

基于贪心算法和变步长搜索的烟幕干扰弹投放策略优化

摘要

以无人机为载体的烟幕干扰弹投放策略设计是加强战场光电对抗水平,推进国防和军队建设的关键。本文从优化角度出发,建立无人机投放策略有效遮蔽时长优化模型,并通过变步长遍历法、贪心算法等方法对多种条件下的优化方案进行求解。为军事部门提供了不同战略条件下烟幕干扰弹投放策略的参考方案。

针对问题一,首先以假目标为原点构筑坐标系,得到无人机、导弹、烟幕干扰弹、烟幕云团的轨迹方程。从光路轨迹模拟及理论分析出发,将烟幕有效遮蔽分为两种情况进行分析:针对导弹位于烟幕云团内部,通过两者距离小于云团半径进行判定。针对烟幕云团位于真目标和导弹之间,在真目标上下底面均匀取多个特征点,通过分析特征点投影是否落于云团投影范围内对遮蔽进行判定。基于上述烟幕有效遮蔽计算方法,带入题目给定参数,对有效遮蔽时长进行求取。最终得到在原问题给定条件下,有效遮蔽时长为1.3926s。

针对问题二,根据轨迹方程可以确定决定投放策略的主要参数间存在等价关系,由于策略的核心就是控制烟幕云团,故将起爆时间与起爆位置设置为决策变量因素,以最大化延长有效遮蔽时长为目标函数,设置起爆时间、起爆范围、起爆高度等条件约束,建立了单个烟幕干扰弹对单个导弹的有效遮蔽时长优化模型。为减少计算量,采用变步长搜索对最优烟幕干扰策略进行求解。解得在无人机 FY1 保持速度1.9339m/s,方位角7.125°,在0s投掷,0.8s引爆烟幕干扰弹的情况下,达到最长有效遮蔽时长4.5976s。

针对问题三,首先考虑各枚烟幕干扰弹间可能存在的遮蔽区间重叠情况,对有效遮蔽总时长进行定义。在初步分析中,将面向无人机运动方向、运动速度的研究简化至对起爆时间的考量。分析不同起爆时间下对应的最佳遮蔽区间,评估无人机主动迎向导弹、面向真目标点飞行两种投放策略。综合上述分析,在前一问模型基础上增添投放时间间隔等约束,建立多枚烟幕干扰弹的有效遮蔽时长优化模型。采取贪心策略,以0.1为步长依次分析三枚烟幕干扰弹的最优投放策略,对模型进行求解。最终求得当无人机 FY1 以139.7308m/s,方向179.7784°飞行,分别在0s,2.4860s,4.9s投放,0s,7.4s,10.7s引爆烟幕干扰弹,可达成最大有效遮蔽总时长7.564s。

针对问题四,考虑到多架无人机的运动相互独立,该模型可拆分为多个问题二所建立模型进行求解。将独立求解结果按照遮蔽区间进行匹配。遍历所有组合方式进行区间匹配,得到最长的有效遮蔽时长。求得最优投放策略的总有效遮蔽时长为11.833s,其余参数详见表9及result2.xlsx。

针对问题五,由于变量较多,模型复杂度较高,出于简化模型考虑,提前对各枚烟幕干扰弹的投放目标进行指定。在实际分析基础上,确定无人机 FY1 的最优投放策略为只针对导弹 M1 进行干扰,其余无人机依据距离由近到远依次对三枚导弹各投放一枚烟幕干扰弹。综合上述分析构建**多枚烟幕干扰弹遮蔽多枚导弹的有效遮蔽时长优化模型**,采取贪心策略以及区间匹配方法对最优投放策略进行求取。得到的最优投放策略的总有效遮蔽时长42.466s,其具体参数详见表11及附录2。

关键词: 烟幕干扰投放策略; 变步长搜索法; 贪心算法

一、 问题重述

1.1 问题背景

烟幕是一种目前应用较为广泛的无源干扰手段，在对抗各类光电侦察和光电制导装置方面发挥着重要作用。其原理是通过化学燃烧或爆炸分散形成气溶胶云团，干扰光辐射的正常传输和光电系统的目标探测。随着相关技术的不断发展，以无人机为载体完成烟幕干扰弹的定点精确抛撒已成为现实。

为延长对保护目标的有效遮蔽时间，实现更为有效的烟幕干扰效果，需研究无人机飞行方向、飞行速度、烟幕干扰弹投放点、烟幕干扰弹起爆点等相关数据，设计烟幕干扰弹的投放策略。

1.2 问题要求

基于上述背景，要求建立数学模型解决下列问题：

假设真实目标为一半径 $7m$ 、高 $10m$ 的圆柱形。来袭的空地导弹有 3 枚，飞行速度均为 $300m/s$ 。飞行方向直指一个为掩护真目标而专门设置的假目标。侦测到导弹同时 5 架无人机受领任务，调整飞行方向，以 $70\sim 140m/s$ 的速度等高度匀速直线飞行。通过投放烟幕干扰弹尽量避免来袭导弹发现真目标。要求每架无人机投放两枚烟幕干扰弹至少需间隔 $1s$ 。烟幕干扰弹起爆后瞬时形成球状烟幕云团，由于采用特定技术，在特殊技术下以 $3m/s$ 的速度匀速下沉。该云团能维持 $20m/s$ 的有效遮蔽时间。无人机初始位置、导弹初始位置、目标位置均已给定。

问题一：利用无人机FY1投放 1 枚烟幕干扰弹实施对单个导弹M1的干扰，假设FY1以的 $120m/s$ 速度朝向假目标方向飞行，受领任务 $1.5s$ 后即投放 1 枚烟幕干扰弹，间隔 $3.6s$ 后起爆。求解该烟幕干扰弹对M1的有效遮蔽时长。

问题二：在问题一的基础上，不对无人机FY1初速、投放时间、起爆时间进行假定，确定FY1的最佳飞行、投放策略，使得遮蔽时间尽可能长。

问题三：在问题二的基础上，考虑利用无人机FY1投放 3 枚烟幕干扰弹，对单个导弹M1实施干扰，并给出最优的投放策略。

问题四：在问题二的基础上，考虑利用FY1、FY2、FY3三架无人机，各投放 1 枚烟幕干扰弹，对单个导弹M1实施干扰，给出烟幕干扰弹的投放策略。

问题五：针对来袭的M1、M2、M3三枚导弹。 5 架无人机全部调用，每架无人机至多投放 3 枚烟幕干扰弹，试求解最优的烟幕干扰弹投放策略。

二、 问题分析

2.1 问题一的分析

按照题目要求，以假目标为原点建立坐标系，设定各导弹、无人机及真目标位置坐标，确定真目标底部、假目标、云团中心指向导弹的方向向量。在此基础上对无人机、导弹、烟幕干扰弹、烟幕云团的轨迹方程进行构建。将烟幕云团有效遮蔽分为两种情况分别进行考虑：导弹位于云团内，或是云团位于真目标和导弹之间，其投影完全挡住真目标。针对第二种情况，为简便运算，在真目标顶面与底面圆周上均匀取特征点，如若在投影平面上特征点投影被云团投影完全覆盖，

则认为达成有效遮挡。基于上述烟幕有效遮蔽的计算方法，以时间为参考量进行遍历搜索。

2.2 问题二的分析

问题二需要求取单枚烟幕干扰弹的最优投放策略。根据轨迹方程，决定投放策略的主要参数：无人机飞行速度、飞行方向、烟幕干扰弹投放时间、起爆时间和起爆位置之间存在等价关系。将起爆时间与起爆位置作为主要决策变量，建立单个烟幕干扰弹对单个导弹的有效遮蔽时长优化模型。考虑到参数在空间中离散程度较高，直接高精度遍历会导致计算量过大，采用变步长搜索法对模型进行求解。

2.3 问题三的分析

问题三在问题二的基础上提高了要求，需要求取单架无人机投放三枚烟幕干扰弹的最优投放策略。考虑到不同烟幕干扰弹的遮蔽区间可能存在重叠，首先对所有烟幕干扰弹的有效遮蔽总时长进行定义。通过散点图连续性的分析，将面向无人机运动方向、运动速度的研究简化至对爆炸时间的考量。分析不同起爆时间下对应的最佳遮蔽区间，评估无人机主动迎向导弹、面向真目标点飞行两种投放策略。出于尽可能扩大有效遮蔽总时长的目的，决定侧重分析无人机朝向真目标点飞行的情况。在上述分析的基础上建立多枚烟幕干扰弹的有效遮蔽时长优化模型，并采取贪心策略依次分析单枚、两枚、三枚烟幕干扰弹的投放策略，对模型进行求解。

2.4 问题四的分析

问题四同样是问题二的延伸，考虑三架无人机各投放一枚烟幕干扰弹的最优投放策略。由于每一架无人机的运动轨迹都独立定向，可将模型拆解为多个问题二模型分别独立求解，将各求解结果按照遮蔽区间进行匹配，遍历所有可能的组合方式，求取最长的有效遮蔽总时长。

2.5 问题五的分析

问题五是上述问题的组合。探索五架无人机投放共计 15 枚烟幕干扰弹，对三枚导弹进行遮蔽的最优投放策略。出于简化模型考虑，在合理的前提下事先对各枚烟幕干扰弹的投放目标进行指定，其中无人机 FY1 只针对导弹 M1 进行投放，其余无人机由近到远依次对三枚导弹各投放一枚烟幕干扰弹。在问题二模型基础上构建最优投放策略，结合问题三、四模型，采取贪心策略和区间匹配方法对模型进行优化。求取具有最长有效遮蔽总时长的投放策略。

三、 模型假设

1. 假设烟幕干扰弹及其起爆后形成的烟幕云团在下落过程中不受空气阻力或是空气流动的影响；
2. 假设烟幕云团可对导弹视野产生完全遮蔽，并忽略其遮蔽效果随时间逐渐衰减的可能，并忽略导弹穿越烟幕云团时对其形状的破坏；
3. 假设导弹与假目标均可视为质点，但真目标不可视为一个质点；
4. 假设重力加速度为常数 $g = 9.8\text{m/s}^2$ ；

5. 忽略无人机在飞行与投放过程中被导弹直接命中坠毁的可能性。

四、 符号说明

表 1 符号说明

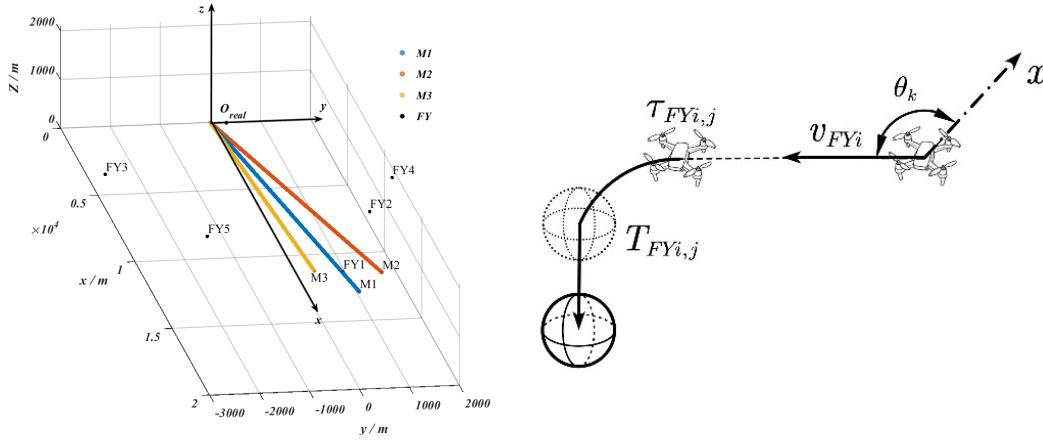
符号	说明	单位
O_{fake}	假目标位置	/
O_{real}	真目标位置	/
i, j, k	无人机、烟幕干扰弹、导弹编号	/
t	无人机受领任务后的时间	s
$X_{Mk}(t), Y_{Mk}(t), Z_{Mk}(t)$	第 k 枚导弹在 t 时刻位置	m
$X_{FY}(t), Y_{FY}(t), Z_{FY}(t)$	无人机在 t 时刻位置	m
$x_{FYi,j}(t), y_{FYi,j}(t), z_{FYi,j}(t)$	烟幕干扰弹在 t 时刻位置	m
$\tau_{FYi,j}$	烟幕干扰弹的投放时间	s
$T_{FYi,j}$	烟幕干扰弹的起爆时间	s
$v_{FYi,j}$	无人机飞行速度	m/s
$\theta_{FYi,j}$	无人机飞行方向	°
g	重力加速度	m/s ²
v_m	导弹飞行速度	m/s
v_s	烟幕云团下降速度	m/s
R_{cloud}	烟幕云团半径	m/s
λ	特征点选取数量	/
t_{start}	开始有效遮蔽时刻	s
t_{end}	结束有效遮蔽时刻	s
t_{shadow}	有效遮蔽时长	s

五、模型建立与求解

5.1 问题一的模型建立与求解

5.1.1 坐标系的建立

为更好的模拟无人机装载烟幕干扰的投放策略，根据题目要求，以假目标位置 O_{fake} 为原点，水平面为 xy 平面，建立坐标系 xyz 。真目标下底面圆心 O_{real} 坐标为 $(0, 200, 0)$ 。初始时刻，3枚导弹分别位于 $M1(20000, 0, 2000)$ 、 $M2(19000, 600, 2100)$ 、 $M3(18000, -600, 1900)$ ，5架无人机分别位于 $FY1(17800, 0, 1800)$ 、 $FY2(12000, 1400, 1400)$ 、 $FY3(6000, -3000, 700)$ 、 $FY4(11000, 2000, 1800)$ 、 $FY5(13000, -2000, 1300)$ 。



在坐标系中，对各模拟对象间的相对位置关系进行研究。针对任意来袭导弹 Mk 。记假目标指向导弹的方向向量为 $e_{Mk,fake}$ ，有：

$$e_{Mk,fake} = \begin{bmatrix} X_{Mk}(t) \\ Y_{Mk}(t) \\ Z_{Mk}(t) \end{bmatrix} - \begin{bmatrix} X_{fake} \\ Y_{fake} \\ Z_{fake} \end{bmatrix} \quad (1)$$

将真目标底部圆心指向导弹的方向向量记为 $e_{Mk,real}$ ，则有：

$$e_{Mk,real} = \begin{bmatrix} X_{Mk}(t) \\ Y_{Mk}(t) \\ Z_{Mk}(t) \end{bmatrix} - \begin{bmatrix} X_{real} \\ Y_{real} \\ Z_{real} \end{bmatrix} \quad (2)$$

以烟幕弹投放时间与烟幕弹爆炸时间为节点，烟幕干扰投掷系统的运作流程可分为三个阶段：首先，无人机装载烟幕弹飞行，抵达预计投放位置，该过程中无人机始终保持等高度的匀速直线运动。其次，烟幕弹在无人机提供的初始速度以及重力作用下，沿无人机原定飞行方向做平抛运动。最后，烟幕干扰弹在空中爆开，形成云团以 $3m/s$ 的速度匀速沉降，在云团范围内提供 $20s$ 的有效遮蔽。该流程具体示意如图2所示。

针对第 i 架无人机投放的第 j 个烟幕干扰弹，假设其投放时间和起爆时间分别为 $\tau_{FYi,j}$ 和 $T_{FYi,j}$ 。当 $t < \tau_{FYi,j}$ 时，无人机以 v_{FYi} 的速度做等高度匀速飞行，其飞行方向与 x 轴夹角记作 θ_{FYi} ；当 $\tau_{FYi,j} \leq t < T_{FYi,j}$ ，此时烟幕干扰弹已投放，但尚未起爆，

烟幕干扰弹的位置记为 $(x_{FYi,j}(t), y_{FYi,j}(t), z_{FYi,j}(t))$ ； $t \geq T_{FYi,j}$ 时，烟幕干扰弹已经起爆，形成烟幕云团，云团位置可同样记为 $(x_{FYi,j}(t), y_{FYi,j}(t), z_{FYi,j}(t))$ 。

对烟幕云团与导弹间的相对位置关系进行研究，记烟幕云团中心指向导弹的方向向量为 $\mathbf{e}_{Mk, YFi,j}$ ，则有：

$$\mathbf{e}_{Mk, YFi,j} = \begin{bmatrix} X_{Mk}(t) \\ Y_{Mk}(t) \\ Z_{Mk}(t) \end{bmatrix} - \begin{bmatrix} x_{FYi,j}(t) \\ y_{FYi,j}(t) \\ z_{FYi,j}(t) \end{bmatrix} \quad (3)$$

5.1.2 轨迹方程的确定

综合上述分析与假设，对各模拟对象的运动轨迹建立方程。

导弹轨迹为直线，运动方向指向假目标，即存在关系式：

$$\begin{bmatrix} X_{Mk} \\ Y_{Mk} \\ Z_{Mk} \end{bmatrix}_{(t)} = \begin{bmatrix} X_{Mk} \\ Y_{Mk} \\ Z_{Mk} \end{bmatrix}_{(0)} - \frac{v_m t}{\|\mathbf{e}_{Mk, fake}\|} \mathbf{e}_{Mk, fake} \quad (4)$$

无人机保持等高度的匀速直线运动，可表示为：

$$\begin{bmatrix} X_{FYi} \\ Y_{FYi} \\ Z_{FYi} \end{bmatrix}_{(t)} = \begin{bmatrix} X_{FYi} \\ Y_{FYi} \\ Z_{FYi} \end{bmatrix}_{(0)} + v_{FYi} t \begin{bmatrix} \cos \theta_{FYi} \\ \sin \theta_{FYi} \\ 0 \end{bmatrix} \quad (5)$$

烟幕干扰弹在脱离无人机后继续沿无人机运动方向做平抛运动，即有：

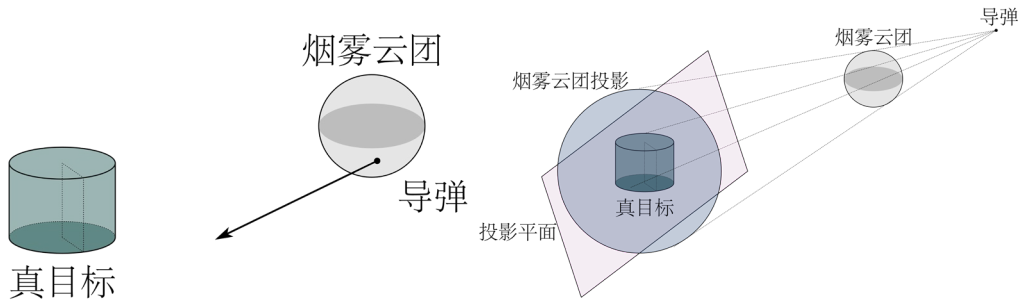
$$\begin{bmatrix} x_{FYi,j} \\ y_{FYi,j} \\ z_{FYi,j} \end{bmatrix}_{(t)} = \begin{bmatrix} X_{FYi} \\ Y_{FYi} \\ Z_{FYi} \end{bmatrix}_{(\tau_{FYi,j})} + (t - \tau_{FYi,j}) \cdot \begin{bmatrix} v_{FYi} \cos \theta_{FYi} \\ v_{FYi} \sin \theta_{FYi} \\ 1/2 \cdot g(t - \tau_{FYi,j}) \end{bmatrix} \quad (6)$$

烟幕干扰弹爆开，烟幕云团成形后，以匀速下沉，故存在关系：

$$\begin{bmatrix} x_{FYi,j} \\ y_{FYi,j} \\ z_{FYi,j} \end{bmatrix}_{(t)} = \begin{bmatrix} x_{FYi,j} \\ y_{FYi,j} \\ z_{FYi,j} \end{bmatrix}_{(T_{FYi,j})} - (t - T_{FYi,j}) \begin{bmatrix} 0 \\ 0 \\ v_s \end{bmatrix} \quad (7)$$

5.1.3 烟幕有效遮蔽的计算方法

综合光路轨迹模拟及理论分析，烟幕云团对真目标达成完美遮蔽效果有两种情况：一种是烟幕云团将导弹完全包裹，另一种是烟幕云团位于真目标与导弹中间，对导弹的可视域进行精准覆盖，使得真目标被完全掩盖。后者又可以表现为在以导弹为源的锥形光源照射下，烟幕云团在投影平面上的投影将真目标的投影完全包裹。



情况一：导弹位于球状烟幕云团内，如图 3 左图所示

该情况下的条件约束为导弹和烟幕云团中心的距离小于10m，即

$$s_{Mk, FYi, j}(t) = \|e_{Mk, FYi, j}(t)\| < R_{cloud} \quad (8)$$

情况二：烟幕云团位于真目标和导弹之间，其投影完全挡住真目标，如图 3 右图所示

将烟幕云团中心和导弹连线，直线方程为

$$\frac{X - X_{Mk}(t)}{X_{Mk}(t) - x_{FYi, j}(t)} = \frac{Y - Y_{Mk}(t)}{Y_{Mk}(t) - y_{FYi, j}(t)} = \frac{Z - Z_{Mk}(t)}{Z_{Mk}(t) - z_{FYi, j}(t)} \quad (9)$$

假设投影平面通过真目标下底面圆心 O_{real} ，其法向量为 $e_{Mk, FYi, j}$ ，则该平面方程可表示为：

$$(X_{Mk}(t) - x_{FYi, j}(t)) \cdot (X - x_{real}) + (Y_{Mk}(t) - y_{FYi, j}(t)) \cdot (Y - y_{real}) + (Z_{Mk}(t) - z_{FYi, j}(t)) \cdot (Z - z_{real}) = 0 \quad (10)$$

将方程(9)和(10)联立，可算得云团中心在投影平面上的位置 O_{proj} ，坐标记作 $(x_{proj, Mk, FYi, j}, y_{proj, Mk, FYi, j}, z_{proj, Mk, FYi, j})$ ，即

$$\begin{bmatrix} x_{proj, Mk, FYi, j} \\ y_{proj, Mk, FYi, j} \\ z_{proj, Mk, FYi, j} \end{bmatrix} (t) = \begin{bmatrix} X_{Mk} \\ Y_{Mk} \\ Z_{Mk} \end{bmatrix} (t) - \frac{\|e_{Mk, real}\| \|e_{Mk, FYi, j}\|}{\|e_{Mk, FYi, j}\|^2} e_{Mk, FYi, j} \quad (11)$$

记导弹距离云团中心的距离为 $s_{Mk, FYi, j}$ ，导弹距离云团投影中心的距离为 $s_{proj, Mk, FYi, j}$ ，如图 4 左图所示，根据相似三角形的性质，可求得烟幕云团形成的投影半径为

$$R_{proj} = \frac{s_{proj, Mk, FYi, j}}{\sqrt{s_{Mk, FYi, j}^2 - R_{cloud}^2}} R_{cloud} \quad (12)$$

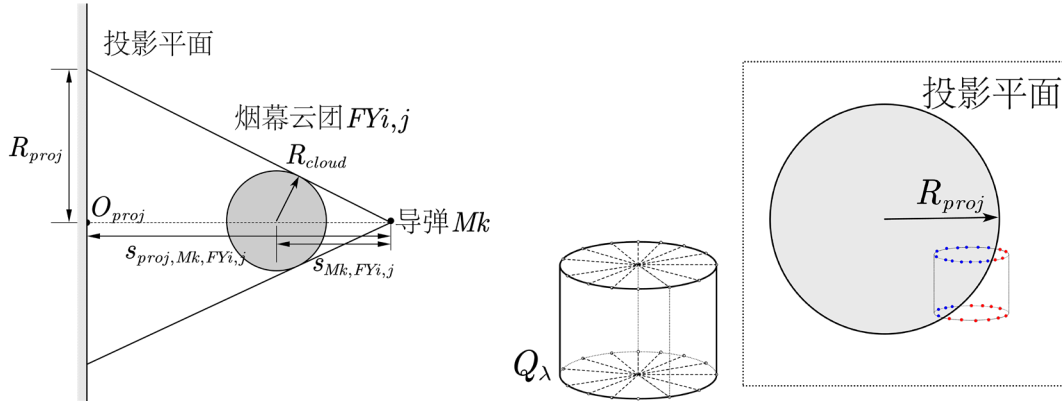


图 4 烟幕云团与真目标的投影示意图

真目标是一个半径7m 高度10m 的圆柱体，为便于计算，在其顶面和底面圆周上均匀取 λ 个特征点 Q_λ ，坐标记作 $(x_{Q_\lambda}, y_{Q_\lambda}, z_{Q_\lambda})$ ，可将特征点指向导弹的方向向量表示为 e_{Mk, Q_λ} 。圆柱体上特征点和导弹连线的直线方程为：

$$\frac{X - X_{Mk}(t)}{X_{Mk}(t) - x_{Q_\lambda}} = \frac{Y - Y_{Mk}(t)}{Y_{Mk}(t) - y_{Q_\lambda}} = \frac{Z - Z_{Mk}(t)}{Z_{Mk}(t) - z_{Q_\lambda}} \quad (13)$$

联立方程(10)和(13)，可得圆柱上特征点在投影平面上的位置为 $(x_{proj, Q_\lambda}, y_{proj, Q_\lambda}, z_{proj, Q_\lambda})$ ，表示为：

$$\begin{bmatrix} x_{proj,Q\lambda} \\ y_{proj,Q\lambda} \\ z_{proj,Q\lambda} \end{bmatrix} = \begin{bmatrix} X_{Mk} \\ Y_{Mk} \\ Z_{Mk} \end{bmatrix}_{(t)} - \frac{\|e_{Mk,real}\| \|e_{Mk,FYi,j}\|}{\|e_{Mk,Q\lambda}\| \|e_{Mk,FYi,j}\|} e_{Mk,Q\lambda} \quad (14)$$

定义在投影平面上，特征点投影位置和云团中心投影位置 O_{proj} 的距离为 $s_{proj,Q\lambda,FYi,j,Mk}$ 。若所有特征点投影位置都满足 $s_{proj,Q\lambda,FYi,j,Mk} \leq R_{proj}$ ，则可认为形成有效遮蔽，否则不为有效遮蔽，如图4右图所示。

5.1.4 问题一的求解与结果

基于上述烟幕有效遮蔽的计算方法，从 $t=0$ 时刻开始，以 Δt 为步长进行遍历搜索，具体步骤如下：

STEP1: 计算时刻 t 时导弹M1和无人机FY1坐标；

STEP2: 如果 $t < \tau_{FY1,1}$ ，说明没有投放烟幕干扰弹，回到STEP1，考虑 $t + \Delta t$ 时刻的情况；

STEP3: 如果 $\tau_{FY1,1} \leq t < T_{FY1,1}$ ，说明已投放烟幕干扰弹，计算该时刻烟幕弹坐标，回到STEP1，考虑 $t + \Delta t$ 的情况；

STEP4: 如果 $t \geq T_{FY1,1}$ ，说明已形成烟幕云团，计算是否形成有效遮挡：若形成有效遮挡，则记录当前时刻；若没有形成遮挡，则继续下一步；

STEP5: 回到STEP1，考虑 $t + \Delta t$ 的情况，直至导弹M1击中假目标；

STEP6: 根据形成有效遮挡的时刻，得到有效遮蔽时长。

根据题目要求和上述求解步骤，设置参数如表2所示。

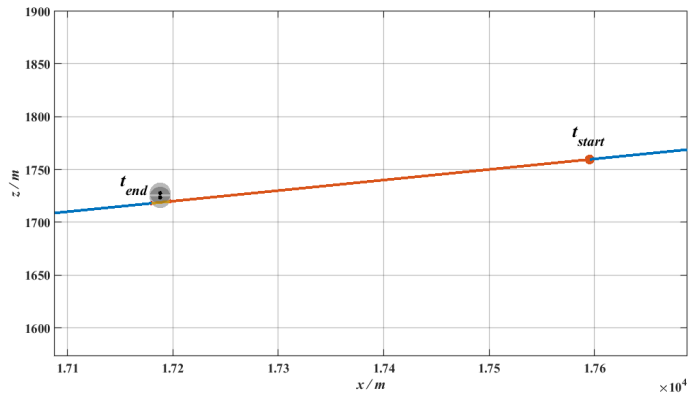
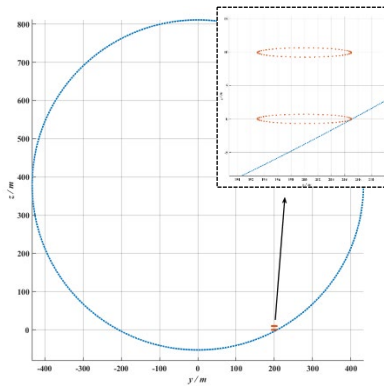
表2 参数设置

YF1速度	YF1方向	投弹时刻	起爆时刻	特征点数量	时间步长
v_{FY1}	θ_{YF1}	$\tau_{FY1,1}$	$T_{FY1,1}$	λ	Δt
120m/s	180°	1.5s	5.1s	12	0.0001s

求解结果如表3所示。

表3 问题一求解结果

有效遮蔽开始时刻 t_{start}	有效遮蔽结束时刻 t_{end}	有效遮蔽时长 t_{shadow}
8.0555s	9.4481s	1.3926s



对结果进行可视化展现。从图 5 可以看到，初始开始遮蔽时，真目标特征点的投影落在云团投影边界内，说明真目标在导弹的可视界域内被云团完全遮挡，符合对烟幕遮蔽有效的定义。图 6 描述了该云团对导弹的有效遮蔽范围，不难看出，导弹被烟幕云团包裹所提供的有效遮蔽时长占比很小，大部分有效遮蔽时间云团都位于导弹和云团中间。结果表明，对于同一颗烟幕干扰而言于情况二比情况一提供了更多的遮挡可能。

值得注意的是，由于无人机、导弹、烟幕干扰弹路径的在水平方向上均具有固定单向性质，单个烟幕干扰弹对单个导弹最多只存在一个遮蔽区间。

5.2 问题二的模型建立与求解

5.2.1 投弹策略的影响因素

根据 5.1.2 建立的轨迹方程 (5) (6) (7)，决定投放策略的主要决策变量——无人机飞行速度、飞行方向、烟幕干扰弹投放时间、起爆时间和起爆位置之间存在等价关系，即有

$$f(v_{FY1}, \theta_{FY1}, \tau_{FY1,1}, T_{FY1,1}) = g(x_{FY1,1}, y_{FY1,1}, z_{FY1,1}, T_{FY1,1}) \quad (15)$$

考虑到本策略的核心即是使得烟幕云团在合适的时机、合适的位置展开，对来袭导弹达成有效干扰，故将起爆时间与起爆位置作为主要参考因素，同时根据起爆时间和起爆坐标，无人机飞行速度、飞行方向、投弹时间可反推得出。

5.2.2 有效遮蔽时长优化模型的建立

本节建立了单个烟幕干扰弹对单个导弹的有效遮蔽时长优化模型，在形成有效遮蔽、无人机速度等约束条件下，确定烟幕干扰弹起爆相关参数，使得烟幕云团对导弹尽可能长时间地形成有效遮蔽。

• 决策变量

烟幕干扰弹的投放策略主要包括无人机飞行速度、飞行方向、烟幕干扰弹投放时间和起爆时间，会显著影响有效遮蔽的时长。由 5.2.1 可知，烟幕干扰弹的起爆时间和起爆坐标等价于上述参数，故将起爆时间 $T_{FY1,1}$ 和起爆坐标 $(x_{FY1,1}, y_{FY1,1}, z_{FY1,1})$ 作为决策变量。

• 目标函数

问题的优化目标为对导弹 M1 的遮蔽时长尽可能长。考虑到遮蔽区间的唯一性，因此，目标函数为

$$\max |t_{end} - t_{start}| = f(x_{FY1,1}, y_{FY1,1}, z_{FY1,1}, T_{FY1,1}) \quad (16)$$

其中， t_{start} 为开始遮蔽的时刻， t_{end} 为结束遮蔽的时刻。

• 约束条件

1. **起爆时间约束：**烟幕干扰弹起爆时必须在导弹命中假目标之前，即

$$0 \leq T_{FY1,1} < \frac{\sqrt{X_{M1}^2(0) + Y_{M1}^2(0) + Z_{M1}^2(0)}}{v_M} = t_{hit} \quad (17)$$

2. **起爆高度约束：**起爆高度不超过无人机 FY1 初始高度，且能在初始时刻立即投弹后，经过时间 $T_{FY1,1}$ 的平抛运动达到此高度，即

$$Z_{FY1}(0) - 1/2 \cdot gT_{FY1,1}^2 \leq z_{FY1,1} \leq Z_{FY1}(0) \quad (18)$$

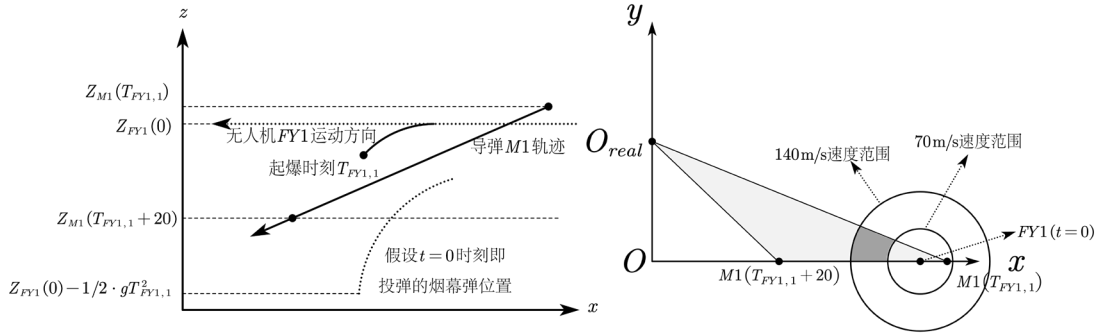
3. **起爆范围约束：**起爆点和无人机初始位置的水平距离必须能在经过时间 $T_{FY1,1}$ 后以 $70 \sim 140 \text{m/s}$ 范围内的速度到达，即

$$v_{FY \min} T_{FY1,1} \leq \sqrt{(X_{FY1(0)} - x_{FY1,1})^2 + (Y_{FY1(0)} - y_{FY1,1})^2} \leq v_{FY \max} T_{FY1,1} \quad (19)$$

4. **起爆形成遮蔽的高度约束：**为形成有效遮蔽，起爆高度应不超过 $T_{FY1,1}$ 时刻导弹高度，且不低于 $T_{FY1,1} + 20 \text{s}$ 时刻的导弹高度，即

$$Z_{M1}(T_{FY1,1} + 20) \leq z_{FY1,1}(T_{FY1,1}) \leq Z_{M1}(T_{FY1,1}) \quad (20)$$

5. **起爆形成遮蔽的范围约束：**为形成有效遮蔽，起爆范围应落在 $T_{FY1,1}$ 时刻导弹位置、 $T_{FY1,1} + 20 \text{s}$ 时刻导弹位置和真目标位置三点构成的三角形内部，如图 8 所示，设深色阴影区域为 Ω 。



综上，本节建立了有效遮蔽时长的优化模型

$$\begin{aligned} \max \quad & |t_{end} - t_{start}| = f(x_{FY1,1}, y_{FY1,1}, z_{FY1,1}, T_{FY1,1}) \\ \text{s.t.} \quad & \begin{cases} 0 \leq T_{FY1,1} < t_{hit} \\ \text{高度约束} \begin{cases} Z_{FY1}(0) - 1/2 \cdot g T_{FY1,1}^2 \leq z_{FY1,1} \leq Z_{FY1}(0) \\ Z_{M1}(T_{FY1,1} + 20) \leq z_{FY1,1}(T_{FY1,1}) \leq Z_{M1}(T_{FY1,1}) \end{cases} \\ \text{范围约束} \begin{cases} v_{FY \min} T_{FY1,1} \leq s_{FY1} \leq v_{FY \max} T_{FY1,1} \\ (x_{FY1,1}, y_{FY1,1}, z_{FY1,1}) \in \Omega \end{cases} \end{cases} \end{aligned} \quad (21)$$

5.2.3 有效遮蔽时长优化模型的求解与结果

本文采用变步长搜索法寻找烟幕干扰弹的最佳起爆时间和起爆位置，在满足时间、起爆高度、起爆范围等约束条件下使得有效遮蔽尽可能长。采用变步长搜索可以在保证求解精度的情况下，减少大量计算时间。

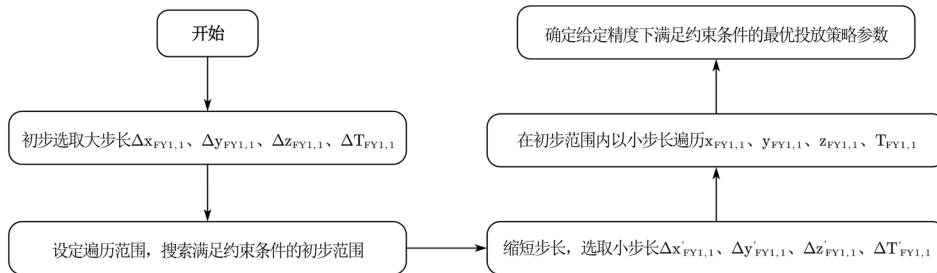


图 9 变步长搜索法寻找最有投放策略流程图

STEP1: 初始选取大步长, 起爆位置和起爆时间初始搜索步长设置如表 4 所示;

表 4 起爆参数初始搜索步长

参数	起爆时间 $T_{FY1,1}$ / s	起爆坐标 (x,y,z) / m
初始步长	0.2	15

STEP2: 设置遍历范围, 进行初次遍历. 根据有效遮蔽时长优化模型的约束条件, 得出起爆参数的初始遍历范围, 如下表所示:

表 5 起爆参数初始遍历范围

参数	起爆时间 $T_{FY1,1}$ / s	起爆坐标 (x,y,z) / m
初始遍历范围	$[0, t_{hit}]$	(x,y,z) in Ω

各参数在对应遍历范围内, 以对应的步长进行遍历, 计算烟幕云团最佳的有效遮蔽时长, 结果如图 10 左图所示。图中任意一点均对应该起爆时间下约束区域内的最长有效遮蔽时长。

可以看到, 当起爆时间在 $(0, 2.1)$ 以及 $(2.1, 5)$ 两个区间内, 有效遮蔽时长均出现了极大值。试结合对应条件下无人机飞行方向进行分析: 第一个区间内无人机运动方向与 x 轴正方向夹角接近 0° , 说明其主动迎向导弹飞行路径进行投掷; 第二个区间内无人机运动方向接近 180° , 转而面向真目标点运动, 由于无人机与导弹速度差距过大, 导致起爆点需距离目标点更近, 烟幕云团在投影平面上投影面积变小, 即有效遮蔽时长变短。

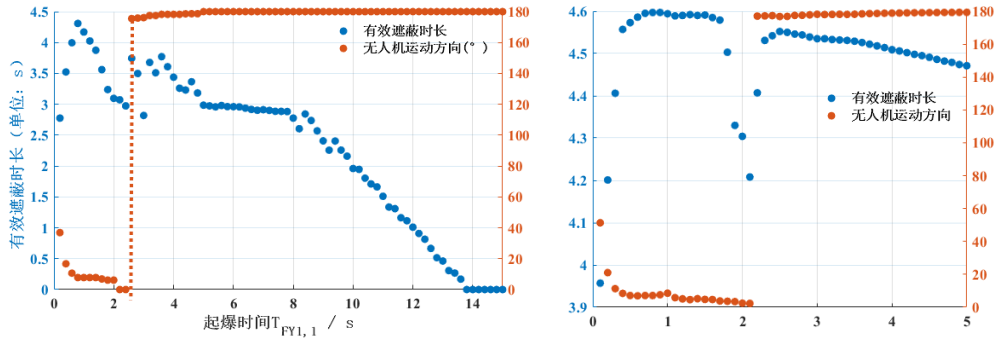


图 10 变步长搜索法得到的不同起爆时间下最长有效遮蔽时长
(左: 大步长, 右: 小步长)

STEP3: 缩短搜索步长, 在新遍历范围内进一步搜索最优的投放策略参数: 起爆时间 $T_{FY1,1}$ 以及起爆坐标 (x,y,z) , 小步长以及新遍历范围参数设定如下。

表 6 小步长与新遍历范围参数设置

参数	起爆时间 $T_{FY1,1}$ / s	起爆坐标 (x,y,z) / m
步长	0.1	2
遍历范围	$[0, 5]$	(x,y,z) in Ω

小步长的遍历结果如图 10 右图所示。与大步长相比, 小步长搜索的峰形状有所变化, 这是因为遍历的离散长度更小, 所包含空间网格更加密集, 故计算精度更高。可以看到, 小步长搜索下有效遮蔽时长的两个峰值在一定范围内具有相当好的连续性。验证了优化模型以及变步长搜索方案的合理性。

STEP4: 遍历搜索结果。经过遍历搜索后，得到最优的烟幕干扰投放策略，即起爆时间和起爆位置，再通过轨迹方程 (5) (6) (7)，得到飞行速度、飞行方向、投弹时刻和投弹位置。问题二的求解结果如表 7 所示。

表 7 问题二的求解结果

FY1 速度 v_{FY1}	FY1 方向 θ_{FY1}	投弹时刻 $\tau_{FY1,1}$	起爆时刻 $T_{FY1,1}$
120.9339m/s	7.1250°	0s	0.8s
投放点	起爆点	有效遮蔽区间	有效遮蔽时长
(17800, 0, 2000)	(17896, 12, 1796.9)	[0.8973s, 5.4949s]	4.5976s

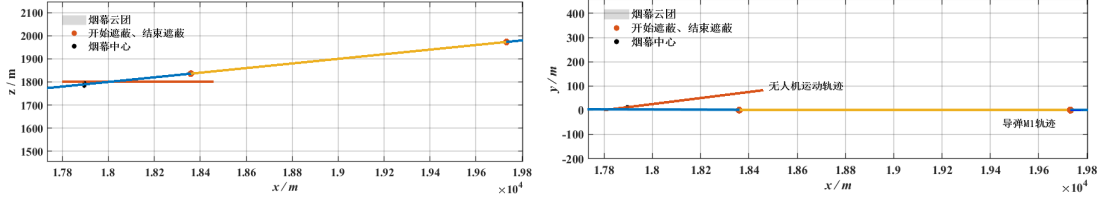


图 11 投放策略轨迹示意图（左侧：正视图；右侧：俯视图）

采用求解结果对投放策略进行计算，导弹、无人机、烟幕干扰弹的运动轨迹如图 11 所示。可以看到，在无人机FY1针对导弹M1 投放烟幕干扰的最佳策略是迎向导弹来袭方向投放，并与 x 轴正方向有一定夹角，该枚烟幕云团对导弹 M1 形成了 4.5976s 时长的有效遮蔽，对应导弹飞行距离约 1300 米，具有良好的干扰作用。

5.3 问题三的建立与求解

5.3.1 单个无人机的多枚烟幕干扰弹投放策略

定义第 i 架无人机投放的第 j 枚烟幕干扰弹对第 k 枚导弹形成的有效遮蔽区间为 $[t_{start, FYi,j, Mk}, t_{end, FYi,j, Mk}]$ ，考虑到各枚干扰弹形成的遮蔽区间可能互相存在重叠，则所有烟幕干扰弹的有效遮蔽时长可表示为

$$t_{shadow} = \sum_{k=1}^3 \bigcup_{i,j} [t_{start, FYi,j, Mk}, t_{end, FYi,j, Mk}] \quad (22)$$

以尽可能延长有效遮蔽时长为评估目标，基于问题二求解结果，探寻运动方向、运动速度以及起爆时间之间的关联。

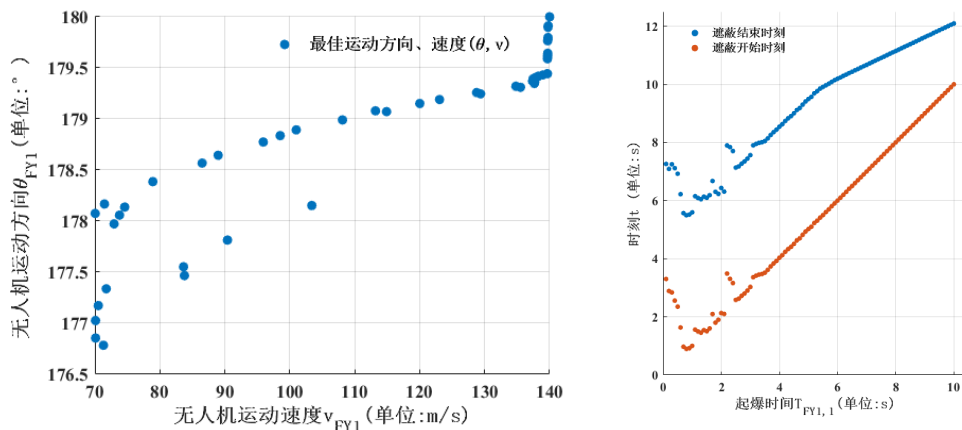


图 12 不同起爆时间对应最佳运动方向速度散点图

如图 12 所示, 在以追寻最长有效遮蔽时间的前提下, 图中每一个散点均标注了某一起爆时间 $T_{FY1,1}$ 对应的无人机最佳运动方向和速度。可以看到, 散点的坐标可大致近似成一条平滑曲线, 说明这一分布具有相当高的连续性。考虑到起爆时间 $T_{FY1,1}$ 在这一过程中同样单调连续增长, 可认为最佳运动方向 θ 与运动速度 v 是一个以起爆时间 $T_{FY1,1}$ 为变量的函数, 即针对 θ 、 v 的研究均可化简为对起爆时间 $T_{FY1,1}$ 的考量。

以起爆时间作为决策变量, 将有效遮蔽时长拆解为遮蔽开始时刻和遮蔽结束时刻两条序列, 求取结果如图 13 所示, 结合图像以及 5.2.2 的分析可以看到, 当起爆时间 $T_{FY1,1}$ 满足 $0 \leq T_{FY1,1} \leq 2.1$, 无人机主动迎向导弹, 遮蔽开始时刻以及遮蔽结束时刻随起爆时间增长成减小趋势。当起爆时间满足 $2.1 \leq T_{FY1,1}$, 无人机转而面向真目标点飞行, 遮蔽开始时刻以及遮蔽结束时刻随起爆时间增长而增大。

考虑到无人机需投掷三枚烟幕干扰弹, 起爆时间间隔需在 1s 以上, 为尽可能延长有效遮蔽时间, 应尽量避免三次烟幕干扰之间的遮蔽时间区间重叠。因此, 虽然最长的有效遮蔽时长在 $T_{FY1,1}$ 为 $(0, 2.1)$ 区间内, 但该范围内各遮蔽区间较为接近, 总体的有效覆盖效果较差。而在 $2.1 \leq T_{FY1,1}$ 范围内, 尽管单枚烟幕干扰的有效遮蔽时长不够长, 但能使三枚烟幕干扰弹做到总体时间更久、更具效率的遮蔽。因此在需要投放多枚烟幕干扰弹时, 应更多考虑无人机朝向真目标点飞行的情况。

5.3.2 多枚烟幕干扰弹的有效遮蔽时长优化模型建立

本节建立单架无人机投放多枚烟幕干扰弹对单个导弹的有效遮蔽时长优化模型, 在形成有效遮蔽、无人机速度、投放时间间隔等约束条件下, 确定烟幕干扰弹起爆相关参数, 使得多个烟幕云团对导弹尽可能长时间地形成有效遮蔽。

• 决策变量

由于无人机出发后速度大小和方向无法改变, 因此利用无人机FY1 投放 3 枚烟幕干扰弹的投放策略共需要确定 8 个参数: 无人机飞行速度 v_{FY1} , 飞行方向 θ_{FY1} , 三枚烟幕弹投放时刻 $\tau_{FY1,1}, \tau_{FY1,2}, \tau_{FY1,3}$, 三枚烟幕弹起爆时刻 $T_{FY1,1}, T_{FY1,2}, T_{FY1,3}$ 。基于问题 2 求解结果, 若假设已知某一枚烟幕弹起爆时刻 $T_{FY1,i}$, 则采用对应的单枚烟幕弹最佳布置策略, 从而可确定 v_{FY1} , θ_{FY1} 和 $\tau_{FY1,i}$, 故将 $T_{FY1,1}, T_{FY1,2}, T_{FY1,3}, \tau_{FY1,2}, \tau_{FY1,3}$ 共 5 个参数作为决策变量。

• 目标函数

问题三要求使用无人机FY1 投放三枚烟幕干扰弹, 有效遮蔽时长可以简化为

$$t_{shadow} = \bigcup_{j=1,2,3} [t_{start,FY1,,j,M1}, t_{end,FY1,,j,M1}] \quad (23)$$

因此, 目标函数为

$$\max t_{shadow} = f(T_{FY1,1}, T_{FY1,2}, T_{FY1,3}, \tau_{FY1,2}, \tau_{FY1,3}) \quad (24)$$

• 约束条件

相较于 5.2.2 有效遮蔽时长优化模型的约束条件 (式(17)-(20)), 新增约束条件如下:

1. 投放时间间隔约束：任意两枚烟幕干扰弹的投放时间应间隔至少1s，即

$$\min\{|\tau_{FY1,1} - \tau_{FY1,2}|, |\tau_{FY1,2} - \tau_{FY1,3}|, |\tau_{FY1,1} - \tau_{FY1,3}|\} \geq 1s \quad (25)$$

2. 投放时刻和起爆时刻先后约束：任意一枚烟幕干扰弹的起爆时刻不能早于投放时刻，即

$$\forall j=1,2,3, T_{FY1,j} - \tau_{FY1,j} \geq 0 \quad (26)$$

综上，本节建立了针对多枚烟幕干扰弹的有效遮蔽时长优化模型。

5.3.3 基于贪心策略的优化模型求解与结果

本节通过贪心策略对多枚烟幕干扰弹的投放策略进行优化，流程如下：

STEP1: 选取 $T_{FY1,1}$ 作为1号烟幕干扰弹的起爆时间，基于问题二的求解方法，计算该时刻下的最佳起爆位置；代入轨迹方程，得到无人机飞行速度 v_{FY1} 、飞行方向 θ_{FY1} 、投放时刻 $\tau_{FY1,1}$ ；

STEP2: 以 $\Delta t = 0.1s$ 为步长，遍历满足约束条件的 $\tau_{FY1,2}$ 和 $T_{FY1,2}$ ，并求解1、2号烟幕干扰弹形成的有效遮蔽总时长 $t_{shadow1,2}$ ；选取有效遮蔽总时长 $t_{shadow1,2}$ 最大时对应的 $\tau_{FY1,2}$ 和 $T_{FY1,2}$ ，作为2号烟幕弹的投放策略；

STEP3: 以 $\Delta t = 0.1s$ 为步长，遍历满足约束条件的 $\tau_{FY1,3}$ 和 $T_{FY1,3}$ ，并求解1、2、3号烟幕干扰弹形成的有效遮蔽总时长 $t_{shadow1,2,3}$ ；选取有效遮蔽总时长 $t_{shadow1,2,3}$ 最大时对应的 $\tau_{FY1,3}$ 和 $T_{FY1,3}$ ，作为3号烟幕弹的投放策略；

STEP4: 基于STEP1-3，得到 $T_{FY1,1}$ 下的3枚烟幕干扰弹投放策略；以 $\Delta t = 0.1s$ 为步长，遍历满足约束条件的 $T_{FY1,1}$ ，可得最长的有效遮蔽总时长。

通过以上步骤，求取得到的最佳投放策略参数如下：

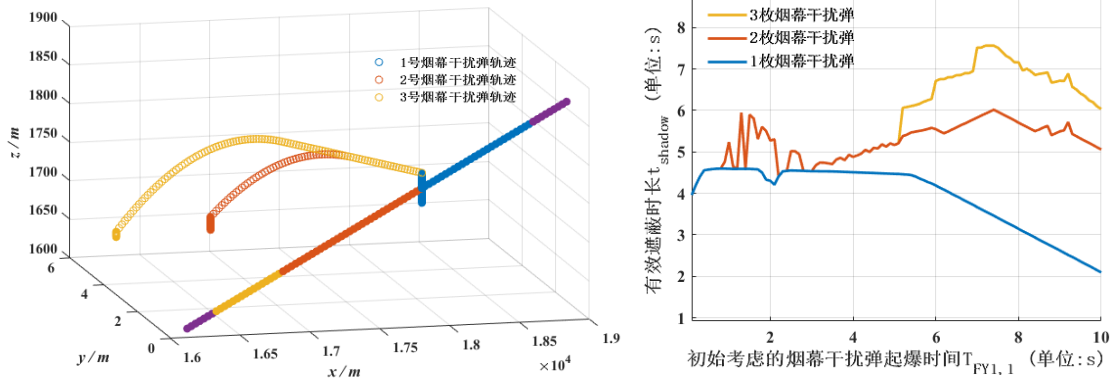
表8 问题三的求解结果

速度 v_{FY1}	方向 θ_{FY1}	烟幕弹编号	投放时刻 $\tau_{FY1,i}$	起爆时刻 $T_{FY1,i}$
139.7308m/s	179.7784°	1	0s	0s
		2	2.4860s	7.4s
		3	4.9s	10.7s
投放点	起爆点	有效遮蔽区间		有效遮蔽时长
(17800, 0, 1800)	(17800, 0, 1800)	[4.842s, 7.404s]		2.562s
(17453, 1.3, 1800)	(16766, 4, 1682)	[7.404s, 10.861s]		3.457s
(17115, 2.6, 1800)	(16305, 6, 1635)	[10.861s, 12.406s]		1.545s
总有效遮蔽时长：7.564s				

将优化模型求解结果及求解过程进行可视化，三枚烟幕干扰弹的投掷轨迹以及每枚烟幕干扰弹有效遮蔽范围在导弹打击路径上的体现如图14所示。

图15以初始决策的烟幕干扰弹起爆时间为因变量，三条曲线分别代表了单枚、两枚、三枚烟幕干扰弹有效遮蔽总时长三种情况。对于单枚烟幕干扰弹，随着起爆时间的推迟，其所达成遮蔽效果逐渐下降。在加入第二枚烟幕干扰弹的遮蔽效果进入评估之后，有效遮蔽总时长有整体提升，甚至后半段的分布也出现改

变，转为上升趋势，原因是新增的烟幕干扰弹在 $t = 0$ 时刻即起爆，而初始决策的烟幕干扰弹起爆时间越晚，与其遮蔽区间的重叠越少。在起爆时间为 $(0, 2.1)$ 的时间段内，有效遮蔽总时长略有增加，原因在于该起爆时间时段内对应的无人机投放策略为迎向导弹来袭方向投放，虽然在单枚情况下具有较长遮蔽时长，但会与新增的遮蔽区间重合较多。加入第三枚烟幕干扰弹后，在两枚的基础上提供了更大范围的有效遮蔽时间覆盖，且普遍出现在初始决策烟幕干扰弹起爆时间较晚的情况；在其他情况下，导弹往往已飞越无人机，由于无人机速度受限，其已无法布置第三枚烟幕干扰弹。



5.4 问题四的模型建立与求解

5.4.1 多架无人机对单枚导弹的有效遮蔽时长优化模型建立

本节建立多架无人机分别投放单枚烟幕干扰弹对单枚导弹的有效遮蔽时长优化模型，在形成有效遮蔽和无人机速度等约束条件下，确定烟幕干扰弹起爆相关参数，使得多架无人机投放的烟幕云团对导弹尽可能长时间地形成有效遮蔽。

• 决策变量

多架无人机的运动相互独立，该模型可拆分为多个问题二所建立模型进行求解，因此将每一枚烟幕干扰弹的起爆时间 $T_{FYi,1}$ 和起爆位置 $(x_{FYi,1}, y_{FYi,1}, z_{FYi,1})$ 作为决策变量。

• 目标函数

问题四要求使用无人机FY1、FY2、FY3分别投放1枚烟幕干扰弹对导弹M1进行干扰，故有效遮蔽总时长可简化为

$$t_{shadow} = \bigcup_{i=1,2,3} [t_{start,FYi,1,M1}, t_{end,FYi,1,M1}] \quad (27)$$

因此，目标函数为

$$\max t_{shadow} = f \left(\begin{matrix} T_{FY1,1}, x_{FY1,1}, y_{FY1,1}, z_{FY1,1} \\ T_{FY2,1}, x_{FY2,1}, y_{FY2,1}, z_{FY2,1} \\ T_{FY3,1}, x_{FY3,1}, y_{FY3,1}, z_{FY3,1} \end{matrix} \right) \quad (28)$$

• 约束条件

相较于 5.2.2 优化模型，无新增类型的约束条件，将式(17)-(20)应用于每一枚烟幕干扰弹即可。

5.4.2 基于区间匹配的优化模型求解与结果

本节通过将该模型拆分为多个问题二所建立模型，分别独立求解，并将独立求解结果按照遮蔽区间进行匹配，使得有效遮蔽总时长最长。具体步骤如下：

STEP1: 分别考虑无人机FY*i*，基于问题二的变步长搜索法，在满足约束条件的情况中，得到不同起爆时间 $T_{FYi,1}$ 下对应的最佳起爆位置 $(x_{FYi,1}, y_{FYi,1}, z_{FYi,1})$ ；

STEP2: 基于轨迹方程(5)-(7)，得到 $(T_{FYi,1}, x_{FYi,1}, y_{FYi,1}, z_{FYi,1})$ 对应的有效遮蔽区间，记作 $[t_{start, FYi, 1}, t_{end, FYi, 1}]$ ；

STEP3: 从每架无人机的求解结果中选取一段有效遮蔽区间，计算总有效遮蔽时长 t_{shadow} ；遍历所有可能的遮蔽区间组合方式，得到最长有效遮蔽时长，即为最终结果。

通过以上步骤，求取得到的最佳投放策略参数如表 9 所示。

表 9 问题四的求解结果

无人机编号	速度 v_{FYi}	方向 θ_{FYi}	投放时刻 $\tau_{FYi,1}$	起爆时刻 $T_{FYi,1}$
FY1	120.9339m/s	7.1250°	0s	0.8s
FY2	136.2362m/s	298.8754°	6.5068s	11.4s
FY3	129.5806m/s	73.7362°	24.1104s	24.8s
投放点	起爆点	有效遮蔽区间	遮蔽时长	
(17800, 0, 1800)	(17896, 12, 1797)	[0.897s, 5.495s]	4.598s	
(12428, 624, 1400)	(12750, 40, 1283)	[11.629s, 15.645s]	4.016s	
(6875, -0.8, 700)	(6900, 85, 698)	[25.127s, 28.346s]	3.219s	
总有效遮蔽时长：11.833s				

将优化模型求解过程及求解结果及进行可视化。图 17 描述了三枚由不同无人机发射烟幕干扰弹的投掷轨迹，并将各自的有效遮蔽范围表现在了导弹打击路径上。

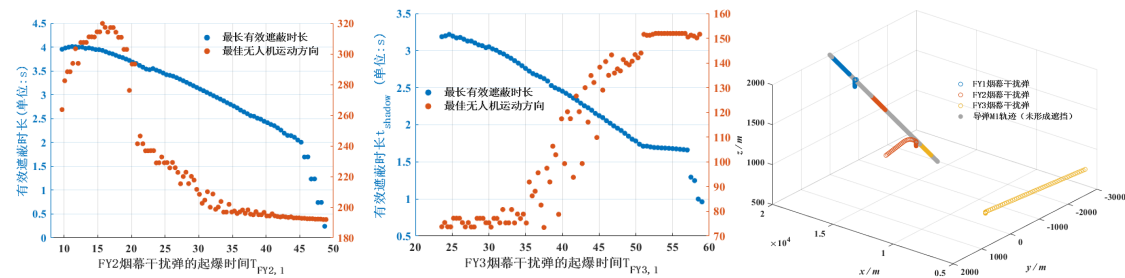


图 16 不同起爆时刻下 FY2、FY3 烟幕弹的最长有效遮蔽时长

图 16 记录了对于不同起爆时刻，FY2、FY3 在最优投放策略下所能达成的最长有效遮蔽时长。可以看到，类似于 FY1，FY2 与 FY3 在总体分布上仍呈现最长有效遮蔽时长随烟幕干扰弹起爆时间的推后而减小的趋势。值得注意的是，无人机 FY2 在其携带烟幕干扰弹起爆时间 $T_{FY2,1}$ 较小时，对应的投放策略依然是主动迎向导弹飞行路径进行投掷。对比之下，无人机 FY3 的运动方向角度保持稳

定上涨，考虑到 FY3 本身距离目标点更近，初始高度值也过低，无法做到迎向导弹路径投放烟幕干扰，只能朝向目标点运动，这一变化趋势是合理的。

5.5 问题五的模型建立与求解

5.5.1 烟幕干扰弹的目标选取策略

本问需求取 5 架无人机共计 15 枚烟幕干扰弹的最优投放策略，变量较多，模型复杂度高。为简化模型，简化计算量，考虑在合理的前提下提前指定每枚烟雾弹的遮蔽目标。

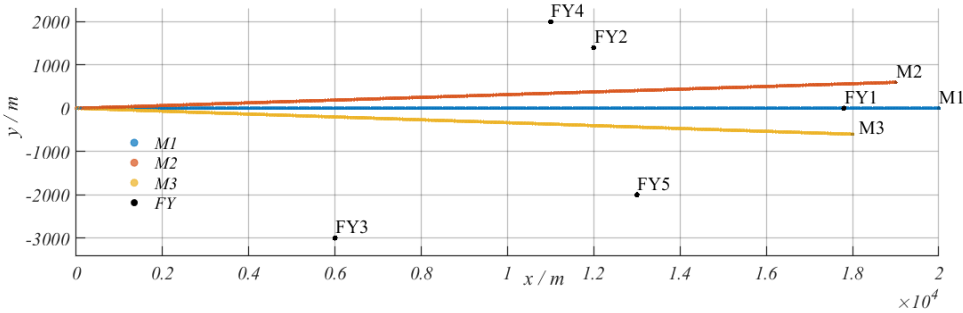


图 18 导弹与无人机初始位置俯视图

各导弹、无人机在水平面上的位置分布如图 18 所示。可以看到除去无人机 FY1 恰好位于导弹 M1 的运动路径上，其余无人机的初始位置均位于三枚导弹运动轨迹的两侧。根据题目要求，开始行动后无人机需保持等高度匀速直线飞行。这意味着无人机 FY1 最多只能选取两个导弹进行干扰，且该情况下总有效遮蔽时长大幅降低，其最具效率的投放方案是其三枚烟幕干扰弹全部用于遮蔽 M1 导弹。对于剩下 4 架无人机而言，题目条件也限制了它们投放的烟幕干扰弹只能遮蔽到不同导弹。另一方面，通过比较问题三和问题四的结论也可以发现，针对同一枚导弹，三架无人机各投放一枚烟幕干扰弹的遮蔽效果要优于单架无人机投放三枚烟幕干扰弹。故它们的最优投放策略是三枚烟幕干扰弹各自针对一枚导弹投放。

考虑各架无人机距离各枚导弹运动轨迹的远近，确定各枚烟幕干扰弹的投放策略如表 10 所示。

表 10 烟幕干扰弹目标选取策略

无人机编号	FY1	FY2	FY3	FY4	FY5
第 1 枚烟幕干扰弹的目标导弹	M1	M2	M3	M2	M3
第 2 枚烟幕干扰弹的目标导弹	M1	M1	M1	M1	M1
第 3 枚烟幕干扰弹的目标导弹	M1	M3	M2	M3	M2

5.5.2 多枚烟幕干扰弹遮蔽多枚导弹的有效遮蔽时长优化模型建立

本节建立多枚烟幕干扰弹遮蔽多枚导弹的有效遮蔽时长优化模型，在满足形成有效遮蔽、无人机速度、投放时间间隔、投放目标顺序等约束条件下，确定烟幕干扰弹起爆相关参数，使得烟幕云团对导弹尽可能长时间地形成有效遮蔽。

• 决策变量

基于问题二的求解过程可知，对于任意给定无人机 FY i ，决定第 1 枚烟幕干

扰弹投放策略的参数为起爆时间 $T_{FYi,1}$ 和起爆位置 $(x_{FYi,1}, y_{FYi,1}, z_{FYi,1})$ ，决定后续第 2、3 枚的烟幕干扰弹投放策略的参数为投放时刻 $\tau_{FYi,2}, \tau_{FYi,3}$ 和起爆时刻 $T_{FYi,2}, T_{FYi,3}$ ，因此将 $T_{FYi,1}, x_{FYi,1}, y_{FYi,1}, z_{FYi,1}, \tau_{FYi,2}, \tau_{FYi,3}, T_{FYi,2}, T_{FYi,3}$ 作为决策变量。

• 目标函数

问题五要求使用无人机FY1、FY2、FY3、FY4、FY5分别投放 3 枚烟幕干扰弹，对导弹M1,M2,M3进行遮蔽，总有效遮蔽时长可表示为

$$t_{shadow} = \sum_{k=1}^3 \bigcup_{i,j} [t_{start, FYi,j, Mk}, t_{end, FYi,j, Mk}] \quad (29)$$

因此，目标函数为

$$\max t_{shadow} = f(T_{FYi,1}, x_{FYi,1}, y_{FYi,1}, z_{FYi,1}, \tau_{FYi,2}, \tau_{FYi,3}, T_{FYi,2}, T_{FYi,3}), \quad (30)$$

$$i = 1, 2, 3, 4, 5$$

• 约束条件

相较于 5.2.2 和 5.3.2 有效遮蔽时长优化模型的约束条件，即式(17)-(20),(25)-(26)，新增约束条件如下：

1. **烟幕干扰弹投放时刻顺序约束：**由于已设定每一枚烟幕干扰弹的目标导弹，因此同一架无人机投放的各烟幕干扰弹应满足先后关系，即

$$\tau_{FYi,1} < \tau_{FYi,2} < \tau_{FYi,3} \quad (31)$$

综上，本节建立了多枚烟幕干扰弹遮蔽多枚导弹的有效遮蔽时长优化模型：

$$\max t_{shadow} = f(T_{FYi,1}, x_{FYi,1}, y_{FYi,1}, z_{FYi,1}, \tau_{FYi,2}, \tau_{FYi,3}, T_{FYi,2}, T_{FYi,3}),$$

$$i = 1, 2, 3, 4, 5, j = 1, 2, 3, k = 1, 2, 3$$

$$s.t. \begin{cases} 0 \leq T_{FYi,j} < t_{hit,k} \\ \text{高度约束} \begin{cases} Z_{FYi}(0) - 1/2 \cdot g T_{FYi,j}^2 \leq z_{FYi,j} \leq Z_{FYi}(0) \\ Z_{Mk}(T_{FYi,j} + 20) \leq z_{FYi,j}(T_{FYi,j}) \leq Z_{Mk}(T_{FYi,j}) \end{cases} \\ \text{范围约束} \begin{cases} v_{FY \min} T_{FYi,j} \leq s_{FYi} \leq v_{FY \max} T_{FYi,j} \\ (x_{FYi,j}, y_{FYi,j}, z_{FYi,j}) \in \Omega_k \end{cases} \\ \min \{ |\tau_{FY1,1} - \tau_{FY1,2}|, |\tau_{FY1,2} - \tau_{FY1,3}|, |\tau_{FY1,1} - \tau_{FY1,3}| \} \geq 1s \\ T_{FY1,j} - \tau_{FY1,j} \geq 0 \\ \tau_{FYi,1} < \tau_{FYi,2} < \tau_{FYi,3} \end{cases} \quad (32)$$

5.5.3 基于贪心策略和区间匹配的优化模型求解与结果

本节对多架无人机多枚烟幕干扰弹的投放策略进行优化，流程如下：

STEP1: 选取无人机FY*i*及其第 1 枚烟幕干扰弹的目标导弹M*k*作为研究对象，使用问题二的**变步长搜索法**得到不同起爆时间 $T_{FYi,1}$ 下的最佳起爆位置 $(x_{FYi,1}, y_{FYi,1}, z_{FYi,1})$ ；代入轨迹方程，得到无人机飞行速度 v_{FYi} 、飞行方向 θ_{FYi} 、投放时刻 $\tau_{FYi,1}$ ；

STEP2: 再选取该无人机第 2、3 枚烟幕干扰弹的目标导弹作为研究对象，采用问题三的**贪心策略**得到不同起爆时间 $T_{FYi,1}$ 下的最佳后续起爆相关参数：投放时刻 $\tau_{FYi,2}, \tau_{FYi,3}$ 和起爆时刻 $T_{FYi,2}, T_{FYi,3}$ ，得到该无人机可对各导弹形成遮蔽的区间 $[t_{start, FYi,j}, t_{end, FYi,j}]$ ；

STEP3: 遍历所有无人机，基于问题四的**区间匹配方法**，从各架无人机中任取 $T_{FYi,1}$ ，计算总有效遮蔽时长 t_{shadow} ；遍历所有可能的遮蔽区间组合方式，得到最长有效遮蔽时长，即为最终结果。

通过上述步骤，求取得到最佳投放策略，其部分关键参数如表 9 所示，其余详见附录二。

表 11 问题五的求解结果

无人机编号	速度 $v_{FYi,j}$	方向 $\theta_{FYi,j}$	烟幕弹编号	投放时刻 τ	起爆时刻 T
FY1	139.72m/s	179.77°	1	0s	0s
FY1			2	2.3s	7.2s
...		
FY5	139.61m/s	143.12°	2	16.9s	24.2s
FY5			3	21.3s	27.7s

投放点	起爆点	有效遮蔽区间	遮蔽时长
(17800, 0, 1800)	(17800, 0, 1800)	[4.842s, 7.404s]	2.562s
(17474, 1.3, 1800)	(16794, 4, 1684)	[7.200s, 10.766s]	3.566s
...
(11113, -584, 1300)	(10297, 28, 1039)	[25.238s, 28.670s]	3.432s
(10621, -215, 1300)	(9907, 31, 1099)	[27.700s, 30.524s]	2.824s

总有效遮蔽时长（已考虑遮蔽区间重叠）：42.466s			
----------------------------	--	--	--

将模型求解结果进行可视化如图 19 所示，可以看到，本策略遮蔽重心主要集中在导弹 M1 上，对导弹 M2、M3 也具有一定的遮蔽效果，总体遮蔽范围分布较大，一定程度上证明了本策略的有效性。

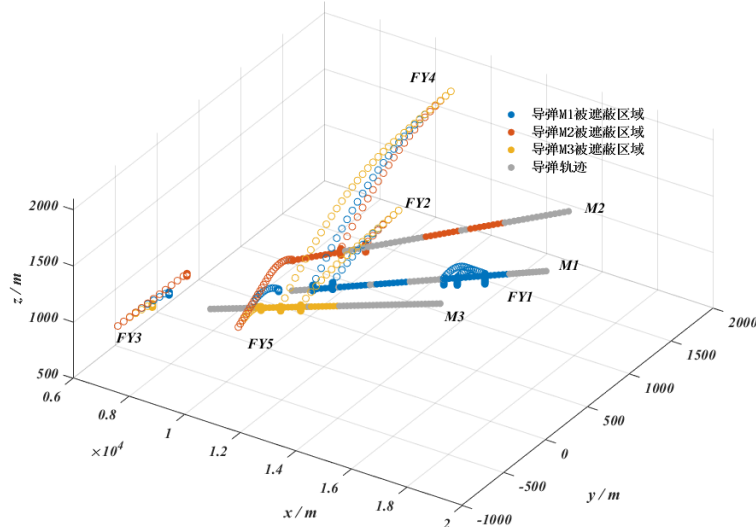


图 19 烟幕干扰弹投放策略与导弹轨迹

六、灵敏度分析

6.1 模型对特征点数量的灵敏度

在上文中，假设特征点数量为 $\lambda = 12$ 。为探究特征点数量对计算精度和计算时间的影响程度，使数量 λ 在 $[8, 12, 120, 720]$ 内进行调整，计算有效遮蔽时长。

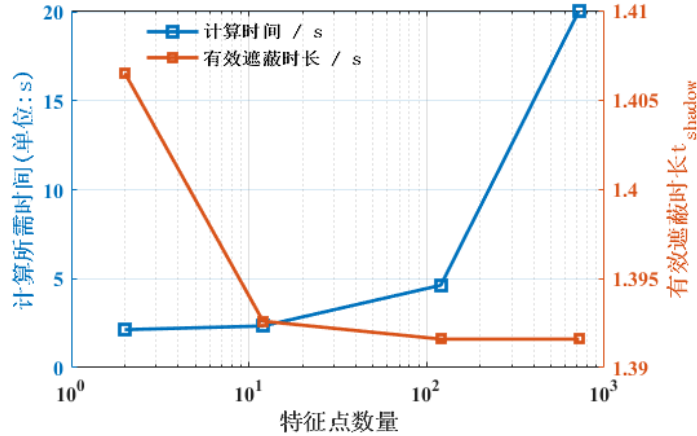


图 20 不同特征点数量的计算精度和计算所需时间

从图 24 中可以看出，当特征点数量超过 10 个时，其有效遮蔽时长的计算精度已趋于稳定，但当特征点选取数量过多时，计算所需时间会极大增加。因此，本文选取特征点数量为 $\lambda = 12$ 是合理的，既能保证计算精度，也能减少计算时间。

七、模型评价与推广

7.1 模型的优点

- 对于不同约束条件下的参数优化，将不同量纲的决策变量转化为对空间域和时间域进行离散，采用变步长搜索法，先在大范围求出结果，后再小范围提高精度；
- 模型考虑了真目标的圆柱形形态，在圆柱体的上下平面均匀采样进行模拟，对遮蔽时间的计算与投放策略的设计更符合真实场景；
- 模型将起爆时间作为主要考虑因素，求解出的遮蔽时间精确度高，投放策略设计合理，整体模型可解释性强。

7.2 模型的缺点

- 模型采用时间与空间步长离散化模拟进行求解，对真目标形状进行离散化采样，尽管综合考虑实际情况，但仍会与实际存在一定误差；
- 当满足约束条件的空间域范围较大时，遍历所需时间较长；

7.3 模型的推广

- 考虑可能会对无人机、烟幕干扰弹运动造成影响的因素，如空气阻力、风向风速等，以提高在实际情况中的有效遮蔽成功率；
- 考虑无人机在飞行过程中改变速度和方向，通过合适机动，以增加形成有效遮蔽的时长。

参考文献

本论文未参考任何文献；
本参赛队未使用任何 AI 工具。

附录

附录一 支撑文件列表

文件一 工作簿: result1.xlsx
文件二 工作簿: result2.xlsx
文件三 工作簿: result3.xlsx
文件四 程序一 (问题一) MATLAB 文件代码 (T1.m)
文件五 程序二 (问题二与问题四) MATLAB 文件代码 (T2_4.m)
文件六 程序三 (问题三) MATLAB 文件代码 (T3.m)
文件七 程序四 (问题五) MATLAB 文件代码 (T5.m)

附录二 问题五求解答案

附录三 程序代码

程序一 问题一代码 (T1.m)
程序二 问题二与问题四代码 (T2_4.m)
程序三 问题三代码 (T3.m)
程序四 问题五代码 (T5.m)

附录二 问题五求解答案

表 1 问题五求解结果

无人机编号	速度 $v_{FY_{i,j}}$	方向 $\theta_{FY_{i,j}}$	烟幕弹编号	投放时刻 τ	起爆时刻 T
FY1	139.72m/s	179.77°	1	0s	0s
FY1	139.72m/s	179.77°	2	2.3341s	7.2s
FY1	139.72m/s	179.77°	3	4.9s	10.7s
FY2	135.00m/s	281.71°	1	4.2936s	7.3s
FY2	135.00m/s	281.71°	2	5.3s	10.5s
FY2	135.00m/s	281.71°	3	9.4s	13.6s
FY3	139.50m/s	86.03°	1	17.9891s	20.7s
FY3	139.50m/s	86.03°	2	19.1s	22.1s
FY3	139.50m/s	86.03°	3	23.1s	23.6s
FY4	139.36m/s	284.08°	1	1.3775s	11.8s
FY4	139.36m/s	284.08°	2	3.1s	14.5s
FY4	139.36m/s	284.08°	3	6.8s	17.6s
FY5	139.61m/s	143.12°	1	14.0842s	19.7s
FY5	139.61m/s	143.12°	2	16.9s	27.7s
FY5	139.61m/s	143.12°	3	21.3s	27.7s

附录三 程序代码

程序一 问题一代码 (T1.m)

```
% config
clear
clc

v_m = 300;
v_s = 3;

Start_M1 = [20000, 0, 2000];
Start_M2 = [19000, 600, 2100];
Start_M3 = [18000, -600, 1900];

Fake = [0,0,0];
Real_center = [0, 200, 0];
R_real = 7;
H_real = 10;
featPoints = [];
for i = 0:60:359
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0]];
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0] + [0, 0, H_real]];
end

Start_FY1 = [17800, 0, 1800];
Start_FY2 = [12000, 1400, 1400];
Start_FY3 = [6000, -3000, 700];
Start_FY4 = [11000, 2000, 1800];
Start_FY5 = [13000, -2000, 1300];

R_cloud = 10;

dt = 0.0001;
g = 9.8;

%%
% T1
tic
t_tot = 0;
t_start = -1;
flag = false;
for t = 0:dt:30
```



```

        if Check_Outside(Start_M1, v_m, Start_FY1, t, 1.5, 5.1, 120, v_s, 180, g,
Real_center, R_cloud, featPoints) && (flag == false)
            t_start = t;
            flag = true;
            continue
        end
        if Check_Outside(Start_M1, v_m, Start_FY1, t, 1.5, 5.1, 120, v_s, 180, g,
Real_center, R_cloud, featPoints) == false && (flag)
            t_tot = t_tot + t - t_start;
            flag = false;
            disp(["start", t_start, "end", t]);
            t_start = 0;
            continue
        end
    end
end
toc

disp(t_tot);

%%
function pos = Pos_M(Start, t, v_M)
    pos(1) = Start(1) - v_M * t * Start(1) / norm(Start);
    pos(2) = Start(2) - v_M * t * Start(2) / norm(Start);
    pos(3) = Start(3) - v_M * t * Start(3) / norm(Start);
end

function pos = Pos_FY(Start, t, v_FY, theta)
    pos(1) = Start(1) + v_FY * t * cosd(theta);
    pos(2) = Start(2) + v_FY * t * sind(theta);
    pos(3) = Start(3);
end

function pos = Pos_Smoke(Start, t, tau, v_FY, theta, g)
    if t <= tau
        pos = Pos_FY(Start, t, v_FY, theta);
        return
    else
        pos_throw = Pos_FY(Start, tau, v_FY, theta);
        pos(1) = pos_throw(1) + v_FY * (t - tau) * cosd(theta);
        pos(2) = pos_throw(2) + v_FY * (t - tau) * sind(theta);
        pos(3) = pos_throw(3) - 0.5 * g * (t - tau)^2;
    end
end
end

```

```

function pos = Pos_Cloud(Start, t, tau, T, v_FY, v_s, theta, g)
    if t < T
        pos = Pos_Smoke(Start, t, tau, v_FY, theta, g);
        return
    else
        pos = Pos_Smoke(Start, T, tau, v_FY, theta, g);
        pos(3) = pos(3) - v_s * (t - T);
        if pos(3) < 0
            pos(3) = 0;
        end
    end
end

function isSuccess = Check_Outside(Start_M, v_M, Start_FY, t, tau, T, v_FY,
v_s, theta, g, Real_center, R_cloud, featPoints)
    tmp = [];
    if t < T
        isSuccess = false;
        return
    end
    pos_M = Pos_M(Start_M, t, v_M);
    pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
    e_cloud_M = pos_M - pos_cloud;
    e_real_M = pos_M - Real_center;

    tmp = [tmp; pos_M; pos_cloud];

    if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
        isSuccess = false;
        return
    end
    if (norm(pos_M - pos_cloud) < 10)
        isSuccess = true;
        return
    end
    pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);
    R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;
    tmp = [tmp; pos_proj];

    for i = 1:2
        e_Q_M = pos_M - featPoints(i, :);

```

```

        pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);

        tmp = [tmp; pos_projQ];

        if norm(pos_projQ - pos_proj) > R_proj
            isSuccess = false;
            return
        end
    end
    isSuccess = true;
end

```

程序二 问题二与问题四代码 (T2_4.m)

```

% config
clear
clc

v_m = 300;
v_s = 3;

Start_M1 = [20000, 0, 2000];
Start_M2 = [19000, 600, 2100];
Start_M3 = [18000, -600, 1900];
T_M1 = norm(Start_M1)/v_m;
T_M2 = norm(Start_M2)/v_m;
T_M3 = norm(Start_M3)/v_m;

Fake = [0,0,0];
Real_center = [0, 200, 0];
R_real = 7;
H_real = 10;
featPoints = [];
for i = 0:60:359
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0]];
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0] + [0, 0, H_real]];
end

Start_FY1 = [17800, 0, 1800];
Start_FY2 = [12000, 1400, 1400];

```

```

Start_FY3 = [6000, -3000, 700];
Start_FY4 = [11000, 2000, 1800];
Start_FY5 = [13000, -2000, 1300];

R_cloud = 10;

dt = 0.1;
dx = 5;
dh = 5;
visual_dt = 0.1;
g = 9.8;

%%
% par
tic
Tlist = dt:dt:50;
NT = numel(Tlist);
best = struct('T',Tlist,'score',zeros(NT,1),'tau',zeros(NT,1),...
             'theta',zeros(NT,1),'v_FY',zeros(NT,1));

StartFY = Start_FY5;
StartM = Start_M3;

if isempty(gcp('nocreate'))
    parpool(12);
end

parfor k = 1:NT
    T = Tlist(k);
    pos_M1 = Pos_M(StartM, T, v_m);
    pos_M2 = Pos_M(StartM, T+20, v_m);

    Pts = sampleTrianglePoints(pos_M1, pos_M2, Real_center, dx,StartFY,T);
    disp(size(Pts,1))
    tmpScore = -inf;
    tmpParam = [0 0 0 0];

    for h = max(pos_M2(3),StartFY(3)-0.5*g*T^2):dh:min(pos_M1(3),StartFY(3))
        tau = T - sqrt(2*(StartFY(3)-h)/g);
        %if ~isreal(tau) || tau<0;continue; end

        for i = 1:size(Pts,1)
            pt = [Pts(i,:),h];
            v_FY = norm(pt(1:2)-StartFY(1:2))/T;

```

```

    %if v_FY>140 || v_FY<70, continue; end

    theta = Theta(pt-StartFY);
    tot    = Total(StartM,v_m,StartFY,tau,T,v_FY,v_s,theta,...
                  g,Real_center,R_cloud,featPoints,1);

    if tot==0
        continue;
    end
    tot    = Total(StartM,v_m,StartFY,tau,T,v_FY,v_s,theta,...
                  g,Real_center,R_cloud,featPoints,0.01);

    if tot > tmpScore
        tmpScore = tot;
        tmpParam = [tau theta v_FY];
    end
end
end

best(k).score = tmpScore;
best(k).T = T;
best(k).tau  = tmpParam(1);
best(k).theta = tmpParam(2);
best(k).v_FY = tmpParam(3);
end

[~,idx] = max([best.score]);
disp(best(idx))
toc

%%
% function
function pos = Pos_M(Start, t, v_M)
    pos(1) = Start(1) - v_M * t * Start(1) / norm(Start);
    pos(2) = Start(2) - v_M * t * Start(2) / norm(Start);
    pos(3) = Start(3) - v_M * t * Start(3) / norm(Start);
    if pos(3) <= 0
        pos = [0,0,0];
    end
end

function pos = Pos_FY(Start, t, v_FY, theta)
    pos(1) = Start(1) + v_FY * t * cosd(theta);
    pos(2) = Start(2) + v_FY * t * sind(theta);
    pos(3) = Start(3);

```

```

end

function pos = Pos_Smoke(Start, t, tau, v_FY, theta, g)
    if t <= tau
        pos = Pos_FY(Start, t, v_FY, theta);
        return
    else
        pos_throw = Pos_FY(Start, tau, v_FY, theta);
        pos(1) = pos_throw(1) + v_FY * (t - tau) * cosd(theta);
        pos(2) = pos_throw(2) + v_FY * (t - tau) * sind(theta);
        pos(3) = pos_throw(3) - 0.5 * g * (t - tau)^2;
    end
end

function pos = Pos_Cloud(Start, t, tau, T, v_FY, v_s, theta, g)
    if t < T
        pos = Pos_Smoke(Start, t, tau, v_FY, theta, g);
        return
    else
        pos = Pos_Smoke(Start, T, tau, v_FY, theta, g);
        pos(3) = pos(3) - v_s * (t - T);
        if pos(3) < 0
            pos(3) = 0;
        end
    end
end

function isSuccess = Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s,
theta, g, Real_center, R_cloud, featPoints)
    if t < T
        isSuccess = false;
        return
    end
    pos_M = Pos_M(Start_M, t, v_M);
    pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
    e_cloud_M = pos_M - pos_cloud;
    e_real_M = pos_M - Real_center;

    if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
        isSuccess = false;
        return
    end
    if (norm(pos_M - pos_cloud) < 10)
        isSuccess = true;
    end
end

```

```

        return
    end
    pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);
    R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;

    for i = 1:12
        e_Q_M = pos_M - featPoints(i, :);
        pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);

        if norm(pos_projQ - pos_proj) > R_proj
            isSuccess = false;
            return
        end
    end
    isSuccess = true;
end

function status = Status(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta,
g, Real_center, R_cloud, featPoints)
    status = configureDictionary("string", "cell");
    if t < T
        return
    end
    pos_M = Pos_M(Start_M, t, v_M);
    pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
    e_cloud_M = pos_M - pos_cloud;
    e_real_M = pos_M - Real_center;

    status{"pos_M"} = pos_M;
    status{"pos_cloud"} = pos_cloud;
    status{"e_cloud_M"} = e_cloud_M;

    if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
        return
    end
    if (norm(pos_M - pos_cloud) < 10)
        return
    end
    pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);

```

```

    R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;

```

```

    status{"pos_proj"} = pos_proj;
    status{"R_proj"} = R_proj;

```

```

    tmp = [];
    for i = 1:12
        e_Q_M = pos_M - featPoints(i, :);
        pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);
        tmp = [tmp; pos_projQ];
        if norm(pos_projQ - pos_proj) > R_proj
            return
        end
    end
    status{"pos_projQ"} = tmp;
end

```

```

function tot = Total(Start_M, v_M, Start_FY, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints, dt)
    tot = 0;
    t_start = -1;
    flag = false;
    for t = T:dt:50
        if Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints)
            if flag ==false
                t_start = t;
                flag = true;
                continue
            end
        else
            if flag==true
                tot = tot + t - t_start;
                break
            end
        end
    end
end

```

```

function sampledPoints = sampleTrianglePoints(P1, P2, P3, spacing,S1,time)

```



```

p1 = P1(1:2); p2 = P2(1:2); p3 = P3(1:2);
minX = min([p1(1), p2(1), p3(1)]);
maxX = max([p1(1), p2(1), p3(1)]);
minY = min([p1(2), p2(2), p3(2)]);
maxY = max([p1(2), p2(2), p3(2)]);

[xGrid, yGrid] = meshgrid(minX:20*spacing:maxX, minY:spacing:maxY);
gridPoints = [xGrid(:), yGrid(:)]; % Nx2
inTri = false(size(gridPoints,1),1);
for i = 1:size(gridPoints,1)
    if ~pointInTriangle(gridPoints(i,:), p1, p2, p3)
        continue
    end
    v_FY = norm(gridPoints(i,1:2)-S1(1:2))/time;
    if v_FY>140 || v_FY<70, continue; end
    inTri(i) = true;
end

sampledPoints = [gridPoints(inTri,:)];
end

function inside = pointInTriangle(pt, A, B, C)
    v0 = C - A; v1 = B - A; v2 = pt - A;
    dot00 = dot(v0,v0); dot01 = dot(v0,v1); dot02 = dot(v0,v2);
    dot11 = dot(v1,v1); dot12 = dot(v1,v2);
    invDenom = 1 / (dot00*dot11 - dot01*dot01);
    u = (dot11*dot02 - dot01*dot12) * invDenom;
    v = (dot00*dot12 - dot01*dot02) * invDenom;
    inside = (u >= 0) && (v >= 0) && (u + v <= 1);
end

function theta = Theta(v)
    xy = v(1:2);
    theta = atan2d(xy(2), xy(1));
    if theta < 0
        theta = theta + 360;
    end
end
end

```

程序三 问题三代码 (T3.m)

```

% config
clear

```

```

clc

v_m = 300;
v_s = 3;

Start_M1 = [20000, 0, 2000];
Start_M2 = [19000, 600, 2100];
Start_M3 = [18000, -600, 1900];
T_M1 = norm(Start_M1)/v_m;
T_M2 = norm(Start_M2)/v_m;
T_M3 = norm(Start_M3)/v_m;

Fake = [0,0,0];
Real_center = [0, 200, 0];
R_real = 7;
H_real = 10;
featPoints = [];
for i = 0:60:359
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0]];
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0] + [0, 0, H_real]];
end

Start_FY1 = [17800, 0, 1800];
Start_FY2 = [12000, 1400, 1400];
Start_FY3 = [6000, -3000, 700];
Start_FY4 = [11000, 2000, 1800];
Start_FY5 = [13000, -2000, 1300];

R_cloud = 10;

dt = 0.1;
dtt = 0.01;
dx = 5;
dh = 5;
visual_dt = 0.1;
g = 9.8;

BestS1 = load("T5_FY2_1.mat");
BestS2 = load("T5_FY2_2.mat");
BestS1
=
[[BestS1.best.T];[BestS1.best.tau];[BestS1.best.theta];[BestS1.best.v_FY]]';
BestS2 = [[BestS2.perTResults.t2]; [BestS2.perTResults.tau2]]';

```

```

%%
%T3_stage1

StartFY = Start_FY2;
StartM = Start_M1;

nT = size(BestS1,1);
t2list_full = 0:dt:50;
Ls = -inf(nT,1);
t2s = NaN(nT,1);
tau2s = NaN(nT,1);
Ts = NaN(nT,1);

parfor T = 1:nT
    bestt = BestS1(T,:);
    period_s1 = TotalPeriod(StartM, v_m, StartFY, bestt(2), bestt(1),
bestt(4), v_s, bestt(3), g, Real_center, R_cloud, featPoints, dtt);
    localL = -inf;
    local_t2 = NaN;
    local_tau2 = NaN;
    for it2 = 1:numel(t2list_full)
        t2 = t2list_full(it2);
        tau2start = 0;
        tau2list = tau2start:dt:t2;
        for i_tau2 = 1:numel(tau2list)
            tau2 = tau2list(i_tau2);
            if abs(tau2 - bestt(2)) < 1
                continue
            end

            tmp = Pos_Cloud(StartFY, t2, tau2, t2, bestt(4), v_s, bestt(3),
g)
            if ~pointInTriangle(tmp, Pos_M(StartM, t2, v_m), Pos_M(StartM,
t2+20, v_m), Real_center)
                continue
            end

            period_s2 = TotalPeriod(StartM, v_m, StartFY, tau2, t2, bestt(4),
v_s, bestt(3), g, Real_center, R_cloud, featPoints, dtt);
            period_s3 = [-10 -10];
            L = UnionLength(period_s1, period_s2, period_s3);
            if L > localL
                localL = L;

```

```

        local_t2 = t2;
        local_tau2 = tau2;
    end
end
end
Ls(T) = localL;
t2s(T) = local_t2;
tau2s(T) = local_tau2;
Ts(T) = bestt(1);
disp([bestt(1), local_t2, local_tau2, localL])
end

perTResults = struct('T', num2cell(Ts'), 'L', num2cell(Ls'), ...
                    't2', num2cell(t2s'), 'tau2', num2cell(tau2s'));
[~, idx] = max(Ls);
globalBest = perTResults(idx);
result = Ls(idx);

%%
% T3_stage2

StartFY = Start_FY2;
StartM = Start_M3;

nT = size(BestS1,1);
t3list_full = 0:dt:50;
Ls = -inf(nT,1);
t3s = NaN(nT,1);
tau3s = NaN(nT,1);
Ts = NaN(nT,1);

parfor T = 1:nT
    best1 = BestS1(T,:);
    best2 = BestS2(T,:);
    period_s1 = TotalPeriod(StartM, v_m, StartFY, best1(2), best1(1),
best1(4), v_s, best1(3), g, Real_center, R_cloud, featPoints, dtt);
    period_s2 = TotalPeriod(StartM, v_m, StartFY, best2(2), best2(1),
best1(4), v_s, best1(3), g, Real_center, R_cloud, featPoints, dtt);
    localL = -inf;
    local_t3 = NaN;
    local_tau3 = NaN;
    for it3 = 1:numel(t3list_full)
        t3 = t3list_full(it3);

```

```

tau3start = 0;
tau3list = tau3start:dt:t3;
for i_tau3 = 1:numel(tau3list)
    tau3 = tau3list(i_tau3);
    if abs(tau3 - best1(2)) < 1
        continue
    end
    if abs(tau3 - best2(2)) < 1
        continue
    end

    tmp = Pos_Cloud(StartFY, t3, tau3, t3, best1(4), v_s, best1(3),
g)
        if ~pointInTriangle(tmp, Pos_M(StartM, t3, v_m), Pos_M(StartM,
t3+20, v_m), Real_center)
            continue
        end

        period_s3 = TotalPeriod(StartM, v_m, StartFY, tau3, t3, best1(4),
v_s, best1(3), g, Real_center, R_cloud, featPoints, dtt);
        L = UnionLength(period_s1, period_s2, period_s3);
        if L > localL
            localL = L;
            local_t3 = t3;
            local_tau3 = tau3;
        end
    end
end
Ls(T) = localL;
t3s(T) = local_t3;
tau3s(T) = local_tau3;
Ts(T) = best1(1);
disp([best1(1), best1(2), best2(1), best2(2), local_t3, local_tau3,
localL])
end

Results = struct('t1', num2cell(Ts), 'tau1', num2cell(BestS1(:,2)'), 'theta',
num2cell(BestS1(:,3)'), 'v_FY', num2cell(BestS1(:,4)'), 'L',
num2cell(Ls), ...
                't2', num2cell(BestS2(:,1)'), 'tau2',
num2cell(BestS2(:,2)'), ...
                't3', num2cell(t3s), 'tau3', num2cell(tau3s));

[~, idx] = max(Ls);
globalBest = Results(idx);

```

```

result = Ls(idx);

%%
% function
function pos = Pos_M(Start, t, v_M)
    pos(1) = Start(1) - v_M * t * Start(1) / norm(Start);
    pos(2) = Start(2) - v_M * t * Start(2) / norm(Start);
    pos(3) = Start(3) - v_M * t * Start(3) / norm(Start);
    if pos(3) <= 0
        pos = [0,0,0];
    end
end

function pos = Pos_FY(Start, t, v_FY, theta)
    pos(1) = Start(1) + v_FY * t * cosd(theta);
    pos(2) = Start(2) + v_FY * t * sind(theta);
    pos(3) = Start(3);
end

function pos = Pos_Smoke(Start, t, tau, v_FY, theta, g)
    if t <= tau
        pos = Pos_FY(Start, t, v_FY, theta);
        return
    else
        pos_throw = Pos_FY(Start, tau, v_FY, theta);
        pos(1) = pos_throw(1) + v_FY * (t - tau) * cosd(theta);
        pos(2) = pos_throw(2) + v_FY * (t - tau) * sind(theta);
        pos(3) = pos_throw(3) - 0.5 * g * (t - tau)^2;
    end
end

function pos = Pos_Cloud(Start, t, tau, T, v_FY, v_s, theta, g)
    if t < T
        pos = Pos_Smoke(Start, t, tau, v_FY, theta, g);
        return
    else
        pos = Pos_Smoke(Start, T, tau, v_FY, theta, g);
        pos(3) = pos(3) - v_s * (t - T);
        if pos(3) < 0
            pos(3) = 0;
        end
    end
end
end

```

```

function isSuccess = Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s,
theta, g, Real_center, R_cloud, featPoints)
    if t < T
        isSuccess = false;
        return
    end
    pos_M = Pos_M(Start_M, t, v_M);
    pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
    e_cloud_M = pos_M - pos_cloud;
    e_real_M = pos_M - Real_center;

    if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
        isSuccess = false;
        return
    end
    if (norm(pos_M - pos_cloud) < 10)
        isSuccess = true;
        return
    end
    pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);
    R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;

    for i = 1:12
        e_Q_M = pos_M - featPoints(i, :);
        pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);

        if norm(pos_projQ - pos_proj) > R_proj
            isSuccess = false;
            return
        end
    end
    isSuccess = true;
end

function status = Status(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta,
g, Real_center, R_cloud, featPoints)
    status = configureDictionary("string", "cell");
    if t < T
        return
    end
    pos_M = Pos_M(Start_M, t, v_M);

```

```

pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
e_cloud_M = pos_M - pos_cloud;
e_real_M = pos_M - Real_center;

status{"pos_M"} = pos_M;
status{"pos_cloud"} = pos_cloud;
status{"e_cloud_M"} = e_cloud_M;

if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
    return
end
if (norm(pos_M - pos_cloud) < 10)
    return
end
pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);
R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;

status{"pos_proj"} = pos_proj;
status{"R_proj"} = R_proj;

tmp = [];
for i = 1:120
    e_Q_M = pos_M - featPoints(i, :);
    pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);
    tmp = [tmp; pos_projQ];
    if norm(pos_projQ - pos_proj) > R_proj
        return
    end
end
status{"pos_projQ"} = tmp;
end

function tot = Total(Start_M, v_M, Start_FY, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints, dt)
    tot = 0;
    t_start = -1;
    flag = false;
    for t = T:dt:60
        if Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints)
            if flag ==false

```



```

        t_start = t;
        flag = true;
        continue
    end
else
    if flag==true
        tot = tot + t - t_start;
        break
    end
end
end
end
end

```

```

function period = TotalPeriod(Start_M, v_M, Start_FY, tau, T, v_FY, v_s,
theta, g, Real_center, R_cloud, featPoints, dt)
    tot = 0;
    Start = -1;
    End = -1;
    flag = false;
    for t = T:dt:60
        if Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints)
            if flag ==false
                Start = t;
                End = Start;
                flag = true;
                continue
            end
        else
            if flag==true
                End = t;
                break
            end
        end
    end
    period = [Start, End];
end

```

```

function sampledPoints = sampleTrianglePoints(P1, P2, P3, spacing,S1,time)

p1 = P1(1:2); p2 = P2(1:2); p3 = P3(1:2);
minX = min([p1(1), p2(1), p3(1)]);
maxX = max([p1(1), p2(1), p3(1)]);

```

```

minY = min([p1(2), p2(2), p3(2)]);
maxY = max([p1(2), p2(2), p3(2)]);

[xGrid, yGrid] = meshgrid(minX:spacing:maxX, minY:spacing:maxY);
gridPoints = [xGrid(:), yGrid(:)];           % Nx2
inTri = false(size(gridPoints,1),1);
for i = 1:size(gridPoints,1)
    if ~pointInTriangle(gridPoints(i,:), p1, p2, p3)
        continue
    end
    v_FY = norm(gridPoints(i,1:2)-S1(1:2))/time;
    if v_FY>140 || v_FY<70, continue; end
    inTri(i) = true;
end

sampledPoints = [gridPoints(inTri,:)];
end

function inside = pointInTriangle(pt, A, B, C)
    v0 = C - A; v1 = B - A; v2 = pt - A;
    dot00 = dot(v0,v0); dot01 = dot(v0,v1); dot02 = dot(v0,v2);
    dot11 = dot(v1,v1); dot12 = dot(v1,v2);
    invDenom = 1 / (dot00*dot11 - dot01*dot01);
    u = (dot11*dot02 - dot01*dot12) * invDenom;
    v = (dot00*dot12 - dot01*dot02) * invDenom;
    inside = (u >= 0) && (v >= 0) && (u + v <= 1);
end

function theta = Theta(v)
    xy = v(1:2);
    theta = atan2d(xy(2), xy(1));
    if theta < 0
        theta = theta + 360;
    end
end

function L = UnionLength(I1, I2, I3)

    intervals = [I1; I2; I3];

    intervals = sortrows(intervals, 1);

    merged = intervals(1,:);
    for i = 2:3

```

```

        cur = intervals(i,:);
        if cur(1) <= merged(end,2)
            merged(end,2) = max(merged(end,2), cur(2));
        else
            merged = [merged; cur];
        end
    end
end

L = sum(merged(:,2) - merged(:,1));
End

```

程序四 问题五代码 (T5.m)

```

% config
clear
clc

v_m = 300;
v_s = 3;

Start_M1 = [20000, 0, 2000];
Start_M2 = [19000, 600, 2100];
Start_M3 = [18000, -600, 1900];
T_M1 = norm(Start_M1)/v_m;
T_M2 = norm(Start_M2)/v_m;
T_M3 = norm(Start_M3)/v_m;

Fake = [0,0,0];
Real_center = [0, 200, 0];
R_real = 7;
H_real = 10;
featPoints = [];
for i = 0:60:359
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0]];
    featPoints = [featPoints; Real_center + [R_real*cosd(i), R_real*sind(i),
0] + [0, 0, H_real]];
end

Start_FY1 = [17800, 0, 1800];
Start_FY2 = [12000, 1400, 1400];
Start_FY3 = [6000, -3000, 700];
Start_FY4 = [11000, 2000, 1800];

```

```

Start_FY5 = [13000, -2000, 1300];

R_cloud = 10;

dt = 0.1;
dtt = 0.01;
dx = 5;
dh = 5;
visual_dt = 0.1;
g = 9.8;

BestFY1 = load("T5_FY1.mat");
BestFY1 = [[BestFY1.Results.theta]', [BestFY1.Results.v_FY]',
[BestFY1.Results.t1]', [BestFY1.Results.tau1]', [BestFY1.Results.t2]',
[BestFY1.Results.tau2]', [BestFY1.Results.t3]', [BestFY1.Results.tau3]'];
BestFY2 = load("T5_FY2.mat");
BestFY2 = [[BestFY2.Results.theta]', [BestFY2.Results.v_FY]',
[BestFY2.Results.t1]', [BestFY2.Results.tau1]', [BestFY2.Results.t2]',
[BestFY2.Results.tau2]', [BestFY2.Results.t3]', [BestFY2.Results.tau3]'];
BestFY3 = load("T5_FY3.mat");
BestFY3 = [[BestFY3.Results.theta]', [BestFY3.Results.v_FY]',
[BestFY3.Results.t1]', [BestFY3.Results.tau1]', [BestFY3.Results.t2]',
[BestFY3.Results.tau2]', [BestFY3.Results.t3]', [BestFY3.Results.tau3]'];
BestFY4 = load("T5_FY4.mat");
BestFY4 = [[BestFY4.Results.theta]', [BestFY4.Results.v_FY]',
[BestFY4.Results.t1]', [BestFY4.Results.tau1]', [BestFY4.Results.t2]',
[BestFY4.Results.tau2]', [BestFY4.Results.t3]', [BestFY4.Results.tau3]'];
BestFY5 = load("T5_FY5.mat");
BestFY5 = [[BestFY5.Results.theta]', [BestFY5.Results.v_FY]',
[BestFY5.Results.t1]', [BestFY5.Results.tau1]', [BestFY5.Results.t2]',
[BestFY5.Results.tau2]', [BestFY5.Results.t3]', [BestFY5.Results.tau3]'];

%%
Maxt = -1;
Maxresult = 0;
t1 = 72;
t2 = 73;
t3 = 207;
t4 = 118;
t5 = 197;
for t = 1:size(BestFY5, 1)
    bestFY5 = BestFY5(t,:);

```

```

    preperiod1 = [];
    preperiod2 = [];
    preperiod3 = [];
    preperiod1 = [preperiod1; BestFY1(t1,9:10);BestFY1(t1,11:12);BestFY1(t1,13:14)];
    preperiod2 = [preperiod2; BestFY2(t2,9:10)];
    preperiod1 = [preperiod1; BestFY2(t2,11:12)];
    preperiod3 = [preperiod3; BestFY2(t2,13:14)];
    preperiod3 = [preperiod3; BestFY3(t3,9:10)];
    preperiod1 = [preperiod1; BestFY3(t3,11:12)];
    preperiod2 = [preperiod2; BestFY3(t3,13:14)];
    preperiod2 = [preperiod2; BestFY4(t4,9:10)];
    preperiod1 = [preperiod1; BestFY4(t4,11:12)];
    preperiod3 = [preperiod3; BestFY4(t4,13:14)];

    period1 = TotalPeriod(Start_M3, v_m, Start_FY5, bestFY5(4), bestFY5(3),
    bestFY5(2), v_s, bestFY5(1), g, Real_center, R_cloud, featPoints, 0.002);
    period2 = TotalPeriod(Start_M1, v_m, Start_FY5, bestFY5(6), bestFY5(5),
    bestFY5(2), v_s, bestFY5(1), g, Real_center, R_cloud, featPoints, 0.002);
    period3 = TotalPeriod(Start_M2, v_m, Start_FY5, bestFY5(8), bestFY5(7),
    bestFY5(2), v_s, bestFY5(1), g, Real_center, R_cloud, featPoints, 0.002);

    BestFY5(t,9:10) = period1;
    BestFY5(t,11:12) = period2;
    BestFY5(t,13:14) = period3;
    result = UnionLength([preperiod1; period2])+ UnionLength([preperiod2;
    period3]) + UnionLength([preperiod3; period1]);
    if result > Maxresult
        Maxt = t/10;
        Maxresult = result;
        disp([Maxt, Maxresult])
    end
end

%%
t1 = 72;
t2 = 73;
t3 = 207;
t4 = 118;
t5 = 197;
% format = [theta, v_FY, t1, tau1, t2, tau2, t3, tau3, p1_left, p1_right,
p2_left, p2_right, p3_left, p3_right];
bestfy1 = load("T5_FY1n.mat").BestFY1(t1,:);
bestfy2 = load("T5_FY2n.mat").BestFY2(t2,:);

```

```

bestfy3 = load("T5_FY3n.mat").BestFY3(t3,:);
bestfy4 = load("T5_FY4n.mat").BestFY4(t4,:);
bestfy5 = load("T5_FY5n.mat").BestFY5(t5,:);
Result = [bestfy1;bestfy2;bestfy3;bestfy4;bestfy5];
%%

% function
function pos = Pos_M(Start, t, v_M)
    pos(1) = Start(1) - v_M * t * Start(1) / norm(Start);
    pos(2) = Start(2) - v_M * t * Start(2) / norm(Start);
    pos(3) = Start(3) - v_M * t * Start(3) / norm(Start);
    if pos(3) <= 0
        pos = [0,0,0];
    end
end

function pos = Pos_FY(Start, t, v_FY, theta)
    pos(1) = Start(1) + v_FY * t * cosd(theta);
    pos(2) = Start(2) + v_FY * t * sind(theta);
    pos(3) = Start(3);
end

function pos = Pos_Smoke(Start, t, tau, v_FY, theta, g)
    if t <= tau
        pos = Pos_FY(Start, t, v_FY, theta);
        return
    else
        pos_throw = Pos_FY(Start, tau, v_FY, theta);
        pos(1) = pos_throw(1) + v_FY * (t - tau) * cosd(theta);
        pos(2) = pos_throw(2) + v_FY * (t - tau) * sind(theta);
        pos(3) = pos_throw(3) - 0.5 * g * (t - tau)^2;
    end
end

function pos = Pos_Cloud(Start, t, tau, T, v_FY, v_s, theta, g)
    if t < T
        pos = Pos_Smoke(Start, t, tau, v_FY, theta, g);
        return
    else
        pos = Pos_Smoke(Start, T, tau, v_FY, theta, g);
        pos(3) = pos(3) - v_s * (t - T);
        if pos(3) < 0
            pos(3) = 0;
        end
    end
end

```

```

    end
end

function isSuccess = Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s,
theta, g, Real_center, R_cloud, featPoints)
    if t < T
        isSuccess = false;
        return
    end
    pos_M = Pos_M(Start_M, t, v_M);
    pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
    e_cloud_M = pos_M - pos_cloud;
    e_real_M = pos_M - Real_center;

    if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
        isSuccess = false;
        return
    end
    if (norm(pos_M - pos_cloud) < 10)
        isSuccess = true;
        return
    end
    pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);
    R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;

    for i = 1:12
        e_Q_M = pos_M - featPoints(i, :);
        pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);

        if norm(pos_projQ - pos_proj) > R_proj
            isSuccess = false;
            return
        end
    end
    isSuccess = true;
end

function status = Status(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta,
g, Real_center, R_cloud, featPoints)
    status = configureDictionary("string", "cell");
    if t < T

```

```

        return
    end
    pos_M = Pos_M(Start_M, t, v_M);
    pos_cloud = Pos_Cloud(Start_FY, t, tau, T, v_FY, v_s, theta, g);
    e_cloud_M = pos_M - pos_cloud;
    e_real_M = pos_M - Real_center;

    status{"pos_M"} = pos_M;
    status{"pos_cloud"} = pos_cloud;
    status{"e_cloud_M"} = e_cloud_M;

    if (dot(e_cloud_M, e_real_M) < 0) && (norm(pos_M - pos_cloud) > 10)
        return
    end
    if (norm(pos_M - pos_cloud) < 10)
        return
    end
    pos_proj = pos_M - e_cloud_M * dot(e_real_M,
e_cloud_M)/norm(e_cloud_M)/norm(e_cloud_M);
    R_proj = norm(pos_proj - pos_M)/(sqrt(norm(pos_M - pos_cloud)*norm(pos_M
- pos_cloud)-100)) * R_cloud;

    status{"pos_proj"} = pos_proj;
    status{"R_proj"} = R_proj;

    tmp = [];
    for i = 1:120
        e_Q_M = pos_M - featPoints(i, :);
        pos_projQ = pos_M - e_Q_M * dot(e_real_M, e_cloud_M)/dot(e_Q_M,
e_cloud_M);
        tmp = [tmp; pos_projQ];
        if norm(pos_projQ - pos_proj) > R_proj
            return
        end
    end
    status{"pos_projQ"} = tmp;
end

function tot = Total(Start_M, v_M, Start_FY, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints, dt)
    tot = 0;
    t_start = -1;
    flag = false;
    for t = T:dt:60

```



```

        if Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints)
            if flag ==false
                t_start = t;
                flag = true;
                continue
            end
        else
            if flag==true
                tot = tot + t - t_start;
                break
            end
        end
    end
end
end
end

```

```

function period = TotalPeriod(Start_M, v_M, Start_FY, tau, T, v_FY, v_s,
theta, g, Real_center, R_cloud, featPoints, dt)
    tot = 0;
    Start = -1;
    End = -1;
    flag = false;
    for t = T:dt:60
        if Check(Start_M, v_M, Start_FY, t, tau, T, v_FY, v_s, theta, g,
Real_center, R_cloud, featPoints)
            if flag ==false
                Start = t;
                End = Start;
                flag = true;
                continue
            end
        else
            if flag==true
                End = t;
                break
            end
        end
    end
    period = [Start, End];
end
end

```

```

function sampledPoints = sampleTrianglePoints(P1, P2, P3, spacing,S1,time)

```

```

p1 = P1(1:2); p2 = P2(1:2); p3 = P3(1:2);
minX = min([p1(1), p2(1), p3(1)]);
maxX = max([p1(1), p2(1), p3(1)]);
minY = min([p1(2), p2(2), p3(2)]);
maxY = max([p1(2), p2(2), p3(2)]);

[xGrid, yGrid] = meshgrid(minX:spacing:maxX, minY:spacing:maxY);
gridPoints = [xGrid(:), yGrid(:)]; % Nx2
inTri = false(size(gridPoints,1),1);
for i = 1:size(gridPoints,1)
    if ~pointInTriangle(gridPoints(i,:), p1, p2, p3)
        continue
    end
    v_FY = norm(gridPoints(i,1:2)-S1(1:2))/time;
    if v_FY>140 || v_FY<70, continue; end
    inTri(i) = true;
end

sampledPoints = [gridPoints(inTri,:)];
end

function inside = pointInTriangle(pt, A, B, C)
    v0 = C - A; v1 = B - A; v2 = pt - A;
    dot00 = dot(v0,v0); dot01 = dot(v0,v1); dot02 = dot(v0,v2);
    dot11 = dot(v1,v1); dot12 = dot(v1,v2);
    invDenom = 1 / (dot00*dot11 - dot01*dot01);
    u = (dot11*dot02 - dot01*dot12) * invDenom;
    v = (dot00*dot12 - dot01*dot02) * invDenom;
    inside = (u >= 0) && (v >= 0) && (u + v <= 1);
end

function theta = Theta(v)
    xy = v(1:2);
    theta = atan2d(xy(2), xy(1));
    if theta < 0
        theta = theta + 360;
    end
end

function L = UnionLength(II)
    II = sortrows(II,1);
    m = II(1,:);
    for i = 2:size(II,1)
        if II(i,1) <= m(end,2)

```

```
        m(end,2) = max(m(end,2),II(i,2));  
    else  
        m = [m; II(i,:)];  
    end  
end  
L = sum(m(:,2)-m(:,1));
```