

# 萤火虫算法 在快递运输网点选址中的应用

班级



讲稿人



时间

10月26日

# 目录

CONTENTS

- 一. 萤火虫算法简介
- 二. 求解问题及数学模型
- 三. 实现思路
- 四. 代码实现
- 五. 改进与参数调优

# 01

## 萤火虫算法介绍





# 萤火虫算法介绍

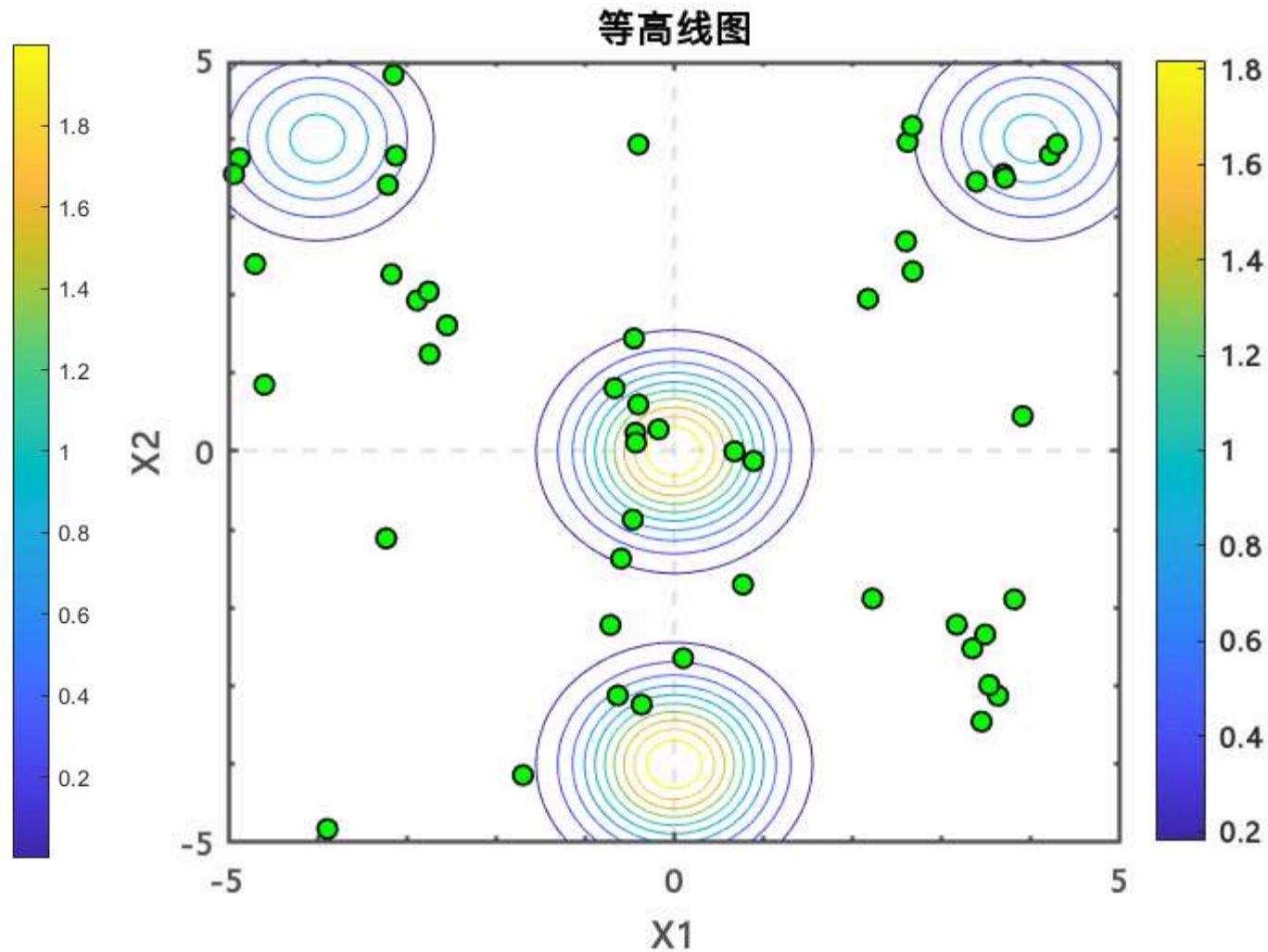
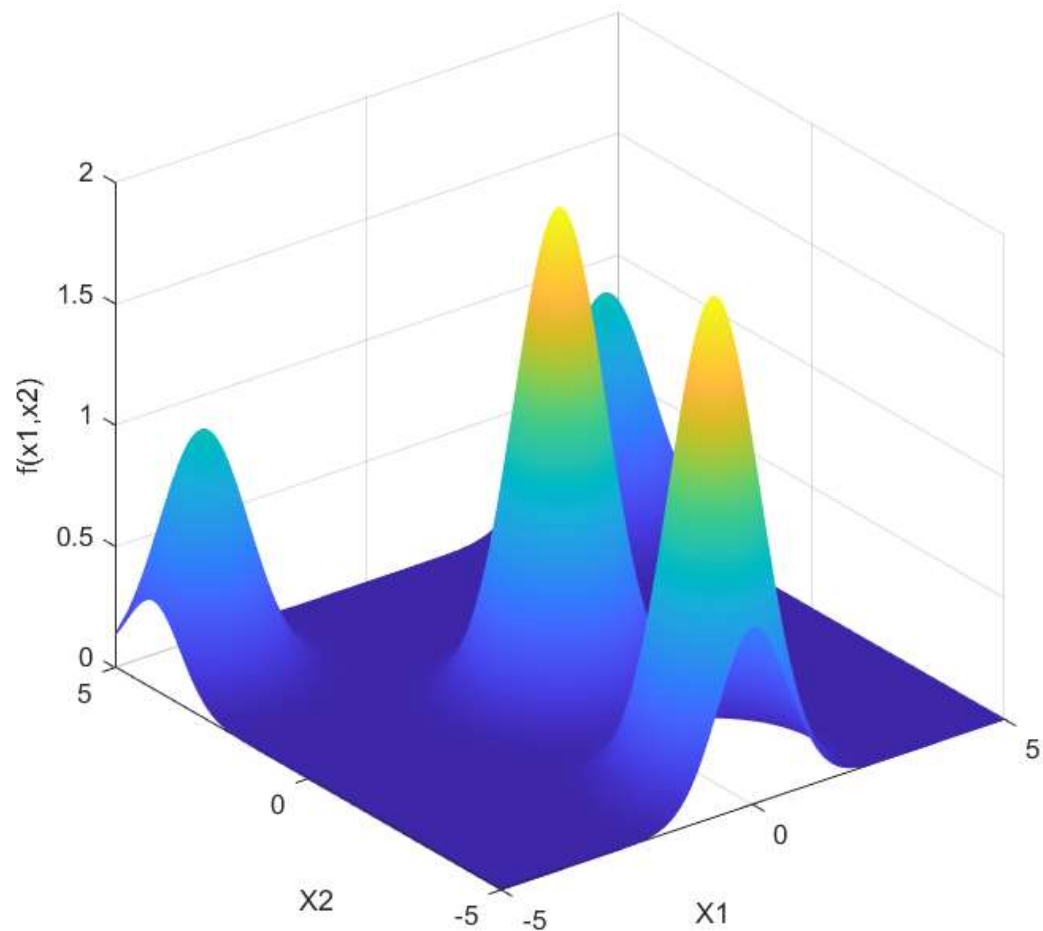


萤火虫算法(Firefly Algorithm, FA)是一种群智能算法领域的流行算法。萤火虫算法的灵感来自每一个萤火虫会通过尾部闪烁的光线相互发出信号来吸引伴侣或警告潜在的捕食者。

剑桥大学的Yang X S.教授在2008年提出了萤火虫算法，在萤火虫算法中每一个萤火虫个体都是一个解，萤火虫尾部的发光亮度由萤火虫所处位置的目标函数值决定，种群中一个萤火虫会受到另一个比其更加亮的萤火虫吸引进行移动，随着个体不断移动，种群不断的进化，最终获取优化问题的解。



# 萤火虫算法介绍

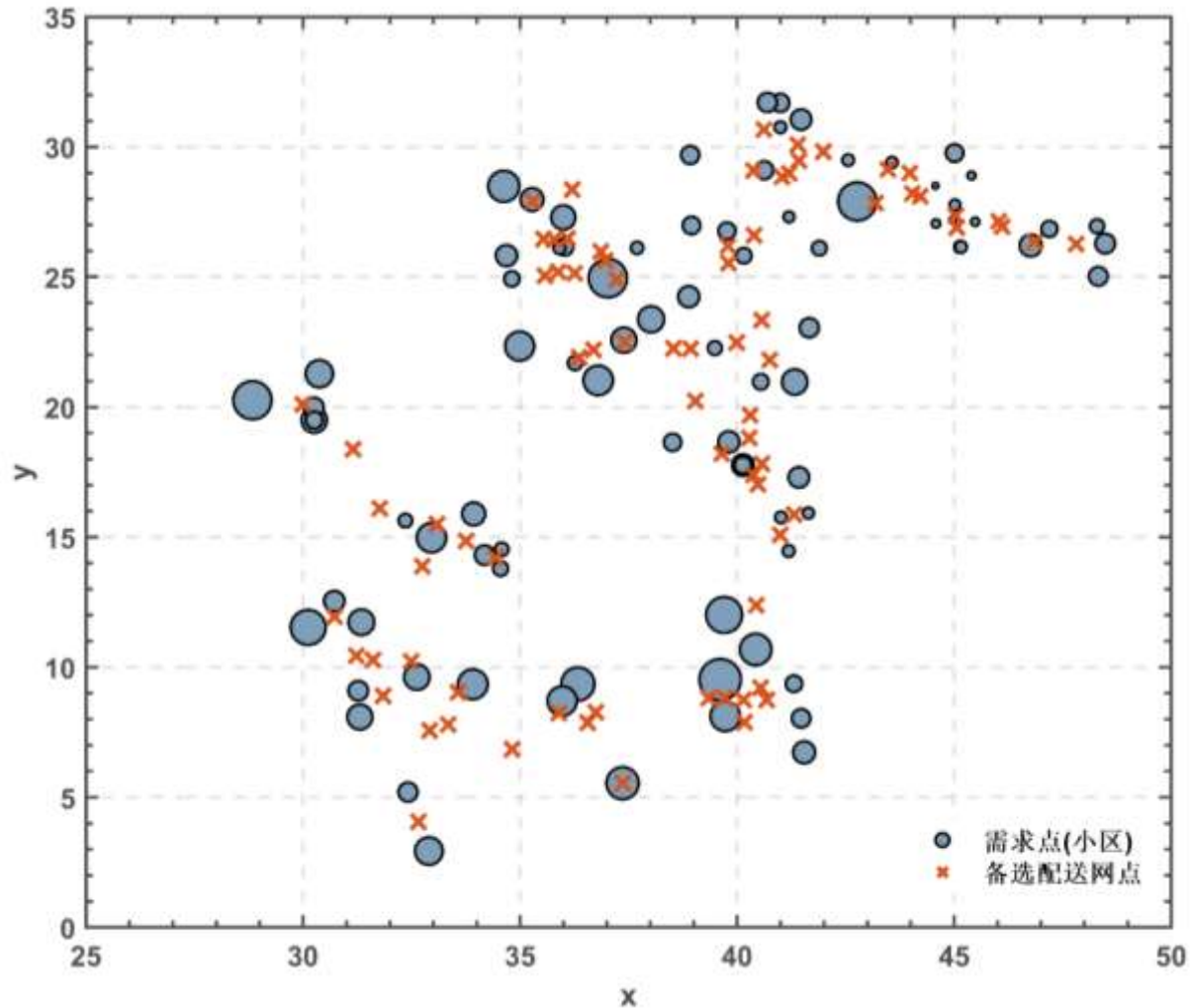


# 02

## 求解问题及数学模型

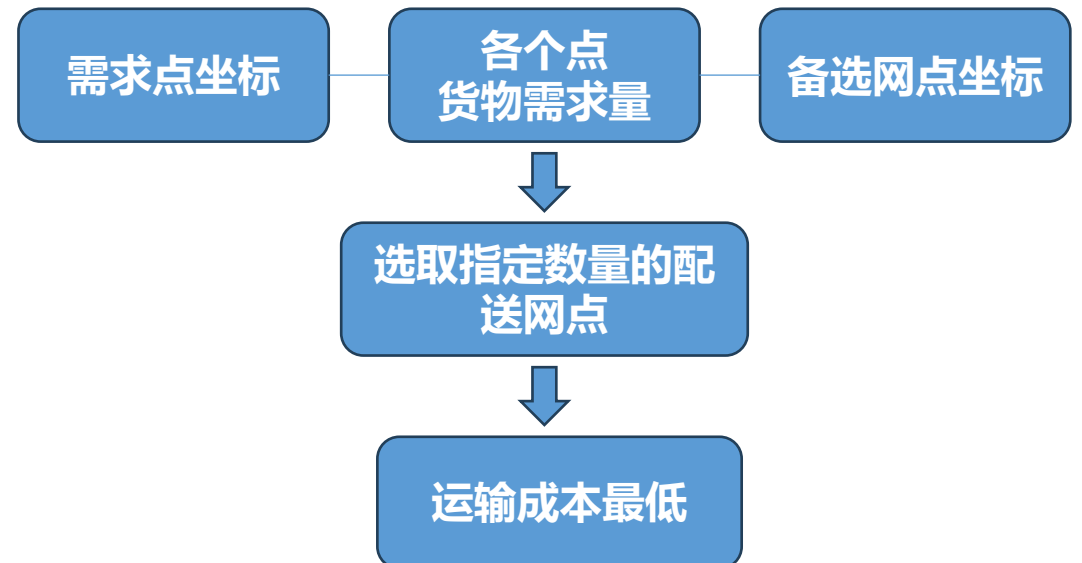


## 二、求解问题



快递网点选址问题，也可以称为配送中心选址规划问题，问题描述如下：

给定所有需求点坐标、需求点的货物需求量、备选网点坐标，需要从备选的网点选取指定个数的物流网点使得在满足供应各个位置所需的要求下，总成本最低。



$$\min F = \sum_{i \in I} \sum_{j \in M_i} h_i d_{ij} z_{ij}$$

需求量与距离  
乘积和最小

目标函数  $\min F$  表示被选中的快递网点  $j$  到由它配送的小区  $i$  的需求量与距离乘积之和的最小值;

$$\text{s.t.} \quad \sum_{j \in M_i} z_{ij} = 1, i \in I$$

表明每个小区仅可被  
一个快递网点配送

$$z_{ij} \leq e_j, i \leq I, j \leq M$$

$$\sum_{j \in M_i} e_j = r$$

表明只能选中  $r$  个  
物流网点

$$z_{ij}, e_j \in \{0, 1\}, i \in I, j \in M$$

符号	含义
$h_i$	小区 $i$ 的需求量
$d_{ij}$	小区 $i$ 到距离其最近快递网点 $j$ 的距离
$z_{ij}$	0-1变量, 标识小区 $i$ 是否被快递网点 $j$ 配送
$e_j$	0-1变量, 标识备选快递网点 $j$ 是否被选中
$r$	被选中的快递网点数量
$I$	小区编号集合
$M$	备选快递网点集合



# 03

## 实现思路



## 三、实现思路

编码方案

基本萤火虫算法是一种连续型的优化算法，而快递运输网点选址是一种典型的离散型优化问题（如：选中编号为3、2、11、6的备选网点，而非选中编号为3.69、2.08、11.65...的备选网点）

为了适应求解问题要求，需要离散型的编码方式。我这里将连续型的编码通过四舍五入的方法转换为整数。每一个萤火虫个体 $x_i$ 是一个解决方案， $x$ 的维度为物流网点数量，每一维的值是一个在备选物流网点范围内整数

举例，假设有10个需求点，5个备选物流网点，从5个备选物流网点选取4个作为物流网点，按上述的编码方案表示如下：

3.64	5.68	2.3	1.9
------	------	-----	-----

连续变量解

四舍五入



4	6	2	11
---	---	---	----

处理后的离散变量解

重复编号随机替换为不重复的值

## 吸引度过小

[illegible]

## 三、实现思路

添加最小吸引力

在基本萤火虫算法中  
萤火虫与萤火虫之间的距离定义为：

$$r_{ij} = \|x_j - x_i\| = \sqrt{\sum_{k=1}^d (x_{j,k} - x_{i,k})^2}$$

萤火虫受到更亮的萤火虫的吸引度定义为：

$$\beta_{ij} = \beta_0 \times \exp(-\gamma r_{ij}^2)$$

萤火虫受到更亮的萤火虫的移动定义为：

$$x_i(t+1) = x_i(t) + \beta_{ij} \times (x_j(t) - x_i(t)) + \alpha \times (rand - 0.5)$$

当前萤火虫个体的位置受到上一代萤火虫位置的影响

表现萤火虫之间的影响是一种群体信息的共享

随机移动有利于萤火虫个体跳出局部极值

在求解离散型问题时，基本萤火虫算法会失效，因为此时萤火虫之间的距离非常大，导致萤火虫受更亮萤火虫的吸引度非常低，进而导致萤火虫算法的移动机制失效，萤火虫随机移动。

◎ 添加最小吸引力机制

$$\beta_{\min} = \beta_{\min^*} \times e^{\left(-\frac{t^2}{p \times \text{MaxG}}\right)}$$

$$\beta_{ij} = (\beta_0 - \beta_{\min}) \times e^{-\gamma r_{ij}^2} + \beta_{\min}$$

# 04

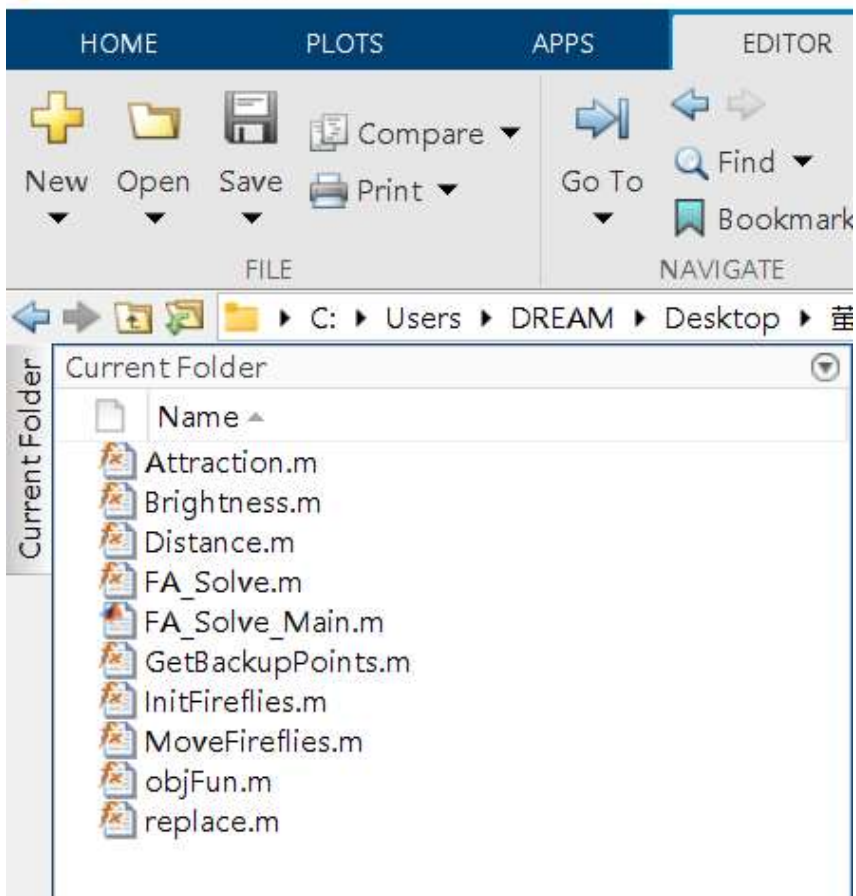
## 代码实现



## 四、代码实现



MATLAB R2023a



博学笃行 与时俱进

```
function x_t1 = MoveFireflies(x_t,I,beta,alpha,m,lb,ub)
```

```
% 萤火虫移动
```

```
% ===== %
```

```
x_t1 = x_t;
```

```
for i=1:length(x_t)
```

```
    for j=1:length(x_t)
```

```
        % 如果个体j比i更加亮, 则i被j吸引
```

```
        if I(j) > I(i)
```

```
            r = rand(1,length(x_t1(i,:)));
```

```
            x_t1(i,:) = x_t1(i,:) + ...
```

```
                m.*r.*(x_t(end,:)-x_t(i,:)) + ...
```

```
                beta(i,j).*(x_t(j,:) - x_t1(i,:)) + ...
```

```
                alpha(i).*(r-0.5);
```

```
        end
```

```
    end
```

```
end
```

```
% 进行取整操作
```

```
x_t1 = round(x_t1);
```

```
% 相同的备选点替换
```

```
for i = 1:size(x_t1,1)
```

```
    x_t1(i,:) = replace(x_t1(i,:),lb,ub);
```

```
end
```

```
% 应用边界缓冲域
```

```
for i = 1:size(x_t1,2)
```

```
    x_t1(x_t1(:, i) < lb, i) = lb; % 将小于下限的值设为下限
```

```
    x_t1(x_t1(:, i) > ub, i) = ub; % 将大于上限的值设为上限
```

```
end
```

```
end
```

# 05

## 改进与参数调优



## 五、算法改进



### 基于寻优偏差度的自适应随机步长机制

- $\partial$ 取值较大时在全局有较好的寻优效果，但在算法迭代的后期不易收敛，
- $\partial$ 取值较小时有较好的收敛性但是容易陷入局部极值中

$$\alpha = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \times \frac{\|x_i - x_{best}\|}{L_{\max}} \times Grand$$

$$L_{\max} = \|x_{worst} - x_{best}\|$$

### 全局导向性移动机制

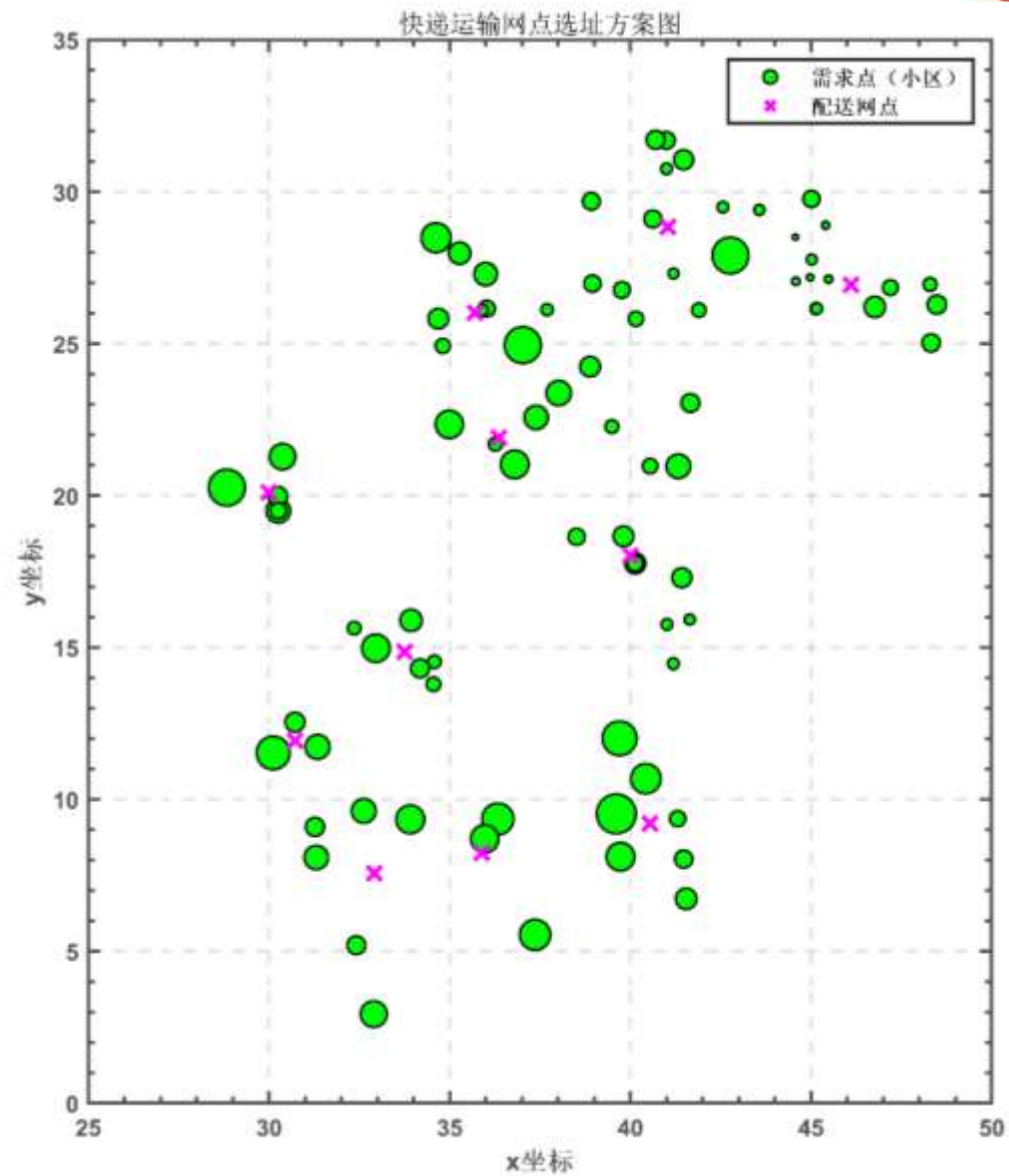
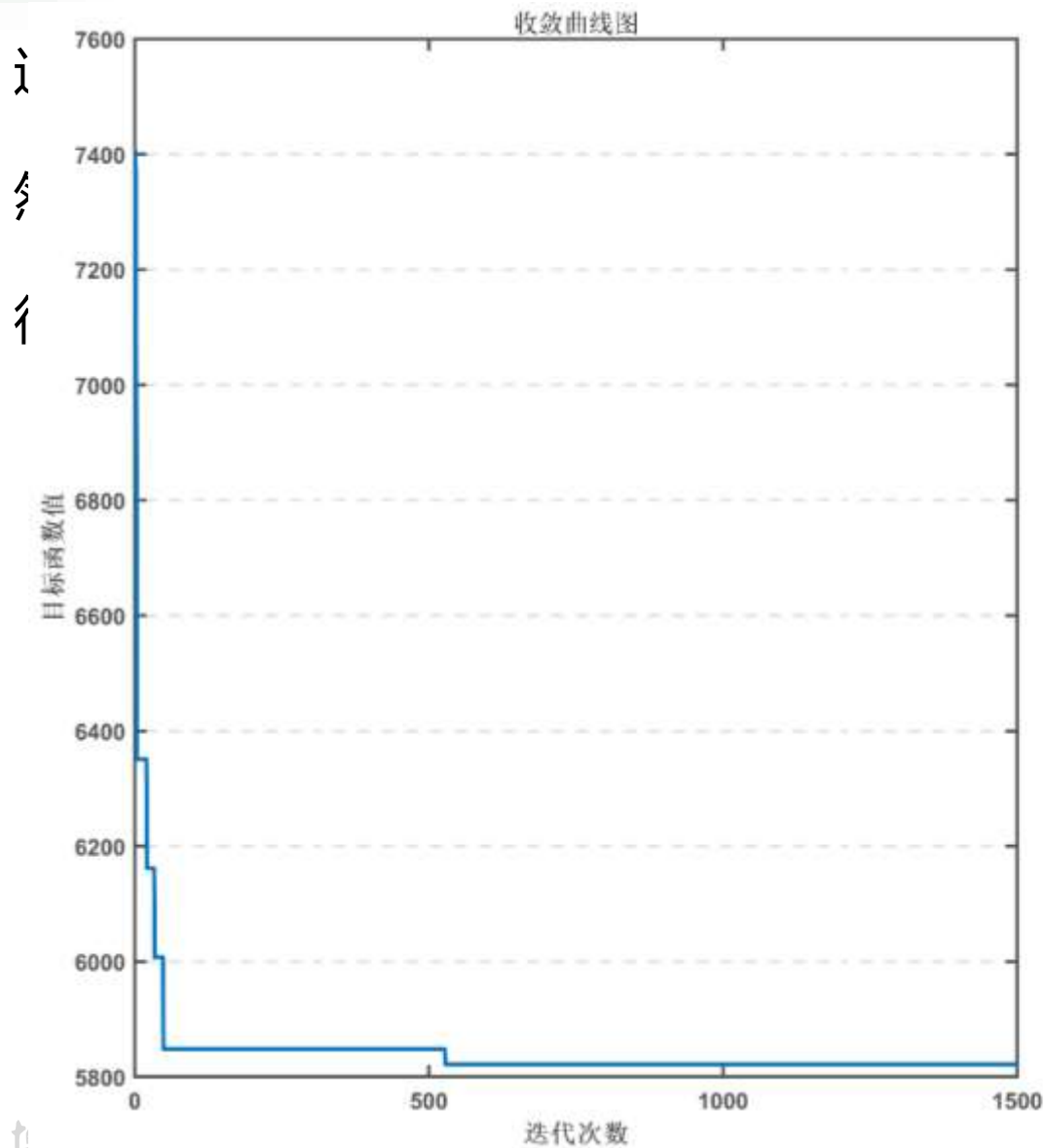
基本萤火虫算法中萤火虫个体向可视范围内比自身更亮的个体移动，从而使整个萤火虫群体向着最优解进化。此时萤火虫寻优只受自身个体周围离自己较近的更亮萤火虫影响，忽略了全局最优值，容易陷入局部极值。为提高算法的寻优性能，改进萤火虫个体位置的移动机制，加入了全局影响策略。

$$x_i(t+1) = x_i(t) + m \times rand \times (x_{best} - x_i(t)) + \beta_{ij} \times (x_j(t) - x_i(t)) + \alpha \times (rand - 0.5)$$

$$m = m_{\max} - \frac{(m_{\max} - m_{\min})}{sum(1:t)} \times \log_{MaxG} t^2$$



# 五、参数调优



值

学

The background features a large, light gray watermark of the Guangzhou University logo, which is a circular emblem containing a stylized figure and the university's name in English.

# 感谢观看