in numerical simulation of saturated rock slope slip [J]. *Mathematical Problems in Engineering*, 2021, 2021: 6682659.

[ 4 ] Gao W. Premium - penalty ant colony optimization and its application in slope stability analysis [J]. *Applied Soft Computing*, 2016, 43: 480-488.

[ 5 ] Zhu J F, Chen C F. Search for circular and noncircular critical slip surfaces in slope stability analysis by hybrid genetic algorithm [J]. *Journal of Central South University*, 2014, 21: 387-397.

[ 6 ] 胡军,王凯凯,黄贵臣.基于自繁衍鱼群算法的土质边坡滑动面搜索[J].*金属矿山*, 2016(2): 132-136. (Hu Jun, Wang Kai-kai, Huang Gui-chen. Searching for the slip surface of soil slopes based on the breeding fish algorithm [J]. *Metal Mine*, 2016(2): 132-136.)

[ 7 ] 李守巨,刘迎曦,何翔,等.基于模拟退火算法的边坡最小安全系数全局搜索方法[J].*岩石力学与工程学报*, 2003, 22(2): 236-240. (Li Shou-ju, Liu Ying-xi, He Xiang, et al. Global search algorithm of minimum safety factor for slope stability analysis based on annealing simulation [J]. *Chinese Journal of Rock Mechanics and Engineering*, 2003, 22 (2) : 236-240.)

- [ 8 ] 潘文超.应用果蝇优化算法优化广义回归神经网络进行企业经营绩效评估[J].太原理工大学学报(社会科学版),2011,29(4):1-5.  
(Pan Wen - chao. Using fruit fly optimization algorithm optimized general regression neural network to construct the operating performance of enterprises model [J]. *Journal of Taiyuan University of Technology (Social Sciences Edition)*, 2011, 29(4): 1-5.)
- [ 9 ] 吴小文,李擎.果蝇算法和5种群智能算法的寻优性能研究[J].火力与指挥控制,2013,38(4):17-20,25.  
(Wu Xiao - wen, Li Qing. Research of optimizing performance of fruit fly optimization algorithm and five kinds of intelligent algorithm [J]. *Fire Control & Command Control*, 2013, 38(4): 17-20, 25.)
- [10] Bishop A W. The use of the slip circle in the stability analysis of slopes[J]. *Géotechnique*, 1955, 5(1): 7-17.
- [11] 王笙屹.土钉墙内部稳定性分析优化算法研究[D].赣州:江西理工大学,2013.  
(Wang Sheng-yi. Study on optimization algorithm of internal stability analysis of soil nailing wall [D]. Ganzhou: Jiangxi University of Science and Technology, 2013.)
- [12] Xiong C B, Lian S D. Structural damage identification based on improved fruit fly optimization algorithm [J]. *KSCE Journal of Civil Engineering*, 2021, 25(3): 1-23.
- [13] 张彩宏,潘广贞.融合禁忌搜索的混合果蝇优化算法[J].计算机工程与设计,2016,37(4):907-913.  
(Zhang Cai - hong, Pan Guang - zhen. Fruit fly optimization algorithm based on Tabu search [J]. *Computer Engineering and Design*, 2016, 37(4): 907-913.)
- [14] Bian H L. Study on chaotic time series prediction algorithm for kent mapping based on particle swarm optimization [J]. *Applied Mechanics and Materials*, 2014, 511/512: 941-944.
- [15] 刘建军,石定元,武国宁.基于Kent映射的混合混沌优化算法[J].计算机工程与设计,2015,36(6):1498-1503.  
(Liu Jian-jun, Shi Ding-yuan, Wu Guo-ning. Hybrid chaotic optimization algorithm based on Kent map [J]. *Computer Engineering and Design*, 2015, 36(6): 1498-1503.)
- [16] Demidova L A, Gorchakov A V. Application of chaotic fish school search optimization algorithm with exponential step decay in neural network loss function optimization [J]. *Procedia Computer Science*, 2021, 186: 352-359.