# 一、 问题重述

## 1.1 问题背景

烟幕干扰弹通过化学燃烧或者爆炸产生气溶胶云团，这片云团能够有效吸收、散射或反射如红外线、雷达波等电磁波，从而遮蔽后方目标，干扰敌方侦察、观瞄和制导系统，使其无法精确锁定和攻击真实目标，降低敌对导弹打击的损失。同时由于烟幕弹具有成本低、效益高的特点，其在军事方面得到了广泛的发展与应用。

图 1: 现实中的烟幕弹运用

本文针对无人机投放烟幕弹干扰导弹定位这一情景，在烟幕弹的性能、导弹以及无人机的运行参数、初始坐标等数据已经给定的条件下，需要综合考虑多个物体的运动轨迹、烟雾弹的扩散模型对不同情况下无人机投放烟幕弹的策略进行时空上的优化，以达到有效遮挡时间最长这一目标。

## 1.2 问题提出

现已经知道了 3 枚导弹、5 架无人机以及真假目标的初始坐标和一些运动参数数据，同时给出了烟幕弹云团的有效遮挡范围及时间，本文需要解决以下问题：

**问题一、二:** 针对单无人机、单烟幕弹干扰单枚导弹的情况，问题一需要考虑运动学以及烟幕弹的效能，分析几何关系，找出有效屏蔽时满足的条件，建立数学模型，算出一组给定的数据下的有效遮挡的时长。问题二从特殊情况拓展到了一般情况，需要综合问题一建立的几何模型，以最大化有效遮挡时长为目标建立优化模型，求解单机单弹的最优投放策略。

**问题三:** 针对单无人机、多烟幕弹干扰单枚导弹的情况，需要规划多枚烟幕弹的投放时序以延长总遮蔽时间。

**问题四:** 不同于问题三单机多弹的情况，同样是投放三枚烟幕弹，但是情况变成了三架无人机各自投放一枚烟幕弹对同一枚导弹进行干扰，需要考虑三架无人机在空间上的协调，给出最优的投放策略。

**问题五:** 是一个多机多弹多目标的场景，需要同时解决任务分配（哪架飞机干扰哪枚导弹）和资源调度（每架飞机投几枚弹、何时投）问题，以实现全局最优的有效遮蔽效果。

中国大学生在线 dxs.moe.gov.cn

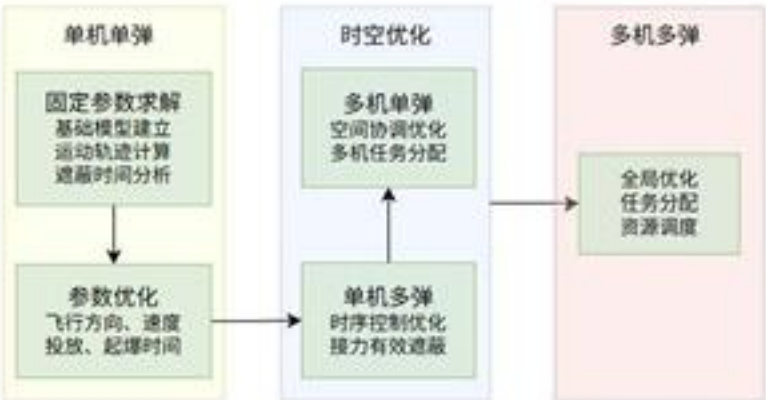



图 2: 各问题关系图

# 二、 问题分析

## 2.1 问题一的分析

问题一是基础情况，要求计算在给定参数下烟幕干扰弹对导弹 M1 的有效遮蔽时长，首先可以利用题目给定的参数算出烟幕弹起爆时刻烟幕弹和导弹的位置作为后续模型的初值，之后需要根据空间几何的知识找到满足使有效遮挡这一要求成立的几何关系，最后就可以根据初值和该几何关系从起爆时刻进行变步长搜索找出满足有效遮挡的时间范围。

## 2.2 问题二的分析

问题二不同于问题一在给定的参数下求解，而是需要延用问题一的分析，把无人机航向速度、烟幕弹投放、引爆时间四个参数作为决策变量，以最大化有效遮挡时间作为目标函数建立单机单弹情况下的策略优化模型，在满足一些物理约束、运动学约束、边界条件的情况下求解出烟幕弹的最优投放策略。

## 2.3 问题三的分析

问题三基于问题二的优化模型，需协调多枚烟幕弹的投放与起爆时序，避免时间重叠或遗漏，通过时序优化与协同控制延长总有效遮蔽时间。由于三枚烟幕弹是由同一无人机发射，其在空间上位置相差不远，考虑有效遮挡时，可能会存在单个云团不能有效遮挡，而多个云团耦合时会形成有效遮挡的情况。

## 2.4 问题四的分析

问题四是多机单弹拦截同一目标的情况，与问题三的单机多弹模式不同，问题四的核心在于多无人机之间的空间协同和时间协调，这里不需要考虑投弹时间间隔，但是三枚导弹爆炸产生的云团应该避免在空间上过度重叠，同时时间上要保持连续。这需要建立多烟幕弹协同遮蔽的判定模型，分析多个云团在空间上的遮蔽效果叠加关系，在考虑各无人机独立决策的前提下，通过协同目标函数确保整体遮蔽效果最大化。

### 2.5 问题五的分析

问题五是最困难的多机多弹多目标的情况，需要确定哪架无人机去拦截哪枚导弹，每个无人机需要投多少枚烟幕弹以及总体投放的策略，直接考虑所有的决策变量，则变量的维度太大，很难直接快速的计算，因此需要把问题简化，拆分成多个步骤，即要先确定资源分配的策略，再决定烟幕弹投放的策略，以提高计算效率。

## 三、 模型假设

1. 将导弹、无人机、烟幕弹、烟幕云团中心以及假目标均视为质点进行运动学分析。
2. 不考虑空气阻力、风速等环境因素对无人机和导弹飞行的影响。
3. 烟幕云团形成后，球心在垂直方向以 $3\mathrm{m}\cdot\mathrm{s}^{-1}$ 的速度匀速下沉。
4. 无人机受领任务和调整飞行姿态的时间忽略不计。
5. 烟幕弹起爆瞬时形成一个半径为 10 米的理想球状云团，且在有效时间内（20 秒）其半径和遮蔽能力保持不变。

## 四、 符号说明

| 符号 | 含义 | 单位 |
|---|---|---|
| $P_{m,i}(t)$ | 导弹 $i$ 在时刻 $t$ 的位置向量 | m |
| $v_{m,i}$ | 导弹 $i$ 的速度向量 | m/s |
| $v_m$ | 导弹速度大小 | m/s |
| $P_{fy,j}(t)$ | 无人机 $j$ 在时刻 $t$ 的位置向量 | m |
| $v_{fy,j}$ | 无人机 $j$ 的速度向量 | m/s |
| $\theta_j$ | 无人机 $j$ 的航向角（与 $x$ 轴夹角） | rad |
| $P_{s,jk}(t)$ | 无人机 $j$ 投放的第 $k$ 枚烟幕弹的位置 | m |
| $t_{d,jk}$ | 烟幕弹投放时间 | s |
| $t_{b,jk}$ | 烟幕弹起爆时间 | s |
| $P_{c,jk}(t)$ | 烟幕弹爆炸后云团中心的位置 | m |
| $v_{\mathrm{sink}}$ | 云团下沉速度 | m/s |
| $g$ | 重力加速度向量 | $\mathrm{m/s^2}$ |
| $\Omega$ | 真实目标圆柱体表面区域 | – |
| $Bool(t)$ | 遮蔽状态布尔函数（1 为遮蔽，0 为未遮蔽） | – |
| $t_{in}$ | 有效遮蔽开始时刻 | s |
| $t_{out}$ | 有效遮蔽结束时刻 | s |

续下页



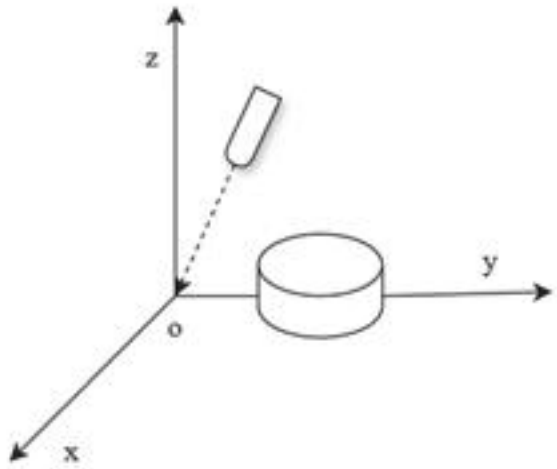
dxs.moe.gov.cn

表 1 符号说明（续）

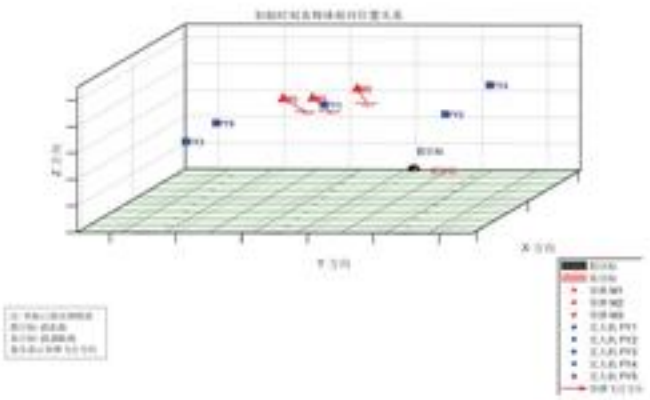
| 符号 | 含义 | 单位 |
|---|---|---|
| $T$ | 总有效遮蔽时长 | s |
| $\Delta t_b$ | 烟幕弹投放至起爆的时间间隔 | s |
| $y_{ij}$ | 无人机 $j$ 投向导弹 $i$ 的烟幕弹数量 | – |

## 五、 模型准备

根据题意，本文的所有求解都是建立在空间直角坐标系的基础上进行的，其中假目标视为一个质点作为坐标系原点，水平面作为坐标系的 $xOy$ 面，竖直向上的方向作为 $z$ 轴的正方向。现可以绘制出本文的空间直角坐标系，同时利用 matlab 对各导弹与各无人机的初始位置进行可视化处理：




(a) 坐标系示意图



(b) 初始坐标可视化图

图 3: 无人机投弹干扰导弹示意图

通过观察初始时刻的各导弹与无人机的位置，可以发现此时无人机群在导弹群前方按照扇形分布，形成了一个闪现拦截带，扇形的角度约为 30°，同时距离目标最远的导弹 M1 的导弹来向与无人机 FY1 几乎重合，这说明 FY1 后续可能对导弹迎面投放烟幕弹，形成烟墙进行阻拦。同时在高度 $z$ 上，无人机群成梯度配置，可以形成低位前伸烟幕拦截网。

### 5.1 运动轨迹方程

要计算后面的有效遮挡时间，需要根据运动学知识求解出各个导弹、各个无人机及烟幕弹和云团的运动轨迹，再利用几何关系去判断是否满足有效遮蔽的条件，下面给出四个物体的运动轨迹函数推导。

中国大学生在线 dxs.moe.gov.cn

### 5.1.1 导弹轨迹

根据题意，所有的导弹均以 300m/s 的速度做匀速直线运动，速度的方向沿着导弹与原点的连线方向，利用 $\mathbf{v}_{m,i}$ 表示导弹 $i$ 的速度向量，$v_m$ 表示导弹的速度大小，其值为 300m/s,$\mathbf{P}_{m,i}(0)$ 表示导弹 $i$ 在初始时刻的位置向量，方向由原点指向导弹，$i$ 的取值为 1，2，3。

根据方向向量的计算方法可知导弹 $i$ 的速度向量可以表示为：

$$\mathbf{v}_{m,i} = \frac{-\mathbf{P}_{m,i}(0)}{\|\mathbf{P}_{m,i}(0)\|} \cdot v_m \quad (i = 1, 2, 3) \tag{5.1}$$

知道了速度向量后根据匀速直线运动的性质可以算出 $t$ 时刻导弹 $i$ 的位置函数即：

$$\mathbf{P}_{m,i}(t) = \mathbf{P}_{m,i}(0) + \mathbf{v}_{m,i} \cdot t \quad (i = 1, 2, 3) \tag{5.2}$$

### 5.1.2 无人机轨迹

在无人机被指派任务后，无人机将在等高度上沿着确定的航向做匀速直线运动，$z$ 方向上的速度分量为 0，因此只需要知道无人机的飞行航向角就可以确定出无人机的速度向量，假设无人机 $j$ 的航线与 $x$ 轴的正方向的夹角为 $\theta_j$，那么速度向量为：

$$\mathbf{v}_{fy,j} = (v_{fy,j}\cos\theta_j, v_{fy,j}\sin\theta_j, 0) \quad (j = 1, \ldots, 5) \tag{5.3}$$

同样的有了无人机的速度向量和初始的位置就能得到无人机的位置函数：

$$\mathbf{P}_{fy,j}(t) = \mathbf{P}_{fy,j}(0) + \mathbf{v}_{fy,j} \cdot t \quad (j = 1, \ldots, 5) \tag{5.4}$$

### 5.1.3 烟幕弹轨迹

无人机在飞行到指定地点时投放烟幕弹，考虑到重力和惯性的双重作用，烟幕弹投放后做抛体运动，忽略空气阻力的前提下，烟幕弹速度在 $xOy$ 平面上的投影由于惯性的作用，将维持在此前无人机的速度 $v_{fy,j}$，而在 $z$ 方向上烟幕弹将在重力加速度 $g$ 下做自由落体运动，将第 $j$ 个无人机发射出的第 $k$ 个烟幕弹的速度定义为 $\mathbf{v}_{\mathrm{sjk}}$，则 $\mathbf{v}_{\mathrm{sjk}}$ 可以表示为：

$$\mathbf{v}_{\mathrm{sjk}} = (v_{fy,j}\cos\theta_j, v_{fy,j}\sin\theta_j, v_{jk,z}) \quad (j = 1, \ldots, 5)(k = 1, 2, 3) \tag{5.5}$$

其中 $v_{jk,z}$ 表示自由落体的速度大小，其值的大小为 $g(t - t_{d,jk})$,$t_{d,jk}$ 是该烟幕弹的投放时间。假设在 $t_{d,jk}$ 时刻，烟幕弹的坐标为 $\mathbf{P}_{s,jk}(t_d)$，由于烟幕弹在抛出瞬间的位移可以忽略不记，所以在投放时烟幕弹的坐标等价于无人机的坐标 $\mathbf{P}_{fy,j}(t_d)$，因此烟幕弹

的位置函数可以表示为：

$$\mathbf{P}_{\mathrm{s,jk}}(t)=\mathbf{P}_{fy,j}(t_{d,jk})+\mathbf{v}_{fy,j}\cdot(t-t_{\mathrm{d,jk}})+\frac{1}{2}\mathbf{g}\cdot(t-t_{\mathrm{d,jk}})^2 \quad t\geq t_{\mathrm{d,jk}} \quad (j=1,\ldots,5)(k=1,2,3) \tag{5.6}$$

其中 $\mathbf{g}=(0,0,-g)$，$g$ 的大小取为 $9.8\,\mathrm{m/s^2}$。

### 5.1.4 云团中心轨迹

烟幕弹爆炸后形成的云团将以团以 3 m/s 的速度匀速下沉，记作 $v_{sank}$，要计算云团的轨迹必须要先计算出云团的起爆位置，而计算起爆位置必须要知道起爆时间，

$$t_{b,jk}=t_{d,jk}+\Delta t_{b,jk} \quad (j=1,2...3)(k=1,2,3) \tag{5.7}$$

其中 $t_{b,jk}$ 表示第 $j$ 架无人机投出的第 $k$ 枚烟幕弹起爆时间,$\Delta t_{b,jk}$ 表示该烟幕弹投放至起爆的时间间隔。起爆点位置即烟幕弹在在起爆时刻的轨迹点，即 $\mathbf{P}_{\mathrm{bjk}}=\mathbf{P}_{\mathrm{sjk}}(t_{\mathrm{bjk}})$。所以云团的位置函数为：

$$\mathbf{P}_{\mathrm{c,jk}}(t)=(P_{\mathrm{b,jk},x},P_{\mathrm{b,jk},y},P_{\mathrm{b,jk},z}-v_{\mathrm{sink}}\cdot(t-t_{\mathrm{b,jk}})) \quad t\geq t_{\mathrm{b,jk}} \quad (j=1,\ldots,5)(k=1,2,3) \tag{5.8}$$

根据上面的四个位置函数就可以计算出各个时刻各物体的空间位置从而进行后面的计算。

## 5.2 有效遮挡判定式

如果把导弹当作一个散发光线的点光源（发射不可见的电磁波），其产生的光线能够覆盖周围空间，要让圆柱形状的真实目标不被导弹探测，那么导弹产生的光线就不能照射到圆柱上，所以有效遮挡的情境即烟幕弹爆炸生成的云团能够屏蔽所有照向圆柱的光线，使圆柱得到很好的隐匿。可以想象到点光源射向一个球体会在球后生成一个圆锥形的部分区域，只要圆柱能够完全落在这一区域内，那么就能有效地屏蔽导弹的定位。

根据示意图，下面将对有效遮挡这一条件进行数学说明。给定一条由点 $\mathbf{P}_1$ 和 $\mathbf{P}_2$ 定义的线段，其中 $\mathbf{P}_1$ 表示的是导弹在某一时刻的位置，$\mathbf{P}_2$ 表示真实目标圆柱上的任意一点，以及此时的云团球体，其球心为 $\mathbf{C}$ ，半径为 $r$ ，判断是否有效遮挡需要判断此线段是否与球体相交。

一个球体 $S(C,r)$ 由其球心 $\mathbf{C}$ 和半径 $r$ 唯一定义。空间中任意一点 $P$ 位于球面上当且仅当满足：

$$|\mathbf{P}-\mathbf{C}|=r \tag{5.9}$$

为避免平方根运算，通常使用平方形式：

$$(\mathbf{P}-\mathbf{C})\cdot(\mathbf{P}-\mathbf{C})=r^2 \tag{5.10}$$

式中 $\cdot$ 表示向量点积运算。这个方程表达了球面上任意点 $P$ 到球心 $\mathbf{C}$ 的距离平方

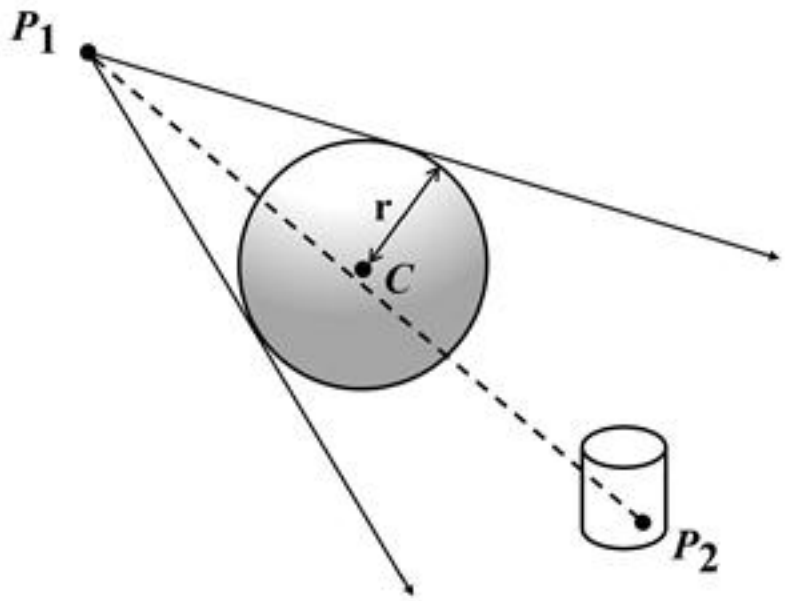



图 4: 有效遮挡示意图

等于半径平方的几何关系。

同时线段 $P_1P_2$ 上的任意点 $\mathbf{P}(t)$ 可以表示为:

$$\mathbf{P}(t) = \mathbf{P}_1 + s(\mathbf{P}_2 - \mathbf{P}_1), \quad s \in [0, 1] \tag{5.11}$$

其中: $s = 0$ 时, $\mathbf{P}(0) = \mathbf{P}_1$ (线段起点); $s = 1$ 时, $\mathbf{P}(1) = \mathbf{P}_2$ (线段终点); $0 < s < 1$ 时, $\mathbf{P}(t)$ 在线段内部; 当 $s < 0$ 或 $s > 1$ 时, $P(t)$ 位于线段的延长线上。

令 $\mathbf{d} = \mathbf{P}_2 - \mathbf{P}_1$, 将线段参数方程 (5.11) 代入球面方程(5.10), 得到:

$$(\mathbf{P}_1 + s\mathbf{d} - \mathbf{C}) \cdot (\mathbf{P}_1 + s\mathbf{d} - \mathbf{C}) = r^2 \tag{5.12}$$

引入向量 $\mathbf{OC} = \mathbf{P}_1 - \mathbf{C}$, 表示从球心指向线段起点的向量, 则方程简化为:

$$(\mathbf{OC} + s\mathbf{d}) \cdot (\mathbf{OC} + s\mathbf{d}) = r^2 \tag{5.13}$$

将上面的公式展开得:

$$s^2(\mathbf{d} \cdot \mathbf{d}) + 2s(\mathbf{d} \cdot \mathbf{OC}) + (\mathbf{OC} \cdot \mathbf{OC}) - r^2 = 0 \tag{5.14}$$

这是一个关于参数 $s$ 的一元二次方程, 标准形式为:

$$as^2 + bs + c = 0 \tag{5.15}$$

其中系数为:

$$a = \mathbf{d} \cdot \mathbf{d} = |\mathbf{d}|^2 \qquad b = 2(\mathbf{OC} \cdot \mathbf{d}) \qquad c = \mathbf{OC} \cdot \mathbf{OC} - r^2 = |\mathbf{OC}|^2 - r^2 \tag{5.16}$$

系数 $a$ 恒为正 (除非 $\mathbf{d} = \mathbf{0}$, 即线段退化为点), $c$ 的符号取决于 $P_1$ 是否在球体内。

利用上面算出的参数表达式，下面根据一元二次方程的判别式进行根的讨论，计算判别式：

$$\Delta = b^2 - 4ac \tag{5.17}$$

根据判别式的值判断相交情况：$\Delta < 0$：无实数解，直线与球体无交点，线段肯定不与球体相交；$\Delta = 0$：一个实数解，直线与球体相切，有一个交点；$\Delta > 0$：两个实数解，直线贯穿球体，有两个交点。

仅考虑直线与球体的交点的个数不足以说明云团有效地遮挡了导弹的光线，例如当导弹的相对位置运行到了云团和真实目标之间时，线段的延长线仍然可能会与云团有交点，但是却不能遮挡导弹，因此还需要考虑交点与线段 $P_1P_2$ 的位置关系，只有当两个实数解：

$$s_1 = \frac{-b - \sqrt{\Delta}}{2a} \qquad s_2 = \frac{-b + \sqrt{\Delta}}{2a} \tag{5.18}$$

满足以下其中一个条件：至少一个交点在线段上，此时 $s_1 \in [0,1]$ 或 $s_2 \in [0,1]$，则线段与球体表面相交。这对应于导弹视线从烟幕云团中穿过的情形；线段完全在球体内部，此时 $s_1 < 0$ 且 $s_2 > 1$，则整个线段都在球体内。这对应于导弹已经完全进入烟幕内部，视线被完全遮蔽的情形。才能说明该点被有效遮挡。而要使整个圆柱被有效遮挡，则圆柱上的任意一点 $P_2$ 都应该满足该条件，所以**有效遮挡判别式**用数学公式表达为：

$$\forall P_2 \in \Omega = \{(x,y,z) \in \mathbb{R}^3 \mid x^2 + (y-200)^2 \le 7^2, \quad 0 \le z \le 10\}, \quad \overline{P_1P_2} \cap C(P_c) \neq \varnothing \tag{5.19}$$

其中：$\Omega = \{(x,y,z) \in \mathbb{R}^3 \mid x^2 + (y-200)^2 \le 7^2, \quad 0 \le z \le 10\}$ 表示圆柱的整个表面。$\overline{P_1P_2}$ 表示线段，$C(P_c)$ 表示以 $C$ 为球心，半径为 10m 的球体。

### 5.2.1 有效遮蔽时间区间说明

由于导弹和云团的运动速度是恒定的，所以遮蔽情况不会发生突变，只会在一段时间内有效遮蔽，一段时间内没有有效遮蔽。记进入有效遮挡的时刻为 $t_{\text{in}}^k$，离开有效遮挡的时间为 $t_{\text{out}}^k$，假设存在 $n$ 个有效遮挡的连续区间那么这些有效区间都应该满足有效遮挡的判定式：

$$\forall t \in [t_{\text{in}}^k, t_{\text{out}}^k], \forall P \in \Omega, \ \overline{P_1P_2} \cap C(P_c(t)) \neq \varnothing$$

同时，满足有效遮挡的时间段应该包含于烟幕弹起爆和烟幕弹浓度降低这两个关键时间点，即求解模型中的时间 $t$ 应该满足：

$$t_{\text{b}} \le t \le t_{\text{b}} + 20 \tag{5.20}$$

再进一步推导可以把上述公式变为：

$$t \in \bigcup_{k=1}^{n} [t_{\text{in}}^k, t_{\text{out}}^k] \subseteq [t_{\text{b}}, t_{\text{b}} + 20]$$

中国大学生在线 dxs.moe.gov.cn

考虑到每个有效遮挡时间段与总体的时间关系后，还应该考虑每个时间段相互之间的关系，即各个时间段之间没有重叠，它们是相互分离的。即：

$$[t_{\text{in}}^{k}, t_{\text{out}}^{k}] \cap [t_{\text{in}}^{k+1}, t_{\text{out}}^{k+1}] = \varnothing \quad k = (1, 2 ... n-1)$$

### 5.2.2 判定式相关参数说明

针对单机单弹的情况，根据上文的分析，已经知道了判定有效遮挡所需数据的计算方法，下面进行一个总结：

$P_2$ 是圆柱表面上的任意一点，可以利用数学表达式 $\forall P_2 \in \Omega = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + (y-200)^2 \leq 7^2, \quad 0 \leq z \leq 10\}$ 表出，对于导弹在某一时刻的位置 $P_1$，由导弹的位置函数可以得出：

$$\mathbf{P}_1 = \mathbf{P}_{m,i}(t) = \mathbf{P}_{m,i}(0) + \mathbf{v}_m \cdot t \tag{5.21}$$

云团的球心可以通过云团的位置函数求出：

$$\begin{cases} \mathbf{P}_{\text{c,jk}}(t) = (P_{\text{b,jk},x}, P_{\text{b,jk},y}, P_{\text{b,jk},z} - v_{\text{sink}} \cdot (t - t_{\text{b,jk}})) \quad t \geq t_{\text{b,jk}} \\ \mathbf{P}_{\text{s,jk}}(t) = \mathbf{P}_{fy,i}(t_{d,jk}) + \mathbf{v}_{fy,i} \cdot (t - t_{\text{d,jk}}) + \frac{1}{2}\mathbf{g} \cdot (t - t_{\text{d,jk}})^2 \quad t \geq t_{\text{d,jk}} \\ \mathbf{v}_{fy,i} = (v_{fy,i} \cos\theta_i, v_{fy,i} \sin\theta_i, 0) \\ \mathbf{P}_{fy,i}(t) = \mathbf{P}_{fy,i}(0) + \mathbf{v}_{fy,i} \cdot t \end{cases} \tag{5.22}$$

### 5.2.3 基于布尔函数的有效遮蔽条件表达

为了便于后文中有效遮挡条件的表述，本文引入了遮蔽状态的布尔函数：

$$Bool(t) = \begin{cases} 1, & \text{if } \forall P_2 \in \Omega, \quad \overline{P_1 P_2} \cap C(P_c(t)) \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{5.23}$$

那么对于有效遮蔽时间区间的开始时刻 $t_{in}$，其前一段微小时间没有达到有效遮蔽，则其相对应的布尔函数值为 0，其后一段微小时间达到了有效遮蔽，则其布尔值输出为 1，用数学表达式可以写为：

$$Bool(t_{in}^{+}) \wedge \neg Bool(t_{in}^{-}) = 1 \tag{5.24}$$

同样的，对于有效屏蔽的结束时刻 $t_{out}$，其前一段微小时间达到了有效遮蔽，则其相对应的布尔函数值为 1，其后一段微小时间没有达到有效遮蔽，则其布尔值输出为 0，用数学表达式可以写为：

$$Bool(t_{out}^{-}) \wedge \neg Bool(t_{out}^{+}) = 1 \tag{5.25}$$

所以满足有效遮蔽的时刻都应该满足：

$$T = \{t \in \mathbb{R}^{+} \mid Bool(t) = 1\} \tag{5.26}$$



中国大学生在线 dxs.moe.gov.cn

有效遮挡时长及所有有效遮挡时间点的集合，即为这个集合的总长度：

$$T = \int_0^{\infty} Bool(t)\,dt \tag{5.27}$$

时间 t 的积分区间为 $[0,\infty]$，这是因为在有效遮挡时间段布尔值才为 1，积分即为这一段的时长，而在非有效遮挡的时段，积分值为 0，不会对结果造成影响，这样考虑可以避免上文关于存在多段有效时间的考虑，使模型的约束减少，计算效率更优。

# 六、 问题一模型建立及求解

问题一针对具体的一个场景，在所有参数都已经给定的情况下（烟幕弹投放时间为 1.5s，起爆时间为 5.1s），需要求解单枚烟幕弹对单枚导弹的有效遮挡时长。需要利用各物体的位置函数以及有效遮挡判定式来确定发生有效遮挡的时间段。

## 6.1 有效时长计算模型

### 6.1.1 导弹 M1 与烟幕云团球心的轨迹

根据模型准备部分的分析，可以得出无人机 FY1 和导弹 M1 在给定的参数下的位置函数，以及烟幕弹和云团的运动轨迹：

$P_2$ 是圆柱表面上的任意一点，可以利用数学表达式 $\forall P_2 \in \Omega$ 表出，对于导弹在某一时刻的位置 $P_1$，这里指的是导弹 $M_1$ 的位置，由导弹的位置函数可以得出：

$$\mathbf{P}_1 = \mathbf{P}_{m,1}(t) = \mathbf{P}_{m,1}(0) + \mathbf{v}_{m,1} \cdot t \tag{6.1}$$

云团的球心可以通过云团的位置函数求出：

$$\begin{cases} \mathbf{P}_{\mathrm{c},11}(t) = (P_{\mathrm{b},11,x}, P_{\mathrm{b},11,y}, P_{\mathrm{b},11,z} - v_{\mathrm{sink}} \cdot (t - t_{\mathrm{b},11})) \quad t \geq t_{\mathrm{b},11} \\ \mathbf{P}_{\mathrm{s},11}(t) = \mathbf{P}_{fy1}(t_{d11}) + \mathbf{v}_{fy1} \cdot (t - t_{\mathrm{d},11}) + \frac{1}{2}\mathbf{g} \cdot (t - t_{\mathrm{d},11})^2 \quad t \geq t_{\mathrm{d},11} \\ \mathbf{v}_{fy,1} = (v_{fy,1}\cos\theta_1, v_{fy,1}\sin\theta_1, 0) \\ \mathbf{P}_{fy,1}(t) = \mathbf{P}_{fy,1}(0) + \mathbf{v}_{fy,1} \cdot t \end{cases} \tag{6.2}$$

### 6.1.2 判定式说明

该问题考虑的是单枚烟幕弹与单枚导弹的情况，所以只需要考虑单个云团的遮挡，而不存在云团间的耦合，所以满足有效遮挡的判别式为：

$$Bool(t) = \begin{cases} 1, & \text{if } \forall P_2 \in \Omega, \quad \overline{P_{m1}P_2} \cap C(P_{c,11}(t)) \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{6.3}$$

中国大学生在线 dxs.moe.gov.cn

上式中的 $C(P_{c,11}(t))$ 是无人机 FY1 发射的云团形成的球体，$C_{11}$ 是球体的球心，其数值根据即为云团中心轨迹的位置函数 $\mathbf{P}_{c,jk}(t)$ 的大小确定。

### 6.1.3 有效时长计算模型汇总

综上, 有效遮挡时间为各有效遮挡时间点的集合长度,, 则在固定参数下求解有效遮蔽时长的计算模型如下;

$$T = \int_0^{\infty} Bool(t)\, dt$$

$$\textbf{s.t.} \begin{cases} \mathbf{P}_1 = \mathbf{P}_{m,1}(t) = \mathbf{P}_{m,1}(0) + \mathbf{v}_{m,1} \cdot t \\ \begin{cases} \mathbf{P}_{c,11}(t) = (P_{b,11,x}, P_{b,11,y}, P_{b,11,z} - v_{sink} \cdot (t - t_{b,11})) \quad t \geq t_{b,11} \\ \mathbf{P}_{s,11}(t) = \mathbf{P}_{fy1}(t_{d11}) + \mathbf{v}_{fy1} \cdot (t - t_{d,11}) + \frac{1}{2}\mathbf{g} \cdot (t - t_{d,11})^2 \quad t \geq t_{d,11} \\ \mathbf{v}_{fy,1} = (v_{fy,1} \cos\theta_1, v_{fy,1} \sin\theta_1, 0) \\ \mathbf{P}_{fy,1}(t) = \mathbf{P}_{fy,1}(0) + \mathbf{v}_{fy,1} \cdot t \end{cases} \\ Bool(t) = \begin{cases} 1, & \text{if } \forall P_2 \in \Omega, \quad \overline{P_{m1}P_2} \cap C(P_{c,11}(t)) \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \end{cases} \tag{6.4}$$

## 6.2 模型的求解

### 6.2.1 有效遮蔽时间区间是一段连续的区间

真目标被烟幕云团有效遮蔽的时间必为一段连续时间区间。理由如下:

导弹 $M(t)$ 沿直线匀速靠近原点，云团球心 $C(t)$ 仅沿竖直方向匀速下沉，这些运动轨迹均为连续且单调的函数。对于给定时刻 $t$，圆柱是否被完全遮蔽，相当于判断：从导弹出发的所有视线是否都被半径固定的球 $B(t)$ 阻挡，是否遮蔽由导弹、云团和圆柱的相对位置关系决定，而这一几何关系是随时间连续演化的。

由于导弹和云团的运动没有振荡或往复，遮蔽关系的变化只能按以下顺序单调发生：初始时圆柱不在阴影之内；随后随着导弹接近和云团下沉，烟幕阴影的投影逐渐扩大并首次完全覆盖圆柱；此后在一段时间内圆柱保持完全遮蔽；最终阴影边界离开圆柱，完全遮蔽消失。整个过程中不存在的跳跃，因为对应的几何覆盖条件在连续单调的运动下只能经历一次进入和一次退出。由此可知，圆柱整体的完全遮蔽持续时间必然构成一个连通区间（在本问题的实际参数下为非空），即一段连续的时间区间。

### 6.2.2 求解定位圆周简化证明

问题一需要判断射向真目标圆柱体上任意一点的视线线段是否与半径为 10 米的烟幕球体相交。而这显然是不可能实现的，在求解过程中，只能对圆柱体进行离散化取点，而直接对整个圆柱体进行离散化取点会导致求解效率低下。

注意到：要判断一个物体是否被完全遮蔽，只需判断它最难被遮蔽的部分即可。对于圆柱体，这个最难被遮蔽的部分是从导弹的视角观察到的物体的外轮廓线（如图 5所

示）。**如果圆柱的外轮廓线被完全遮蔽，那么物体表面上所有在轮廓线内部的点也必然被遮蔽**。对于圆柱体而言，其轮廓线通常由上下底面圆周的两段圆弧和两条侧面的竖直母线构成（如图 6所示）。

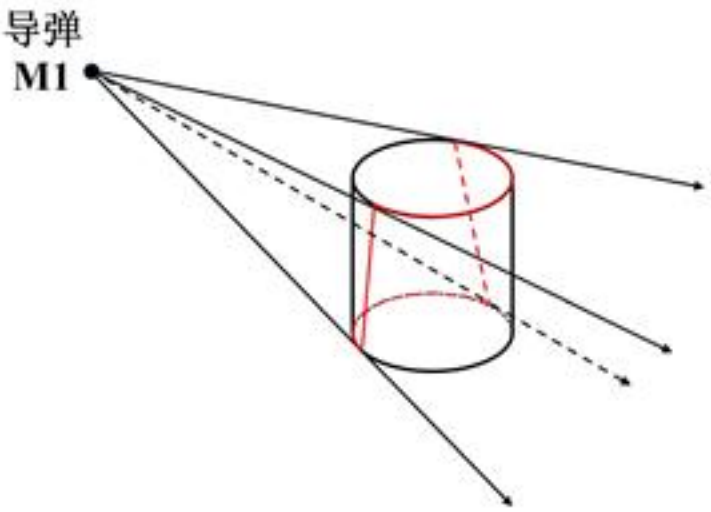



图 5: 导弹视角下观察圆柱

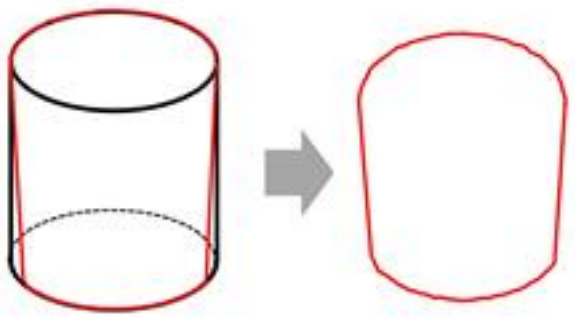


图 6: 导弹视角下圆柱体的外轮廓

因此，我们**将判断整个圆柱体是否被烟幕云团遮蔽简化为：判断圆柱体在导弹的视角下的外轮廓是否被烟幕云团完全遮蔽**。

经过进一步的分析，我们发现利用烟幕球体的凸性，**可以再次简化这个问题：只需判断圆柱体上下两个底面圆周是否被完全遮蔽，即可判定整个圆柱体是否被完全遮蔽。具体证明如下。**

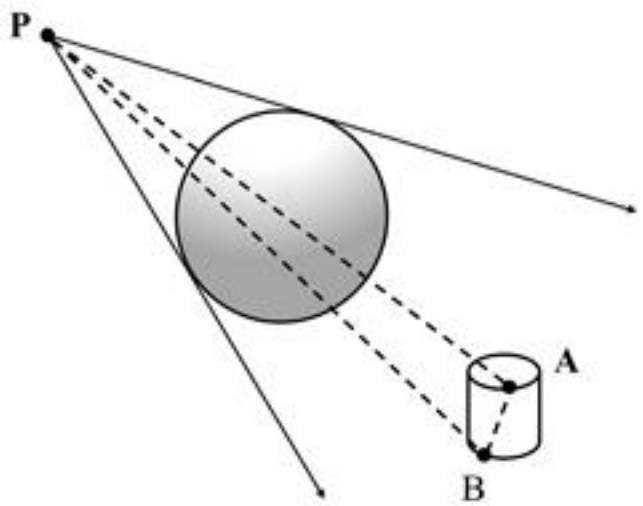



图 7: 圆柱体被遮蔽示意图

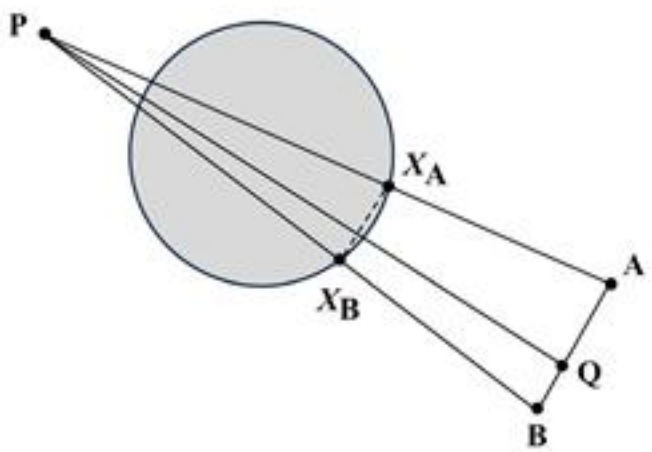



图 8: 截面 $PAB$ 示意图

**命题：** 若导弹位置为 $P$，且对于任意分别位于上底和下底圆周上的两点 $A$（上底）和 $B$（下底）对应的射线 $PA$ 与 $PB$ 都与烟幕球体 $S$ 相交（即被遮挡），则对于线段 $AB$ 上任意一点 $Q$，射线 $PQ$ 也与烟幕球体 $S$ 相交（也被遮挡）。

**证明.** 因为 $PA$ 与 $PB$ 都与 $S$ 相交，这说明平面 $PAB$ 和球体 $S$ 的交集不为空，它们的交集为一个圆盘。在平面 $PAB$ 上，令：

$$D := S \cap \text{平面} PAB$$

中国大学生在线 dxs.moe.gov.cn

区域 $D$ 是一个闭的圆盘。要证明原命题，即证明在二维平面内，从点 $P$ 指向位于 $A$ 与 $B$ 之间任一点 $Q$ 的线段也与圆盘 $D$ 相交。

令 $X_A$ 为 $PA$ 与 $D$ 的某个交点。同理取 $X_B$ 为 $PB$ 与 $D$ 的某一个交点。则有 $X_A, X_B \in D$。

因为 $D$ 是凸的，线段 $X_AX_B$ 完全包含在 $D$ 内。

因为 $X_A \in PA$ 且 $X_B \in PB$。因此 $X_AX_B$ 是位于三角形 $PAB$ 内部的一条连线（或在其边界上）。从 $P$ 指向边 $AB$ 上任意一点 $Q$ 的线段必与线段 $X_AX_B$ 相交。

又由于线段 $X_AX_B \subset D$，所以线段 $PQ$ 和线段 $X_AX_B$ 交点也属于 $D$。因此射线 $PQ$ 与 $D$ 相交，得证。 □

### 6.2.3 圆周离散化及采样点数的选取

根据前文的证明，我们只需判断圆柱体的上下底面圆周是否被完全遮蔽。在数值计算中，我们通过在圆周上均匀选取有限个采样点来近似模拟连续的圆周。

采样点数（记为 $n$）的选取需要在计算精度和效率之间取得平衡。为了量化几何近似的精度，我们计算了内接正 $n$ 边形与圆的面积相对误差。从表 2可以看出，当 $n = 300$ 时，相对误差仅为 0.0073%，此时正 300 边形在几何上已经能够很好地逼近圆。

表 2: 不同 $n$ 值下正 $n$ 边形与圆的面积相对误差

| **取点个数** ($n$) | 100 | 200 | 300 | 500 | 1000 |
|---|---|---|---|---|---|
| **相对误差** (%) | 0.06578 | 0.01645 | 0.00731 | 0.00263 | 0.00066 |

因此，在模型的求解过程中，我们暂时选取 $n = 300$，即在上下底面圆周上各均匀取 300 个点进行遮蔽判断。**该选择的合理性将在后文的结果验证部分通过敏感性分析进一步确认。**

### 6.2.4 二分法查找有效遮挡时间段的开始和结束时刻

上文已经说明了有效遮挡的时间段是一段单一连续的时间段，故本问可以通过**二分查找法**来搜寻有效遮挡时间段的开始时刻 $t_{in}$ 和结束时刻 $t_{out}$。

算法包含两个阶段：第一步，需要找到一个在有效遮蔽区间内部的时刻 $t^*$；第二步，利用该时刻作为端点，通过二分法分别逼近有效遮蔽区间的开始时刻 $t_{\text{in}}$ 和结束时刻 $t_{\text{out}}$。

**Step 1 算法初始化:** 定义初始点集 $S_0 = \{t_b, t_b{+}10, t_b{+}20\}$，设置时间精度 $\varepsilon_0 = 10^{-5}$，定义初始时间网格尺寸 $\delta_0 = 10$（即 $S_0$ 中相邻两点的初始间距）。

**Step 2 迭代点集:** 若当前迭代的网格尺寸 $\delta_k < \varepsilon_0$，则算法终止，转至 **step4**。基于有序集合 $S_k = \{s_1, s_2, \ldots, s_m\}$，生成新的测试点集 $S_{new} = \{\frac{s_i + s_{i+1}}{2}\}_{i=1}^{m-1}$。对每一个满足 $t \in S_{new}$ 的 $t$ 计算遮蔽状态 $C(t)$。

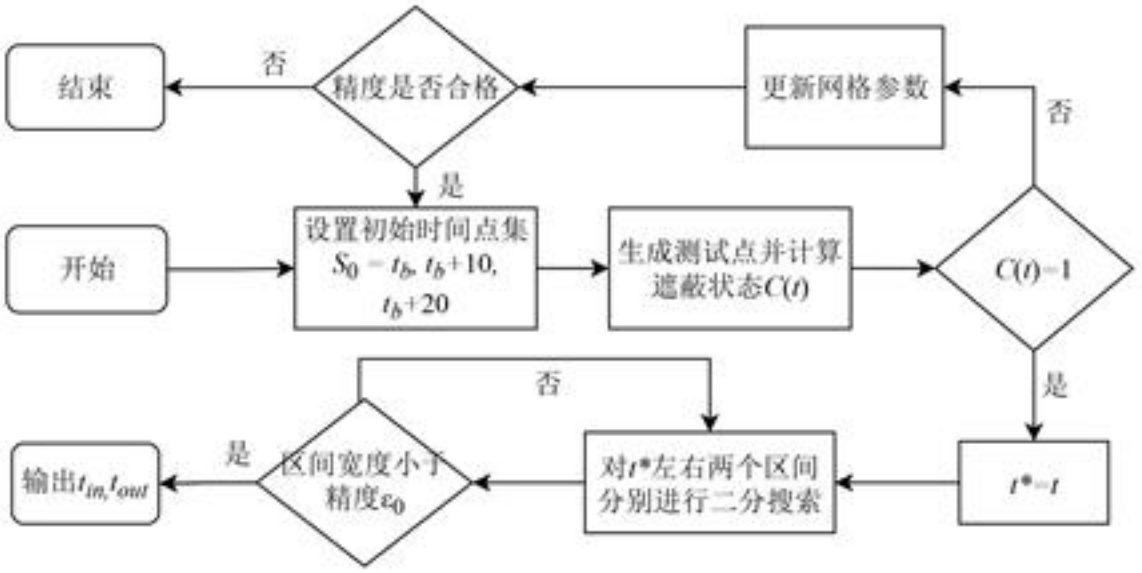



图 9: 问题一算法总体流程图

**Step 3 检验新的测试点集：** 若 $S_{new}$ 中存在 $t$ 满足 $C(t)=1$，则令 $t^*=t$，成功找到有效遮挡时间段内的一个时刻，结束算法。

若不存在这样的 $t$，则通过合并点集来更新 $S_{k+1}=S_k \cup S_{new}$，并更新网格尺寸 $\delta_{k+1}=\delta_k/2$ 转至 **step2** 继续迭代。

**Step 4 终止条件：** 若在有限次迭代后，仍未找到满足 $C(t)=1$ 的点，即在网格精度高于 $\varepsilon_0$ 前仍未找到满足 $C(t)=1$ 的点，则可以判定有效遮蔽区间的长度小于当前搜索精度 $\varepsilon_0$，算法终止。

### 6.2.5 用二分查找法求解有效遮蔽区间的起始时刻 $t_in$

二分查找法的示意图以及具体步骤如下：

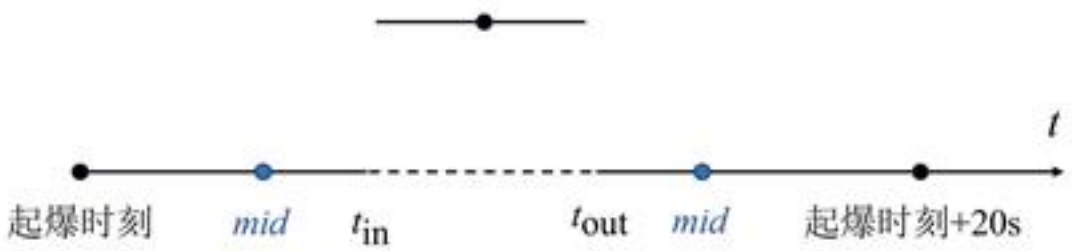



图 10: 二分查找法示意图

**Step 1 初始化搜索区间：** 设定初始搜索区间为 $[left, right]=[t_b, t^*]$，其中 $t_b$ 是烟幕起爆时刻。设定时间精度 $\varepsilon=10^{-5}$。初始化起始时刻的当前值为 $t_{\text{in}}=t^*$。

**Step 2 计算中间点：** 根据当前区间 $[left, right]$，计算中点 $mid=(left+right)/2$。计算 $C(mid)$ 的值来判断在 $mid$ 时刻真目标是否被完全遮蔽。

**Step 3 更新区间：** 如果 $C(mid)=1$，说明 $mid$ 时刻仍在有效遮蔽期内，真正的起始时刻可能等于或早于 $mid$。我们记录下这个可能的新边界 $t_{\text{in}}=mid$，并向左继续搜索，更新区间为 $[left, mid]$。

如果 $C(mid)=0$，说明 $mid$ 时刻已经早于有效遮蔽期的起始点，起始时刻必然在 $mid$ 之后。目标值位于 $[mid, right]$，更新区间为 $[mid, right]$。

**Step 4 求解结果：** 循环迭代直至区间宽度 $right - left < \varepsilon$。迭代结束时，$t_{\text{in}}$ 的最终值即为有效遮蔽区间的精确起始时刻。

#### 6.2.6 用二分查找法求解有效遮蔽区间的结束时刻 $t_{out}$

求解结束时刻 $t_{\text{out}}$ 的过程与求解 $t_{\text{in}}$ 的方法完全类似，同样采用二分查找法，**其不同之处仅在于：**

1. 初始搜索区间设定为 $[left, right] = [t^*, t_b + 20]$。
2. 区间更新逻辑 (Step 3)：

如果 $C(mid) = 1$，说明有效遮蔽尚未结束，真正的结束时刻位于或晚于 $mid$，因此向右搜索，更新区间为 $[mid, right]$。

如果 $C(mid) = 0$，说明 $mid$ 时刻已超出有效遮蔽范围，结束时刻必然早于 $mid$，因此向左搜索，更新区间为 $[left, mid]$。

算法的其余步骤与终止条件均与求解 $t_{\text{in}}$ 的过程相同。

### 6.3 结果展示

通过前文所述的计算方法，得到有效遮挡时间段的开始时刻 $t_{in}$ 和结束时刻 $t_{out}$。然后将求得的起止时刻相减，即可得到问题一需要求得的有效遮蔽时长 $T = t_{\text{out}} - t_{\text{in}}$。我们最终求得的**烟幕干扰弹对 $M1$ 的有效遮蔽时长为 1.391643s**。

### 6.4 结果的分析与验证

#### 6.4.1 离散化点数 $n$ 的敏感性分析

为验证模型求解过程中所选取的圆周离散化点数 $n = 300$ 的合理性与结果的稳定性，我们进行了一项敏感性分析。我们分别取 $n = \{100, 200, 300, 500, 1000\}$，并重复问题一的完整求解过程，得到的有效遮蔽时间结果如表 3所示。

表 3: 不同 $n$ 值下的有效遮蔽时长

| **取点个数 ($n$)** | 100 | 200 | 300 | 500 | 1000 |
|---|---|---|---|---|---|
| **有效时长 (s)** | 1.391646 | 1.391644 | **1.391643** | 1.391643 | 1.391643 |

从上表可以看出，当 $n \geq 300$ 时，计算得到的有效遮蔽时长在小数点后六位上保持一致，表明结果已经收敛。这证明我们选择 $n = 300$ 不仅保证了计算结果的精确性和稳定性，也避免了因 $n$ 值过大而导致的额外计算开销。因此，我们认为基于 $n = 300$ 计算出的**有效遮蔽时长 1.391643s** 是可靠的。

## 七、 问题二模型建立及求解

问题二在问题一建立的数学模型基础上，从固定参数计算提升到多参数的优化，这是一个典型的非线性优化问题。在实际军事应用中，烟幕干扰弹的投放策略优化具有重

要意义　　通过精确控制无人机的飞行参数和烟幕弹的引爆时间，可以在最合适的时间、最合适的位置形成有效遮蔽。本问针对这一背景进行最优策略的研究，下面对问题二的优化模型的建立进行详细说明：

## 7.1 单机单弹优化模型的建立

### 7.1.1 决策变量定义

本问需要决策无人机 FY1 往哪边飞，以多大的速度飞，何时投放烟幕弹，何时引爆烟幕弹这四个问题。因此本文定义了一个四维决策空间：

$$\mathcal{X} = (\theta, v_{fy}, t_d, t_b) \in \mathbb{R}^4 \tag{7.1}$$

其中各变量的物理含义和约束范围如下：航向角 $\theta$：决定了无人机的飞行方向，直接影响烟幕弹的投放位置。$\theta \in [0, 2\pi]$ 覆盖所有可能的方向选择。飞行速度 $v_{fy}$：影响无人机到达投放点的时间，$v$ $[70, 140]m/s$ 反映了无人机的性能限制。投放时间 $t_d$：决定了烟幕弹何时开始自由飞行，$t_d \geq 0$ 表示在受领任务后的某个时间点投放。

### 7.1.2 目标函数设置

研究烟幕弹的投放策略最终的目的在于延长有效遮挡的时间，所以应该选取最大化有效遮挡时间作为目标函数，因此求解最优策略的目标函数 $f(\mathbf{x})$ 的表达式为：

$$\max \quad f(\mathcal{X}) = t_{\text{out}} - t_{\text{in}} = \int_0^{\infty} Bool(t)\, dt \tag{7.2}$$

### 7.1.3 约束条件分析

**速度约束：** 无人机 FY1 的飞行速度 $v_{fy}$ 必须是一个在 70 m/s 到 140 m/s 的区间内的常数。

$$70 \leq v_{fy} \leq 140 \tag{7.3}$$

**飞行方向角约束：** 无人机的飞行方向角 $\theta$ 可以是水平面内的任意方向，$\theta \in [0, 2\pi]$ 就能覆盖所有可能的飞行方向：

$$0 \leq \theta < 2\pi \tag{7.4}$$

**等高度飞行约束：** 无人机须在初始高度 1800 米进行等高度飞行，其飞行速度的垂直分量为 0。

**时间约束：** 烟幕弹的投放时间 $t_d$ 必须在任务开始后，烟幕弹的起爆时间 $t_b$ 必须晚于或等于其投放时间 $t_d$，因此两者应该满足：

$$t_d, t_b \geq 0 \quad , \quad t_d \leq t_b \tag{7.5}$$

中国大学生在线 dxs.moe.gov.cn

**有效遮蔽窗口约束：** 烟幕云团仅在起爆后的 20 秒内可以提供有效遮蔽。因此，最终计算出的有效遮蔽时间区间 $[t_{in}, t_{out}]$ 必须完全包含在 $[t_b, t_b+20]$ 这个时间窗口内:

$$[t_{in}, t_{out}] \subseteq [t_b, t_b+20] \tag{7.6}$$

**有效遮挡判定约束：** 根据第一问的分析，单一烟幕弹对于导弹 M1 的有效遮挡判定是根据定义的布尔函数确定的，所以:

$$Bool(t) = \begin{cases} 1, & \text{if } \forall P_2 \in \Omega, \quad \overline{P_{m1}P_2} \cap C(P_{c,11}(t)) \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{7.7}$$

### 7.1.4 模型汇总

综上求解本问题的优化模型为:

$$\begin{aligned}
&max \quad f(\mathcal{X}) = \int_0^{\infty} Bool(t)\,dt \\
&\text{s.t.} \begin{cases}
v_{fy} \in [70,140], t_b \geq t_d+1, t_d \geq 0, \theta \in [0,2\pi] \\
t_b \leq t \leq t_b+20 \\
\begin{cases}
\mathbf{P}_{m,1}(t) = \mathbf{P}_{m,1}(0) + \mathbf{v}_{m,1} \cdot t \\
\mathbf{P}_{\mathrm{c},11}(t) = (P_{\mathrm{b},x}, P_{\mathrm{b},y}, P_{\mathrm{b},z} - v_{\mathrm{sink}} \cdot (t-t_{\mathrm{b}})) \quad t \geq t_{\mathrm{b}} \\
\mathbf{P}_{\mathrm{s},11}(t) = \mathbf{P}_{fy,1}(t_d) + \mathbf{v}_{fy,1} \cdot (t-t_{\mathrm{d}}) + \frac{1}{2}\mathbf{g} \cdot (t-t_{\mathrm{d}})^2 \quad t \geq t_{\mathrm{d}} \\
\mathbf{v}_{fy,1} = (v_{fy,1}\cos\theta, v_{fy,1}\sin\theta, 0) \\
\mathbf{P}_{fy,1}(t) = \mathbf{P}_{fy,1}(0) + \mathbf{v}_{fy,1} \cdot t
\end{cases} \\
Bool(t) = \begin{cases} 1, & \text{if } \forall P_2 \in \Omega, \quad \overline{P_{m1}P_2} \cap C(P_{c,11}(t)) \neq \varnothing \\ 0, & \text{otherwise} \end{cases}
\end{cases}
\end{aligned} \tag{7.8}$$

## 7.2 优化模型求解

### 7.2.1 多起点变步长搜索寻优

由于本问题的目标函数有效遮蔽时长 $f(\mathcal{X})$ 没有解析表达式，其函数值需要通过模拟仿真才能得到，这导致它的梯度信息难以计算，传统的梯度下降等优化算法不再适用。因此，我们设计了一种多起点变步长搜索算法。通过在可行域内进行多次随机采样来保证解的全局探索性，并对每个采样点采用自适应变步长的坐标搜索方法进行局部寻优，从而有效平衡了全局搜索与局部收敛的效率，并且能够防止陷入局部最优。

**Step 1 初始化与全局采样：** 为避免陷入局部最优，我们首先在整个可行解空间内进行全局随机抽样，生成 $N$ 个初始解作为局部搜索的起点。

每一个初始解 $\mathcal{X}^{(i)} = (\theta, v_{fy}, t_d, t_b)$ 都是通过在其四个决策变量的各自可行域，进行独立的均匀随机采样而生成的。这保证了初始种群在整个多维参数空间中的分布是均匀

的，从而更有可能找到全局最优解。

**Step 2 生成测试解：** 对每一个初始解 $\mathcal{X}^{(i)}$，执行一次变步长坐标搜索进行局部寻优。令当前迭代的最优解为 $\mathcal{X}_{best}$，并初始化步长衰减等级 $k=0$。算法迭代执行直至 $k$ 达到最大预设值 $K_{max}$。

依次遍历四个决策变量维度 $(j=1,2,3,4)$。在当前维度 $j$ 上，生成两个测试解：

$$\mathcal{X}^{+}=\mathcal{X}_{best}+\frac{\Delta_j}{2^k}\cdot\mathbf{e}_j$$

$$\mathcal{X}^{-}=\mathcal{X}_{best}-\frac{\Delta_j}{2^k}\cdot\mathbf{e}_j$$

其中 $\Delta_j$ 是维度 $j$ 的初始搜索步长，$\mathbf{e}_j$ 是该维度的单位基向量。

**Step 3 择优更新：** 若 $f(\mathcal{X}^{+})>f(\mathcal{X}_{best})$ 或 $f(\mathcal{X}^{-})>f(\mathcal{X}_{best})$，则更新 $\mathcal{X}_{best}$ 为更优的解

若本轮搜索未能找到更优解，则对步长进行衰减，$k\leftarrow k+1$

否则，保持 $k$ 不变，以更新后的 $\mathcal{X}_{best}$ 为中心开始新一轮的坐标轮换探索。

**Step 4 确定全局最优解：**

将 **Step 2** 的局部寻优过程应用于全部 $N$ 个初始点，会得到 $N$ 个对应的局部最优点。最后，通过比较这 $N$ 个点的目标函数值，选取其中最大者，即为最终求得的全局最优投放策略。

### 7.2.2 单个无人机单个干扰弹的投放最优策略

根据以上算法步骤，我们最终求出：无人机 FY1 在受领任务 0.8977s 后投放一枚烟幕干扰弹，间隔 0.0497s 后起爆，这样有效遮蔽时长最长，**最长有效遮蔽时长为 4.5873s。飞行方向：5.168°**（0.0902 弧度），无人机朝 $x$ 轴正方向略偏向 $y$ 轴正方向飞行。**飞行速度为 136.3663m/s**，无人机以接近其速度的上限飞行，以求尽快抵达最优投放区域。烟幕干扰弹**投放点坐标** $(x,y,z)$ **为 (17921.9184, 11.0270, 1800.0000)**，烟幕干扰弹**起爆点坐标** $(x,y,z)$ **为 (17928.6682, 11.6375, 1799.9879)**。

## 7.3 结果的分析与验证

为验证本文所采用的多起点变步长搜索算法求解这个优化模型的可靠性与准确性，我们采用粒子群优化算法对问题二进行独立的再次求解。粒子群算法在解空间中进行全局搜索，同样适用于求解目标函数复杂、非凸的优化问题。将两种算法求解得到的最优投放策略进行对比，具体结果如表 4所示：

中国大学生在线 dxs.moe.gov.cn

表 4: 两种优化算法最优策略对比

| 决策变量/目标 | 变步长搜索 (本文算法) | 粒子群优化 (PSO) |
|---|---|---|
| **最长有效遮蔽时长 (s)** | **4.587300** | **4.587772** |
| 飞行方向 (弧度) | 0.0902 | 0.0880 |
| 飞行速度 (m/s) | 136.3663 | 137.8825 |
| 投放时间 $t_d$ (s) | 0.8977 | 0.9389 |
| 起爆延迟 $\Delta t_b$ (s) | 0.0497 | 0.0106 |

从表 4可以看出，两种截然不同的优化算法所求得的最长有效遮蔽时长分别为 4.5873 秒和 4.5878 秒，相对误差仅为 0.01%。这表明我们的多起点变步长搜索算法在求解这个优化模型上的准确性和可靠性很高。

另外，两种算法给出的最优策略在核心特征上保持一致。比如，无人机的飞行速度都接近其性能上限（140 m/s），飞行方向角都非常小（约 0.09 弧度/5 角度，即朝 $x$ 轴正向略偏向 $y$ 轴正方向飞行），投放时间都在 0.9 秒左右。**从这些相似之处我们可以得出最优策略的一些规律：**要最有效遮挡其视线，理想的烟幕云团就应该位于导弹到真目标这条视线路径上。并且无人机 FY1 的初始位置在导弹到真目标和导弹到假目标这两个直线附近。因此，**最直接高效的投放策略就是迎着导弹来的方向飞过去**，这样就能最快地将烟幕干扰弹部署到导弹的必经之路上，形成一道迎面的烟幕云团。微小的角度偏差（5.1°）是为了更好地对准圆柱体轮廓而非仅仅是其中心点所做出的精细调整。

# 八、 问题三模型建立及求解

问题三在问题二的基础上，从单架无人机投放一枚烟幕弹提升到了单架无人机投放多枚烟幕弹，这就导致需要综合考虑三枚烟幕弹的投放时间序列、引爆时间序列、飞机的航向速度等，确定出一个能够有效衔接并且最大化总有效遮挡时间的方案，在本问中，优化模型中关于时间的变量从两个单独的时间变为了两个相互关联的时间序列。下面对这种单机多弹的优化模型的建立进行说明。

## 8.1 单机多弹优化模型的建立

### 8.1.1 决策变量的定义

同问题一一样，本问只考虑一架无人机，所以设该无人机的航向和速度分别为 $\theta$ 和 $v_{fy}$，而起爆时间和引爆时间都变成了三个，因此令三枚烟幕弹的投放时间和起爆时间分别为 $t_{d1}1, t_{d2}, t_{d3}$ 和 $t_{b1}, t_{b2}, t_{b3}$。所以本问的决策空间变为：

$$\mathcal{X} = (\theta, v_{fy}, t_{d,1}, t_{d,2}, t_{d,3}, t_{b,1}, t_{b,2}, t_{b,3}) \tag{8.1}$$

这些决策变量都应该约束在相应的范围内，下文将给出详细说明。

### 8.1.2 目标函数设置

前文已经说明了单枚烟幕弹对同一个导弹的有效遮挡时间是一段单一连续的时间段，但是在引入另外两枚烟幕弹后，不同的烟幕弹对同一导弹的有效遮挡时间段可能会由于投放时间序列和引爆时间序列的决策而发生重叠或者衔接，因此本问中有效遮挡的时间段不再由单一的两个时间点决定，而是需要综合三枚烟幕弹共同的有效遮挡效果。

原有的判定式是判断导弹视线 $P_{m1}P_2$ 是否与单个球体 $C(P_c)$ 相交。现在，我们**需要判断视线是否与任意一个烟幕云团相交**。在时刻 $t$，真目标圆柱体被有效遮蔽的条件是：对于圆柱体表面上的任意一点 $P_2$，连接导弹位置 $P_{m1}(t)$ 与 $P_2$ 的视线线段 $P_{m1}P_2$，至少与这 3 个烟幕云团中的一个相交。

新的有效遮挡判定式为：

$$\forall P_2 \in \Omega, \quad \exists k \in \{1,2,3\} \quad \text{s.t.} \quad \overline{P_{m1}(t)P_2} \cap C(P_{c,k}(t)) \neq \varnothing \tag{8.2}$$

其中 $C(P_{c,k})$ 代表 FY1 投下的第 $k$ 个有效云团。

基于此，我们重新定义布尔函数 $Bool(t)$：

$$Bool(t) = \begin{cases} 1, & \text{if } \forall P_2 \in \Omega, \quad \left(\bigcup_{k=1}^{N(t)} C(P_{c,k})\right) \cap \overline{P_{m1}(t)P_2} \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{8.3}$$

这里的 $N(t)$ 表示在时刻 $t$ 处于有效期内的云团总数。这个函数 $Bool(t) = 1$ 的时刻 $t$，就构成了总的有效遮蔽时间。

令 $T$ 为所有满足 $Bool(t) = 1$ 的时刻 $t$ 的集合。这个集合可能是由多个不连续的时间段组成的。

$$T = \{t \in \mathbb{R}^+ \mid Bool(t) = 1\} \tag{8.4}$$

目标函数即为这个集合的总长度：

$$\max \quad f(\mathcal{X}) = \int_0^{\infty} Bool(t)\, dt \tag{8.5}$$

### 8.1.3 约束条件分析

**物理约束：**决策变量以及一些量的取值范围应该符合题意与物理实际，所以应该满足：

$$\begin{cases} 70 \leq v_{fy} \leq 140 \quad , \quad 0 \leq \theta \leq 2\pi \quad , \quad v_{fy,z} = 0 \\ t_{d,k}, t_{b,k} \geq 0 \quad k = (1,2,3) \\ t_{b,k} \geq t_{d,k} \quad k = (1,2,3) \end{cases} \tag{8.6}$$

**有效遮蔽窗口约束：**烟幕弹爆炸产生的云团只在起爆后的 20s 内可以提供有效遮

中国大学生在线 dxs.moe.gov.cn

挡，因此每一枚烟幕弹的有效遮挡时长应该含与起爆时刻到起爆 20s 这两个时刻之间：

$$[t_{in,k}, t_{out,k}] \subseteq [t_{b,k}, t_{b,k}+20] \quad (k=1,2,3) \tag{8.7}$$

**投放延迟约束：**根据题意虽然同一烟幕弹投放时间和引爆时间没有影响，但同一架无人机投放多枚烟幕弹，其投放间隔必须在 1s 以上：

$$t_{d,k} \le t_{d,k+1} - 1 \quad (k=1,2) \tag{8.8}$$

### 8.1.4 模型汇总

综上，单架无人机投放多枚烟幕弹的策略优化模型为：

$$\max \quad f(\mathcal{X}) = \int_0^{\infty} Bool(t)\,dt$$

$$\text{s.t.}\begin{cases}
\begin{cases}
70 \le v_{fy} \le 140 \quad , \quad 0 \le \theta \le 2\pi \quad , \quad v_{fy,z} = 0 \\
t_{d,k}, t_{b,k} \ge 0 \quad k=(1,2,3) \\
t_{b,k} \ge t_{d,k} \quad k=(1,2,3)
\end{cases} \\
[t_{in,k}, t_{out,k}] \subseteq [t_{b,k}, t_{b,k}+20] \quad (k=1,2,3) \\
t_{d,k} \le t_{d,k+1} - 1 \quad (k=1,2) \\
\begin{cases}
\mathbf{P}_{m,1}(t) = \mathbf{P}_{m,1}(0) + \mathbf{v}_{m,1} \cdot t \\
\mathbf{P}_{\mathrm{c,1k}}(t) = (P_{\mathrm{b,k},x}, P_{\mathrm{b,k},y}, P_{\mathrm{b,k},z} - v_{\mathrm{sink}} \cdot (t - t_{\mathrm{b,k}})) \quad t \ge t_{\mathrm{b,k}} \\
\mathbf{P}_{\mathrm{s,1k}}(t) = \mathbf{P}_{fy}(t_{d,k}) + \mathbf{v}_{fy} \cdot (t - t_{\mathrm{d,k}}) + \frac{1}{2}\mathbf{g} \cdot (t - t_{\mathrm{d,k}})^2 \quad t \ge t_{\mathrm{d,k}} \\
\mathbf{v}_{fy1} = (v_{fy}\cos\theta, v_{fy}\sin\theta, 0) \\
\mathbf{P}_{fy}(t) = \mathbf{P}_{fy}(0) + \mathbf{v}_{fy} \cdot t
\end{cases} \\
Bool(t) = \begin{cases}
1, \quad \text{if } \forall P_2 \in \Omega, \quad \overline{P_{m1}P_2} \cap C(P_{c,k}) \ne \varnothing \quad (k=1,2,3) \\
0, \quad \text{otherwise}
\end{cases}
\end{cases} \tag{8.9}$$

## 8.2 优化模型求解

本问共有八个决策变量，一般的算法解决不了高维变量空间的模型，而且目标函数是没有解析表达式的，仅是一些逻辑表达，函数值需要通过数值仿真计算，梯度信息难以获取。由于差分进化算法能够处理高维、非线性优化问题同时不依赖梯度信息，适合复杂仿真计算的特点，本问选择利用差分进化法进行求解。

中国大学生竞赛 dxs.moe.gov.cn

### 8.2.1 差分进化算法

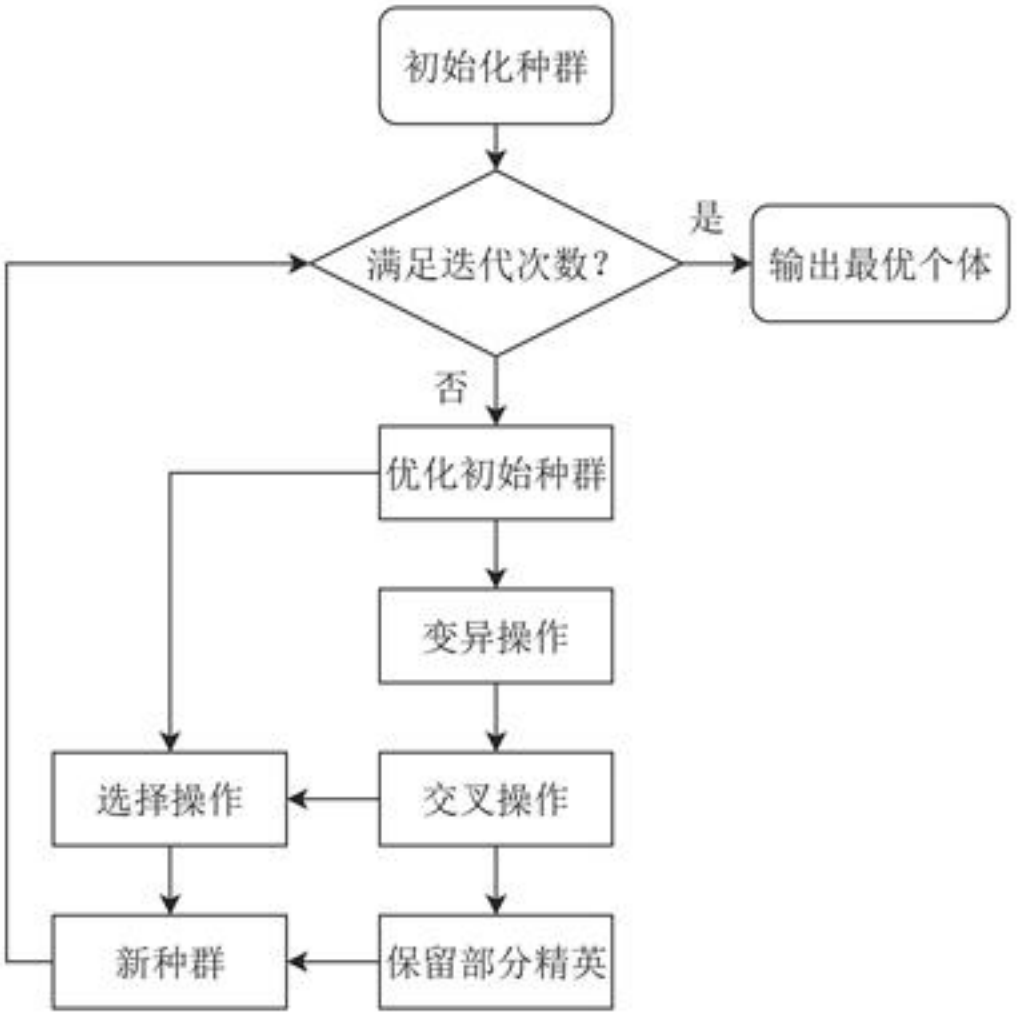



图 11: 差分进化算法流程图

**Step 1 初始化：** 设定种群规模 $N = 100$、最大迭代次数 $G = 500$、变异因子 $F = 0.5$ 及交叉概率 $CR = 0.9$ 等关键参数。每个个体为一个 8 维决策向量 $\mathcal{X} = (\theta, v_{fy}, t_{d,1}, t_{d,2}, t_{d,3}, t_{b,1}, t_{b,2}, t_{b,3})$。在可行域内随机生成初始种群，并确保其满足所有的约束条件。

**Step 2 个体评估与约束处理：** 对种群中的每个个体 $\mathcal{X}_i$ 通过模拟仿真计算其对应的总有效遮蔽时长 $f(\mathcal{X}_i)$。采用惩罚函数法处理约束，将约束违反度整合到适应度评估中，使种群向可行域搜索。

**Step 3 差分进化操作：** 对种群中的每一个个体 $\mathcal{X}_i$ 进行如下变异和交叉操作以生成试验向量 $U_i$。

变异：从当前种群中随机选择三个与 $\mathcal{X}_i$ 不同的个体 $\mathcal{X}_{r1}, \mathcal{X}_{r2}, \mathcal{X}_{r3}$，根据式 8.10生成变异向量 $V_i$。

$$V_i = \mathcal{X}_{r1} + F \cdot (\mathcal{X}_{r2} - \mathcal{X}_{r3}) \tag{8.10}$$

交叉：将变异向量 $V_i$ 与目标向量 $\mathcal{X}_i$ 进行交叉，生成试验向量 $U_i$。本文采用二项式交叉，即以交叉概率 $CR$ 决定试验向量的每个维度是继承自变异向量还是目标向量，并确保至少有一个维度发生交换，以引入新信息。

**Step 4 选择：** 计算试验向量 $U_i$ 的目标函数值 $f(U_i)$，如果 $f(U_i)$ 优于 $f(\mathcal{X}_i)$，则在下一代中用 $U_i$ 替换 $\mathcal{X}_i$；否则，保留原个体 $\mathcal{X}_i$。

**Step 5 迭代与输出：** 重复执行 **Step 2** 至 **Step 4**，直到达到预设的最大迭代次数 $G_{max}$。算法终止后，输出整个进化过程中记录下的最优个体 $\mathcal{X}_{best}$，该个体对应的决策变量组合即为所求的无人机最优投放策略，其对应的目标函数值即为最长有效遮蔽时间。

### 8.2.2 求解结果

根据上述流程，本问利用差分进化算法求解出的无人机 FY1 投放 3 枚烟幕弹干扰导弹 M1 的最优策略下对应的**最大有效遮挡时间**是 **6.93586s**，**无人机航向方向角**为：**13.57377°**，**无人机速度**大小为：**116.81318m/s**。三枚烟幕弹的具体投放策略见表 5。

表 5: 烟幕干扰弹投放与效能数据

| 烟幕弹序号 | 投放坐标 (m) | | | 起爆坐标 (m) | | |
|---|---|---|---|---|---|---|
| | X | Y | Z | X | Y | Z |
| 1 | 17800 | 0 | 1800 | 17800 | 0 | 1800 |
| 2 | 17915.99463 | 13.80460 | 1800 | 17915.99463 | 13.80460 | 1800 |
| 3 | 22048.21768 | 505.58318 | 1800 | 23913.0268 | 727.51534 | 533.54687 |

| 烟幕弹序号 | 生效时间 (s) | 失效时间 (s) | 有效干扰时长 (s) |
|---|---|---|---|
| 1 | 0.79116 | 5.41117 | 2.55600 |
| 2 | 1.00000 | 5.07793 | 4.37985 |
| 3 | 0.00000 | 0.00000 | 0.00000 |

分析上表中数据可以看出，第二发烟幕干扰弹于 1s 极早期开始生效，持续到了 5.07793s，贡献了 4.07793s 的有效遮蔽，占到了总有效遮挡时间的极大部分，说明第二枚烟幕弹的投放策略是极优的，但是该参数下的无人机行进轨迹会很早的与导弹相遇，使无人机相对于目标的位置比导弹更远，之后投放的烟幕弹不能够再提供有效遮挡，这也是第三枚烟幕弹始终没有发挥作用的原因。导致这种现象出现的原因可能与无人机的初始位置有关，在其他的初始位置可能可以找到真正的全局最优的投放策略达到最大的有效遮挡时长。

## 九、 问题四模型建立及求解

问题四要求利用 FY1、FY2、FY3 三架无人机各投放一枚烟幕弹以干扰来袭的 M1 导弹。此问题从单无人机的任务规划，转变为一个协同优化问题，与第三问不同的地方在于：三个无人机各自投弹不需要考虑不同烟幕弹的投放时间的制约，三个烟幕弹的投放时间是相互独立的。需要为三架无人机分别规划出最优的飞行策略和投弹时机，使得三枚烟幕弹形成的遮蔽效果能够同实现对真目标总有效遮蔽时间的最大化。

## 9.1 多机单弹优化模型的建立

### 9.1.1 决策变量的定义

本问仍然以最大化有效遮挡时间为目标，但现在有三架无人机分别投一枚烟雾弹，而不是一架无人机独自投三枚，因此需要决策的参数是三架无人机各自的航向、速度以及烟幕弹的投放与引爆时间，即决策空间为:

$$\mathbf{x}_j = (\theta_j, v_{fy,j}, t_{d,j}, t_{b,j}), \quad \text{总变量} \quad \mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \in \mathbb{R}^{12}. \tag{9.1}$$

上式中 $j$ 的取值为 $j = 1, 2, 3$，代表了三架无人机。

### 9.1.2 目标函数的设置

同问题三一样，问题四的目标也是最大化有效遮挡时间，因此在形式上，问题四的目标函数与问题三完全一致:

$$\max \quad f(\mathcal{X}) = \int_0^{\infty} Bool(t)\, dt, \tag{9.2}$$

其中, 布尔函数的定义需要进行调整，无人机 FY1 投下的第 $k$ 个有效云团 $C(P_{c,k})$ 应该变为第 $j$ 架无人机投下的云团 $C(P_{c,j})$ 即:

$$Bool(t) = \begin{cases} 1, & \forall P_2 \in \Omega,\ \exists j \in \{1,2,3\},\ \overline{P_{m1}(t)P_2} \cap C(P_{cj}(t)) \neq \varnothing \\ 0, & \text{otherwise.} \end{cases} \tag{9.3}$$

### 9.1.3 约束条件分析

由于三架无人机投放烟幕弹同样需要满足物理约束以及烟幕弹爆炸产生的云团的有效期的约束，这与上文的模型中的分析并无太大差别，这里不再赘述。即满足:

$$\begin{cases} 70 \le v_{fy,j} \le 140 \quad , \quad 0 \le \theta_j \le 2\pi \quad , \quad v_{fy,j,z} = 0 \\ t_{d,j}, t_{b,j} \ge 0 \quad j = (1,2,3) \\ t_{b,j} \ge t_{d,j} \quad j = (1,2,3) \end{cases} \tag{9.4}$$

### 9.1.4 模型汇总

综上，多架无人机各自投放单枚烟幕弹的最优投放策略求解模型为:

$$\max \quad f(\mathcal{X}) = \int_0^{\infty} Bool(t)\, dt$$

中国大学生在线 dxs.moe.gov.cn

$$
\text{s.t.}\begin{cases}
\begin{cases}
70 \le v_{fy,j} \le 140 \quad , \quad 0 \le \theta_j \le 2\pi \quad , \quad v_{fy,j,z} = 0 \\
t_{d,j}, t_{b,j} \ge 0 \quad j = (1,2,3) \\
t_{b,j} \ge t_{d,j} \quad j = (1,2,3)
\end{cases} \\
[t_{in,j}, t_{out,j}] \subseteq [t_{b,j}, t_{b,j} + 20] \quad (j = 1,2,3) \\
\begin{cases}
\mathbf{P}_{m,1}(t) = \mathbf{P}_{m,1}(0) + \mathbf{v}_{m,1} \cdot t \\
\mathbf{P}_{\mathrm{c,j}}(t) = (P_{\mathrm{b,j},x}, P_{\mathrm{b,j},y}, P_{\mathrm{b,j},z} - v_{\mathrm{sink}} \cdot (t - t_{\mathrm{b,j}})) \quad t \ge t_{\mathrm{b,j}} \\
\mathbf{P}_{\mathrm{s,j}}(t) = \mathbf{P}_{fy,j}(t_{d,j}) + \mathbf{v}_{fy,j} \cdot (t - t_{\mathrm{d,j}}) + \frac{1}{2}\mathbf{g} \cdot (t - t_{\mathrm{d,j}})^2 \quad t \ge t_{\mathrm{d,j}} \\
\mathbf{v}_{fy,j} = (v_{fy,j} \cos\theta_j, v_{fy,j} \sin\theta_j, 0) \\
\mathbf{P}_{fy,j}(t) = \mathbf{P}_{fy,j}(0) + \mathbf{v}_{fy,j} \cdot t
\end{cases} \\
Bool(t) = \begin{cases}
1, \quad \forall P_2 \in \Omega,\ \exists j \in \{1,2,3\},\ \overline{P_{m1}(t)P_2} \cap C(P_{cj}(t)) \ne \varnothing \\
0, \quad \text{otherwise.}
\end{cases}
\end{cases}
\tag{9.5}
$$

## 9.2 优化模型求解

### 9.2.1 基于预测拦截点的协同优化策略

问题 4 的优化模型涉及 3 个无人机，决策变量较多，如果直接进行求解会导致效率低下。

**由问题 2 的结果分析可知：**要最有效遮挡其视线，理想的烟幕云团应该近似位于导弹到真目标这条视线路径上，同时，由于真目标距离原点的距离相对于真目标距离导弹以及原点距离导弹的距离很小，导弹到真目标的这条线段和导弹到假目标（即原点）的线段很接近，所以近似有：理想的烟幕云团应该近似位于导弹到假目标（即原点）这条视线路径上。因此，最直接高效的投放策略就是迎着导弹来的方向飞过去，然后将烟幕干扰弹投放到导弹到假目标（即原点）这条视线路径上，形成一个迎面的烟幕云团。

基于以上分析，**我们考虑将烟幕干扰弹投到导弹到假目标（即原点）的线段上，然后再对这个结果进行微调，从而得到烟幕干扰弹的投放策略**。为进一步简化并获得高质量的初始解，我们提出一个更强的约束作为寻优起点：在某一时刻 $t$，烟幕干扰弹的中心与导弹的位置完全重合。即：

$$
P_{s,j}(t) = P_{m1}(t) \tag{9.6}
$$

通过一些运动学分析建立一组不等式来求得能够得到满足条件的一些解集。具体而言：我们希望知道是否存在某一组满足所有约束条件的决策变量 $(v_{fy,j}, \theta_j, t_{d,j})$，能使得在某一时刻 $t$，烟幕干扰弹的中心与导弹的位置完全重合。建立不等式组和求解的具体过程如下：

为简化求解过程，我们引入比例系数 $s \in [0,1]$，定义为烟幕弹自由落体时间占无人机总飞行时间 $t$ 的比值。因此，投放时间 $t_{d,j}$ 可表示为 $t_{d,j} = t(1-s)$。

令 $P_{s,j}(t) = (x_s, y_s, z_s)$ 和 $P_{m1}(t) = (x_{m1}, y_{m1}, z_{m1})$，其中：

$$
\begin{cases}
x_s(t) = x_{fy,j}(0) + v_{fy,j}\cos(\theta_j)\,t, \\
y_s(t) = y_{fy,j}(0) + v_{fy,j}\sin(\theta_j)\,t, \\
z_s(t) = z_{fy,j}(0) - \frac{1}{2}g(t - t_{d,j})^2
\end{cases}
\tag{9.7}
$$

重合条件 $P_{s,j}(t) = P_{m1}(t)$ 可分解为以下方程组：

$$
\begin{cases}
x_{fy,j}(0) + v_{fy,j}\cos(\theta_j)\cdot t = x_{m1}(t) \\
y_{fy,j}(0) + v_{fy,j}\sin(\theta_j)\cdot t = y_{m1}(t) \\
z_{fy,j}(0) - \frac{1}{2}g(s\cdot t)^2 = z_{m1}(t)
\end{cases}
\tag{9.8}
$$

该方程组为我们提供了一种高效的求解路径。对于任意一个假定的拦截时刻 $t$，由 Z 轴方程可直接解出所需的自由落体时间比例 $s$：

$$
s = \frac{1}{t}\sqrt{\frac{2(z_{fy,j}(0) - z_{m1}(t))}{g}}
\tag{9.9}
$$

其次，联立 X 轴和 Y 轴方程，可解出无人机为在时刻 $t$ 到达目标水平位置 $(x_{m1}(t), y_{m1}(t))$ 所需的恒定飞行速度大小 $v_{fy,j}$：

$$
v_{fy,j} = \frac{1}{t}\sqrt{(x_{m1}(t) - x_{fy,j}(0))^2 + (y_{m1}(t) - y_{fy,j}(0))^2}
\tag{9.10}
$$

然而，并非任意时刻 $t$ 都存在可行策略。一个解要想在物理上成立，必须同时满足以下所有约束条件：

$$
\begin{cases}
z_{fy,j}(0) \geq z_{m1}(t) \\
0 \leq s \leq 1 \\
70 \leq v_{fy,j} \leq 140
\end{cases}
\tag{9.11}
$$

**最终可以求得可行时刻 $t$ 的解集，只需选取在这个解集中最小的那个时刻 $t$ 即可，理由如下：**

随着时刻 $t$ 的增长、导弹持续向原点方向移动，所以导弹距离原点的距离会逐渐减小。因此真目标、导弹和原点的夹角会持续增大，即导弹对于真目标圆柱体的切向线速度逐渐增大，并且导弹和真目标的距离在减小。所以导弹对于真目标圆柱体的切向角速度在逐渐增大。这会导致导弹以更快的速度掠过烟幕云团，使得有效遮蔽时间变短，因此需要选取尽可能小的 $t$。

得到的解是基于导弹和烟幕干扰弹重合这一理想化近似，需要进一步微调以最大化对真实圆柱目标的有效遮蔽时长。我们**采用精细化的局部网格搜索来对上文得到的近似最优解进行微调**，算法具体步骤如下：

**step1 定义搜索空间：**以上文得到的近似最优解 $\mathcal{X}_0 = (\theta_0, v_{fy,0}, t_{d,0}, \Delta t_{b,0})$ 作为搜索

空间的中心。在这个中心附近定义一个搜索域。

**step2 网格离散化与遍历：** 各维度的搜索域范围及离散化如下：

航向角 $\theta$ 在弧度区间 $[\theta_0 - 0.2, \theta_0 + 0.2]$ 内以 0.008 弧度为步长进行采样。

飞行速度 $v_{fy}$ 在区间 $[\max\{v_{fy,0} - 20, 70\}, \min\{v_{fy,0} + 20, 140\}]$ m/s 内均匀采样 50 个点。

投放时间 $t_d$ 在 $[t_{d,0} - 5, t_{d,0} + 5]$ 秒区间内均匀采样 80 个点。

引爆延迟 $\Delta t_b$ 在 $[\Delta t_{b,0} - 5, \Delta t_{b,0} + 5]$ 秒区间内均匀采样 80 个点。

**step3 候选策略评估与择优：** 对于网格中的每一个策略 $\mathcal{X}_{cand}$ 进行模拟仿真，计算出该策略下对真目标的有效遮蔽时长 $T$。动态更新最大的有效遮蔽时长和决策变量。

**step4 输出：** 遍历所有网格节点后，算法输出最终记录的 $\mathcal{X}_{best}$，作为该架无人机在当前情形下的最优投放策略。

### 9.2.2 求解结果

根据对预测拦截点的分析，本文在采用局部网格搜索进行求解时，效率得到了极大的提高。最终利用局部网格搜索求解出**最大总有效遮蔽时间达到 11.125936s**，其余结果可见附件 `result2.xlsx`。各烟幕弹的遮蔽时间段衔接良好，形成了连续的有效遮蔽效果，各无人机投放烟幕弹的具体策略见表 6：

表 6: 烟幕干扰弹投放策略

| 烟幕弹序号 | 投放坐标 (m) | | | 起爆坐标 (m) | | |
|---|---|---|---|---|---|---|
| | X | Y | Z | X | Y | Z |
| 1 | 17763.38330 | 1.17214 | 1800 | 17552.65402 | 7.91777 | 1762.82904 |
| 2 | 13251.30020 | 260.33090 | 1400 | 13521.74556 | 14.01253 | 1361.06668 |
| 3 | 6455.35305 | -208.59043 | 700 | 6498.39015 | 55.23604 | 654.78273 |
| **烟幕弹序号** | **生效时间 (s)** | | | **失效时间 (s)** | | **有效干扰时长 (s)** |
| 1 | 3.49922 | | | 8.02193 | | 4.52271 |
| 2 | 16.51588 | | | 20.41043 | | 3.89455 |
| 3 | 35.41727 | | | 38.12596 | | 2.70869 |
| **无人机序号** | **航向 (°)** | | | **速度 (m/s)** | | |
| 1 | 178.16654 | | | 76.54968 | | |
| 2 | 317.67312 | | | 129.77384 | | |
| 3 | 80.73514 | | | 87.996834 | | |

分析表 6 可知，三架无人机（FY1、FY2、FY3）通过合理规划投放策略，使得投下的三枚烟幕弹在时间上形成了较好的衔接，最终实现了约 11.1 秒的总有效遮蔽时间。从结果可以看出：各无人机投放的烟幕弹虽然作用区间不重叠，但覆盖在导弹来袭路径上的时间段分布合理，最大限度地延长了真目标被遮蔽的总时长。同时，优化策略利用了预测拦截点和局部搜索的方法，既保证了烟幕云团基本位于导弹和目标的视线附近，又通过微调提升了遮蔽时长。

### 9.3 结果的验证

为验证本文所提出的预测起爆点和局部网格搜索进行优化的有效性和解的质量，我们采用差分进化算法对问题四进行独立的对比求解。差分进化算法能够在复杂的解空间中进行全局搜索，可以有效避免陷入局部最优解，因此非常适合用来验证本文选取的求解算法是否合适。

最终，用差分进化算法求解得到的总最大有效遮蔽时间是 11.645260s，相对误差为 4.46%，说明我们的算法在求解这个问题上误差较小。同时，该方法利用了问题的物理特性，大幅缩小了搜索空间，避免了全局进化算法的大量迭代开销，在运算效率上具有显著优势。

## 十、 问题五模型建立及求解

### 10.1 多机多弹优化模型建立

#### 10.1.1 决策变量的定义

问题五涉及多架无人机和多枚烟幕弹，决策变量的数量将显著增加。对于每架无人机 $j \in \{1,\ldots,5\}$，如果它被分配了任务并决定投放 $y_j$ 枚烟幕弹（$0 \le y_j \le 3$），则需要决策该无人机的飞行航向角、速度和烟幕干扰弹的投放时间和起爆时间。

为了简化模型，我们假设每架无人机如果参与任务，都会确定一个飞行方向和速度，并在该方向和速度下投放所有烟幕弹，而不会在飞行途中临时改变飞行参数。

因此，决策变量可以定义为：

$$\mathcal{X} = \{(\theta_j, v_{fy,j}, t_{d,jk}, t_{b,jk}) \mid j = 1,\ldots,5; k = 1,\ldots,3\} \tag{10.1}$$

其中：$j$ 表示无人机编号，$k$ 表示无人机 $j$ 投放的第 $k$ 枚烟幕弹。如果某架无人机决定只投放 1 枚或 2 枚烟幕弹，那么相应的 $t_{d,jm}$ 和 $t_{b,jm}$ 将被设为一些无效值或通过其他方式处理。为了方便表示，我们可以在模型中假定每架无人机都考虑投放 3 枚，然后通过约束条件来限制实际投放的数量。

#### 10.1.2 目标函数设置

目标仍然是最大化对真实目标的总有效遮蔽时间。即：

$$\max \quad f(\mathcal{X}) = \int_0^{\infty} Bool(t)\, dt, \tag{10.2}$$

但是现在需要同时考虑 3 枚来袭导弹。这意味着，我们需要在任何给定时刻 $t$，判断对于所有来袭导弹 $i \in \{1,2,3\}$ 射向真目标的所有视线是否都被所有有效的烟幕云团所遮蔽。

中国大学生在线 dxs.moe.gov.cn

因此，布尔函数 $Bool(t)$ 的定义需要进一步修改：

$$Bool(t)=\begin{cases}1, & \text{if } \forall i\in\{1,2,3\},\forall P_2\in\Omega,\quad \exists j\in\{1,\ldots,5\},\exists k\in\{1,\ldots,3\}\\ & \quad \text{s.t.}\quad \overline{P_{m\,i}(t)P_2}\cap C(P_{c,jk}(t))\neq\varnothing\\ 0, & \text{otherwise}\end{cases}\tag{10.3}$$

其中，$P_{m\,i}(t)$ 是第 $i$ 枚导弹在时刻 $t$ 的位置，$C(P_{c,jk}(t))$ 是无人机 $j$ 投放的第 $k$ 枚烟幕弹在时刻 $t$ 形成的云团球体。

### 10.1.3 约束条件分析

除了问题三和问题四中已有的如物理约束、运动学约束的约束外，问题五还需考虑以下约束：

**任务分配约束**：每枚导弹至少被一架无人机干扰，每架无人机最多投放 3 枚烟幕弹：

$$\sum_{j=1}^{5}a_{ij}\geq 1\quad \forall i\in\{1,2,3\}\qquad \sum_{k=1}^{3}b_{jk}\leq 3\quad \forall j\in\{1,\ldots,5\}\tag{10.4}$$

其中 $a_{ij}$ 为二进制变量，表示无人机 $j$ 是否干扰导弹 $i$；$b_{jk}$ 表示无人机 $j$ 是否投放第 $k$ 枚烟幕弹。这两个约束可以来限制实际的投弹数量。

**投弹间隔约束**：根据题意同一无人机投放多枚烟幕弹的时间间隔至少为 1 秒：

$$t_{d,j(k+1)}>t_{d,jk}+1\quad \forall j\in\{1,\ldots,5\},k\in\{1,2\}\tag{10.5}$$

**运动学约束**：根据模型准备部分，已经求出了各个物体的位置函数，即：

$$\begin{cases}\mathbf{P}_{m,1}(t)=\mathbf{P}_{m,1}(0)+\mathbf{v}_{m,1}\cdot t\\ \mathbf{P}_{\mathrm{c,j}}(t)=(P_{\mathrm{b,j},x},P_{\mathrm{b,j},y},P_{\mathrm{b,j},z}-v_{\mathrm{sink}}\cdot(t-t_{\mathrm{b,j}}))\quad t\geq t_{\mathrm{b,j}}\\ \mathbf{P}_{\mathrm{s,j}}(t)=\mathbf{P}_{fy,j}(t_{d,j})+\mathbf{v}_{fy,j}\cdot(t-t_{\mathrm{d,j}})+\frac{1}{2}\mathbf{g}\cdot(t-t_{\mathrm{d,j}})^2\quad t\geq t_{\mathrm{d,j}}\\ \mathbf{v}_{fy,j}=(v_{fy,j}\cos\theta_j,v_{fy,j}\sin\theta_j,0)\\ \mathbf{P}_{fy,j}(t)=\mathbf{P}_{fy,j}(0)+\mathbf{v}_{fy,j}\cdot t\end{cases}\tag{10.6}$$

### 10.1.4 模型汇总

综上，问题五针对多机多弹的投放策略优化模型可表述为：

$$\max\quad f(\mathcal{X})=\int_0^{\infty}Bool(t),dt$$

中国大学生在线 dxs.moe.gov.cn

$$
\text{s.t.}\begin{cases}
\mathbf{P_1} = \mathbf{P}_{m,i}(t) = \mathbf{P}_{m,i}(0) + \mathbf{v}_m \cdot t \\
\begin{cases}
\mathbf{P}_{\mathrm{c,jk}}(t) = (P_{\mathrm{b,jk},x}, P_{\mathrm{b,jk},y}, P_{\mathrm{b,jk},z} - v_{\mathrm{sink}} \cdot (t - t_{\mathrm{b,jk}})) \quad t \geq t_{\mathrm{b,jk}} \\
\mathbf{P}_{\mathrm{s,jk}}(t) = \mathbf{P}_{fy,i}(t_{d,jk}) + \mathbf{v}_{fy,i} \cdot (t - t_{\mathrm{d,jk}}) + \frac{1}{2}\mathbf{g} \cdot (t - t_{\mathrm{d,jk}})^2 \quad t \geq t_{\mathrm{d,jk}} \\
\mathbf{v}_{fy,i} = (v_{fy,i}\cos\theta_i, v_{fy,i}\sin\theta_i, 0) \\
\mathbf{P}_{fy,i}(t) = \mathbf{P}_{fy,i}(0) + \mathbf{v}_{fy,i} \cdot t
\end{cases} \\
70 \leq v_{fy,j} \leq 140, \quad 0 \leq \theta_j \leq 2\pi, \quad t_{d,jk}, t_{b,jk} \geq 0, \quad t_{b,jk} \geq t_{d,jk} \\
\sum_{j=1}^{5} a_{ij} \geq 1 \quad \forall i \\
\sum_{k=1}^{3} b_{jk} \leq 3 \quad \forall j \\
t_{d,j(k+1)} \geq t_{d,jk} + 1 \quad \forall j, k \\
Bool(t) = \begin{cases}
1, \quad \text{if } \forall i \in \{1,2,3\}, \forall P_2 \in \Omega, \quad \exists j \in \{1,\ldots,5\}, \exists k \in \{1,\ldots,3\} \\
\qquad \text{s.t.} \quad \overline{P_{m\,i}(t)P_2} \cap C(P_{c,jk}(t)) \neq \varnothing \\
0, \quad \text{otherwise}
\end{cases}
\end{cases}
\tag{10.7}
$$

## 10.2 模型求解

由于问题五的决策变量维度非常高（最多可达到 45 维），约束复杂，属于大规模组合优化问题，利用传统的优化算法难以直接求解。所以本文采用分层优化策略结合改进的遗传算法进行求解。分析各无人机与各导弹的初始位置示意图:

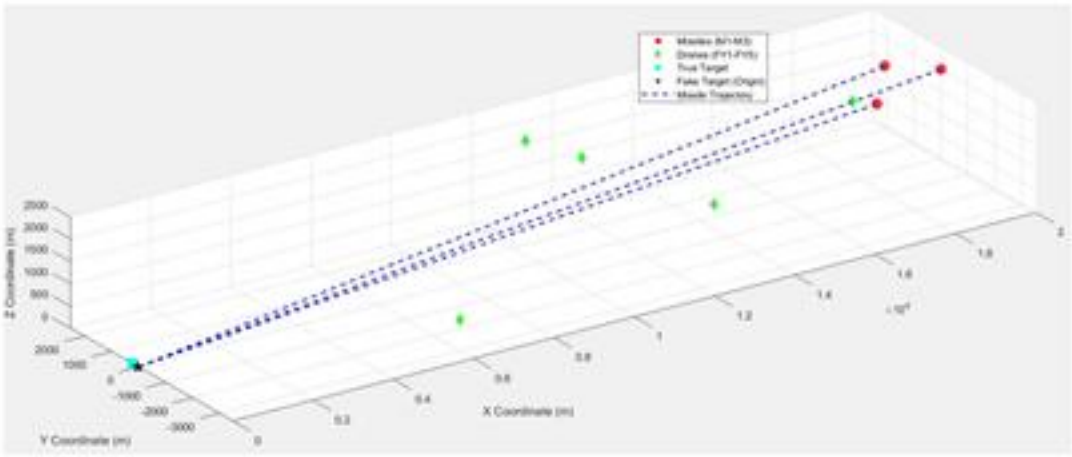



图 12: 各无人机与各导弹的初始位置示意图

关于这幅图，很容易产生一个想法：无人机应该去拦截在空间上距离它最近的那枚导弹，即 FY3 和 FY5 应该去拦截 M3，FY1 拦截 M1，FY2 和 FY4 拦截 M2。从算法角度来讲，如果能够提前确定各无人机应该投放几发烟幕弹去分别拦截哪枚导弹，那么会极大的提升后续的计算效率，以便更快地去确定无人机的航向、速度，烟幕弹的投放、引爆时间。下面将对如何确定每架无人机干扰哪枚导弹，以及每架无人机投放烟幕弹的数量进行说明。

### 10.2.1 任务分配说明

这一步的目的在于确定最优的任务分配方案，即**每架无人机 $j$ 干扰哪枚导弹 $i$，以及每架无人机投放烟幕弹的数量 $y_j$**。为了定义这个最优，这里假设无人机都以固定的 140m/s 进行匀速直线运动，航向沿着能够截击导弹的直线飞到导弹航迹正前方的一个参考点，这个参考点记为 $P_{ref}$，根据以上的假设能够快速计算该烟幕弹理论上最大的有效遮挡时长。令这个理论上最大的有效遮挡时长为 $c_{ij}$，下面说明 $c_{ij}$ 的计算方法；

$$
\begin{aligned}
T_{\text{fly},ij} &= \frac{\|\mathbf{P}_{\text{fy},i}(0) - \mathbf{P}_{\text{ref},j}\|}{140}, \\
T_{\text{cloud}} &= 20 \\
c_{ij} &= \max\big(0, T_{\text{cloud}} - T_{\text{fly},ij}\big)
\end{aligned}
\tag{10.8}
$$

$P_{ref,j}$ 可以取导弹 $i$ 初位置沿 $x$ 负方向偏移 5 k，无人机 $j$ 以最大速度 140 m/s 到该参考点即刻释放 1 枚弹并立即起爆，云团的寿命为 20s、之后按 3 m/s 匀速下沉。如果无人机在导弹袭来该点时已经在该点引爆烟幕弹，那么理论上可用时间窗口被飞行时间挤占的值还未到达 20s，理论最大有效遮挡时间就为 $T_{\text{cloud}} - T_{\text{fly},ij}$，而如果导弹比无人机先经过此点，那么烟幕弹就不能提供有效的遮挡，则 $c_{ij} = 0$。

如果把无人机 $j$ 发射向导弹 $i$ 的烟幕弹数量记为 $y_{ij}$，显然 $y_{ij}$ 是一个十五维的变量，也可以写成一个 $3 \times 5$ 的矩阵 $Y_{ij}$，那么通过改变矩阵元素的取值，计算无人机群对导弹群拦截的理论上的最大有效遮挡时间，就可以初步决定出最优的任务分配。寻找 $Y_{ij}$ 的步骤如下:

无人机群对导弹群拦截的理论上的有效遮挡时间可以记作 $\sum_{i=1}^{3}\sum_{j=1}^{5} c_{ij}\, y_{ij}$

在满足导弹的拦截已经无人机发射导弹数量的前提下，求解最优的分配方案的优化模型为:

$$
max \quad \sum_{i=1}^{3}\sum_{j=1}^{5} c_{ij}\, y_{ij}
$$

$$
\textbf{s.t.} \begin{cases}
\sum_{j=1}^{5} y_{ij} \le 3 \\
\sum_{i=1}^{3} y_{ij} \ge 1 \\
y_{ij} \le 3, y_{ij} \in N
\end{cases}
\tag{10.9}
$$

这是一个整数线性规划模型，可以利用分支定界法对其进行求解得到 $Y_{ij}$。

### 10.2.2 投放策略求解

有了最优的分配方案之后，最初的优化问题可以分解为 3 个独立的规模更小的子问题。每个子问题都只对应一枚来袭导弹。

子问题 1：优化导弹 M1 的无人机群的投放策略，目标是最大化对 M1 的有效遮蔽时长。

子问题 2：优化导弹 M2 的无人机群的投放策略，目标是最大化对 M2 的有效遮蔽

时长。

子问题 3：优化导弹 M3 的无人机群的投放策略，目标是最大化对 M3 的有效遮蔽时长。

求解出来之后再对这三个子问题得到的三个有效遮蔽时间区间取交集，最终的总有效遮蔽时间为这三个集合交集的总长度。

### 10.2.3 求解结果

通过上面的策略结合问题三的求解思路,求解得到总最长有效遮蔽时间为 **21.0770s**。其他的具体数据见附录以及文件 `result3.xlsx`。

# 十一、 模型的评价与推广

## 11.1 模型的评价

**模型的优点：**

1. 本文对导弹、无人机、云团及烟幕弹的运动情况做了详细分析，给出了精确的运动轨迹方程，同时没有简单地假设“导弹与云团中心距离小于 10 米即遮蔽”，而是敏锐地抓住了问题的核心　　需要遮蔽的是一个圆柱体，而不是一个点，体现了建模过程的数学严谨性于思考的深度。

2. 创新性地引入了布尔函数来清晰地定义遮蔽状态，这使得逻辑非常清晰，同时对模型的推导以及算法的实现给出了详细的说明，文章层级分明、逻辑很严谨。

3. 绘制了许多可视化图像让论文更加地直观，易于读者的理解。

**模型的缺点：**

1. 符号系统略显繁琐，定义了很多包含大量下标的符号，包含大量下标和字母的符号虽然精确，但严重影响了阅读的流畅性。

2. 算法流程说明太过冗长，本文对每一个算法的使用都进行了详细的说明，虽然易于读者理解，但是使文章的篇幅过于长会引起阅读的疲劳。

## 11.2 模型的推广

本文的模型是在理想情况下进行，没有考虑到空气的阻力、云团的扩散，在实际的问题研究中，这些都是不可忽略的因素。同时可以设计更高效的策略来在更大的范围内找到最优的策略使效益最大。

# 参考文献

[1]DeepSeek,DeepSeek-V3.1, 深度求索（DeepSeek），2025.09.05-2025.09.07）；

[2] 王庆华. 防空火箭烟幕弹干扰 [J]. 光电技术应用，2008，23（6）；

[3] 杨甲胜，耿晓蕾，顾文慧，等. 对烟幕遮蔽干扰投放参数的修正分析 [J]. 光电技术应用，2007，22（1）；

[4] 顾文慧，高红杰, 烟幕投放控制分析与设计 [J]. 光电技术应用,2014，29（2）

# 附 录

## A 支撑材料目录

| | | |
|---|---|---|
| AI使用说明 | DOCX 文档 | 119 KB |
| fifth | MATLAB Code | 3 KB |
| first | MATLAB Code | 2 KB |
| fourth_DE | MATLAB Code | 5 KB |
| fourth_grid | MATLAB Code | 4 KB |
| IntervalInter | MATLAB Code | 1 KB |
| IntervalUnion | MATLAB Code | 1 KB |
| isOverwhelm | MATLAB Code | 1 KB |
| result1 | XLSX 工作表 | 9 KB |
| result2 | XLSX 工作表 | 9 KB |
| result3 | XLSX 工作表 | 10 KB |
| second_greed_solve | MATLAB Code | 2 KB |
| second2_pso | MATLAB Code | 3 KB |
| third | MATLAB Code | 4 KB |

图 1: 支撑材料目录

| 无人机编号 | 无人机运动方向 | 无人机运动速度 (m/s) | 烟幕干扰弹编号 | 烟幕干扰弹投放点的x坐标 (m) | 烟幕干扰弹投放点的y坐标 (m) | 烟幕干扰弹投放点的z坐标 (m) | 烟幕干扰弹起爆点的x坐标 (m) | 烟幕干扰弹起爆点的y坐标 (m) | 烟幕干扰弹起爆点的z坐标 (m) | 有效干扰时长 (s) | 干扰的导弹编号 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FY1 | 179.09533 | 77.426826 | 1 | 17748.2697 | 0.926862 | 1800 | 17494.52913 | 4.923641 | 1747.361354 | 2.401596 | M1 |
| FY1 | 179.09533 | 77.426826 | 2 | 17670.85253 | 2.02934 | 1800 | 17417.11086 | 6.046119 | 1747.361354 | 3.68903 | M1 |
| FY1 | 179.09533 | 77.426826 | 3 | 17593.43535 | 3.261818 | 1800 | 17339.69378 | 7.268597 | 1747.361354 | 1.796792 | M1 |
| FY2 | 219.709224 | 73.177298 | 1 | 11103.46301 | 652.945053 | 1400 | 10737.9662 | 351.922684 | 1196.863813 | 3.335216 | M2 |
| FY2 | 219.709224 | 73.177298 | 2 | 11044.16796 | 606.182987 | 1400 | 10681.70514 | 305.170328 | 1196.863813 | 0 | M2 |
| FY2 | 219.709224 | 73.177298 | 3 | 10987.6729 | 559.440821 | 1400 | 10625.40809 | 258.417962 | 1196.863813 | 0 | M2 |
| FY3 | 78.122825 | 86.312749 | 1 | 6590.431802 | -192.698020 | 700 | 6590.431802 | -192.698020 | 700.000000 | 2.424903 | M3 |
| FY3 | 78.122825 | 86.312749 | 2 | 6608.196204 | -108.193140 | 700 | 6608.196204 | -108.193140 | 700 | 0 | M3 |
| FY3 | 78.122825 | 86.312749 | 3 | 6625.960605 | -23.728260 | 700 | 6625.960605 | -23.728260 | 700 | 0 | M3 |
| FY4 | 310.827001 | 134.705711 | 1 | 10360.07807 | 1351.793477 | 1800 | 12376.96178 | 406.293436 | 1378.400671 | 3.709183 | M2 |
| FY4 | 310.827001 | 134.705711 | 2 | 11648.1423 | 1249.834925 | 1800 | 12665.528 | 304.364883 | 1378.400671 | 0 | M2 |
| FY4 | 310.827001 | 134.705711 | 3 | 11736.20852 | 1147.906372 | 1800 | 12953.09453 | 202.436331 | 1378.400671 | 0 | M2 |
| FY5 | 263.664827 | 105.287847 | 1 | 12767.7556 | 178.346657 | 1300 | 12778.92299 | 8.717779 | 1287.124635 | 3.740325 | M1 |
| FY5 | 263.664827 | 105.287847 | 2 | 12788.15767 | 73.701760 | 1300 | 12797.30906 | -96.927118 | 1287.124635 | 0 | M1 |
| FY5 | 263.664827 | 105.287847 | 3 | 12774.51973 | -30.943136 | 1300 | 12755.68712 | -200.572014 | 1287.124635 | 0 | M1 |
| | 注：以x轴为正向，逆时针为为正，取值0~360（度）。 | | | | | | | | | | |

图 2: 问题五详细数据

中国大学生在线 dxs.moe.gov.cn

## B 代码 (matlab)

```
%文件名: first.m
%作用: 求解第一问, 计算有效遮蔽时间

clear;
clc;


g = 9.8;
drop_time = 3.6;
fly_time = 1.5;
last_time = 20;
sample_num = 2*200000;
sample_dots = zeros(sample_num,3);
idx = 1;
for h = [0,10]
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end

function [t1,t2] = get_last_time(v_fy1_vec,v_fy1_num,fly_time,drop_time
    ,sample_dots)
    if v_fy1_num>140 || v_fy1_num<70
        t1 = 0;
        t2 = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
    if drop_time>65 || drop_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
    if drop_time+fly_time>65
```

中国大学生在线 dxs.moe.gov.cn

```
        t1 = 0;
        t2 = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;
    fy1_pos0 = [17800,0,1800];
    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %%先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    while ~time_is_found
        if length(candidate_time)>100
            t1 = 0;
            t2 = 0;
            return;
        end
        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dots)
                time_is_found = true;
                overwhelm_time = t;
                break;
            end
        end
        if ~time_is_found
            new_candidate_time = zeros(1,2*length(candidate_time)-1);
            for i = 2:2:2*length(candidate_time)-2
```

中国大学生在线 dxs.moe.gov.cn

```
            new_candidate_time(i) = (candidate_time(i/2)+
                candidate_time(i/2+1))/2;
        end
        for i = 1:2:2*length(candidate_time)-1
            new_candidate_time(i) = candidate_time((i+1)/2);
        end
        candidate_time = new_candidate_time;
    end
end
start_time1 = fly_time+drop_time;
end_time1 = overwhelm_time;
while end_time1-start_time1>1e-8
    mid = (end_time1+start_time1)/2;
    m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m1_pos,smoke_pos,sample_dots)
        end_time1 = mid;
    else
        start_time1 = mid;
    end
end
t1 = (end_time1+start_time1)/2;

start_time2 = overwhelm_time;
end_time2 = end_time;
while end_time2-start_time2>1e-8
    mid = (end_time2+start_time2)/2;
    m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m1_pos,smoke_pos,sample_dots)
        start_time2 = mid;
    else
        end_time2 = mid;
    end
end
t2 = (end_time2+start_time2)/2;
end

m1_pos0 = [20000,0,2000];
```

中国大学生在线 dxs.moe.gov.cn

```
v_m1_vec = -m1_pos0/norm(m1_pos0);
v_m1_num = 300;
fy1_pos0 = [17800,0,1800];
v_fy1_vec = -fy1_pos0;
v_fy1_vec(3) = 0;
v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);
v_fy1_num = 120;

smoke_pos = fy1_pos0+(drop_time+fly_time).*v_fy1_num.*v_fy1_vec;
smoke_pos(3) = smoke_pos(3) - 0.5*g*drop_time^2;


[t1,t2] = get_last_time(v_fy1_vec,v_fy1_num,fly_time,drop_time,
    sample_dots);
fprintf("烟幕干扰弹投放点的坐标是(%f ,%f ,%f)\n",fy1_pos0+(fly_time).*
    v_fy1_num.*v_fy1_vec);
fprintf("烟幕干扰弹起爆点的坐标是(%f ,%f ,%f)\n",smoke_pos);
fprintf("烟幕干扰弹开始生效的时刻是%fs,失效的时刻是%fs, 有效遮蔽时间是%fs\
    n",t1,t2,t2-t1);
```

```
%文件:second_greed_solve.m
%作用: 使用贪心算法结合变步长搜索求解第二问

clear;
clc;

% if isempty(gcp('nocreate'))
%    parpool(8);
% end

sample_num = 2*300;
sample_dots = zeros(sample_num,3);
idx = 1;
for h = [0,10]
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end

function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dots)
```

中国大学生在线 dxs.moe.gov.cn

```
    [sample_num,~] = size(sample_dots);
    sample_num = floor(sample_num/2);

    d = sample_dots - repmat(missile_pos,sample_num*2,1);
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(repmat(oc,sample_num*2,1).*d,2);
    c = dot(oc,oc)-100;
    discriminant = b.^2 - 4 * a * c;

    if any(discriminant<0)
        overwhelmed = false;
        return;
    end
    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);
    overwhelmed = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed);
    if(norm(smoke_pos-missile_pos)<10)
        overwhelmed = false;
    end
end



function t = get_last_time(v_fy1_vec,v_fy1_num,fly_time,drop_time,
    sample_dots)
    if v_fy1_num>140 || v_fy1_num<70
        t = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t = 0;
        return;
    end
    if drop_time>65 || drop_time<0
        t = 0;
        return;
    end
```

中国大学生在线 dxs.moe.gov.cn

```
    if drop_time+fly_time>65
        t = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;
    fy1_pos0 = [17800,0,1800];
    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    start_time1 = fly_time+drop_time;
    end_time2 = end_time;

    while ~time_is_found
        if length(candidate_time)>32
            t1 = 0;
            t2 = 0;
            t = t2-t1;
            return;
        end
        idx = 2;
        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dots)
                time_is_found = true;
                overwhelm_time = t;
                start_time1 = candidate_time(idx-1);
```

```
            end_time2 = candidate_time(idx+1);
            break;
        end
        idx = idx+2;
    end
    if ~time_is_found
        new_candidate_time = zeros(1,2*length(candidate_time)-1);
        for i = 2:2:2*length(candidate_time)-2
            new_candidate_time(i) = (candidate_time(i/2)+
                candidate_time(i/2+1))/2;
        end
        for i = 1:2:2*length(candidate_time)-1
            new_candidate_time(i) = candidate_time((i+1)/2);
        end
        candidate_time = new_candidate_time;
    end
end

end_time1 = overwhelm_time;
while end_time1-start_time1>1e-7
    mid = (end_time1+start_time1)/2;
    m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m1_pos,smoke_pos,sample_dots)
        end_time1 = mid;

    else
        start_time1 = mid;
    end
end
t1 = (end_time1+start_time1)/2;

start_time2 = overwhelm_time;

while end_time2-start_time2>1e-7
    mid = (end_time2+start_time2)/2;
    m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m1_pos,smoke_pos,sample_dots)
```

```
            start_time2 = mid;
        else
            end_time2 = mid;
        end
    end
    t2 = (end_time2+start_time2)/2;
    t = t2-t1;
end

space = [2*pi,70,10,10]/100;
lb = [0,70,0,0];
ub = [2*pi,140,10,10];

longest_t = 0;
best_x = zeros(1,4);

for epoch = 1:1
    longest_ts = zeros(2000,1);
    best_xs = zeros(2000,4);
    parfor batch = 1:2000
        x = lb+rand(1,4).*(ub-lb);
        while get_last_time( [ cos(x(1)) , sin(x(1)),0],x(2),x(3),x(4),
            sample_dots)<=0
            x = lb+rand(1,4).*(ub-lb);
        end
        k = 0;
        while k<8
            ans_ischanged = false;
            for i = 1:4
                x_temp = x;
                x_temp(i) = x_temp(i)+space(i)/2^k;
                t = get_last_time( [ cos(x_temp(1)) , sin(x_temp(1)),0],
                    x_temp(2),x_temp(3),x_temp(4),sample_dots);
                if t>longest_ts(batch)
                    best_xs(batch,:) = x_temp;
                    longest_ts(batch) = t;
                    ans_ischanged = true;
                end
                x_temp = x;
                x_temp(i) = x_temp(i)-space(i)/2^k;
                t = get_last_time( [ cos(x_temp(1)) , sin(x_temp(1)),0],
```

中国大学生在线 dxs.moe.gov.cn

```
                    x_temp(2),x_temp(3),x_temp(4),sample_dots);
                if t>longest_ts(batch)
                    best_xs(batch,:) = x_temp;
                    longest_ts(batch) = t;
                    ans_ischanged = true;
                end
            end
            if ~ans_ischanged
                k = k+1;
            end
        end
    end
    [max_t,idx] = max(longest_ts);
    if max_t>longest_t
        longest_t = max_t
        best_x = best_xs(idx,:);
    end
    disp(epoch)
end
v_fy1_vec = [cos(best_x(1)),sin(best_x(1)),0];
drop_pos = [17800,0,1800]+ v_fy1_vec*best_x(2)*(best_x(3));
explode_pos = [17800,0,1800] + v_fy1_vec*best_x(2)*(best_x(3)+best_x(4)
    ) - 9.8*0.5*best_x(4)^2.*[0,0,1];
for i = 1:50000
    x = (1+randn(1,4)/200).*best_x;
    t = get_last_time( [cos(x(1)) , sin(x(1)),0],x(2),x(3),x(4),
        sample_dots);
    if (t-longest_t>0.000625)
        disp(t);
        disp(x);
    end
end
```

```
%文件:second_pso.m
%作用: 使用粒子群算法求解第二问

clear;
clc;

% 初始化采样点
sample_num = 2*300;
```

中国大学生在线 dxs.moe.gov.cn

```
sample_dots = zeros(sample_num,3);
idx = 1;
for h = [0,10]
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end

% 定义目标函数
ObjectiveFunction = @(x) -get_last_time([cos(x(1)), sin(x(1)), 0], x(2)
    , x(3), x(4), sample_dots);

nVar = 4;           % [角度, 速度, 飞行时间, 投放时间]
lb = [0, 70, 0, 0];
ub = [2*pi, 140, 65, 65];

maxIter = 1000;
nParticles = 600;
w = 0.9;
w_damp = 0.996;
c1 = 2;
c2 = 2;


vel_max = (ub - lb) * 0.05;
vel_min = -vel_max;

% 初始化粒子结构体
pa.pos = [];
pa.v = [];
pa.cost = [];
pa.best.pos = [];
pa.best.cost = [];

% 创建粒子群
pa = repmat(pa, nParticles, 1);


globalBest.cost = inf;
globalBest.pos = [];
```

中国大学生在线 dxs.moe.gov.cn

```
parfor i = 1:nParticles
    while true
        pa(i).pos = lb + (ub-lb) .* rand(1, nVar);
        if pa(i).pos(3) + pa(i).pos(4) <= 65
            initial_cost = ObjectiveFunction(pa(i).pos);
            if initial_cost ~= 0 || i>0.1*nParticles
                break;
            end

        end
    end

    % 初始化速度
    pa(i).v = zeros(1, nVar);

    % 计算初始代价值
    pa(i).cost = initial_cost;

    % 初始化个体最优
    pa(i).best.pos = pa(i).pos;
    pa(i).best.cost = pa(i).cost;
end
for i = 1:nParticles
    % 更新全局最优
    if pa(i).best.cost < globalBest.cost
        globalBest = pa(i).best;
    end
end

% 存储每次迭代的全局最优值，用于绘图
bestCosts = zeros(maxIter, 1);


for it = 1:maxIter
    for i = 1:nParticles
        % 更新速度
        pa(i).v = w * pa(i).v ...
            + c1 * rand(1, nVar) .* (pa(i).best.pos - pa(i).pos) ...
            + c2 * rand(1, nVar) .* (globalBest.pos - pa(i).pos);

        % 限制速度
```

中国大学生在线 dxs.moe.gov.cn

```
        pa(i).v = max(pa(i).v, vel_min);
        pa(i).v = min(pa(i).v, vel_max);

        % 更新位置
        pa(i).pos = pa(i).pos + pa(i).v;

        % 边界处理
        pa(i).pos = max(pa(i).pos, lb);
        pa(i).pos = min(pa(i).pos, ub);

        if pa(i).pos(3) + pa(i).pos(4) > 65
            sum_val = pa(i).pos(3) + pa(i).pos(4);
            pa(i).pos(3) = pa(i).pos(3) * 65 / sum_val;
            pa(i).pos(4) = pa(i).pos(4) * 65 / sum_val;
        end

        pa(i).cost = ObjectiveFunction(pa(i).pos);
    end

    for i = 1:nParticles
        % 更新个体最优
        if pa(i).cost < pa(i).best.cost
            pa(i).best.pos = pa(i).pos;
            pa(i).best.cost = pa(i).cost;

            % 更新全局最优
            if pa(i).best.cost < globalBest.cost
                globalBest = pa(i).best;
            end
        end
    end

    bestCosts(it) = globalBest.cost;
    fprintf('迭代次数 %d: 最优值为 %f\n', it, -globalBest.cost);
    w = w * w_damp;
end


best_solution = globalBest.pos;
longest_t = -globalBest.cost;

```

中国大学生在线 dxs.moe.gov.cn

```
fprintf('最优持续时间: %f s\n', longest_t);
fprintf('最优解 (x): \n');
fprintf('速度方向(弧度): %f\n', best_solution(1));
fprintf('速度大小: %f\n', best_solution(2));
fprintf('投弹时间: %f s\n', best_solution(3));
fprintf('引爆延迟: %f s\n', best_solution(4));

% 绘制收敛曲线
figure;
plot(-bestCosts, 'LineWidth', 2);
xlabel('迭代次数');
ylabel('最优持续时间 (s)');
title('PSO 收敛曲线');
grid on;



function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dots)
    [sample_num,~] = size(sample_dots);

    d = sample_dots - repmat(missile_pos,sample_num,1);
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(repmat(oc,sample_num,1).*d,2);
    c = dot(oc,oc)-100;
    discriminant = b.^2 - 4 * a * c;

    if any(discriminant<0)
        overwhelmed = false;
        return;
    end

    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);

    overwhelmed_points = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed_points(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed_points);
end

```

中国大学生在线 dxs.moe.gov.cn

```
function t = get_last_time(v_fy1_vec,v_fy1_num,fly_time,drop_time,
    sample_dots)
    if v_fy1_num>140 || v_fy1_num<70
        t = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t = 0;
        return;
    end
    if drop_time>65 || drop_time<0
        t = 0;
        return;
    end
    if drop_time+fly_time>65
        t = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;
    fy1_pos0 = [17800,0,1800];
    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %%先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    while ~time_is_found
        if length(candidate_time)>32
            t1 = 0;
            t2 = 0;
            t = t2-t1;
```

```
            return;
        end
        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dots)
                time_is_found = true;
                overwhelm_time = t;
                break;
            end
        end
        if ~time_is_found
            new_candidate_time = zeros(1,2*length(candidate_time)-1);
            for i = 2:2:2*length(candidate_time)-2
                new_candidate_time(i) = (candidate_time(i/2)+
                    candidate_time(i/2+1))/2;
            end
            for i = 1:2:2*length(candidate_time)-1
                new_candidate_time(i) = candidate_time((i+1)/2);
            end
            candidate_time = new_candidate_time;
        end
    end
    start_time1 = fly_time+drop_time;
    end_time1 = overwhelm_time;
    while end_time1-start_time1>1e-8
        mid = (end_time1+start_time1)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dots)
            end_time1 = mid;
        else
            start_time1 = mid;
        end
    end
    t1 = (end_time1+start_time1)/2;

    start_time2 = overwhelm_time;
    end_time2 = end_time;
```

中国大学生在线 dxs.moe.gov.cn

```
    while end_time2-start_time2>1e-7
        mid = (end_time2+start_time2)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dots)
            start_time2 = mid;
        else
            end_time2 = mid;
        end
    end
    t2 = (end_time2+start_time2)/2;
    t = t2-t1;
end
```

```
%文件:third.m
%作用: 使用差分进化算法求解第三题
clear;
clc;

if isempty(gcp('nocreate'))
    parpool(8);
end
sample_num = 5*50;
sample_dots = zeros(sample_num,3);
idx = 1;
for h = linspace(0,10,5)
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end
%%

function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dots)
    [sample_num,-] = size(sample_dots);
    d = sample_dots - repmat(missile_pos,sample_num,1);
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(repmat(oc,sample_num,1).*d,2);
    c = dot(oc,oc)-100;
```

中国大学生在线 dxs.moe.gov.cn

```
    discriminant = b.^2 - 4 .* a * c;

    if any(discriminant<0)
        overwhelmed = false;
        return;
    end
    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);
    overwhelmed = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed);
    % if(norm(smoke_pos-missile_pos)<10)
    %     overwhelmed = false;
    % end
end



function [t1,t2] = get_last_time(v_fy1_vec,v_fy1_num,fly_time,drop_time
    ,sample_dot)
    if v_fy1_num>140 || v_fy1_num<70
        t1 = 0;
        t2 = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
    if drop_time>65 || drop_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
    if drop_time+fly_time>65
        t1 = 0;
        t2 = 0;
        return;
    end
```

中国大学生在线 dxs.moe.gov.cn

```
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;
    fy1_pos0 = [17800,0,1800];
    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %%先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    start_time1 = start_time;
    end_time2 = end_time;
    while ~time_is_found
        if length(candidate_time)>63
            t1 = 0;
            t2 = 0;
            return;
        end
        idx = 2;

        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dot)
                time_is_found = true;
                overwhelm_time = t;
                start_time1 = candidate_time(idx-1);
                end_time2 = candidate_time(idx+1);
                break;
            end
            idx = idx+2;
        end
```

中国大学生在线 dxs.moe.gov.cn

```
        if ~time_is_found
            new_candidate_time = zeros(1,2*length(candidate_time)-1);
            for i = 2:2:2*length(candidate_time)-2
                new_candidate_time(i) = (candidate_time(i/2)+
                    candidate_time(i/2+1))/2;
            end
            for i = 1:2:2*length(candidate_time)-1
                new_candidate_time(i) = candidate_time((i+1)/2);
            end
            candidate_time = new_candidate_time;
        end
    end
    end_time1 = overwhelm_time;
    while end_time1-start_time1>1e-8
        mid = (end_time1+start_time1)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dot)
            end_time1 = mid;
        else
            start_time1 = mid;
        end
    end
    t1 = (end_time1+start_time1)/2;

    start_time2 = overwhelm_time;
    while end_time2-start_time2>1e-8
        mid = (end_time2+start_time2)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dot)
            start_time2 = mid;
        else
            end_time2 = mid;
        end
    end
    t2 = (end_time2+start_time2)/2;
end

```

中国大学生在线 dxs.moe.gov.cn

```
variSize = 8;
maxPopSize = 100;
maxIters = 500;
actualIter = 0;
ub = [2*pi,140,65,65,65,65,65,65];
lb = [0,70,0,0,0,0,0,0];
F = 0.5;
CR = 0.9;
elitRemain = round(maxPopSize/3);
phiPenality = 1e1;

%%初始化种群

Pop = zeros(maxPopSize,variSize);
for i = 1:maxPopSize
    x = lb+rand(1,variSize).*(ub-lb);
    while allCons(x)>0
        x = lb+rand(1,variSize).*(ub-lb);
    end
    Pop(i,:) = x;
end


%%主循环
minFit = zeros(maxIters,1);
minAns = zeros(maxIters,variSize);
minconsPhi = zeros(maxIters,1);
newFit = zeros(maxPopSize,1);
newConsphis = zeros(maxPopSize,1);

for iter = 1:maxIters
    disp(iter);
    if all(newFit==0)
        Fit = myObj(Pop,sample_dots);
    else
        Fit = newFit;
    end
    if all(newConsphis == 0)
        consphis = allCons(Pop);
    else
        consphis = newConsphis;
```

中国大学生在线 dxs.moe.gov.cn

```
    end
    feasibleIdx = find(consphis < 1e-8);

    if ~isempty(feasibleIdx)
        % 2. 如果存在可行解，则在这些可行解中，根据目标函数值 Fit 找到最优的
        %   [minF_i, tempIdx] = min(Fit(feasibleIdx)); % 找到可行解中的最小值和它的局部索引

        [minF_i, tempIdx] = min(Fit(feasibleIdx));

        %   minIdx 是最优解在原始种群 Pop 中的全局索引
        minIdx = feasibleIdx(tempIdx);
    else
        % 3. 如果没有任何可行解，则退而求其次，在所有个体中找到约束违反度 consphis 最小的
        minIdx = 1;

        %   此时的 minF_i 是这个约束违反度最小的个体的目标函数值
        minF_i = 0;
    end
    minFit(iter) = minF_i;
    minAns(iter,:) = Pop(minIdx,:);
    minconsPhi(iter) = consphis(minIdx);
    actualIter = iter;
    if iter>20 && -sum(minFit(iter-19:iter)-minFit(iter-20:iter-1),"all")<1e-6;
        break;
    end
    % if iter>1 && (minFit(iter-1) - minFit(iter) < 1e-7) && minconsPhi(iter) < 1e-8
    %     break;
    % end
    newPop = Pop;
    parfor i = 1:maxPopSize
        [i1,i2,i3] = getDiffIdx(1:maxPopSize,i);
        v = Pop(i1,:) + F * (Pop(i2,:) - Pop(i3,:)); %变异向量
        u = Pop(i,:);
        mustCross = randi(variSize);
        %%
        needCross = rand(1,variSize)<CR;
```

```
        needCross(mustCross) = true;
        u(needCross) = v(needCross);
        u = max(min(u, ub), lb);
        uf = myObj(u,sample_dots);
        uf = reshape(uf,1,1);
        uConsPhi = allCons(u);
        uConsPhi = reshape(uConsPhi,1,1);
        if uConsPhi<1 && consphis(i)<1
            if uf<Fit(i)
                newPop(i,:) = u;
            end
        else
            if uConsPhi<consphis(i)
                newPop(i,:) = u;
            end
        end
    end
    elitCandidates = Pop(consphis<2,:);
    elitFit = Fit(consphis<2);
    elitCons = consphis(consphis<2);
    if ~isempty(elitCandidates)
        [~, sortedIdx] = sort(elitFit, 'ascend');
        elitCount = min(length(sortedIdx),elitRemain);
        elitPreserve = elitCandidates(sortedIdx(1:elitCount), :);
        elitFitPreserve = elitFit(sortedIdx(1:elitCount));
        elitConsPreserve = elitCons(sortedIdx(1:elitCount));
        newFit = myObj(newPop,sample_dots);
        newConsphis = allCons(newPop);

        isViolated = (newConsphis > 1e-5);
        sortMat = [isViolated, newConsphis, newFit];

        [~, sortedIdx] = sortrows(sortMat);


        numToReplace = elitCount;
        worstIdx = sortedIdx(end - numToReplace + 1 : end);

        newPop(worstIdx, :) = elitPreserve;
        newFit(worstIdx) = elitFitPreserve;
        newConsphis (worstIdx) = elitConsPreserve;
```

中国大学生在线 dxs.moe.gov.cn

```
    end
    Pop = newPop;
    disp(minFit(actualIter));
end


function [i1,i2,i3] = getDiffIdx(candidates,i)
    candidates = setdiff(candidates,i);
    i1 = candidates(randi(length(candidates)));
    candidates = setdiff(candidates,i1);
    i2 = candidates(randi(length(candidates)));
    i3 = candidates(randi(length(candidates)));
end


function fits = myObj(Pop,sample_dots)
    [sample_num,~] = size(sample_dots);
    popsize = size(Pop);
    len = popsize(1);
    fits = zeros(len,1);
    v_fy1_vec = [cos(Pop(:,1)),sin(Pop(:,1)),zeros(len,1)];
    v_fy1_num = Pop(:,2);
    fly_time1 = Pop(:,3);
    drop_time1 = Pop(:,4);

    fly_time2 = Pop(:,5)+fly_time1+1;
    drop_time2 = Pop(:,6);

    fly_time3 = Pop(:,7)+fly_time2+1;
    drop_time3 = Pop(:,8);

    for i = 1:len
        sol_union = [];
        for j = 1:sample_num
            [t1,t2] = get_last_time(v_fy1_vec(i,:),v_fy1_num(i),fly_time1
                (i),drop_time1(i),sample_dots(j,:));
            [t3,t4] = get_last_time(v_fy1_vec(i,:),v_fy1_num(i),fly_time2
                (i),drop_time2(i),sample_dots(j,:));
            [t5,t6] = get_last_time(v_fy1_vec(i,:),v_fy1_num(i),fly_time3
                (i),drop_time3(i),sample_dots(j,:));
            if isempty(sol_union)
```

dxs.moe.gov.cn

```
            if j==1
                sol_union = IntervalUnion([t1,t2;t3,t4;t5,t6]);
            else
                break;
            end
        else
            sol_union = IntervalInter(sol_union,IntervalUnion([t1,t2;
                t3,t4;t5,t6]));
        end
    end
    fits(i) = sum(sol_union(:,1)-sol_union(:,2),"all");
  end
end

function consphis = allCons(Pop)
    popsize = size(Pop);
    len = popsize(1);
    ub = repmat([2*pi,140,65,65,65,65,65,65],len,1);
    lb = repmat([0,70,0,0,0,0,0,0],len,1);


    fly_time1 = Pop(:,3);
    drop_time1 = Pop(:,4);

    fly_time2 = Pop(:,5)+fly_time1+1;
    drop_time2 = Pop(:,6);

    fly_time3 = Pop(:,7)+fly_time2+1;
    drop_time3 = Pop(:,8);

    function smoke_z = get_smoke_z(drop_time,popsize)
        g = 9.8;
        smoke_z0 = 1800*ones(popsize(1),1);
        smoke_z = smoke_z0 - 0.5*g*drop_time.^2;
    end

    consphis = sum((Pop<lb)+(Pop>ub),2);
    consphis = consphis+(fly_time1+drop_time1 > 65)...
    +(fly_time2+drop_time2 > 65)...
    +(fly_time3+drop_time3 > 65)...
    +(get_smoke_z(drop_time1,popsize)<10)...
```

中国大学生在线 dxs.moe.gov.cn

```
    +(get_smoke_z(drop_time2,popsize)<10)...
    +(get_smoke_z(drop_time3,popsize)<10);
end
g = 9.8;
u = minAns(actualIter,:);
v_fy_vec = [cos(u(1)),sin(u(1)),0];
v_fy_num = u(2);
fy1_pos0 = [17800,0,1800];

drop_time1 = u(3);
drop_time2 = drop_time1+1+u(5);
drop_time3 = drop_time2+1+u(7);
explode_time1 = drop_time1+u(4);
explode_time2 = drop_time2+u(6);
explode_time3 = drop_time3+u(8);
m1_pos0 = [20000,0,2000];
v_m1_num = 300;
v_m1_vec = -m1_pos0/norm(m1_pos0);

time_steps = 1000000;
state = zeros(sample_num,3);
fullyOW1 = false;
fullyOW2 = false;
fullyOW3 = false;
partOW = false;
start_t1 = 0;
end_t1 = 0;
start_t2 = 0;
end_t2 = 0;
start_t3 = 0;
end_t3 = 0;

start_t4 = 0;
end_t4 = 0;
for t = linspace(0,30,time_steps)
    m1_pos = m1_pos0+t*v_m1_vec*v_m1_num;
    for j = 1:sample_num
        if t > explode_time1
            smoke1_pos = fy1_pos0+explode_time1*v_fy_vec*v_fy_num
                +[0,0,-0.5*g*(explode_time1-drop_time1)^2]+[0,0,-3*(t-
                explode_time1)];
```

```
            if(isOverwhelm(m1_pos,smoke1_pos,sample_dots(j,:)))
                state(j,1)=1;
            else
                state(j,1)=0;
            end
        end

        if t > explode_time2
            smoke2_pos = fy1_pos0+explode_time2*v_fy_vec*v_fy_num
                +[0,0,-0.5*g*(explode_time2-drop_time2)^2]+[0,0,-3*(t-
                explode_time2)];
            if(isOverwhelm(m1_pos,smoke2_pos,sample_dots(j,:)))
                state(j,2)=1;
            else
                state(j,2)=0;
            end
        end

        if t > explode_time3
            smoke3_pos = fy1_pos0+explode_time3*v_fy_vec*v_fy_num
                +[0,0,-0.5*g*(explode_time3-drop_time3)^2]+[0,0,-3*(t-
                explode_time3)];
            if(isOverwhelm(m1_pos,smoke3_pos,sample_dots(j,:)))
                state(j,3)=1;
            else
                state(j,3)=0;
            end
        end
    end

    if all(state(:,1)==1)
        if ~fullyOW1
            fprintf("%fs时第一个烟幕干扰弹开始生效\n",t);
            fullyOW1 = true;
        end
    else
        if fullyOW1
            fprintf("%fs时第一个烟幕干扰弹开始失效\n",t);
            fullyOW1 = false;
        end
    end
```

中国大学生在线 dxs.moe.gov.cn

```
    if all(state(:,2)==1)
        if ~fullyOW2
            fprintf("%fs时第二个烟幕干扰弹开始生效\n",t);
            fullyOW2 = true;
        end
    else
        if fullyOW2
            fprintf("%fs时第二个烟幕干扰弹开始失效\n",t);
            fullyOW2 = false;
        end
    end

    if all(state(:,3)==1)
        if ~fullyOW3
            fprintf("%fs时第三个烟幕干扰弹开始生效\n",t);
            fullyOW3 = true;
        end
    else
        if fullyOW3
            fprintf("%fs时第三个烟幕干扰弹开始失效\n",t);
            fullyOW3 = false;
        end
    end

    if all(sum(state,2)>=1) && ~all(state(:,1)==1) && ~all(state(:,2)
        ==1) && ~all(state(:,3)==1)
        if ~partOW
            fprintf("%fs时只有多个烟幕干扰弹才能生效\n",t);
            partOW = true;
        end
    else
        if partOW
            fprintf("%fs时即使有多个烟幕干扰弹都无法生效\n",t);
            partOW = false;
        end
    end

end

fprintf("最大有效遮蔽时间是%f\n",-minFit(actualIter))
```

中国大学生在线 dxs.moe.gov.cn

```
fprintf("无人机速度方向方向角为: %f° ,速度大小为: %fm/s\n",u(1)*360/pi,u(2));

fy1_fly_pos = fy1_pos0+u(3)*u(2).*v_fy_vec;
fy1_explode_pos = fy1_fly_pos+u(4)*u(2)*v_fy_vec+[0,0,-0.5*u(4)^2*g];
fprintf("第一颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第一颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy1_fly_pos,fy1_explode_pos);

fy1_fly_pos = fy1_pos0+(u(3)+u(5)+1)*u(2).*v_fy_vec;
fy1_explode_pos = fy1_fly_pos+u(6)*u(2)*v_fy_vec+[0,0,-0.5*u(6)^2*g];
fprintf("第二颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第二颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy1_fly_pos,fy1_explode_pos);

fy1_fly_pos = fy1_pos0+(u(3)+u(5)+u(7)+2)*u(2).*v_fy_vec;
fy1_explode_pos = fy1_fly_pos+u(8)*u(2)*v_fy_vec+[0,0,-0.5*u(8)^2*g];
[t1,t2] = get_last_time(v_fy_vec,u(2),u(3)+u(5)+u(7)+2,u(8),sample_dots);
fprintf("第三颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第三颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy1_fly_pos,fy1_explode_pos);
```

```
%文件: fourth_DE.m
%作用: 使用差分进化算法求解第四问。

clear;
clc;

if isempty(gcp('nocreate'))
    parpool(8);
end
sample_num = 3*10;
sample_dots = zeros(sample_num,3);
idx = 1;
for h = linspace(0,10,3)
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end
%%

function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dot)
```

中国大学生在线 dxs.moe.gov.cn

```
    d = sample_dot - missile_pos;
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(oc.*d,2);
    c = dot(oc,oc)-100;
    discriminant = b.^2 - 4 * a * c;

    if any(discriminant<0)
        overwhelmed = false;
        return;
    end
    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);
    overwhelmed = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed);
    % if(norm(smoke_pos-missile_pos)<10)
    %     overwhelmed = false;
    % end
end



function [t1,t2] = get_last_time(fy1_pos0,v_fy1_vec,v_fy1_num,fly_time,
    drop_time,sample_dot)
    if v_fy1_num>140 || v_fy1_num<70
        t1 = 0;
        t2 = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
    if drop_time>65 || drop_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
```

中国大学生在线 dxs.moe.gov.cn

```
    if drop_time+fly_time>65
        t1 = 0;
        t2 = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;
    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %%先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    start_time1 = start_time;
    end_time2 = end_time;
    while ~time_is_found
        if length(candidate_time)>63
            t1 = 0;
            t2 = 0;
            return;
        end
        idx = 2;

        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dot)
                time_is_found = true;
                overwhelm_time = t;
                start_time1 = candidate_time(idx-1);
                end_time2 = candidate_time(idx+1);
```

```
            break;
        end
        idx = idx+2;
    end
    if ~time_is_found
        new_candidate_time = zeros(1,2*length(candidate_time)-1);
        for i = 2:2:2*length(candidate_time)-2
            new_candidate_time(i) = (candidate_time(i/2)+
                candidate_time(i/2+1))/2;
        end
        for i = 1:2:2*length(candidate_time)-1
            new_candidate_time(i) = candidate_time((i+1)/2);
        end
        candidate_time = new_candidate_time;
    end
end
end_time1 = overwhelm_time;
while end_time1-start_time1>1e-8
    mid = (end_time1+start_time1)/2;
    m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m1_pos,smoke_pos,sample_dot)
        end_time1 = mid;
    else
        start_time1 = mid;
    end
end
t1 = (end_time1+start_time1)/2;

start_time2 = overwhelm_time;
while end_time2-start_time2>1e-8
    mid = (end_time2+start_time2)/2;
    m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m1_pos,smoke_pos,sample_dot)
        start_time2 = mid;
    else
        end_time2 = mid;
    end
```

中国大学生在线 dxs.moe.gov.cn

```
    end
    t2 = (end_time2+start_time2)/2;
end

%角度，速度，drop_time,explode_time
variSize = 12;
maxPopSize = 800;
maxIters = 300;
actualIter = 0;
ub = [2*pi,2*pi,2*pi,140,140,140,65,65,65,65,65,65];
lb = [0,0,0,70,70,70,0,0,0,0,0,0];
F = 0.5;
CR = 0.9;
elitRemain = round(maxPopSize/30);


%%初始化种群

Pop = zeros(maxPopSize,variSize);
for i = 1:maxPopSize
    x = lb+rand(1,variSize).*(ub-lb);
    while allCons(x)>0
        x = lb+rand(1,variSize).*(ub-lb);
    end
    Pop(i,:) = x;
end

%best1 = [0.0902,136.3663,0.8977,0.0497];
%Pop(1:end/4,[1,4,7,10]) = repmat(best1,maxPopSize/4,1);
%Pop(1:end/4,[2,3]) = repmat([2*pi-0.6747,1.2490],maxPopSize/4,1);

%%主循环
minFit = zeros(maxIters,1);
minAns = zeros(maxIters,variSize);
minconsPhi = zeros(maxIters,1);
newFit = zeros(maxPopSize,1);
newConsphis = zeros(maxPopSize,1);

for iter = 1:maxIters
    disp(iter);
    if all(newFit==0)
```

中国大学生在线 dxs.moe.gov.cn

```
        Fit = myObj(Pop,sample_dots);
    else
        Fit = newFit;
    end
    if all(newConsphis == 0)
        consphis = allCons(Pop);
    else
        consphis = newConsphis;
    end
    feasibleIdx = find(consphis < 1e-8);

    if ~isempty(feasibleIdx)
        % 2. 如果存在可行解，则在这些可行解中，根据目标函数值 Fit 找到最优的
        %   [minF_i, tempIdx] = min(Fit(feasibleIdx)); % 找到可行解中的最小值和它的局部索引

        [minF_i, tempIdx] = min(Fit(feasibleIdx));

        %   minIdx 是最优解在原始种群 Pop 中的全局索引
        minIdx = feasibleIdx(tempIdx);
    else
        % 3. 如果没有任何可行解，则退而求其次，在所有个体中找到约束违反度 consphis 最小的
        minIdx = 1;

        %   此时的 minF_i 是这个约束违反度最小的个体的目标函数值
        minF_i = 0;
    end
    disp(minF_i);
    minFit(iter) = minF_i;
    minAns(iter,:) = Pop(minIdx,:);
    minconsPhi(iter) = consphis(minIdx);
    actualIter = iter;
    if iter>15 && -sum(minFit(iter-14:iter)-minFit(iter-15:iter-1),"all")<(1e-5)/15
        break;
    end
    % if iter>1 && (minFit(iter-1) - minFit(iter) < 1e-7) && minconsPhi(iter) < 1e-8
    %   break;
```

```
    % end
    newPop = Pop;
    parfor i = 1:maxPopSize
        [i1,i2,i3] = getDiffIdx(1:maxPopSize,i);
        v = Pop(i1,:) + F * (Pop(i2,:) - Pop(i3,:)); %变异向量
        u = Pop(i,:);
        mustCross = randi(variSize);
        needCross = rand(1,variSize)<CR;
        needCross(mustCross) = true;
        u(needCross) = v(needCross);
        u = max(min(u, ub), lb);
        uf = myObj(u,sample_dots);
        uf = reshape(uf,1,1);
        uConsPhi = allCons(u);
        uConsPhi = reshape(uConsPhi,1,1);
        if uConsPhi<1 && consphis(i)<1
            if uf<Fit(i)
                newPop(i,:) = u;
            end
        else
            if uConsPhi<consphis(i)
                newPop(i,:) = u;
            end
        end
    end
    elitCandidates = Pop(consphis<2,:);
    elitFit = Fit(consphis<2);
    elitCons = consphis(consphis<2);
    if ~isempty(elitCandidates)
        [~, sortedIdx] = sort(elitFit, 'ascend');
        elitCount = min(length(sortedIdx),elitRemain);
        elitPreserve = elitCandidates(sortedIdx(1:elitCount), :);
        elitFitPreserve = elitFit(sortedIdx(1:elitCount));
        elitConsPreserve = elitCons(sortedIdx(1:elitCount));
        newFit = myObj(newPop,sample_dots);
        newConsphis = allCons(newPop);

        isViolated = (newConsphis > 1e-5);
        sortMat = [isViolated, newConsphis, newFit];

        [~, sortedIdx] = sortrows(sortMat);
```

中国大学生在线 dxs.moe.gov.cn

```
        numToReplace = elitCount;
        worstIdx = sortedIdx(end - numToReplace + 1 : end);

        newPop(worstIdx, :) = elitPreserve;
        newFit(worstIdx) = elitFitPreserve;
        newConsphis (worstIdx) = elitConsPreserve;
    end
    Pop = newPop;
end


function [i1,i2,i3] = getDiffIdx(candidates,i)
    candidates = setdiff(candidates,i);
    i1 = candidates(randi(length(candidates)));
    candidates = setdiff(candidates,i1);
    i2 = candidates(randi(length(candidates)));
    i3 = candidates(randi(length(candidates)));
end


function fits = myObj(Pop, sample_dots)
    fy1_pos0 = [17800, 0, 1800];
    fy2_pos0 = [12000, 1400, 1400];
    fy3_pos0 = [6000, -3000, 700];
    [sample_num, ~] = size(sample_dots);
    popsize = size(Pop);
    len = popsize(1);
    fits = zeros(len, 1);
    v_fy1_vec = [cos(Pop(:, 1)), sin(Pop(:, 1)), zeros(len, 1)];
    v_fy2_vec = [cos(Pop(:, 2)), sin(Pop(:, 2)), zeros(len, 1)];
    v_fy3_vec = [cos(Pop(:, 3)), sin(Pop(:, 3)), zeros(len, 1)];
    v_fy1_num = Pop(:, 4);
    v_fy2_num = Pop(:, 5);
    v_fy3_num = Pop(:, 6);
    fly_time1 = Pop(:, 7);
    drop_time1 = Pop(:, 10);
    fly_time2 = Pop(:, 8);
    drop_time2 = Pop(:, 11);
    fly_time3 = Pop(:, 9);
```

中国大学生在线 dxs.moe.gov.cn

```
    drop_time3 = Pop(:, 12);

    for i = 1:len
        % 为每个采样点计算其总的有效遮蔽区间（并集）
        unions_per_sample = cell(sample_num, 1);
        for j = 1:sample_num
            [t1, t2] = get_last_time(fy1_pos0, v_fy1_vec(i, :), v_fy1_num(i), fly_time1(i), drop_time1(i), sample_dots(j, :));
            [t3, t4] = get_last_time(fy2_pos0, v_fy2_vec(i, :), v_fy2_num(i), fly_time2(i), drop_time2(i), sample_dots(j, :));
            [t5, t6] = get_last_time(fy3_pos0, v_fy3_vec(i, :), v_fy3_num(i), fly_time3(i), drop_time3(i), sample_dots(j, :));
            unions_per_sample{j} = IntervalUnion([t1, t2; t3, t4; t5, t6]);
        end

        % 计算所有采样点有效区间的交集
        final_intersection = unions_per_sample{1};
        for j = 2:sample_num
            if isempty(final_intersection)
                break;
            end
            final_intersection = IntervalInter(final_intersection, unions_per_sample{j});
        end

        if isempty(final_intersection)
            fits(i) = 0;
        else
            fits(i) = sum(final_intersection(:, 1) - final_intersection(:, 2), "all");
        end
    end
end

function consphis = allCons(Pop)
    popsize = size(Pop);
    len = popsize(1);
    ub = repmat([2*pi,2*pi,2*pi,140,140,140,65,65,65,65,65,65],len,1);
    lb = repmat([0,0,0,70,70,70,0,0,0,0,0,0],len,1);

```

中国大学生在线 dxs.moe.gov.cn

```
    fly_time1 = Pop(:,7);
    drop_time1 = Pop(:,10);

    fly_time2 = Pop(:,8);
    drop_time2 = Pop(:,11);

    fly_time3 = Pop(:,9);
    drop_time3 = Pop(:,12);

    function smoke_z = get_smoke_z(h,drop_time,popsize)
        g = 9.8;
        smoke_z0 = h*ones(popsize(1),1);
        smoke_z = smoke_z0 - 0.5*g*drop_time.^2;
    end
    h = [1800,1400,700];
    consphis = sum((Pop<lb)+(Pop>ub),2)...
    +(fly_time1+drop_time1 > 65)...
    +(fly_time2+drop_time2 > 65)...
    +(fly_time3+drop_time3 > 65)...
    +(get_smoke_z(h(1),drop_time1,popsize)<10)...
    +(get_smoke_z(h(2),drop_time2,popsize)<10)...
    +(get_smoke_z(h(3),drop_time3,popsize)<10);
end
g = 9.8;
u = minAns(actualIter,:);


v_fy1_vec = [cos(u(1)),sin(u(1)),0];
v_fy1_num = u(4);
fy1_pos0 = [17800,0,1800];

v_fy2_vec = [cos(u(2)),sin(u(2)),0];
v_fy2_num = u(5);
fy2_pos0 = [12000,1400,1400];

v_fy3_vec = [cos(u(3)),sin(u(3)),0];
v_fy3_num = u(6);
fy3_pos0 = [6000,-3000,700];


```

中国大学生在线 dxs.moe.gov.cn

```
drop_time1 = u(7);
drop_time2 = u(8);
drop_time3 = u(9);
explode_time1 = u(10)+drop_time1;
explode_time2 = u(11)+drop_time2;
explode_time3 = u(12)+drop_time3;
m1_pos0 = [20000,0,2000];
v_m1_num = 300;
v_m1_vec = -m1_pos0/norm(m1_pos0);

time_steps = 500000;
state = zeros(sample_num,3);
fullyOW1 = false;
fullyOW2 = false;
fullyOW3 = false;
partOW = false;
start_t1 = 0;
end_t1 = 0;
start_t2 = 0;
end_t2 = 0;
start_t3 = 0;
end_t3 = 0;

start_t4 = 0;
end_t4 = 0;
for t = linspace(0,10,time_steps)
    m1_pos = m1_pos0+t*v_m1_vec*v_m1_num;
    for j = 1:sample_num
        if t > explode_time1
            smoke1_pos = fy1_pos0+explode_time1*v_fy1_vec*v_fy1_num
                +[0,0,-0.5*g*(explode_time1-drop_time1)^2]+[0,0,-3*(t-
                explode_time1)];
            if(isOverwhelm(m1_pos,smoke1_pos,sample_dots(j,:)))
                state(j,1)=1;
            else
                state(j,1)=0;
            end
        end

        if t > explode_time2
            smoke2_pos = fy2_pos0+explode_time2*v_fy2_vec*v_fy2_num
```

```
            +[0,0,-0.5*g*(explode_time2-drop_time2)^2]+[0,0,-3*(t-
            explode_time2)];
        if(isOverwhelm(m1_pos,smoke2_pos,sample_dots(j,:)))
            state(j,2)=1;
        else
            state(j,2)=0;
        end
    end

    if t > explode_time3
        smoke3_pos = fy3_pos0+explode_time3*v_fy3_vec*v_fy3_num
            +[0,0,-0.5*g*(explode_time3-drop_time3)^2]+[0,0,-3*(t-
            explode_time3)];
        %%
        if(isOverwhelm(m1_pos,smoke3_pos,sample_dots(j,:)))
            state(j,3)=1;
        else
            state(j,3)=0;
        end
    end
end

if all(state(:,1)==1)
    if ~fullyOW1
        fprintf("%fs时第一个烟幕干扰弹开始生效\n",t);
        fullyOW1 = true;
    end
else
    if fullyOW1
        fprintf("%fs时第一个烟幕干扰弹开始失效\n",t);
        fullyOW1 = false;
    end
end

if all(state(:,2)==1)
    if ~fullyOW2
        fprintf("%fs时第二个烟幕干扰弹开始生效\n",t);
        fullyOW2 = true;
    end
else
    if fullyOW2
```

中国大学生在线 dxs.moe.gov.cn

```
            fprintf("%fs时第二个烟幕干扰弹开始失效\n",t);
            fullyOW2 = false;
        end
    end

    if all(state(:,3)==1)
        if ~fullyOW3
            fprintf("%fs时第三个烟幕干扰弹开始生效\n",t);
            fullyOW3 = true;
        end
    else
        if fullyOW3
            fprintf("%fs时第三个烟幕干扰弹开始失效\n",t);
            fullyOW3 = false;
        end
    end

    if all(sum(state,2)>=1) && ~all(state(:,1)==1) && ~all(state(:,2)
        ==1) && ~all(state(:,3)==1)
        if ~partOW
            fprintf("%fs时只有多个烟幕干扰弹才能生效\n",t);
            partOW = true;
        end
    else
        if partOW
            fprintf("%fs时即使有多个烟幕干扰弹都无法生效\n",t);
            partOW = false;
        end
    end

end


fprintf("总最大有效遮蔽时间是%f\n",-minFit(actualIter))
fprintf("第一架无人机(FY1)速度方向方向角为: %f° ,速度大小为: %fm/s\n",u
    (1)*180/pi,u(4));
fy1_fly_pos = fy1_pos0+u(4)*u(7).*v_fy1_vec;
fy1_explode_pos = fy1_fly_pos+u(4)*u(10)*v_fy1_vec+[0,0,-0.5*u(10)^2*g
    ];
fprintf("第一颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第一颗烟幕干扰弹的起爆坐
    标是(%f ,%f, %f)\n",fy1_fly_pos,fy1_explode_pos);
```

```
fprintf("第二架无人机(FY2)速度方向方向角为: %f° ,速度大小为: %fm/s\n",u(2)*180/pi,u(5));
fy2_fly_pos = fy2_pos0+u(5)*u(8).*v_fy2_vec;
fy2_explode_pos = fy2_fly_pos+u(5)*u(11)*v_fy2_vec+[0,0,-0.5*u(11)^2*g];
fprintf("第二颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第二颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy2_fly_pos,fy2_explode_pos);

fprintf("第三架无人机(FY3)速度方向方向角为: %f° ,速度大小为: %fm/s\n",u(3)*180/pi, u(6)); % 使用 u(6)
fy3_fly_pos = fy3_pos0+u(6)*u(9).*v_fy3_vec;
fy3_explode_pos = fy3_fly_pos+u(6)*u(12).*v_fy3_vec+[0,0,-0.5*u(12)^2*g]; % 使用 v_fy3_vec
fprintf("第三颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第三颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy3_fly_pos,fy3_explode_pos);
```

```
%文件: fourth_grid.m
%作用: 求解第四问


% 先用解析法求一个初始解, 再在这个初始解周围的等距网格搜索

function [theta,v,t1,t2] = get_vt1t1(fy_pos0)
    for t = 0:0.005:60
        v_m1 = [20000,0,2000]/norm([20000,0,2000])*300;
        m1_pos = [20000-v_m1(1)*t,0,2000-v_m1(3)*t];
        fy2m1 = m1_pos-fy_pos0;
        fy2m1(3) = 0;
        v = norm(fy2m1)/t;
        if(v < 140 && v>70 &&m1_pos(3)<=fy_pos0(3))
            v_vec = (fy2m1)/norm(fy2m1);
            theta = atan2(v_vec(2),v_vec(1));
            s_square = (fy_pos0(3)-(fy_pos0(1)+v_vec(1)*t)/10)*2/9.8/t^2;
            if s_square>=0 && s_square<1
                s = sqrt((fy_pos0(3)-(fy_pos0(1)+v_vec(1)*t)/10)*2/9.8/t^2);
                t1 = (1-s)*t;
                t2 = s*t;
                break;
            else
```

中国大学生在线 dxs.moe.gov.cn

```
                continue;
            end
        end
    end
end
clear;
clc;
sample_num = 2*50;
sample_dots = zeros(sample_num,3);
idx = 1;
for h = [0,10]
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end


[theta,v,t1,t2] = get_vt1t1([17800, 0, 1800]);
[longest_t1,tl1,tr1,best_x1] = grid_search([17800, 0, 1800],theta,v,t1,
    t2,sample_dots);

[theta,v,t1,t2] = get_vt1t1([12000, 1400, 1400]);
[longest_t2,tl2,tr2,best_x2] = grid_search([12000, 1400, 1400],theta,v,
    t1,t2,sample_dots);

[theta,v,t1,t2] = get_vt1t1([6000, -3000, 700]);
[longest_t3,tl3,tr3,best_x3] = grid_search([6000, -3000, 700],theta,v,
    t1,t2,sample_dots);



A = [tl1,tr1];
B = [tl2,tr2];
C = [tl3,tr3];

%单个长度
LA = tr1 - tl1;
LB = tr2 - tl2;
LC = tr3 - tl3;

```

中国大学生在线 dxs.moe.gov.cn

```
%两两交集
LAB = max(0, min(tr1,tr2) - max(tl1,tl2));
LAC = max(0, min(tr1,tr3) - max(tl1,tl3));
LBC = max(0, min(tr2,tr3) - max(tl2,tl3));

%三者交集
LABC = max(0, min([tr1,tr2,tr3]) - max([tl1,tl2,tl3]));

%容斥原理
time_sum = LA + LB + LC - LAB - LAC - LBC + LABC;

theta1 = atan2(best_x1(2),best_x1(1));
theta2 = atan2(best_x2(2),best_x2(1));
theta3 = atan2(best_x3(2),best_x3(1));
u = [theta1,theta2,theta3,best_x1(4),best_x2(4),best_x3(4),best_x1(5),
    best_x2(5),best_x3(5),best_x1(6),best_x2(6),best_x3(6)];
g = 9.8;

v_fy1_vec = [cos(u(1)),sin(u(1)),0];
v_fy1_num = u(4);
fy1_pos0 = [17800,0,1800];

v_fy2_vec = [cos(u(2)),sin(u(2)),0];
v_fy2_num = u(5);
fy2_pos0 = [12000,1400,1400];

v_fy3_vec = [cos(u(3)),sin(u(3)),0];
v_fy3_num = u(6);
fy3_pos0 = [6000,-3000,700];


drop_time1 = u(7);
drop_time2 = u(8);
drop_time3 = u(9);
explode_time1 = u(10)+drop_time1;
explode_time2 = u(11)+drop_time2;
explode_time3 = u(12)+drop_time3;
m1_pos0 = [20000,0,2000];
v_m1_num = 300;
v_m1_vec = -m1_pos0/norm(m1_pos0);

```

中国大学生在线 dxs.moe.gov.cn

```
time_steps = 500000;
state = zeros(sample_num,3);
fullyOW1 = false;
fullyOW2 = false;
fullyOW3 = false;
partOW = false;
start_t1 = 0;
end_t1 = 0;
start_t2 = 0;
end_t2 = 0;
start_t3 = 0;
end_t3 = 0;

start_t4 = 0;
end_t4 = 0;
for t = linspace(0,65,time_steps)
    m1_pos = m1_pos0+t*v_m1_vec*v_m1_num;
    for j = 1:sample_num
        if t > explode_time1
            smoke1_pos = fy1_pos0+explode_time1*v_fy1_vec*v_fy1_num
                +[0,0,-0.5*g*(explode_time1-drop_time1)^2]+[0,0,-3*(t-
                explode_time1)];
            if(isOverwhelm(m1_pos,smoke1_pos,sample_dots(j,:)))
                state(j,1)=1;
            else
                state(j,1)=0;
            end
        end

        if t > explode_time2
            smoke2_pos = fy2_pos0+explode_time2*v_fy2_vec*v_fy2_num
                +[0,0,-0.5*g*(explode_time2-drop_time2)^2]+[0,0,-3*(t-
                explode_time2)];
            if(isOverwhelm(m1_pos,smoke2_pos,sample_dots(j,:)))
                state(j,2)=1;
            else
                state(j,2)=0;
            end
        end

        if t > explode_time3
```

中国大学生在线 dxs.moe.gov.cn

```
            smoke3_pos = fy3_pos0+explode_time3*v_fy3_vec*v_fy3_num
                +[0,0,-0.5*g*(explode_time3-drop_time3)^2]+[0,0,-3*(t-
                explode_time3)];
            %%
            if(isOverwhelm(m1_pos,smoke3_pos,sample_dots(j,:)))
                state(j,3)=1;
            else
                state(j,3)=0;
            end
        end
    end

    if all(state(:,1)==1)
        if ~fullyOW1
            fprintf("%fs时第一个烟幕干扰弹开始生效\n",t);
            fullyOW1 = true;
        end
    else
        if fullyOW1
            fprintf("%fs时第一个烟幕干扰弹开始失效\n",t);
            fullyOW1 = false;
        end
    end

    if all(state(:,2)==1)
        if ~fullyOW2
            fprintf("%fs时第二个烟幕干扰弹开始生效\n",t);
            fullyOW2 = true;
        end
    else
        if fullyOW2
            fprintf("%fs时第二个烟幕干扰弹开始失效\n",t);
            fullyOW2 = false;
        end
    end
    if all(state(:,3)==1)
        if ~fullyOW3
            fprintf("%fs时第三个烟幕干扰弹开始生效\n",t);
            fullyOW3 = true;
        end
    else
```

中国大学生在线 dxs.moe.gov.cn

```
        if fullyOW3
            fprintf("%fs时第三个烟幕干扰弹开始失效\n",t);
            fullyOW3 = false;
        end
    end

    if all(sum(state,2)>=1) && ~all(state(:,1)==1) && ~all(state(:,2)==1) && ~all(state(:,3)==1)
        if ~partOW
            fprintf("%fs时只有多个烟幕干扰弹才能生效\n",t);
            partOW = true;
        end
    else
        if partOW
            fprintf("%fs时即使有多个烟幕干扰弹都无法生效\n",t);
            partOW = false;
        end
    end

end


fprintf("总最大有效遮蔽时间是%f\n",-myObj(u,sample_dots))
fprintf("第一架无人机(FY1)速度方向方向角为: %f° ,速度大小为: %fm/s\n",u(1)*180/pi,u(4));
fy1_fly_pos = fy1_pos0+u(4)*u(7).*v_fy1_vec;
fy1_explode_pos = fy1_fly_pos+u(4)*u(10)*v_fy1_vec+[0,0,-0.5*u(10)^2*g];
fprintf("第一颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第一颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy1_fly_pos,fy1_explode_pos);

fprintf("第二架无人机(FY2)速度方向方向角为: %f° ,速度大小为: %fm/s\n",u(2)*180/pi,u(5));
fy2_fly_pos = fy2_pos0+u(5)*u(8).*v_fy2_vec;
fy2_explode_pos = fy2_fly_pos+u(5)*u(11)*v_fy2_vec+[0,0,-0.5*u(11)^2*g];
fprintf("第二颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第二颗烟幕干扰弹的起爆坐标是(%f ,%f, %f)\n",fy2_fly_pos,fy2_explode_pos);

fprintf("第三架无人机(FY3)速度方向方向角为: %f° ,速度大小为: %fm/s\n",u(3)*180/pi, u(6)); % 使用 u(6)
```

中国大学生在线 dxs.moe.gov.cn

```
fy3_fly_pos = fy3_pos0+u(6)*u(9).*v_fy3_vec;
fy3_explode_pos = fy3_fly_pos+u(6)*u(12).*v_fy3_vec+[0,0,-0.5*u(12)^2*g
    ]; % 使用 v_fy3_vec
fprintf("第三颗烟幕干扰弹的投放坐标是(%f ,%f, %f),第三颗烟幕干扰弹的起爆坐
    标是(%f ,%f, %f)\n",fy3_fly_pos,fy3_explode_pos);




function fits = myObj(Pop, sample_dots)
    fy1_pos0 = [17800, 0, 1800];
    fy2_pos0 = [12000, 1400, 1400];
    fy3_pos0 = [6000, -3000, 700];
    [sample_num, ~] = size(sample_dots);
    popsize = size(Pop);
    len = popsize(1);
    fits = zeros(len, 1);
    v_fy1_vec = [cos(Pop(:, 1)), sin(Pop(:, 1)), zeros(len, 1)];
    v_fy2_vec = [cos(Pop(:, 2)), sin(Pop(:, 2)), zeros(len, 1)];
    v_fy3_vec = [cos(Pop(:, 3)), sin(Pop(:, 3)), zeros(len, 1)];
    v_fy1_num = Pop(:, 4);
    v_fy2_num = Pop(:, 5);
    v_fy3_num = Pop(:, 6);
    fly_time1 = Pop(:, 7);
    drop_time1 = Pop(:, 10);
    fly_time2 = Pop(:, 8);
    drop_time2 = Pop(:, 11);
    fly_time3 = Pop(:, 9);
    drop_time3 = Pop(:, 12);

    for i = 1:len
        % 为每个采样点计算其总的有效遮蔽区间（并集）
        unions_per_sample = cell(sample_num, 1);
        for j = 1:sample_num
            [~,t1, t2] = get_last_time(fy1_pos0, v_fy1_vec(i, :),
                v_fy1_num(i), fly_time1(i), drop_time1(i), sample_dots(j,
                 :));
            [~,t3, t4] = get_last_time(fy2_pos0, v_fy2_vec(i, :),
                v_fy2_num(i), fly_time2(i), drop_time2(i), sample_dots(j,
                 :));
            [~,t5, t6] = get_last_time(fy3_pos0, v_fy3_vec(i, :),
```

```
            v_fy3_num(i), fly_time3(i), drop_time3(i), sample_dots(j,
             :));
        if(t1>t2)
            t1 = 0;
            t2 = 0;
        end
        unions_per_sample{j} = IntervalUnion([t1, t2; t3, t4; t5, t6
            ]);
    end

    % 计算所有采样点有效区间的交集
    final_intersection = unions_per_sample{1};
    for j = 2:sample_num
        if isempty(final_intersection)
            break;
        end
        final_intersection = IntervalInter(final_intersection,
            unions_per_sample{j});
    end

    if isempty(final_intersection)
        fits(i) = 0;
    else
        fits(i) = sum(final_intersection(:, 1) - final_intersection
            (:, 2), "all");
    end
end
end


function [t1,t2] = get_last_time_1dot(fy1_pos0,v_fy1_vec,v_fy1_num,
    fly_time,drop_time,sample_dot)
    if v_fy1_num>140 || v_fy1_num<70
        t1 = 0;
        t2 = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t1 = 0;
        t2 = 0;
```

中国大学生在线 dxs.moe.gov.cn

```
        return;
    end
    if drop_time>65 || drop_time<0
        t1 = 0;
        t2 = 0;
        return;
    end
    if drop_time+fly_time>65
        t1 = 0;
        t2 = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;
    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %%先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    start_time1 = start_time;
    end_time2 = end_time;
    while ~time_is_found
        if length(candidate_time)>63
            t1 = 0;
            t2 = 0;
            return;
        end
        idx = 2;

        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
```

中国大学生在线 dxs.moe.gov.cn

```
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dot)
                time_is_found = true;
                overwhelm_time = t;
                start_time1 = candidate_time(idx-1);
                end_time2 = candidate_time(idx+1);
                break;
            end
            idx = idx+2;
        end
        if ~time_is_found
            new_candidate_time = zeros(1,2*length(candidate_time)-1);
            for i = 2:2:2*length(candidate_time)-2
                new_candidate_time(i) = (candidate_time(i/2)+
                    candidate_time(i/2+1))/2;
            end
            for i = 1:2:2*length(candidate_time)-1
                new_candidate_time(i) = candidate_time((i+1)/2);
            end
            candidate_time = new_candidate_time;
        end
    end
    end_time1 = overwhelm_time;
    while end_time1-start_time1>1e-8
        mid = (end_time1+start_time1)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dot)
            end_time1 = mid;
        else
            start_time1 = mid;
        end
    end
    t1 = (end_time1+start_time1)/2;

    start_time2 = overwhelm_time;
    while end_time2-start_time2>1e-8
        mid = (end_time2+start_time2)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
```

中国大学生在线 dxs.moe.gov.cn

```
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dot)
            start_time2 = mid;
        else
            end_time2 = mid;
        end
    end
    t2 = (end_time2+start_time2)/2;
end

function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dots)
    [sample_num,~] = size(sample_dots);

    d = sample_dots - repmat(missile_pos,sample_num,1);
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(repmat(oc,sample_num,1).*d,2);
    c = dot(oc,oc)-100;
    discriminant = b.^2 - 4 * a * c;

    if any(discriminant<0)
        overwhelmed = false;
        return;
    end
    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);
    overwhelmed = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed);
    if(norm(smoke_pos-missile_pos)<10)
        overwhelmed = false;
    end
end



function [t,t1,t2] = get_last_time(fy1_pos0,v_fy1_vec,v_fy1_num,
    fly_time,drop_time,sample_dots)
    if v_fy1_num>140 || v_fy1_num<70
```

中国大学生在线 dxs.moe.gov.cn

```
        t1 = 0;
        t2 = 0;
        t = 0;
        return;
    end
    if fly_time>65 || fly_time<0
        t1 = 0;
        t2 = 0;
        t = 0;
        return;
    end
    if drop_time>65 || drop_time<0
        t1 = 0;
        t2 = 0;
        t = 0;
        return;
    end
    if drop_time+fly_time>65
        t1 = 0;
        t2 = 0;
        t = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    m1_pos0 = [20000,0,2000];
    v_m1_vec = -m1_pos0/norm(m1_pos0);
    v_m1_num = 300;

    v_fy1_vec = v_fy1_vec/norm(v_fy1_vec);

    smoke_pos0 = fy1_pos0+(fly_time+drop_time).*v_fy1_num.*v_fy1_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
```

中国大学生在线 dxs.moe.gov.cn

```
    start_time1 = fly_time+drop_time;
    end_time2 = end_time;

    while ~time_is_found
        if length(candidate_time)>32
            t1 = 0;
            t2 = 0;
            t = t2-t1;
            return;
        end
        idx = 2;
        for t = candidate_time(2:2:end-1)
            m1_pos = m1_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m1_pos,smoke_pos,sample_dots)
                time_is_found = true;
                overwhelm_time = t;
                start_time1 = candidate_time(idx-1);
                end_time2 = candidate_time(idx+1);
                break;
            end
            idx = idx+2;
        end
        if ~time_is_found
            new_candidate_time = zeros(1,2*length(candidate_time)-1);
            for i = 2:2:2*length(candidate_time)-2
                new_candidate_time(i) = (candidate_time(i/2)+
                    candidate_time(i/2+1))/2;
            end
            for i = 1:2:2*length(candidate_time)-1
                new_candidate_time(i) = candidate_time((i+1)/2);
            end
            candidate_time = new_candidate_time;
        end
    end

    end_time1 = overwhelm_time;
    while end_time1-start_time1>1e-7
        mid = (end_time1+start_time1)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
```

中国大学生在线 dxs.moe.gov.cn

```
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dots)
            end_time1 = mid;

        else
            start_time1 = mid;
        end
    end
    t1 = (end_time1+start_time1)/2;

    start_time2 = overwhelm_time;

    while end_time2-start_time2>1e-7
        mid = (end_time2+start_time2)/2;
        m1_pos = m1_pos0+mid*v_m1_num*v_m1_vec;
        smoke_pos = smoke_pos0;
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m1_pos,smoke_pos,sample_dots)
            start_time2 = mid;
        else
            end_time2 = mid;
        end
    end
    t2 = (end_time2+start_time2)/2;
    t = t2-t1;
end



function [longest_t,best_tl,best_tr,best_x] = grid_search(fy_pos0,theta
    ,v,t1,t2,sample_dots)
    best_x = zeros(1:4);
    best_tl = 0;
    best_tr = 0;
    longest_t = 0;
    for theta_change = -0.2:0.008:0.2
        for v_change = (linspace(max(70,v-20),min(140,v+20),50)-v)
            for t1_change = linspace(max(0,t1-5),t1+5,80)-t1
                for t2_change = linspace(max(0,t2-5),t2+5,80)-t2
```

```
                    new_theta = theta+theta_change;
                    new_v = v+v_change;
                    new_t1 = t1*t1_change;
                    new_t2 = t2*t2_change;
                    v_vec = [cos(new_theta),sin(new_theta),0];
                    [t,tl,tr]= get_last_time(fy_pos0,v_vec,new_v,new_t1,
                        new_t2,sample_dots);

                    if t>longest_t
                        best_tl = tl;
                        best_tr = tr;
                        longest_t = t
                        best_x = [v_vec,new_v,new_t1,new_t2];
                    end
                end
            end
        end
    end
end
```

```
%文件: fifth.m
%作用: 求解第五问
clear;
clc;
function [theta,v,t1,t2] = get_vt1t1(fy_pos0,m_pos0)
    g = 9.8;
    for t = 0:0.005:60
        v_m = -m_pos0/norm(m_pos0)*300;
        m_pos = m_pos0+v_m*t;
        fy2m = m_pos-fy_pos0;
        fy2m(3) = 0;
        v = norm(fy2m)/t;
        if(v < 140 && v>70 &&m_pos(3)<=fy_pos0(3))
            v_vec = (fy2m)/norm(fy2m);
            theta = atan2(v_vec(2),v_vec(1));
            s_square = (fy_pos0(3)-m_pos(3))/(0.5*g*t^2);
            if s_square>=0 && s_square<1
                s = sqrt(s_square);
                t1 = (1-s)*t;
                t2 = s*t;
                break;
```

中国大学生在线 dxs.moe.gov.cn

```
            else
                continue;
            end
        end
    end
end

sample_num = 2*300;
sample_dots = zeros(sample_num,3);
idx = 1;
for h = [0,10]
    for theta = linspace(0,2*pi,sample_num/2)
        sample_dots(idx,:) = [7*cos(theta),7*sin(theta)+200,h];
        idx = idx+1;
    end
end

m1_pos0 = [20000, 0, 2000];
m2_pos0 = [19000, 600, 2100];
m3_pos0 = [18000, -600, 1900];

fy1_pos0 = [17800, 0, 1800];
fy2_pos0 = [12000, 1400, 1400];
fy3_pos0 = [6000, -3000, 700];
fy4_pos0 = [11000, 2000, 1800];
fy5_pos0 = [13000, 2000, 1300];

m_pos0 = [m1_pos0;m2_pos0;m3_pos0];
fy_pos0 = [fy1_pos0;fy2_pos0;fy3_pos0;fy4_pos0;fy5_pos0];

v_m_pos = -m_pos0./sqrt(sum(m_pos0.^2,2));

dis = zeros(3,5);
for j = 1:5
    for i = 1:3
        t = (m_pos0(i,3)-fy_pos0(j,3))/(-v_m_pos(i,3))/300;
        mi_pos = m_pos0(i,:)+v_m_pos(i,:)*t*300;
        dis_mi2fy = norm([fy_pos0(j,1)-mi_pos(1),fy_pos0(j,2)-mi_pos(2)
            ]);
        dis(i,j) = dis_mi2fy;
    end
```

中国大学生在线 dxs.moe.gov.cn

```
end


all_time = 0;
for j = 1:5
    [min_dis,idx] = min(dis(:,j));
    [theta,v,fly_t,drop_time] = get_vt1t1(fy_pos0(j,:),m_pos0(idx,:));
    [longest_t1,longest_t2,longest_t3,best_ans] = grid_search(m_pos0(idx
        ,:),fy_pos0(j,:),theta,v,fly_t,drop_time,sample_dots);
    theta = best_ans(1);
    v = best_ans(2);
    fly_t = best_ans(3);
    drop_time = best_ans(4);
    fly_pos1 = fy_pos0(j,:)+[cos(theta),sin(theta),0]*fly_t*v;

    fprintf("无人机FY%d的速度方向是%f°,速度大小是%fm/s\n",j,theta/pi*180,
        v);
    explode_pos1 = fly_pos1+[cos(theta),sin(theta),0]*drop_time*v
        -[0,0,0.5*9.8*drop_time^2];
    fprintf("第一颗烟幕弹抛出坐标: (%f,%f,%f)\n,爆炸坐标: (%f,%f,%f)\n对
        导弹M%d造成%fs的有效遮挡\n" ,fly_pos1,explode_pos1,idx,longest_t1
        );

    fly_pos2 = fy_pos0(j,:)+[cos(theta),sin(theta),0]*(fly_t+1)*v;
    explode_pos2 = fly_pos2+[cos(theta),sin(theta),0]*drop_time*v
        -[0,0,0.5*9.8*drop_time^2];
    fprintf("第二颗烟幕弹抛出坐标: (%f,%f,%f)\n,爆炸坐标: (%f,%f,%f)\n对
        导弹M%d造成%fs的有效遮挡\n" ,fly_pos2,explode_pos2,idx,longest_t2
        );

    fly_pos3 = fy_pos0(j,:)+[cos(theta),sin(theta),0]*(fly_t+2)*v;
    explode_pos3 = fly_pos3+[cos(theta),sin(theta),0]*drop_time*v
        -[0,0,0.5*9.8*drop_time^2];
    fprintf("第三颗烟幕弹抛出坐标: (%f,%f,%f)\n,爆炸坐标: (%f,%f,%f)\n对
        导弹M%d造成%fs的有效遮挡\n" ,fly_pos3,explode_pos3,idx,longest_t3
        );
    all_time = all_time+longest_t3+longest_t2+longest_t1
    fprintf("\n\n\n")
end


```

中国大学生在线 dxs.moe.gov.cn

```
function [longest_t1,longest_t2,longest_t3,best_ans] = grid_search(m_pos0,fy_pos0,theta,v,fly_t,drop_time,sample_dots)
    longest_t1 = get_last_time(m_pos0,fy_pos0,[cos(theta),sin(theta),0],v,fly_t,drop_time,sample_dots);
    longest_t2 = get_last_time(m_pos0,fy_pos0,[cos(theta),sin(theta),0],v,fly_t+1,drop_time,sample_dots);
    longest_t3 =get_last_time(m_pos0,fy_pos0,[cos(theta),sin(theta),0],v,fly_t+2,drop_time,sample_dots);

    best_ans = [theta,v,fly_t,0];
    for d_theta = linspace(-0.3,0.3,20)
        for d_v = linspace(-20,20,20)
            for d_fly_t = linspace(-6,6,20)
                for d_drop_time = linspace(-6,6,20)
                    new_theta = d_theta+theta;
                    new_fly_t = d_fly_t+fly_t;
                    new_v = d_v+v;
                    new_drop_time = d_drop_time+drop_time;
                    new_theta = max(min(new_theta,pi),-pi);
                    new_fly_t = max(min(new_fly_t,65),0);
                    new_drop_time = max(min(new_drop_time,65),0);
                    new_v = max(min(new_v,140),70);
                    new_longest_t1 = get_last_time(m_pos0,fy_pos0,[cos(new_theta),sin(new_theta),0],new_v,new_fly_t,new_drop_time,sample_dots);
                    new_longest_t2 = get_last_time(m_pos0,fy_pos0,[cos(new_theta),sin(new_theta),0],new_v,new_fly_t+1,new_drop_time,sample_dots);
                    new_longest_t3 = get_last_time(m_pos0,fy_pos0,[cos(new_theta),sin(new_theta),0],new_v,new_fly_t+2,new_drop_time,sample_dots);
                    if new_longest_t1+new_longest_t2+new_longest_t3 > longest_t1+longest_t2+longest_t3
                        longest_t1 = new_longest_t1;
                        longest_t2 = new_longest_t2;
                        longest_t3 = new_longest_t3;
                        best_ans = [new_theta,new_v,new_fly_t,new_drop_time];
                    end
                end
            end
```

中国大学生在线 dxs.moe.gov.cn

```
        end
    end
end



function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dots)
    [sample_num,~] = size(sample_dots);
    sample_num = floor(sample_num/2);

    d = sample_dots - repmat(missile_pos,sample_num*2,1);
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(repmat(oc,sample_num*2,1).*d,2);
    c = dot(oc,oc)-100;
    discriminant = b.^2 - 4 * a * c;

    if any(discriminant<0)
        overwhelmed = false;
        return;
    end
    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);
    overwhelmed = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed);
    if(norm(smoke_pos-missile_pos)<10)
        overwhelmed = false;
    end
end



function t = get_last_time(m_pos0,fy_pos0,v_fy_vec,v_fy_num,fly_time,
    drop_time ,sample_dots)
    if v_fy_num>140 || v_fy_num<70
        t = 0;
        return;
    end
```

中国大学生在线 dxs.moe.gov.cn

```
    if fly_time>65 || fly_time<0
        t = 0;
        return;
    end
    if drop_time+fly_time>65
        t = 0;
        return;
    end
    smoke_valid_time = 20;
    g = 9.8;
    v_m1_vec = -m_pos0/norm(m_pos0);
    v_m1_num = 300;
    v_fy_vec = v_fy_vec/norm(v_fy_vec);

    smoke_pos0 = fy_pos0+(fly_time+drop_time).*v_fy_num.*v_fy_vec;
    smoke_pos0(3) = smoke_pos0(3) - 0.5*g*drop_time^2;

    start_time = fly_time+drop_time;
    end_time = smoke_valid_time+start_time;
    %先找一个时间点，这时目标被遮蔽。
    %这个点的候选时间点
    candidate_time = [start_time,(start_time+end_time)/2,end_time];
    time_is_found = false;
    overwhelm_time = 0;
    start_time1 = fly_time+drop_time;
    end_time2 = end_time;

    while ~time_is_found
        if length(candidate_time)>32
            t1 = 0;
            t2 = 0;
            t = t2-t1;
            return;
        end
        idx = 2;
        for t = candidate_time(2:2:end-1)
            m_pos = m_pos0+t*v_m1_num*v_m1_vec;
            smoke_pos = smoke_pos0;
            smoke_pos(3) = smoke_pos(3)-3*(t-fly_time-drop_time);
            if isOverwhelm(m_pos,smoke_pos,sample_dots)
                time_is_found = true;
```

```
            overwhelm_time = t;
            start_time1 = candidate_time(idx-1);
            end_time2 = candidate_time(idx+1);
            break;
        end
        idx = idx+2;
    end
    if ~time_is_found
        new_candidate_time = zeros(1,2*length(candidate_time)-1);
        for i = 2:2:2*length(candidate_time)-2
            new_candidate_time(i) = (candidate_time(i/2)+
                candidate_time(i/2+1))/2;
        end
        for i = 1:2:2*length(candidate_time)-1
            new_candidate_time(i) = candidate_time((i+1)/2);
        end
        candidate_time = new_candidate_time;
    end
end

end_time1 = overwhelm_time;
while end_time1-start_time1>1e-7
    mid = (end_time1+start_time1)/2;
    m_pos = m_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
    smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
    if isOverwhelm(m_pos,smoke_pos,sample_dots)
        end_time1 = mid;

    else
        start_time1 = mid;
    end
end
t1 = (end_time1+start_time1)/2;

start_time2 = overwhelm_time;

while end_time2-start_time2>1e-7
    mid = (end_time2+start_time2)/2;
    m_pos = m_pos0+mid*v_m1_num*v_m1_vec;
    smoke_pos = smoke_pos0;
```

中国大学生在线 dxs.moe.gov.cn

```
        smoke_pos(3) = smoke_pos(3)-3*(mid - fly_time-drop_time);
        if isOverwhelm(m_pos,smoke_pos,sample_dots)
            start_time2 = mid;
        else
            end_time2 = mid;
        end
    end
    t2 = (end_time2+start_time2)/2;
    t = t2-t1;
end
```

```
%文件: IntervalInter.m
%作用: 计算两个规范化区间并集的交集。
function inter = IntervalInter(union1, union2)

    if isempty(union1) || isempty(union2)
        inter = zeros(0, 2);
        return;
    end
    p1 = 1;
    p2 = 1;
    inter = zeros(0, 2);

    while p1 <= size(union1, 1) && p2 <= size(union2, 1)
        iv1 = union1(p1, :);
        iv2 = union2(p2, :);

        start = max(iv1(1), iv2(1));
        stop = min(iv1(2), iv2(2));

        if start < stop
            inter = [inter; [start, stop]];
        end

        % 移动终点较小的那个区间的指针
        if iv1(2) < iv2(2)
            p1 = p1 + 1;
        else
            p2 = p2 + 1;
        end
    end
```

中国大学生在线 dxs.moe.gov.cn

```
end
```

```
%文件: IntervalUnion.m
%作用: 将多个区间合并成一个规范的集合
function merged = IntervalUnion(intervals)
    % 将多个区间合并成一个规范的集合
    if isempty(intervals)
        merged = zeros(0, 2);
        return;
    end

    intervals = sortrows(intervals, 1);
    merged = intervals(1, :);

    %遍历剩余区间并合并
    for i = 2:size(intervals, 1)
        current_interval = intervals(i, :);
        pre_interval = merged(end, :);

        if current_interval(1) <= pre_interval(2)
            merged(end, 2) = max(pre_interval(2), current_interval(2));
        else
            merged = [merged; current_interval];
        end
    end
end
```

```
%文件: isOverwhelm.m
%作用: 判断导弹与目标之间的视线是否被烟幕遮挡

function overwhelmed = isOverwhelm(missile_pos,smoke_pos,sample_dots)
    [sample_num,-] = size(sample_dots);
    sample_num = floor(sample_num/2);

    d = sample_dots - repmat(missile_pos,sample_num*2,1);
    oc = missile_pos - smoke_pos;
    a = sum(d .* d,2);
    b = 2*sum(repmat(oc,sample_num*2,1).*d,2);
    c = dot(oc,oc)-100;
    discriminant = b.^2 - 4 * a * c;

```

中国大学生在线 dxs.moe.gov.cn

```
    if any(discriminant<0)
        overwhelmed = false;
        return;
    end
    sqrt_d = sqrt(discriminant);
    t1 = (-b - sqrt_d) ./ (2 * a);
    t2 = (-b + sqrt_d) ./ (2 * a);
    overwhelmed = (t1 >= 0 & t1 <= 1) | (t2 >= 0 & t2 <= 1);
    overwhelmed(t1 < 0 & t2 > 1) = true;
    overwhelmed = all(overwhelmed);
    % if(norm(smoke_pos-missile_pos)<10)
    %     overwhelmed = false;
    % end
end
```

中国大学生在线
dxs.moe.gov.cn