

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

学 校 同济大学

参赛队号 20102470005

1.孟潇

队员姓名 2.张文娟

3.王慧慧

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

题 目 无人机集群追逃博弈模型研究

摘 要：

本文研究了无人机集群的协同对抗问题，这属于约束条件下的追逃博弈问题。以一对多、多对多的追逃博弈问题为背景，在矩形边界下，研究红方无人机集群 P 拦截单个或多个蓝方突防无人机 E 问题。具体为，当 E 的速度更快时，研究 P 和 E 的支付函数、路径规划的最优机动决策和多个无人机集群的协同机制等问题。建立 P 和 E 的离散时间追逃博弈模型，引入阿波罗尼奥斯圆的可逃离范围定理，给出不同限定条件下 P 和 E 的最优机动决策以及 P 的协同机制，得到 P 和 E 的支付函数。

针对问题一，本文通过构架一对二追逃博弈模型，以突防速度尽可能快为最优机动策略，以一定能突防 CD 边界为条件，确定逃离区边界界栅的形态，计算逃离区的面积。首先讨论 P 集群的最大扫描范围，根据约束条件给出三角形方案和均匀分布的圆形方案。通过快速蒙特·卡罗算法估计两种方案的扫描范围，得出均匀分布的圆形方案扫描范围最大。因此，本题采用圆型方案，并且将其构成的花瓣形扫描范围合理简化成半径为 470 m 的圆形。建立离散时空的一对二追逃博弈模型，设定在离散时间下，红蓝双方轮流决策。给出三种 E 的机动策略模型：暴力突破、近似直线突破和欺骗性突防，并分别给出 P 的机动策略。这三种模型层层递进，逐步优化，复杂度和精确度也逐步增高。引入阿波罗尼奥斯定理，证明不可逃离角的存在性，将追逃博弈问题转化为几何分析问题，给出三种情况下的支付函数算法，分别计算可逃离区的面积为 776.35 km^2 ， 1358.89 km^2 ， 1528.55 km^2 。文章给出了不同限定条件下所有逃离区的具体坐标，详见附件。

针对问题二，本文通过问题一构架的一对二追逃博弈模型，分析得出当 M 大于临界值 M_{\min} 时，近似突围模型和欺骗性突围模型的逃离区中都包含 AB 中点的位置；当 M 小于 M_{\min} 时，近似突围法的逃离区中不包含 AB 中点的位置，而欺骗性突围法的逃离区有可能包含 AB 中点位置。找到临界状态后，采用几何分析法，利用不可逃离角的证明，得到问题求解的临界状态为当 E 与 P_1 、 P_2 的夹角恰好为 1.854 ，计算得出 M 的值为 133 m 。

针对问题三，本文构架了一对多追逃博弈模型。通过快速蒙特·卡罗算法估计得到结论，无人机集群最大扫描半径为 530 m ，最小为 415 m ，分别在无人机数为 7 和 3 取得。在模型中，因运载机速度大于无人机速度，所以第二波次的 P 最先拦截到 E ，则设定第一个波次发射的无人机数量为 3 ，第二个波次发射无人机的数量为 7 。由问题一、二求得逃离区的方式，得知 E 从 AB 中点出发一定被拦截，所以 E 直接采用近似直线运动模型突防。 P 的拦截策略为：设定距离 AB 中点 5 km 的地方为运载机 Y 初始位置，在以 Y 为中心，半径

为 2 km 圆的任意对称位置释放第一波 P, 求解运动方程, 得到运载机的机动决策为 80.2° , 在 t 约为 100.4 s, 距离决策方向线与 AB 中线的交点处还有 3.6 km 处发射第二波次无人机群。带宽上限的临界状态为在红方采用最差拦截策略时, 蓝方依旧不能通过。若只有一个波次时通过一对多追逃博弈模型可得带宽上限为 79.8 km; 若考虑两个波次时, 通过的带宽上限为 120 km。

针对问题四, 本文构架了多对多追逃博弈模型, 给出 P 的最优拦截策略称为“分配-包围-围困-收缩-抓捕”分配: 将多对多转化为一对多博弈问题; 包围: 最快的保证满足追捕者包围的三个条件; 围困: 协同合作改变集群阵型实现分散包围以满足围困条件; 收缩: 使用合适的非劣策略来尽可能收缩包围圈; 抓捕: 径直朝向包围圈的中心运动。给出 E 的最优突防策略为尽可能使不可逃脱角大于 1.854, 若非, 不断的更改其突破口, 尽可能的保证其不处于 P 的包围状态, 当 P、E 双方相对距离不变时, 在不限制时间的情况下, 保持和 P 相反的角度运动。

关键词: 追逃博弈模型; 协同对抗; 阿波罗尼奥斯圆; 界栅; 动态规划

一、问题重述

1.1 问题背景

无人机作战是空中作战的发展趋势之一。无人机集群作战作为智能无人机作战的重要形式，正在崭露头角，它将生物群体运动理论应用于无人机集群的系统控制，对敌方实施电子干扰或攻击。目前针对无人机集群作战的指挥决策研究中的单方策略优化较多，而多方策略优化较少。因此，研究无人机集群协同对抗问题是智能无人机作战的重要任务。

1.2 问题提出

本题目是研究红、蓝双方的无人机集群在带宽限定的约束平面区域内的协同对抗问题，利用博弈理论可以较好地进行作战指挥决策分析，并能将传统方法中最优研究从仅考虑“单边”发展为“双边”兼顾，使得决策更贴近现实。假设双方对抗区域约定为图 2-1 所示的矩形区域，攻击纵深即之间的距离为 L ，通道带宽即之间的距离记为 M 。

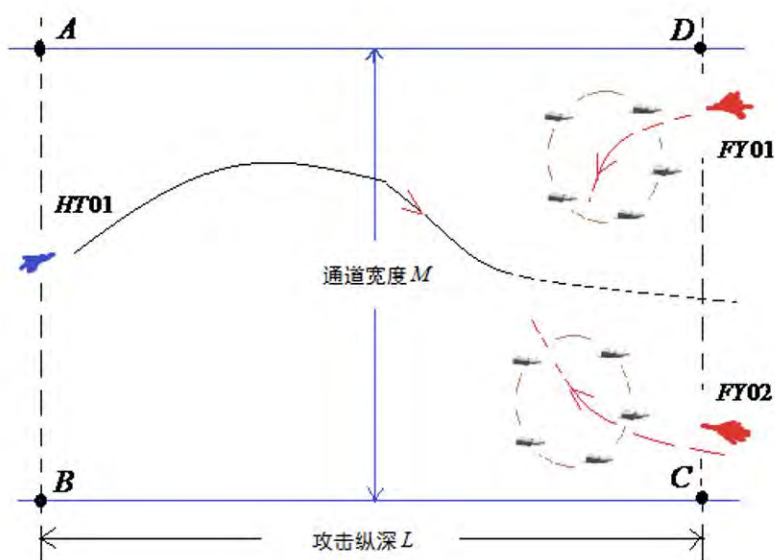


图 1-1 红方两个无人机集群拦截一架蓝方突防无人机示意图

双方航行需要满足的约束如下：

- (1) 蓝方无人机的飞行轨迹不能越过AD、BC两边；
- (2) 红蓝双方无人机的速度保持不变，蓝方无人机的速度为 $v_e = 250 \text{ m/s}$ ，红方无人机的速度为 $v_p = 200 \text{ m/s}$ ，红方运载机的速度为 $v_r = 300 \text{ m/s}$
- (3) 由于无人机在转弯时受到结构和控制系统的限制，飞行器在转弯时需要满足蓝方无人机的最小转弯半径 $R_e = 500 \text{ m}$ ；红方无人机的最小转弯半径 $R_p = 350 \text{ m}$ ；红方运载机最小转弯半径 $R_r = 1000 \text{ m}$ 。
- (4) 红方无人机集群需要满足任何两架无人机的间距大于 30 m ，每一架无人机与本集群中至少两架无人机的距离不超过 200 m 。

(5) 运载机到蓝方无人机的距离应大于5 km, 运载机与红方无人机中最近的一架的距离范围为100 m~10 km, 同时为安全需要, 与蓝方的突防无人机的距离需大于5 km。

(6) 红方每个无人机集群的无人机数量不少于3架。

(7) 蓝方任2架突防无人机的间距需大于30 m。

需要通过建立数学模型, 为红蓝双方博弈指定满足上述约束条件的航迹规划模型和算法。解决以下几个问题:

问题 1: 在满足条件 (1) ~ (5) 并且初始红方 2 个无人机集群的圆周中心分别位于给定的 G_1 和 G_2 点的情况下, 制定红蓝双方非合作目标的追逃博弈策略, 计算出蓝方无人机的逃离区。综合考虑以下优化目标:

(A) 蓝方能够突击成功; (B) 逃离区面积最大; (C) 突击时间最短。

讨论算法的复杂性和有效度, 并绘出蓝方可出发的位置和最佳突防策略。

问题 2: 在满足条件(1)~(5)并且初始红方 2 个无人机集群的圆周中心分别位于非给定 G_1 和 G_2 点的情况下, 蓝方突防无人机从边界 AB 的中心点出发, 分析通道带宽 M 的下限 M_{min} 的存在性, 使得当实际通道带宽 M 比 M_{min} 大时, 蓝方无人机一定能突破红方无人机集群的拦截, 并且讨论蓝方无人机时间最短的突防策略。综合考虑以下优化目标:

(A) 蓝方定够突击成功; (B) 突击时间按尽可能小; (C) M 尽可能小。

讨论算法的复杂性和有效度, 并绘出蓝方最佳突防策略。

问题 3: 在满足条件 (1) - (6) 的情况下, 蓝方突防无人机从边界 AB 的中心点出发, 红方每架运载机可分两个波次共发射 10 架无人机, 组成两个无人机集群遂行拦截任务, 计算红方两架运载机两个波次发射的无人机数量、每架运载机第二波次发射的时刻和位置以及第二波次发射的无人机集群的中心位置, 以实现最优的拦截效果; 进一步具体建模分析是通道带宽 M 的上限 M_{min} 的存在性, 当实际通道带宽 M 小于 M_{min} , 红方无人机一定能拦截成功。综合考虑以下优化目标:

(A) 红方定能拦截成功; (B) M 尽可能大。

讨论算法的复杂性和有效度, 并绘出红方的布阵策略。

问题 4: 在满足条件 (1) - (7) 的情况下, 给定通道带宽 M , 蓝方 3 架突防无人机从 AB 一侧突防, 红方 5 架运载机各携带 10 架无人机拦截, 每架运载机分两个波次发射无人机。蓝方希望尽可能多的无人机突防成功, 红方则希望成功拦截尽可能多的蓝方无人机。试讨论红方最优拦截策略和蓝方最优突防策略。

综合考虑以下优化目标:

(A) 蓝方尽可能多的无人机通过; (B) 红方尽可能多的拦截蓝方无人机。

讨论博弈模型的建立, 给出红蓝双方的博弈策略。

二、模型假设

2.1 模型的基本假设

2.1.1 题目对模型的假设

1、红蓝双方的无人机均在同一高度上机动飞行，即本题只讨论平面上的红蓝双方对抗问题。

2、双方无人机均装载有探测装置和数据链，能够准确探测对方无人机和己方无人机的实时位置。

3、运载机可瞬间布设好无人机集群，且满足相关约束条件。

2.1.2 为便于问题的研究而追加的假设

1、忽略无人机和运载机自身大小对路径长度的影响，将无人机视为质点；

2、为使红蓝双方尽可能敏捷,无人机只以最大速度、最小转弯半径运动和转弯。

3、假设蓝方和红方的起始运动方向垂直AB、CD向前。

4、每个无人机的运动符合牛顿运动学定律，且摩擦力、空气阻力以及无人机内部构件的误差等完全忽略。

2.2 模型符号说明

符号	意义说明
E	突防者（蓝方）
P	拦截者（红方）
R	集群的捕获半径
$posi(t)=(x_i(t),y_i(t)) \ i=\{e,p\}$	状态（位置）
ω	逃离区
θ	不可逃离角
$S(posi(t))$	允许状态（位置）集合
$u(posi(t))$	机动决策(旋转方向)
$U(posi(t))$	允许机动决策集合
$p(posi(t))$	机动策略（决策的集合）
$P(posi(t))$	允许机动策略集合
$T(posi(t),u(posi(t)))$	状态转移方程（位置变化）
$V(posi(t))$	指标函数
φ_i	速度方向和视线角的偏差

*	阿波罗尼奥圆
δ	视线角

三、问题一

3.1 问题分析

问题一是研究一对多的非合作目标的追逃博弈问题。追逃问题根据适用问题的情况将当前时刻对弈双方的所有可选对策空间区分为抓获区和逃离区。本题就突防方面而言，双方状态 pos_i （也就是位置坐标信息）有两种情况：当 E 处于抓获区内时，只要 P 使用恰当的策略，E 都会被 P 捕获；当 E 位置处于逃离区时，不论 P 使用何种策略，E 采用合适的策略一定能成功。抓获区和逃离区的分界线被称作界栅，确定界栅的形态和在界栅上对抗双方所选择执行的最优策略就是定性追逃博弈的核心问题^[1]。

本题如图 3-1 所示，要求解蓝方无人机处于矩形区域 ABCD 内哪些位置时，无论红方的两个无人机集群采用什么样的追击策略，蓝方无人机总能采用合适的策略以躲避红方的拦截，实现成功突防，并且讨论蓝方无人机相应的最优突防策略，也就是求解追踪博弈问题的逃离区 ω ，确定界栅的形态和位置并且讨论在界栅上蓝方无人机选择执行的最优策略。

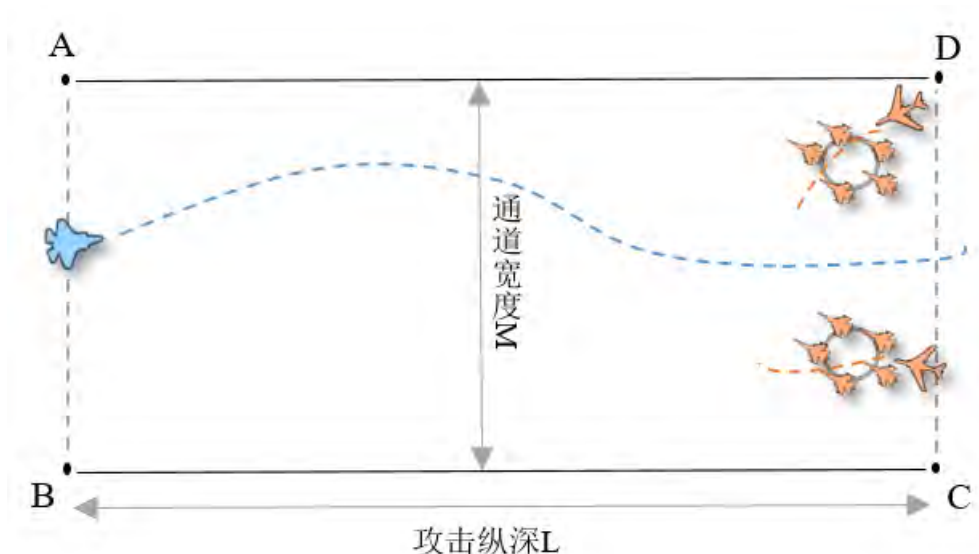


图 3-1 红方两个无人机集群拦截一架蓝方突防无人机示意图

根据题目和参考文献给出解题分为如下几个步骤：

第一步：为了方便本文的后续研究，对此问题进行合理的假设，建立二维空间坐标系，分析无人机基础飞行机动动作。

第二步：讨论红方无人机（集群）阵型的几种可能情况，并分别分析每个阵型对应的无人机扫描范围，得到使红色无人机（集群）扫描范围最大的阵型。

第三步：建立追逃博弈模型；确定红蓝双方最优机动决策，分为三种：强力突防模型；近似直线突围模型；欺骗性突围模型。

第四步：计算支付函数，计算 $\omega = \omega_1 \cup \omega_2 \cup \omega_3$ ，对结果进行优化。

第五步：讨论算法的复杂性和有效度，并绘出蓝方最佳突防策略。

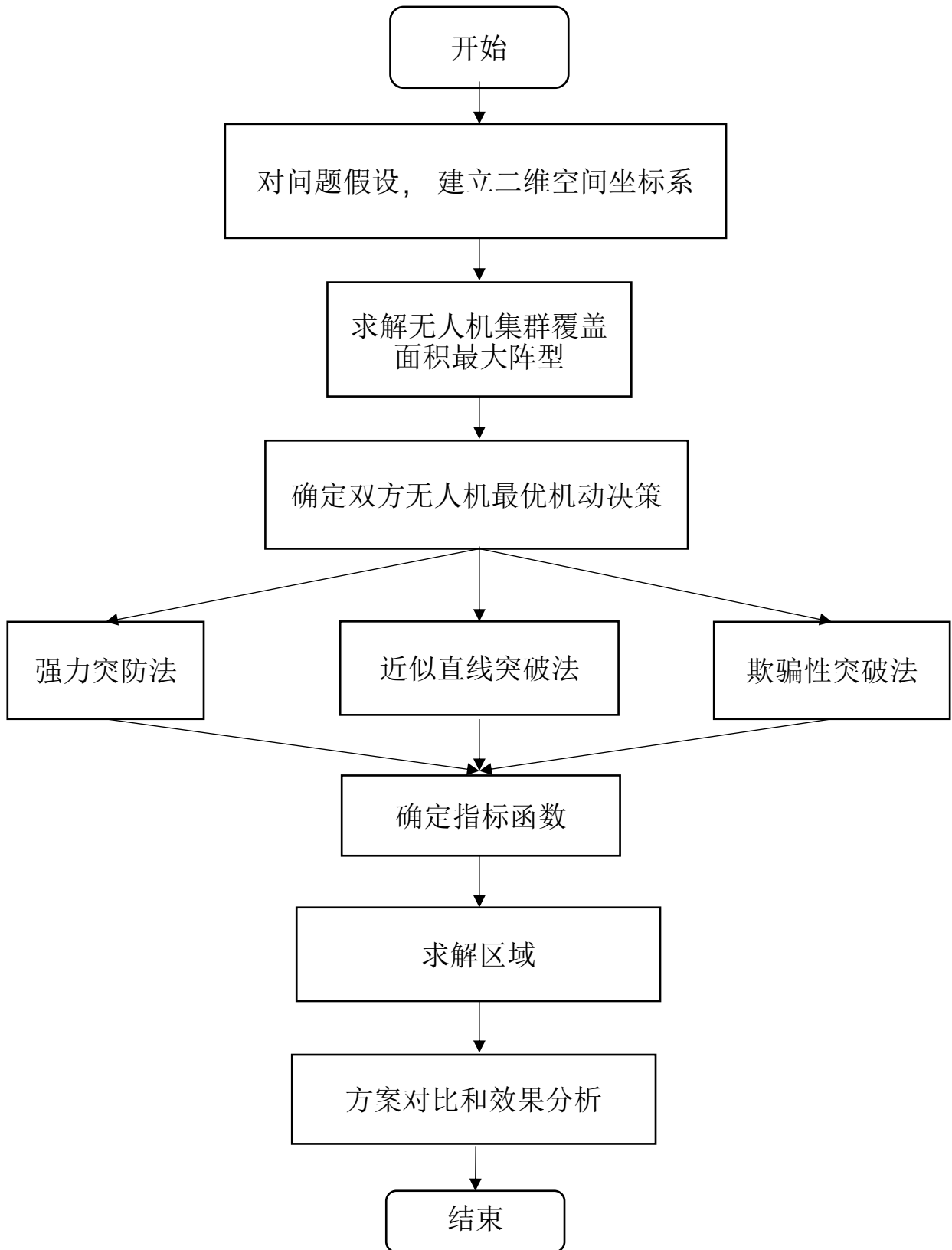


图 3-2 问题一建模流程图

3.2 模型建立

3.2.1 建立坐标系

坐标系的选取和定义及其坐标系之间的变换是构建无人机模型在二维空间中运动的基础和框架。为了方便本文的后续研究，现介绍本文所使用的二维空间坐标系及变量方便后续研究。坐标系如图 3-3,平面坐标系 O_{xy} ：在对抗区ABCD中建立坐标系，原点 O 假设在对抗区ABCD中的固定点A。

$$\begin{cases} x_{p_1} = v_{p_1} \cos \theta_{1p_1}, y_{p_1} = v_{p_1} \sin \theta_{1p_1} \\ x_{p_2} = v_{p_2} \cos \theta_{2p_2}, y_{p_2} = v_{p_2} \sin \theta_{2p_2} \\ x_e = v_e \cos \theta_e, y_e = v_e \sin \theta_e \end{cases} \quad (1)$$

一个逃跑逃跑者 E 和两个追踪者 P 的追捕动态方程。其中 θ 表示对应无人机的运动方向角度。 v_e 和 v_p 分别为逃跑者和追捕者的速度。

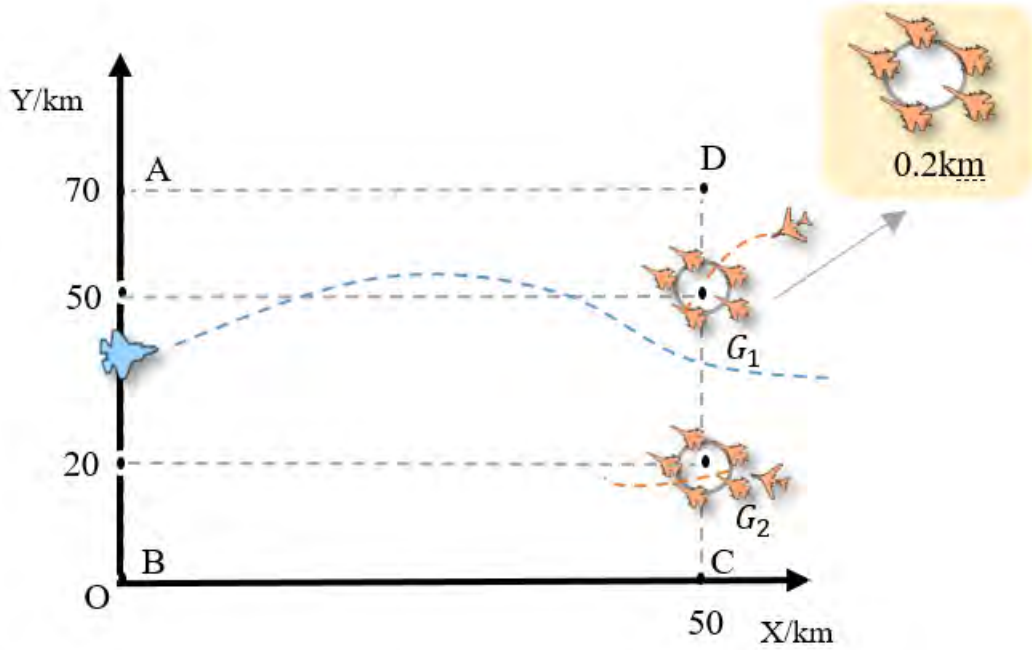


图 3-3 二维坐标系示意图

本问题中所设计采用的机动动作库的控制变量为时间的变化，每个机动动作是无人机在二维空间中能够做出的基本无人机机动动作，在实现基本无人机飞行机动动作的基础上，通过简单的根据飞行需要对一段时间内对基本无人机动作进行组合，就能够合成一些复杂的机动动作。

以下就是整个论文要用到的基础无人机飞行动作(以红方无人机角度):

1. 平飞

平飞时，无人机飞行参数条件如下： $\theta = 0$ 。即无人机的转弯角度为 0。

2. 左转弯

左转弯时，无人机飞行参数条件如下： $\frac{\pi}{2} < \theta \leq \pi$ 。即无人机的转弯角度为在 $\frac{\pi}{2}$ 和 π 之间。

3.右转弯

右转弯时，无人机飞行参数条件如下： $0 < \theta \leq \frac{\pi}{2}$ 。即无人机的转弯角度为在 0 和 $\frac{\pi}{2}$ 之间。

3.2.2 讨论无人机集群阵型

为了让红方无人机的扫描范围更大，在满足约束条件 4 的情况下，对无人机集群阵型进行讨论。

方案一：三角形方案

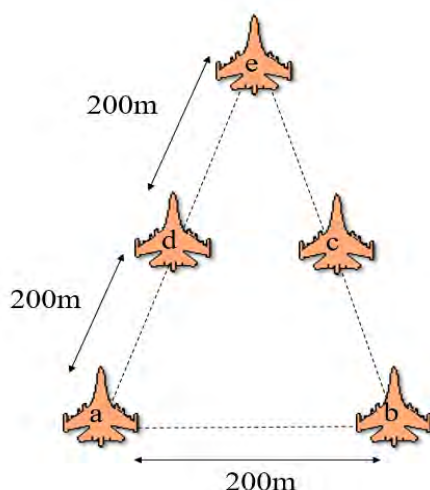


图 3-4 三角型方案示意图

考虑在满足条件约束 4 的临界情况，也就是每一架无人机与本集群中至少两架无人机的距离等于 200m，以获得最大面积。如图 3-4 所示，a,b,c,d,e 分别代表的是五架无人机，在以a圆心，半径为 200m 的圆上取到两点 b 和 d，此时 d 还需要一架无人机和其相距 200m，为了在这一个步骤让面积尽可能大，所以在以 d 为圆心，200m 为半径得圆内寻找 e 时，选择离 a 最远的点，得到 a、d、e 三点共线，此时 e 和 b 都只有一架无人机与之距离 200m,所以做以这两点为圆心，半径为 200m 的圆，交点则为 c。假定方案 1 的搜索面积为 S_1 。

方案二：圆形方案

同样考虑在满足条件约束 4 的临界情况，也就是每一架无人机与本集群中至少两架无人机的距离等于200m，以获得最大面积，很容易联想的等边五边形，也就是均匀分布的圆形，如图 3-5 所示。假定方案 2 的扫描面积为 S_2 。

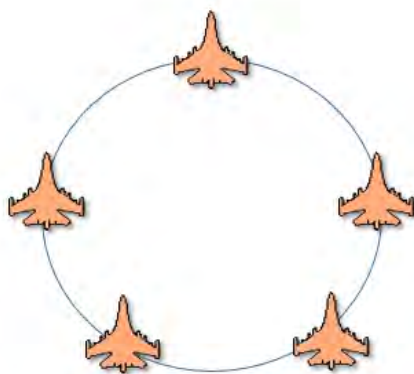


图 3-5 圆形方案示意图

为了得出方案一和方案二阵型哪个更好，需要求解分别以五架无人机为圆心、半径为300m的圆两两相交的重叠区域的面积。由于多圆相交的面积计算求解复杂，采用快速蒙特·卡罗方法对两方案面积比值估计，试验次数越多结果越准确，将实验次数设置为5000，最终计算结果方案一和方案二的面积比值为 $\frac{S_1}{S_2} = \frac{1.8268}{2.1184} = 0.864$ ，得出方案二的效果优于方案一，所以在本题中采用方案二作为无人机阵列。为了简化模型，经过计算，将方案二的整个搜索区域看成一个半径 $R_t = 470\text{m}$ 的圆，实验结果如图 3-6 所示。

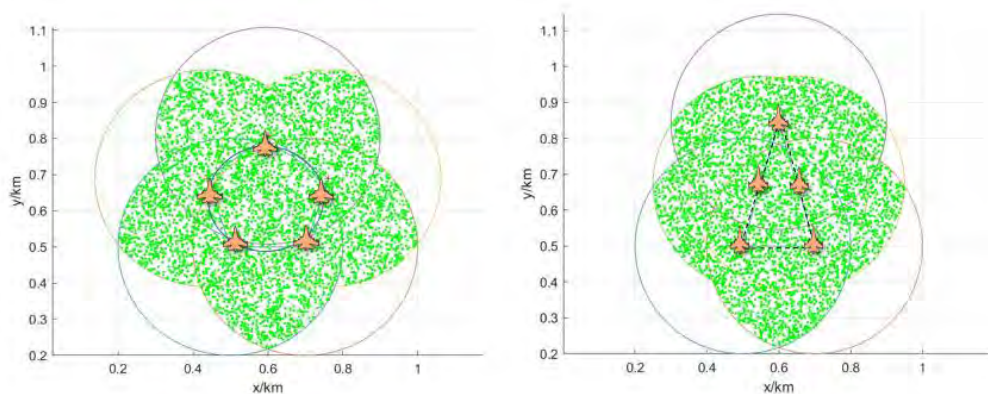


图 3-6 两种方案扫描范围对比图

3.4 确定双方无人机机动决策

3.4.1 一对二追逃博弈运动模型

建立一对二追逃博弈模型，它包括两个拦截者 P 和一个突防者 E，P 和 E 有相反的支付函数。追逃博弈模型如图 3-7 所示：

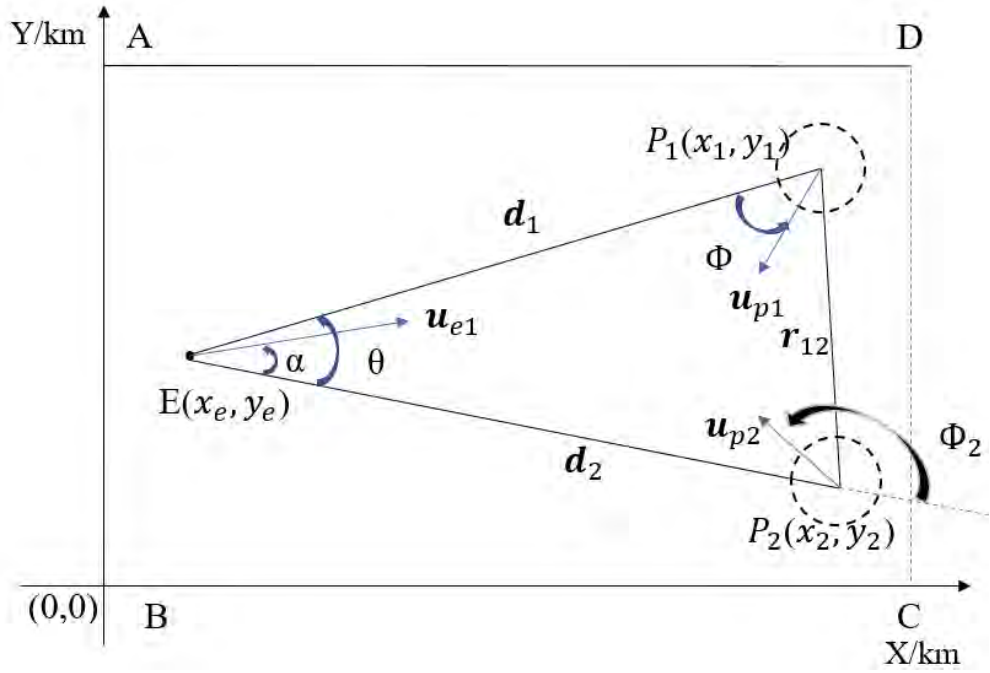


图 3-7 一对二追讨博弈模型图

离散时间下，P 和 E 的运动学模型为^[2]：

$$\begin{cases} P1(k+1) = P1(k) + u1(k) \\ P2(k+1) = P2(k) + u2(k) \\ E(k+1) = E(k) + v(k) \end{cases} \quad (2)$$

如图所示 3-7 所示，设 E 与 P₁、P₂ 的相对距离分别为 d₁、d₂，P₁ 和 P₂ 间的距离为 r₁₂，三者间夹角 $\angle P_1EP_2 = \theta \geq 0$ 。E、P₁ 和 P₂ 的相对运动方向分别设为 α 、 β_1 和 β_2 。由前文分析得到 P 的拦截半径为 470m，则当 $\min\{d_1, d_2\} \leq 470$ m 时，认为拦截成功。在离散时间下，采取红蓝双方轮流决策，并且假定 E 先决策，P 观察到 E 的运动后再决策，然后 E 再根据 P 的行动再决策，反复进行知道满足结束条件。

在本模型中，由题目所给条件，E 的速度比 P 快，即 $\gamma = \frac{v_{pmax}}{v_{emax}} = 0.8 < 1$ ，而 E 的机动能力比 P 差，即

$$R_{emin} > R_{pmax} \quad (3)$$

3.4.2 确定双方无人机机动决策

(1) 蓝方强力突破

首先考虑最简单的情况，若红方在靠近 CD 的区域对 CD 进行强力突防，即只按照垂直于 CD 的方向航行，在某个范围内红方一定无法拦截蓝方，经过计算得到图 3-8 所示区域。当蓝方从该区域内任意位置开始以垂直 CD 的方向往 CD 飞行，此时红方也立刻开始转向平行于 CD 的方向进行追截，红方无法对蓝方拦截成功。该区域边界如图所示，计算公式如下：

$$|MD| = \frac{(|DG_1| - R_p - R_t) + \frac{\pi}{2} \times R_p}{v_p} * v_e \quad (4)$$

$$|NF| = \frac{\left(\frac{|G_1 G_2|}{2} - R_p - R_t\right) + \frac{\pi}{2} \times R_p}{v_p} v_p * v_e \quad (5)$$

$$|QC| = \frac{(|CG_2| - R_p - R_t) + \frac{\pi}{2} \times R_p}{v_p} * v_e \quad (6)$$

经过计算， $|MD| = 25.1\text{km}$ ， $|NF| = 18.41\text{km}$ ， $|QC| = 25.01\text{ km}$ 。

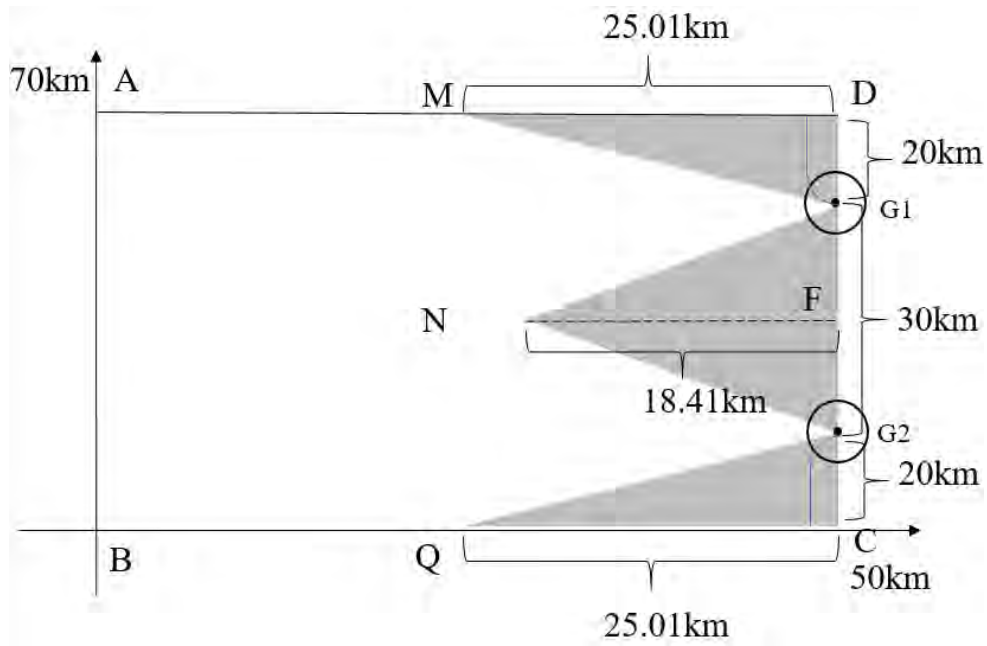


图 3-8 强力突防模型结果

(2) 蓝方近似直线运动

若蓝方不仅能垂直 CD 方向跑，还能倾斜近似直线跑，则这种情况下的博弈双方考虑的因素更多，为了方便理解，先考虑一对一追逃博弈模型。在此引入阿波罗尼奥斯圆 (Apollonius circle) 的概念，它是平面上两点的距离之比等于常数且不为 1 的点的轨迹。

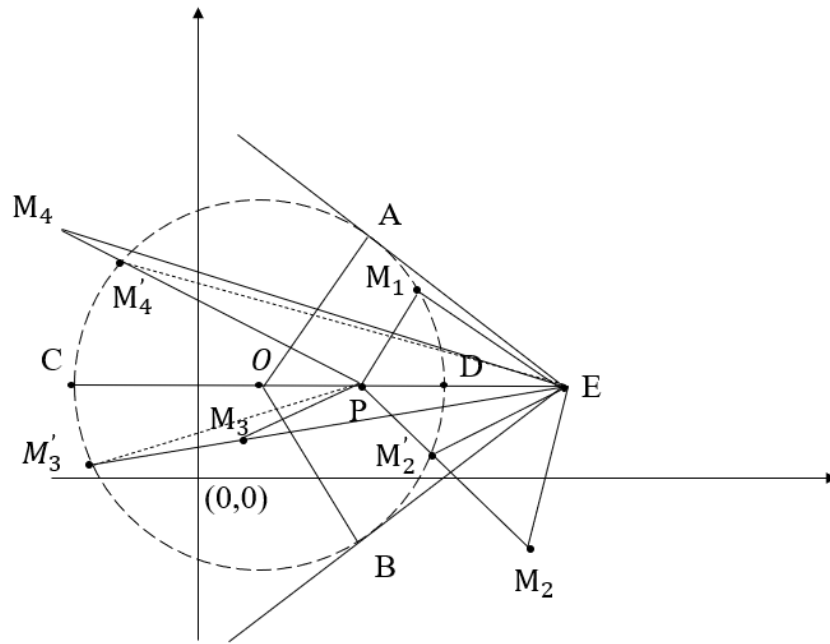


图 3-9 阿波罗尼奥斯圆示意图

如图 3-9 所示, P、E 分别表示拦截者和突防者的位置, 在 P 和 E 都以恒速运动并且 $\gamma = \frac{|PM|}{|EM|}$ ($\gamma < 1$) 的情况下, $M(x_m, y_m)$ 为平面中的任意点, 则 M 的轨迹即形成阿波罗尼奥斯圆, 并且 P 拦截 E 的拦截范围就是他们之间所形成的阿波罗尼奥斯圆^[3], 即 E 的突防路径必须经过阿波罗尼奥斯圆才有可能被 P 拦截, 并且被拦截点位于阿波罗尼奥斯圆内。证明如下:

设 M_1 为阿波罗尼奥斯圆上的任意点, 若 E 的运动方向为 EM_1 , 设 P 的运动方向为 PM_1 , 则 $\gamma = \frac{|PM_1|}{|EM_1|}$. 设经过时间 t_1 后 E 到达点 M_1 , 则:

$$v_e \times t_1 = |EM_1| \quad (7)$$

又知 $\gamma = \frac{v_{pmax}}{v_{emax}}$, 因此:

$$\frac{v_p \times t_1}{v_e \times t_1} = \frac{v_p}{v_e} = \gamma = \frac{|PM_1|}{|EM_1|}, v_p \times t_1 = |PM_1| \quad (8)$$

即, P 刚好在 M_1 点拦截到 E。

(2) 若 P 的运动方向在 $\angle AEB$ 之外, 则 E 的运动路径不会经过阿波罗尼奥斯圆, 设 M_2 为 E 运动方向上的任意点, PM_2 为 P 的追捕方向, M_2' 为 PM_2 与阿波罗尼奥斯圆的交点, 则有 $\frac{|PM_2'|}{|EM_2'|} = \gamma$. 设 P 经过时间 t_2 到达 M_2' , 可得:

$$v_e \times t_2 = |EM_2'| \quad (9)$$

若 P 和 E 分别用时间 t_{p2} 和 t_{e2} 到达点 M_2 , 则有

$$\begin{aligned}
t_{e2} &= \frac{|EM_2|}{v_e} < \frac{|EM_2'| + |M_2M_2'|}{v_e} = \frac{|EM_2'|}{v_e} + \frac{|M_2M_2'|}{v_e} = t_2 + \frac{|M_2M_2'|}{v_e} = \frac{|PM_2'|}{v_p} + \frac{|M_2M_2'|}{v_e} \\
&< \frac{|PM_2'|}{v_p} + \frac{|M_2M_2'|}{v_p} = \frac{|M_2M_2'| + |PM_2'|}{v_p}
\end{aligned} \tag{10}$$

因此，这种情况下P无法拦截E。

(3)若E的运动方向在 $\angle AEB$ 内， M_3 为E运动路径上位于阿波罗尼奥斯圆内部的任意点， M_3' 为E运动路径与阿波罗尼奥斯圆的一个交点，设P的追捕方向为 PM_3 ，此时有：

$\frac{|PM_3'|}{|EM_3'|} = \gamma$ ，设P和E分别用时间 t_{p3} 和 t_{e3} 到达点 M_3 ，则：

$$t_{p3} = \frac{|PM_3|}{v_p} < \frac{|PM_3'|}{v_p} = \frac{|EM_3'|}{v_e} = t_{e3} \tag{11}$$

即在圆内，即使双方都采用最优策略，P也确保能追捕到逃跑者E。

(4)若E的运动方向在 $\angle AEB$ 内， M_4 点为E运动路径上位于阿波罗尼奥斯圆外面的任意点，E想最快到达 M_4' 必然经过阿波罗尼奥斯圆，则过程中就被捕获。

由上述定理可以得到实时的阿波罗尼奥斯圆*的圆心 (x_*, y_*) 和半径 R_* 。

圆心坐标：

$$x_* = \frac{x_p - \gamma^2 \times x_e}{1 - \gamma^2} \tag{12}$$

$$y_* = \frac{y_p - \gamma^2 \times y_e}{1 - \gamma^2} \tag{13}$$

半径：

$$R_* = \frac{\gamma \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2}}{1 - \gamma^2} \tag{14}$$

由公式可知如果 γ 越小，即E比P的速度越快，则 R_* 就越小，此时P有更大范围的路径突防，也给P带来了更大的拦截难度，这非常符合实际情况。

显然若E朝C运动，P也会朝C运动，它们将同时在C点相遇。为了求出E的逃离区需要使用不可逃离角 θ ，有图3-9可知，图中E的不可逃离角为 $\angle AEB$ ，EA和EB分别为E点到*的切线，由

$$\sin(\varphi) = \frac{|QB|}{|QE|} = \frac{\frac{\gamma}{1-\gamma^2} \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2}}{\sqrt{\left(x_e - \frac{x_p - \gamma^2 x_e}{1-\gamma^2}\right)^2 + \left(y_e - \frac{y_p - \gamma^2 y_e}{1-\gamma^2}\right)^2}} = \gamma$$

$$\vartheta = 2\angle OEB = 2\arcsin\gamma \quad (15)$$

设PE方向角为 β ，E 的逃离区就是 $\omega = (\beta + \varphi - \pi, \beta + \pi - \varphi) \in (-\pi, \pi)$ 。

现在考虑追逃双方的策略，若E朝着向量EC的方向飞行，即在直角坐标系中的机动决策 $u(\text{pos}_e(t)) = \beta + \pi - \alpha$ ，则P的最优策略为向量PC的方向，即在直角坐标系中的机动决策 $u(\text{pos}_p(t)) = \beta + \Phi$ 。

α 和 Φ 有两种情况，当 $\alpha \leq \Phi$ ，由正弦定理，在 ΔPEC 中，有：

$$\frac{\sin(\Phi)}{\sin(\alpha)} = \frac{|EC|}{|PC|} = \gamma \quad (16)$$

得到：

$$\Phi = \arcsin\left(\frac{\sin(\alpha)}{\gamma}\right) \quad (17)$$

如图 3-10， $\alpha \leq \Phi$ 就是E就是朝着被捕获的方向走，若E朝着逃离方向（即非*覆盖区域）走， $\alpha > \Phi$ ，如图 3-10 所示。此时，即使P不能捕获到E，它仍旧可以选择机动决策 $u(\text{pos}_p(t)) = \beta + \Phi$ 来逼近 E。设经过 Δt 时间，E 运动到E'，则P的最优策略显然是直接向E航行，因

$$\frac{\sin(\pi - \Phi - \alpha)}{\sin(\Phi)} = \frac{|EP|}{v_e \times \Delta t} \quad (18)$$

得到：

$$\Phi = \arccot\left(\frac{|EP|}{v_e \times \Delta t \times \sin(\alpha)} - \cot\alpha\right) \quad (19)$$

则P的机动决策 $u(\text{pos}_p(t)) = \Phi$ 。

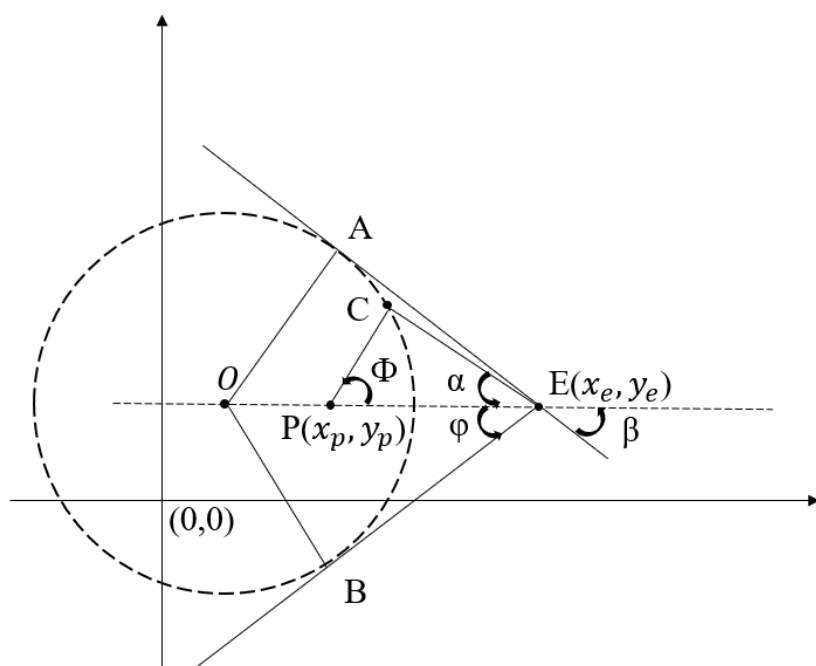


图 3-10 一对一追逃博弈

上面是考虑一对一追逃博弈，下面根据题意考虑一对二追逃博弈模型，如图 3-11 所示：

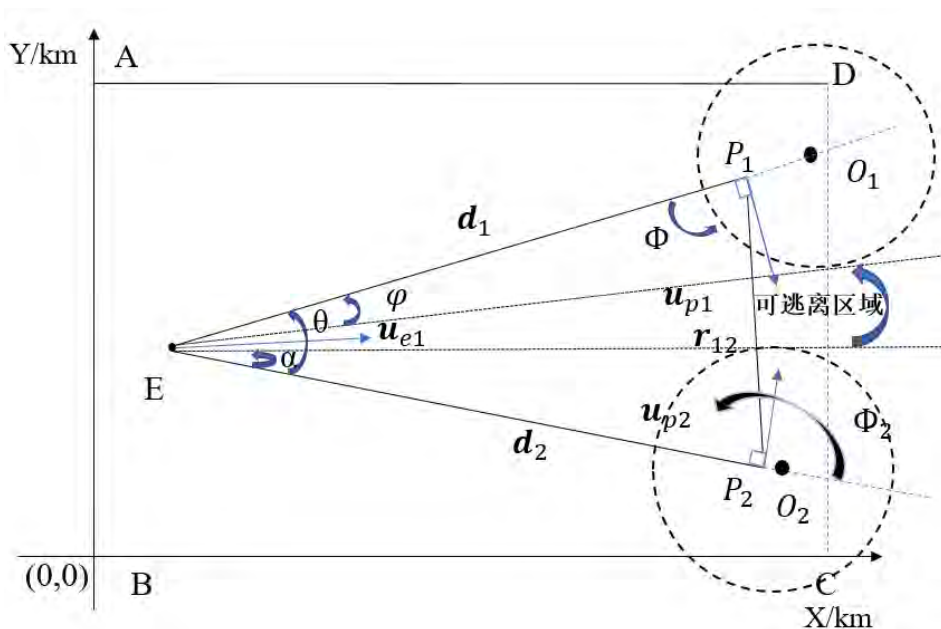


图 3-11 一对二追逃博弈

圆 O_1 和圆 O_2 分别为 P_1 、 P_2 对应的*, EA 和 EB 分别为 E 点到两圆的切线，由一对一追逃博弈模型里的推导可知 $\angle AEP_1 = \angle BEP_2 = \arcsin y$

当 $\angle BEP_1 > \arcsin y$ 时，即会出现图示的可逃离区域。则 E 只需要在可逃离区域内运动， P_1 、 P_2 就无法拦截到。也就是在此情况下， E 的逃脱条件为 $\theta > 2\arcsin y$ 。 E 的最优

机动策略集合 $U_e(\text{pos}_e(t), \text{pos}_p(t)) = (\varphi, \theta - \varphi)$ 。而P只能尽可能接近E，机动策略如一对一模型中最优机动策略。

注意在此时，我们把 P_1 、 P_2 当成了质点，也就是没有考虑搜索范围，在后文计算指标函数时将进一步考虑。又由几何关系里的圆与圆的位置关系有三种：相离、相切和相交，同时只要E冲破ABCD区域的CD边界，即为逃脱成功，所以总结可以使E逃脱的情况有：（A）两圆相离；（B）至少一圆与AD和BC不相交；（C）两圆相交或相切但是交点在CD外。

（3）蓝方欺骗性突防

在E近似直线突破中，我们没有考虑若E能在策略中随情况改变机动决策。当 $\theta \leq 2\arcsin\gamma$ 时，两圆间没有可逃离区域，随着时间的增加，P越来越接近E，E想要突防则必须要使 θ 变大，而P则需要每个机动决策集合中选择使 θ 最小的。则 θ 即可以作为E、P博弈的赢得函数。

如图3-12所示，设在 Δt 内， $u_e = \alpha$ ，蓝方从E点运动到 E' ，则P想要使 $\theta' < \theta$ ，必须使 P_1 、 P_2 分别运动到 P_1' 、 P_2' 。则蓝方的最优策略为让 θ' 最大。

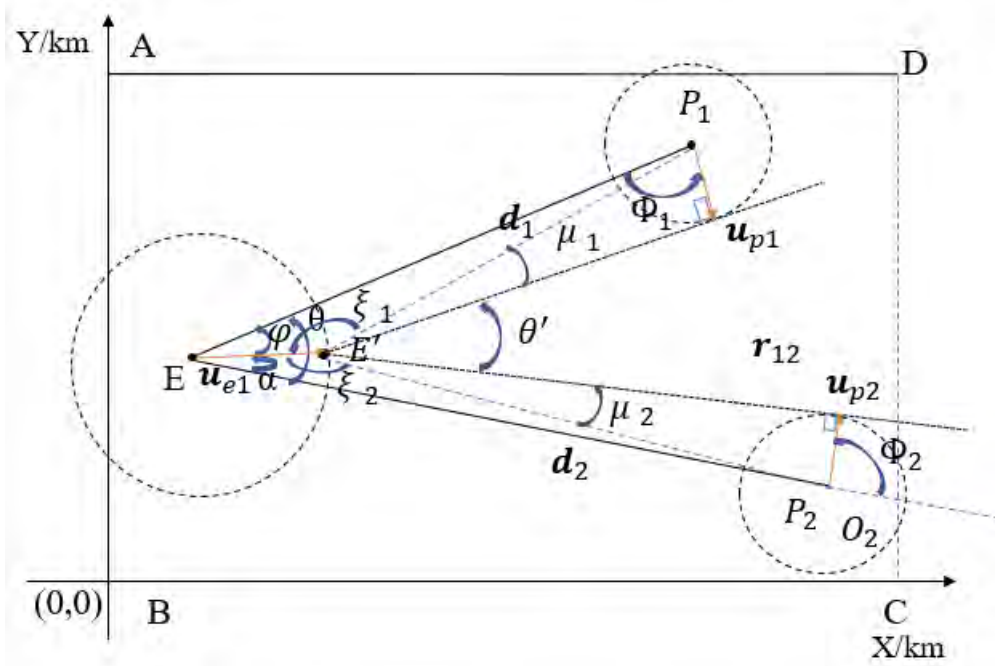


图 3-12 蓝方无人机欺骗性突防分析图

如图，在 $\triangle EE'P_2$ 中，由余弦定理得到 $|E'P_2| = \sqrt{v_e^2 \Delta t^2 + d_2^2 - 2v_e \Delta t \times d_2 \cos \alpha}$

并且有

$$\xi_2 = \arccos\left(\frac{v_e \Delta t - d_2 \cos \alpha}{|E'P_2|}\right) \quad (20)$$

$$\mu_2 = \arcsin\left(\frac{v_p \Delta t}{|E'P_2|}\right)$$

$$\xi_1 = \arccos\left(\frac{v_e \Delta t - d_1 \cos(\theta - \alpha)}{|E'P_1|}\right)$$

$$\mu_1 = \arcsin\left(\frac{v_p \Delta t}{|E'P_1|}\right)$$

其中

$$|E'P_1| = \sqrt{v_e^2 \Delta t^2 + d_1^2 - 2v_e \Delta t \times d_2 \cos(\theta - \alpha)}$$

最终求得

$$\begin{aligned} \theta' = 2\pi - \arccos\left(\frac{v_e \Delta t - d_2 \cos \alpha}{|E'P_2|}\right) - \arcsin\left(\frac{v_p \Delta t}{|E'P_2|}\right) - \arccos\left(\frac{v_e \Delta t - d_1 \cos(\theta - \alpha)}{|E'P_1|}\right) \\ - \arcsin\left(\frac{v_p \Delta t}{|E'P_1|}\right) \end{aligned} \quad (21)$$

由图 3-12 可知, 当 $d_1 = d_2$ 时, $\alpha = \frac{\theta}{2}$, $\max(\theta') = \theta$; 当 $d_1 < d_2$ 时, $\alpha < \frac{\theta}{2}$; 当 $d_1 > d_2$ 时, $\alpha > \frac{\theta}{2}$ 。由公式可见 θ' 的计算相当复杂, 所以我们进行多次尝试, 发现当

$$\sin \alpha = \frac{d_1 - d_2 \cos \theta}{r_{12}} \quad (22)$$

$$\cos \alpha = \frac{d_2 \sin \theta}{r_{12}} \quad (23)$$

其中 $r_{12} = \sqrt{d_1^2 + d_2^2 - 2d_1 d_2 \cos \theta}$ 效果最好。即 $u_e(\text{pos}_e(t), \text{pos}_p(t)) = \alpha$ 。由图 3-12 红方的机动决策

$$u_{p1}(\text{pos}_p(t), \text{pos}_p(t)) = \frac{3}{2}\pi + \alpha - \theta - \xi_1 - \mu_1 \quad (24)$$

当 $d_1 > d_2$ 时,

$$u_{p2}(\text{pos}_e(t), \text{pos}_p(t)) = -\frac{\pi}{2} + \alpha + \xi_2 + \mu_2 \quad (25)$$

3.4.3 确定支付函数

由上述分析得知, 本问题简化成当E处于属于ABCD的 pos_e , 由公式分别计算E和P1、P2两个无人机集群的阿波罗尼奥思圆, 根据两个圆在ABCD中不同区域的位置关系, 来计算出模型的支付函数V, 根据V判断E是否能逃离的问题。当V为1即为可逃离, 为0即不可逃离。最后计算满足 $V = 1$ 的E的集合。即 $\exists E$ 的位置 $\text{pos}_e \in ABCD$, 使得

$V(*_{e,p1} \cup *_{e,p2}) = 1$, 求解 $S(\text{pos}_e)$ 。

针对 3.4.2 的不同 3 对抗策略有 3 种不同的指标函数。

对于强力突围法，它的求解方案和结果在 3.4.2 已经讨论，不再赘述。

对于近似直线突围法，经过计算，此种情况下的结果从CD最远的地方以 v_e 直线跑到CD的时间远小于 360s,所以本方案不用考虑时间。它的指标函数 $V(*_{e,p1} \cup *_{e,p2})$ 的求解算法如表 1。

表 1 近似直线突围法

近似直线突围法求解算法步骤和分析
1.对ABCD区域进行抽样
2.初始化参数
3.for $E \in ABCD$ do,对 ABCD 区域内的点做遍历
4.计算E与 P_1, P_2 的 $*_1, *_2$ 的圆心坐标 (x_1, y_1) 、 (x_2, y_2) 和半径 R_1, R_2
5.判断 $*_1, *_2$ 位置关系，若 $\theta > 2\arcsin y$,则两者相离必定可逃离， $V = 1$ ，若两者相交或相切，即 $ R_1 + R_2 < (x_1, y_1), (x_2, y_2) $ ，则分 3 种情况：①判断两圆是否至少有一圆不和边界 AD 和 BC 产生交点，若是 $V = 1$ ；②否则判断两圆所有交点或切点是否在 CD 边界外，若是 $V = 1$ ；③否则 $V = 0$
6.存储 $V = 1$ 的点于 ω_2
7.end

对于欺骗性突围法，在这种情况下需要动态的考虑，经过 3.4.3 的分析可知，在追逃双方都采取最优策略的情况下，也无法再使两个阿波罗尼奥圆相离，若要使得满足可逃离条件，则只有可能是蓝方通过欺骗性策略将红方两个无人机集群欺骗到一边导致满足两圆至少有一圆不和边界AD和BC产生交点，产生可逃离范围，然后蓝方成功逃离。它的指标函数 $V(*_{e,p1} \cup *_{e,p2})$ 的求解算法如表 2。

表 2 欺骗性直线突围法

欺骗性突围法求解算法步骤和分析
1.对ABCD区域进行抽样
2.初始化参数
3.for $E \in (ABCD - \omega_1 - \omega_2)$ do(对前两种方法的不可逃脱区域内点遍历)
4.计算E与 P_1, P_2 的 $*_1, *_2$ 的圆心坐标 (x_1, y_1) 、 (x_2, y_2) 和半径 R_1, R_2
5.while($*_1, *_2$ 不和边界AD和BC产生交点)
6.求 $d_1, d_2, \angle P_1 E P_2$,
7.求 $u_{p1}(\text{pos}_e(t), \text{pos}_p(t))$, $u_{p2}(\text{pos}_e(t), \text{pos}_p(t))$, $u_e(\text{pos}_e(t), \text{pos}_p(t))$

8.根据运动学模型，计算 $\text{pos}_e(t)$ 、 $\text{pos}_{p1}(t)$ 和 $\text{pos}_{p2}(t)$

9. $\text{count_time}++$

10.end

11.判断 count_time 加 E 沿着与边界相离的*的切线方向到边界的时间是否小于 360s，是则满足可逃离点的要求，放入 ω_3

12.end

3.5 模型结果对比与分析

通过 Matlab 对算法进行建模，得到模型结果如下所示，模型均采用 1:50 比例尺。

3.5.1 模型结果

(1) 蓝方近似直线

首先，建立了没有考虑红方无人机扫描范围的简化模型，模型结果如图 3-13 (a) 所示。当蓝方无人机处于矩形区域 ABCD 中的黑色区域时，无论红方无人机集群采用什么拦截策略，蓝方无人机总能采用合适的策略实现成功突防。计算该区域的面积为 1381.55 km^2 。

进一步精化模型，考虑红方无人机集群扫描范围，得到的模型结果如图 3-13 (b) 所示。与图 3-11 相比，蓝方无人机可以逃脱的区域有所缩小（红色框标注），经计算，该区域的面积为 1358.89 km^2 。

(2) 蓝方欺骗性突破

最后，对蓝方无人机欺骗性突破策略进行模拟仿真，得到模型结果如图 3-13 (c) 所示。当蓝方无人机处于图中灰色区域时，可以通过欺骗性突破策略实现成功突围。对该策略进行模拟时，为弥补实验精度造成的误差，加强了对突围时间的要求，将蓝方无人机实施欺骗性策略的时间限制在 50 s 内。若蓝方无人机从某处开始在 50 秒的时间内无法成功将诱骗红方无人机集群到一边，产生可逃脱范围，该处仍为不可逃脱点。图 3-14 给出了几处蓝方无人机处于欺骗性突破区域的逃脱轨迹。经计算，通过欺骗性突破，蓝方可突破区域增加 169.66 km^2 ，共 1528.55 km^2 。

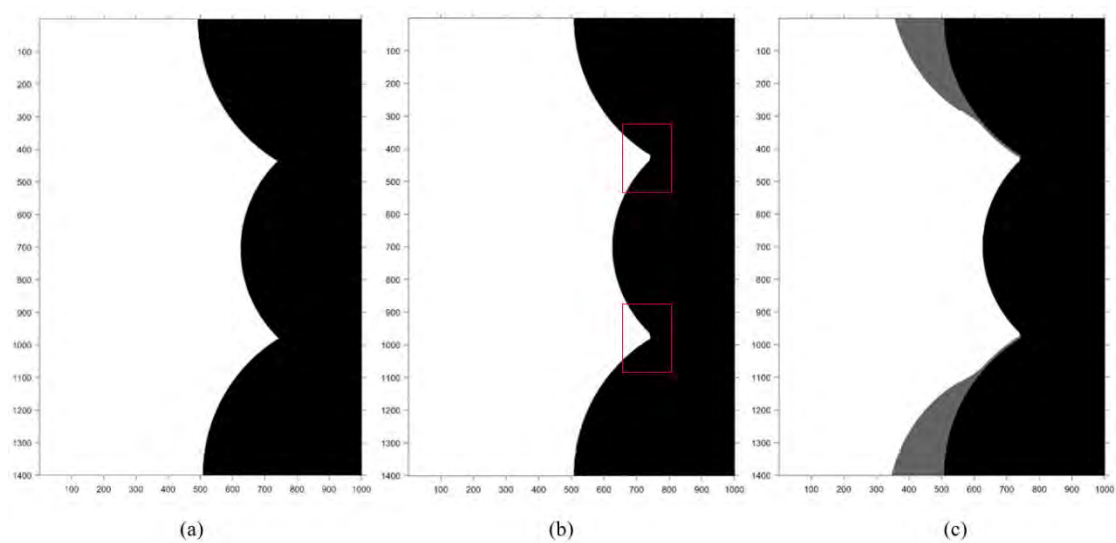


图 3-13 三种优化算法结果示意图

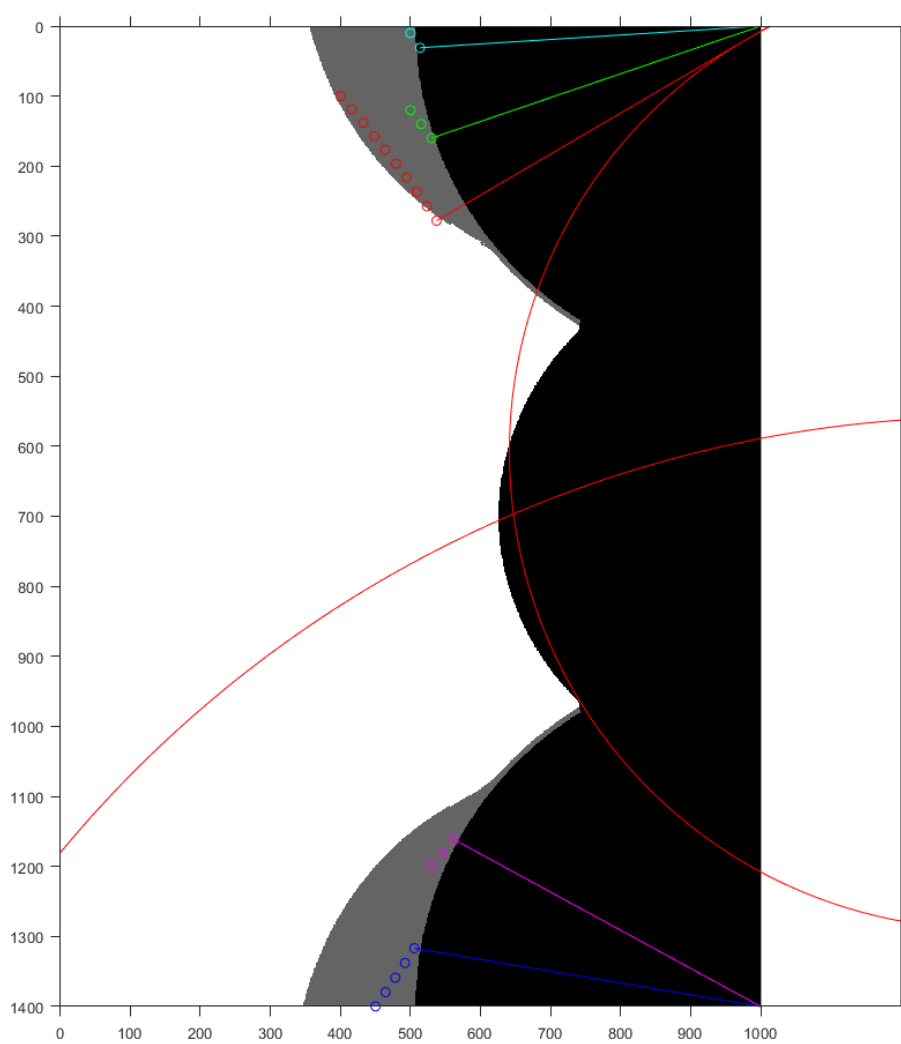


图 3-14 蓝方欺骗性突防策略轨迹图

3.5.2 结果分析

模型建立后可以看到，在三种不同的算法下，蓝方无人机逃脱区域相似的，说明本文为解决问题所采用的算法具有非常高的准确性。三种算法层层递进，算法逐步优化，复杂度和精确度也逐步增高，从而得到最终的结果。其中，对于近似直线突围策略，当蓝方无人机处于该策略形成的可逃脱区时，蓝方无人机最优突防策略是沿着所处位置与阿波罗尼奥斯圆的切线方向逃脱；对于蓝方欺骗性策略，当蓝方无人机处于该策略形成可逃脱区时，最优突防策略是当完成欺骗策略后，蓝方沿着与阿波罗尼奥斯圆的切线方向逃脱。

四、问题二

4.1 问题分析

问题二是在问题一一对二追逃博弈问题基础上，考虑红方 2 个无人机集群在 CD 边界上的圆周中心 G_1 、 G_2 未给定的情况，给定蓝方无人机的初始点，建模分析是否存在一个通道带宽 M 的下限，当实际通道带宽比 M_{min} 大时，蓝方无人机一定能突破红方无人机集群的拦截；并且给出此种情形下蓝方无人机时间最短的突防策略。它和问题一的不同在于，问题一求蓝方无人机能够突防的出发点集合，也就是在给定 M 、 G_1 、 G_2 下求逃离区，而本问题求解的是给定蓝方出发点，求蓝方出发点在逃离区内 M 的最小临界值。当 $M \geq M_{min}$ 时，蓝方一定能通过，也就是不论蓝方采取何种突防策略（近似直线突围、欺骗性突围），均能通过；而当 $M < M_{min}$ ，蓝方不一定能通过，分为两种情况：蓝方不论采取何种突围法都不能通过；蓝方采取欺骗性突围能够通过。以问题一的 $M = 70 \text{ km}$ ， $G_1 G_2 = 30 \text{ km}$ ， $CG_1 = 20$ ， $DG_2 = 20$ 的条件为例，根据 3.5 的结果，无论是近似直线突围还是欺骗性突围，AB 中点都不在逃离区内，因此所求 $M_{min} > 70 \text{ km}$ 。

要求解 M 的临界值，必须要找到此时 E 、 P_1 、 P_2 的临界状态。

根据题目和参考文献，给出求解临界值 M_{min} 的步骤如下：

第一步：修改问题一模型的初始条件；

第二步：确定红蓝双方机动决策 $u(\text{pos}_e(t), \text{pos}_p(t))$ ；

第三步：理论分析在何种状态下满足临界条件；

第四步：找到临界状态后，利用不可逃离角 θ 计算在临界状态下的 M ；

第五步：分析结果与绘制结果图。

4.2 模型建立

4.2.1 确定红蓝双方机动决策

根据 4.1 分析，当 $M \geq M_{min}$ 时，近似突围和欺骗性突围的逃离区中都包含 AB 中点 (K)；当 $M < M_{min}$ ，近似突围的逃离区中不包含 K ，而欺骗性突围的逃离区有可能包含 K 。则问题简化成已知 $\text{pos}_e(0)$ ，找到 $\text{pos}_{p_1}(0)$ 和 $\text{pos}_{p_2}(0)$ 在 CD 边界上， $K \notin \omega_2$ ，使得 M 为临界值 M_{min} 。考虑在现实中，我们很容易想到如果 P_1 距离 E 比 P_2 距离 E 显著远时，因为 P 、 E 速度比 γ 小于 1，所以 E 因为其速度更快，可以轻易从 P_1 、 P_2 之间穿过。因此，对于 P 来说最好的策略就是 P_1 、 P_2 在起始点位置尽量保持与 E 之间距离相等，使得 $b = d_1/d_2 =$

1, 具体验证在后面的 θ 分析里。知道 $b = 1$, 则红方的 $\text{pos}_{p_1}(0)$ 和 $\text{pos}_{p_2}(0)$ 必定是对称分布在CD上, 由阿波罗尼奥斯圆定理, 我们可以知道在 P_1 、 P_2 相对应的 $*_1$ 、 $*_2$ 也必定是关于CD中线对称的。

4.2.2 求解临界状态下的 M_{\min}

在解决问题1时我们引入了不可逃离角 θ , 由 θ 的定义我们知道当 $\theta > \theta$ 时, 蓝方突防成功; 当 $\theta = \theta$ 时, 蓝方处于突防的临界状态; $\theta < \theta$ 时, 蓝方需要采用欺骗性突防才可能成功。注意, 这里我们没有考虑两圆与AD和BC边界不相交时, 因为这时不可能为 M_{\min} 。因此本题要求解 M_{\min} 双方的状态即为当 $\theta = \theta$ 时, 蓝方恰好不能突防成功的状态。由图4-1所示, G_1 , G_2 处于CD两点时可以夹角 θ 最大, 若此时的 $\theta = \theta$, 则可以求得使蓝方恰好不能近似直线突防成功的 M 最小值。

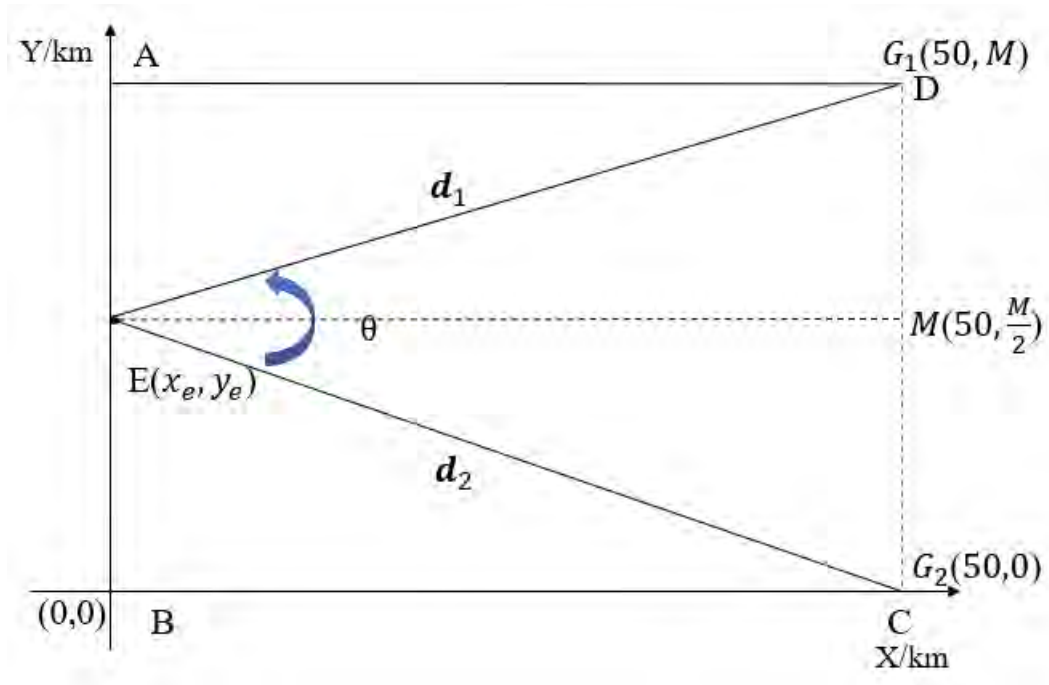


图 4-1 临界状态下一对二追逃博弈示意图

因为

$$\theta = 2\arcsin y \quad (26)$$

则由正切公式:

$$\frac{M}{2} = L \times \arcsin y \quad (27)$$

可以得到

$$M = 2L\arcsin y \quad (28)$$

计算得到 $M_{\min} = 133.2514 \text{ km}$ 。

4.2.3 实验结果分析

通过 Matlab 对计算结果进行验证， $M_{min} = 133.2514 \text{ km}$ 是正确结果。

图 4-2（左）显示，当 $M=133.2514 \text{ km}$ 时，红方无人机集群处于能够有合适策略拦截蓝方无人机的临界状态，此种情形下蓝方无人机最优突防策略为沿着图 4-2（右）中的 EC 或 ED 轨迹直线飞行。

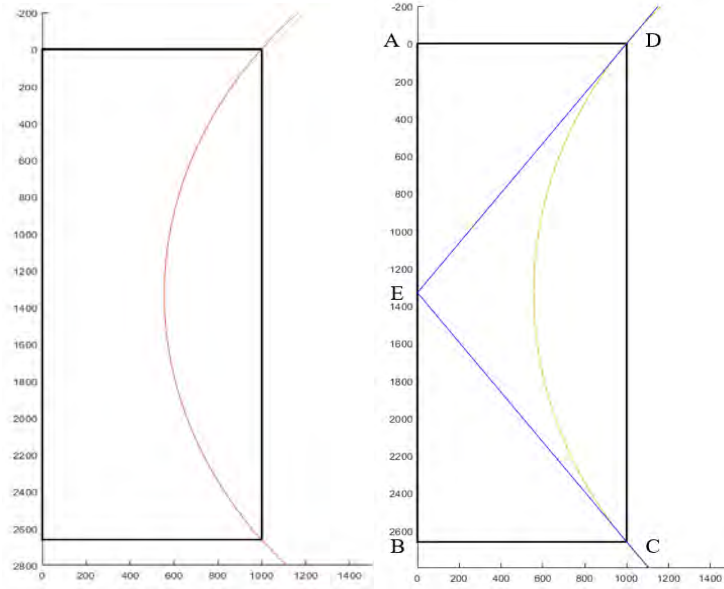


图 4-2 $M = M_{min}$ 状态实验结果验证

图 4-3 显示，当 M 略大于 M_{min} 时，两个红方无人机集群无论如何布防，蓝方无人机存在合适的策略突防成功。

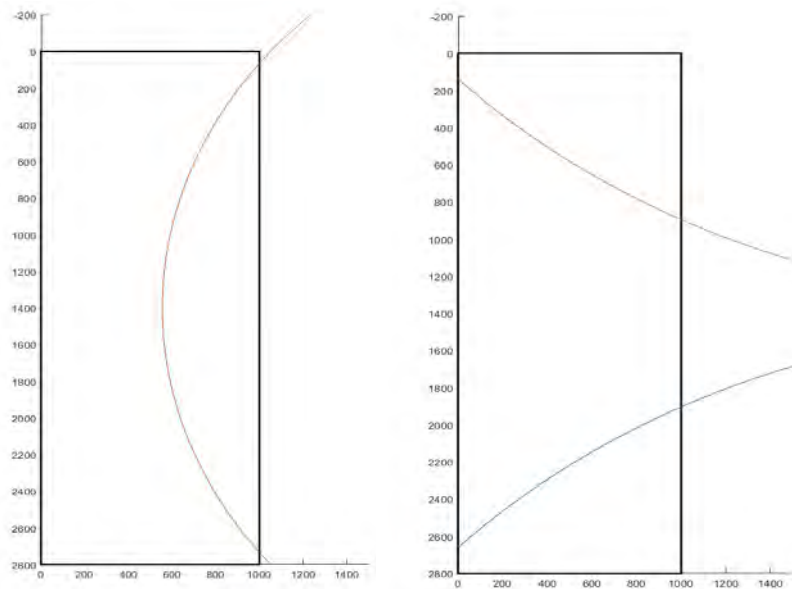


图 4-3 $M > M_{min}$ 状态时结果示意图

图 4-4 显示，当 M 略小于 M_{min} 时，蓝方无人机并不能一定脱逃成功，且阿波罗尼奥斯圆包围更多矩形区域，验证了本文求得的 M_{min} 为临界值。

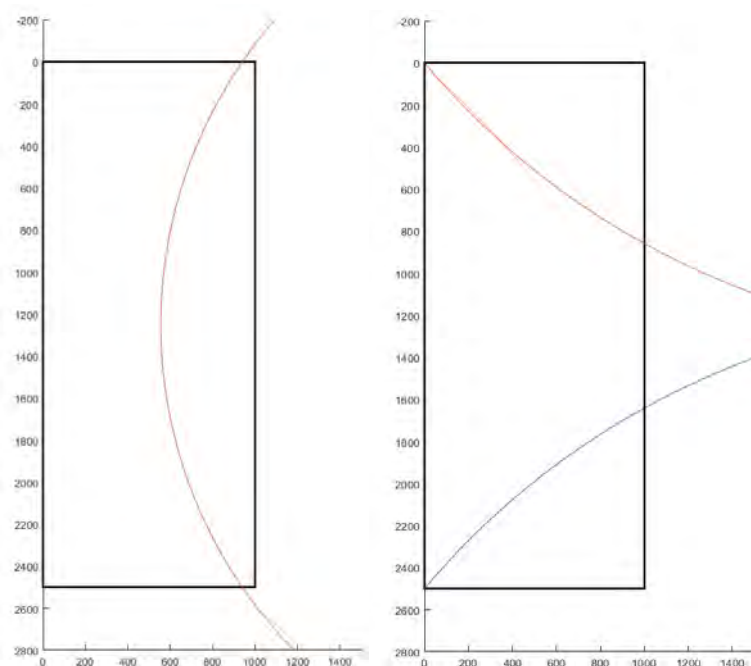


图 4-4 $M < M_{min}$ 状态时结果示意图

五、问题三

5.1 问题分析

问题三是在问题一的一对二追逃博弈问题基础上考虑了约束条件（6），并且红方运载机增加了一个波次释放无人机集群。根据问题一和问题二的结果，已知 M ，只要 G_1 、 G_2 点设置在 CD 边界上，蓝方无人机从 AB 中点出发，必定突防失败，所以问题三就可以理解为红方通过第一个波次一定能拦截到蓝方无人机，增加一个波次的目的可以实现最优拦截效果。

由题意，给定蓝方起始点和通道带宽 M ，红方每架运载机可分两个波次共发射 10 架无人机，组成两个无人机集群执行拦截任务。由于题目限定对于运载机每个波次发射的无人机数量不小于 3，无人机的数量影响了无人机集群的扫描范围，讨论红方两架运载机两个波次发射的无人机数量，使得无人机集群扫描面积达到最大。设定红蓝双方的最优机动策略，根据问题一、二中的结论，在满足运载机和无人机集群的位置约束，利用数学理论计算出每架运载机第二波次发射的时刻、位置和无人机集群的中心位置。进一步建模计算通道带宽的上限，当实际通道带宽小于 M_{max} 时，红方无人机集群总是能拦截成功，当实际通道带宽大于 M_{max} 时，红方不一定可以成功拦截。计算通道带宽上限可以从问题二相反的角度进行分析。

根据题目和参考文献，给出讨论和建模的步骤如下：

第一步：确定两个波次的无人机阵型；

第二步：确定红蓝双方机动决策 $u(\text{pos}_e(t), \text{pos}_p(t))$ ；

第三步：根据约束条件，讨论第二波次发射的无人机集群的中心位置；

第四步：建模分析，找到临界状态，利用阿波罗尼奥圆计算在临界状态下的 M_{max} ；

第五步：分析结果利用仿真方法来验证上述结果是否正确。

5.2 模型建立

5.2.1 确定两个波次的无人机阵型

根据 3.2.2 推出，当无人机数量确定时，扫描范围最大的无人机集群在圆上的均匀排布。由题意，考虑无人机阵型为三角形、正方形、等六边形、等七边形，如图 5-1 所示，利用快速蒙特卡罗算法进行方案对比，当无人机数量为 7 的时候，扫描范围最大，参考问题一将其扫描范围简化为 $R_{max} = 530\text{ m}$ 的圆形；数量为 3 的时候扫描范围最小，扫描范围可简化为 $R_{min} = 415\text{ m}$ 的圆。在本题条件下，推断出 E 肯定不能突防成功，并且运载机速度大于红方无人机的速度，红方为了实现最优拦截，需要第二波次的无人机进行来进行最快的拦截，所以第二波次的无人机的扫描范围为最大，此时无人机数量为 7。由于红方运载机最多承载 10 架无人机，所以红方运载机第一波次释放 3 架无人机。

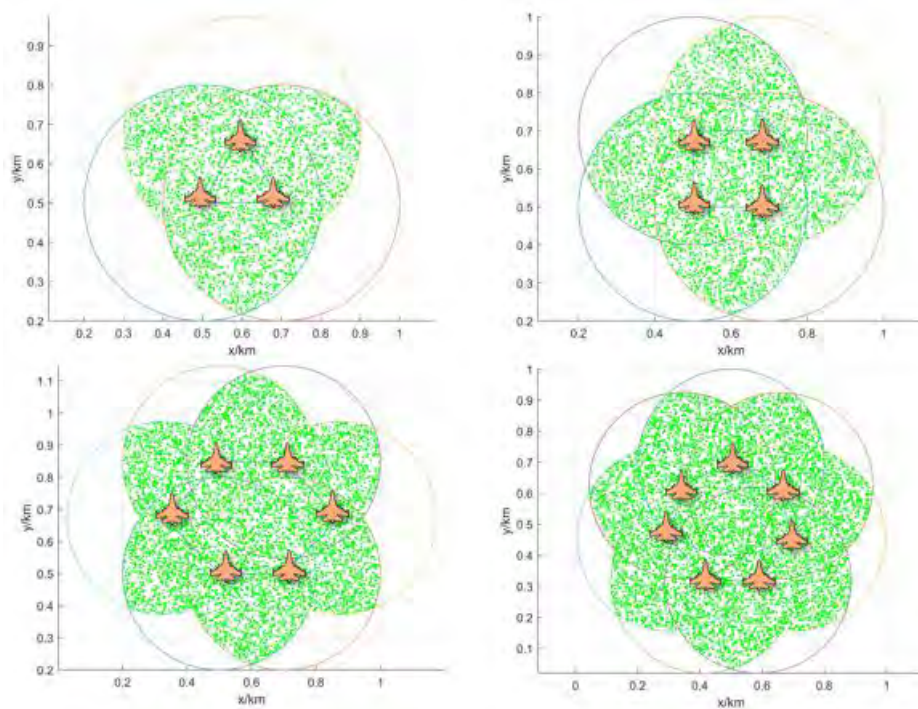


图 5-1 不同无人机数量下扫描范围分析图

5.2.2 红蓝双方机动决策

由于蓝方不可能突防成功，所以直接采用强力突防法，对于红方来说，模型结果如图 5-2 所示。

当 $t = 101.6 \text{ s}$ ，运载机已经与E在同一点，考虑当运载机距离蓝方无人机 3.6 km 时，已经能发射无人机将其拦截，所以时间 $t \approx 101.6 - \frac{3.6}{0.3} = 100.4 \text{ s}$ 。

5.3 求解 M_{max}

5.3.1 模型理论分析

接下来分析在上述对 $\text{pos}_{p1}(0)$ 和 $\text{pos}_{p2}(0)$ 的限制下，E、 P_1 、 P_2 在未给定M的ABCD中几种状态。

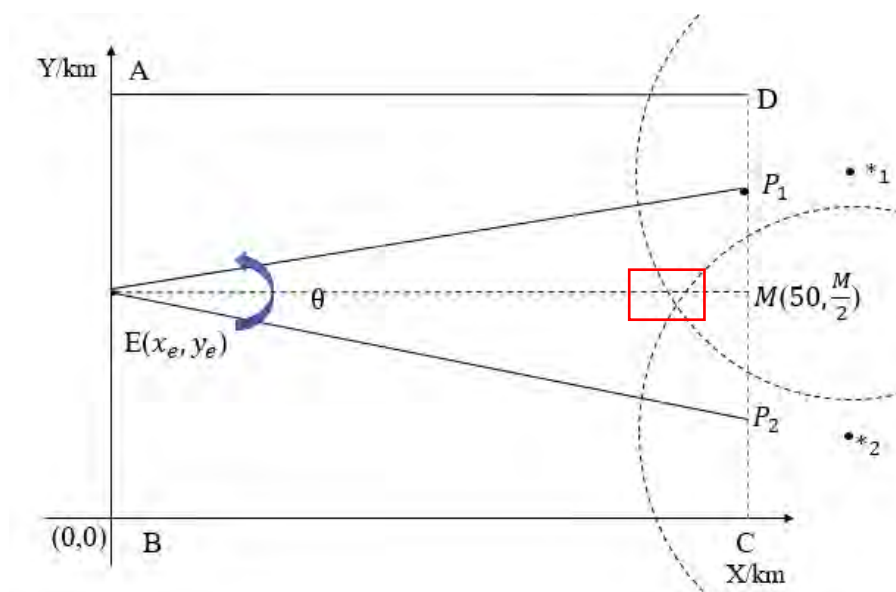


图 5-3 (a) $M < M_{max}$ 时状态示意图

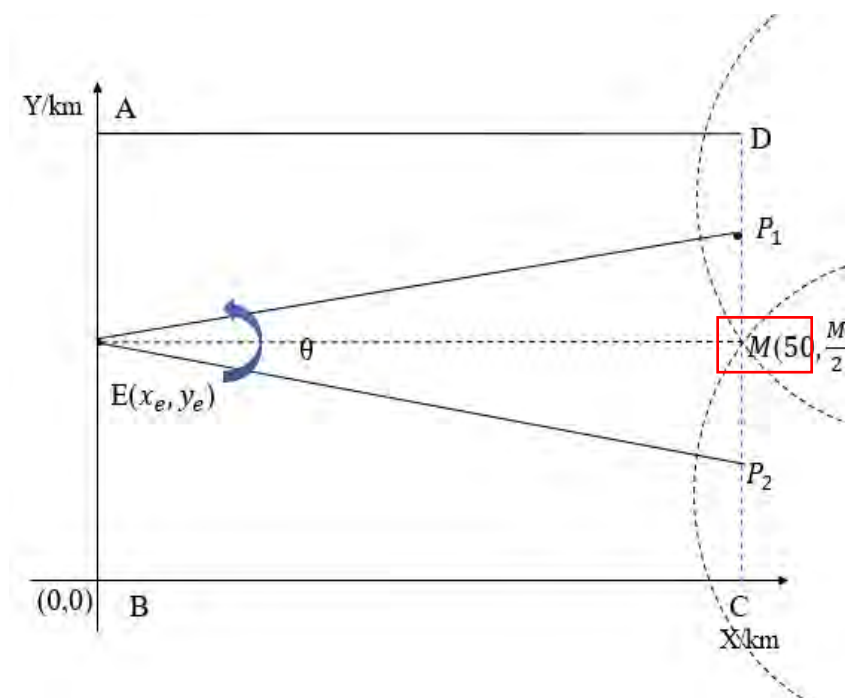


图 5-3 (b) $M = M_{max}$ 时状态示意图

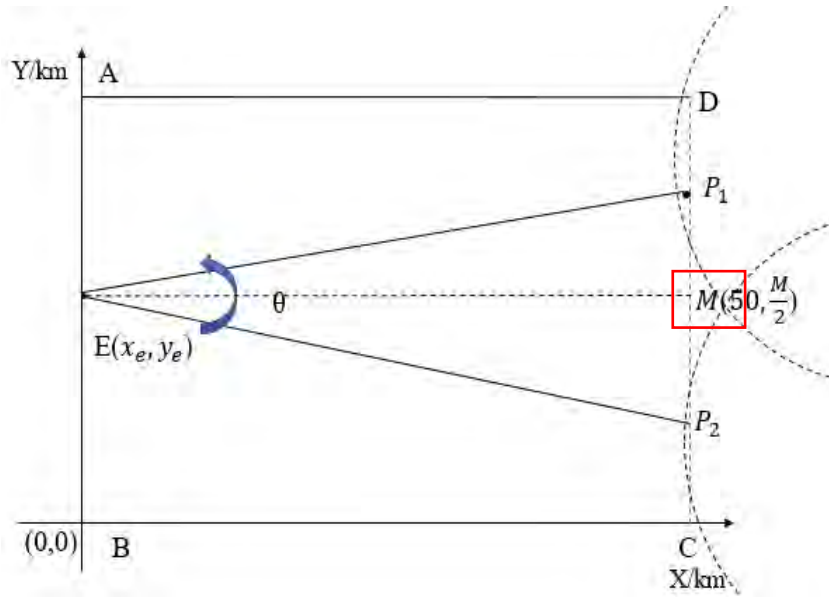


图 5-3 (c) $M > M_{max}$ 时状态示意图

经过前面分析可知对称情况下 $*_1$ 、 $*_2$ 的交点必定在CD的中线上，如图 5-3(a)所示，当 $*_1$ 、 $*_2$ 处于如图所示位置时，可以看到采用近似直线法肯定无法突围成功，因为两圆在ABCD区域内相交，若要找到M的最大值，则必须要使两圆的交点顺着x轴正向往后移；当交点超过CD边界时，如图 5-3(c)所示，则AB中点包含在近似直线法逃离区内；而当交点刚好在CD边界上时，则形成了临界状态，即近似突围法的逃离区中刚好不包含AB中点的位置，此时使 $M_{max} = y_{p1}(0) - y_{p2}(0)$ 。

5.4 求解临界状态下的 M_{max} 和分析

建立 4.2.1 所述坐标系，经过上述分析，根据阿波罗尼奥斯圆定理，得到发射第一波无人机的临界状态如图 5-4 所示。

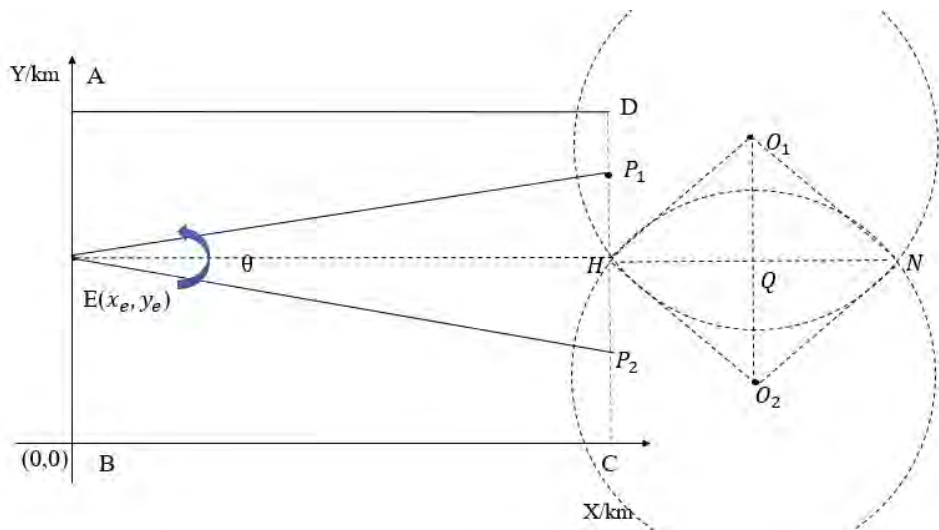


图 5-4 无人机的临界状态图

图 5-5 结果验证图

图 5-5 为 M_{max} 结果的仿真图，可以验证结果的正确性。

六、问题四

6.1 问题分析

在问题三的中，当红方的每架运载机分两波对蓝方的无人机进行追捕时，在满足题目约束条件五的情况下，利用运载机 $v_r > v_e$ 的优势，在合适的时间、合适的位置释放第二波运载机就可以对蓝方无人机进行抓捕。问题四是在问题三的基础上对红蓝双方无人机的数目进行增加，在满足蓝方无人机突防数目最多的情况，以及红方无人机拦截数目最多的情况下，红蓝双方的最优拦截策略和最优突防策略。因此问题四可以理解为多对多的追捕博弈问题，由于多个追捕者和多个逃跑者的追捕问题的研究是一个非常复杂以及困难的工作，以及数学建模时间的限制，很难为其很快的构建一个数学形式的解决方案，所以本论文基于参考论文^[2]针对多对多的追捕博弈问题提出了一种分层追捕算法（如图 5-1）进行讨论，并提出红方的最优拦截策略和蓝方的最优突防策略。

假设每一红方无人机（集群）至少追捕一个蓝方，而且只能捕获一个蓝方，则多个红方无人机（集群）可以转化为两个层次上的问题。第一层确定拦截到蓝方无人机数量最大的原则。第二层在基于问题三的追捕策略上增加对红方无人机（集群）形态改变的考虑，红方无人机（集群）实现对蓝方无人机的围捕，对应的蓝方无人机实现不同于问题一、二、三中近似直线/欺骗性运动的自由运动进行突破。由于在追捕过程中，红方无人机（集群）的数量总是大于蓝方无人机的数量，则在第一层将多对多追捕-逃跑博弈分层分解为一对二博弈后，必定存在没有参与追捕的红方无人机（集群），此时这些无人机（集群）根据“活跃区域”原则进行压迫式追捕。

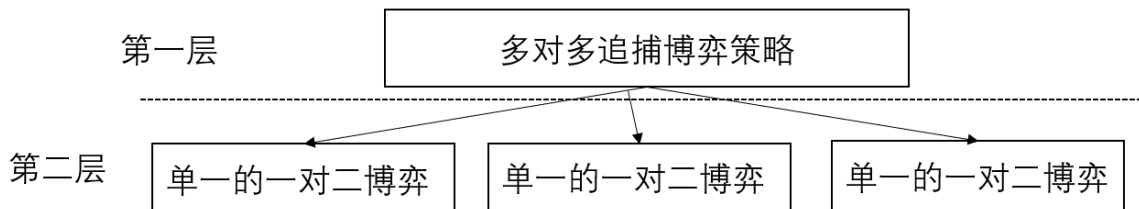


图 6-1 分层算法

根据题目和参考文献，讨论红方的最优拦截策略和蓝方的最优突防策略步骤如下：

第一步：讨论红方无人机（集群）变换的阵型，在约束条件的制约下，使其可以变换为围捕蓝方无人机的集群阵型。

第二步：根据分层追捕算法以红方运载体为追捕者。根据拦截到蓝方无人机数量最大的原则，进行多对多追捕-逃跑博弈分层分解为一对二博弈分析。

第三步：分析红方最优拦截策略和蓝方最优突防策略。

第四步：总结红方最优拦截策略和蓝方最优突防策略。

6.2 模型建立

6.2.1 讨论红方无人机（集群）变换阵型。

根据问题一、三中对每一波红方无人机（集群）的阵型讨论，可以推出当红方运载机可以发送两波无人机（集群）时，不考虑两波无人机（集群）之间进行协作拦截时，无人机（集群）阵型为近似的圆形时，会有最大的扫描面积。本问题考虑在约束条件四、五下，两波无人机（集群）或者不同运载机（最多两架）发出的无人机（集群）之间可以进行协作拦截时，分析红方无人机（集群）的最大扫描面积时的阵型。

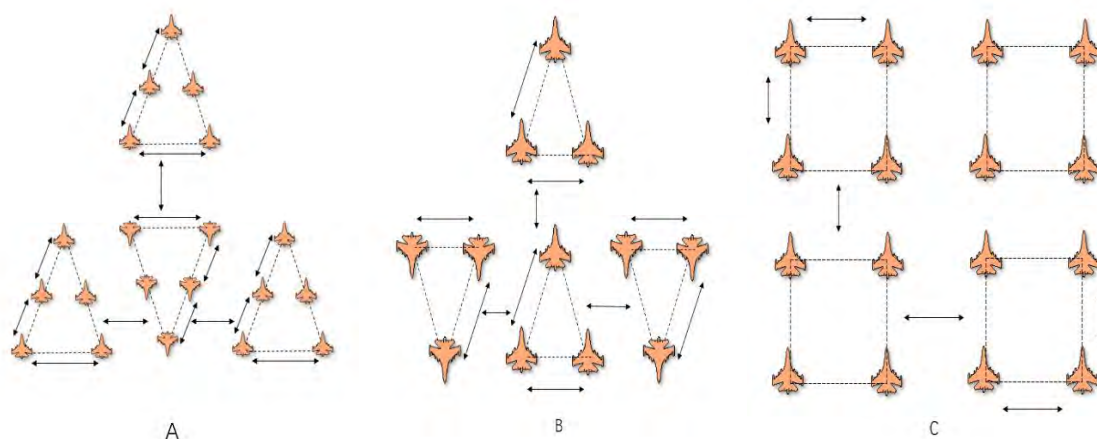


图 6-2 无人机协同条件下的集群形态对比图

考虑在约束条件四、五下，两波无人机（集群）或者不同运载机（最多两架）发出的无人机（集群）之间可以进行协作拦截时，如图 6-2 所示，图 A、B、C 中每一架无人机与各自集群中至少两架无人机的距离等于 200 m，并且各集群之间无人机的距离满足大于 30 m 且小于 200 m。可以很直观的判断出当每波发送 5 架无人机，5 架无人机组成的集群形状是问题一中的三角型方案，并且两架运载机发送的两波无人机（集群）进行协作时，此时的扫描面积最大。同样的，采用快速蒙特·卡罗方法证实了结论的正确性。

6.2.2 基于分层追捕算法分解多对多追捕逃逸问题变为一对二追捕逃逸问题。

由于多个红方运载机的加入，使得红蓝双方追捕逃跑问题的复杂度呈指数增加，直接求解追捕者的最优追捕策略是非常困难的，为了简化问题的复杂度，根据参考文献[3]假设追捕过程中红方的运载机或无人机（集群）长时间的保持追捕蓝方，提出一种适用当前问题的分层追捕算法（如图 6-3）。

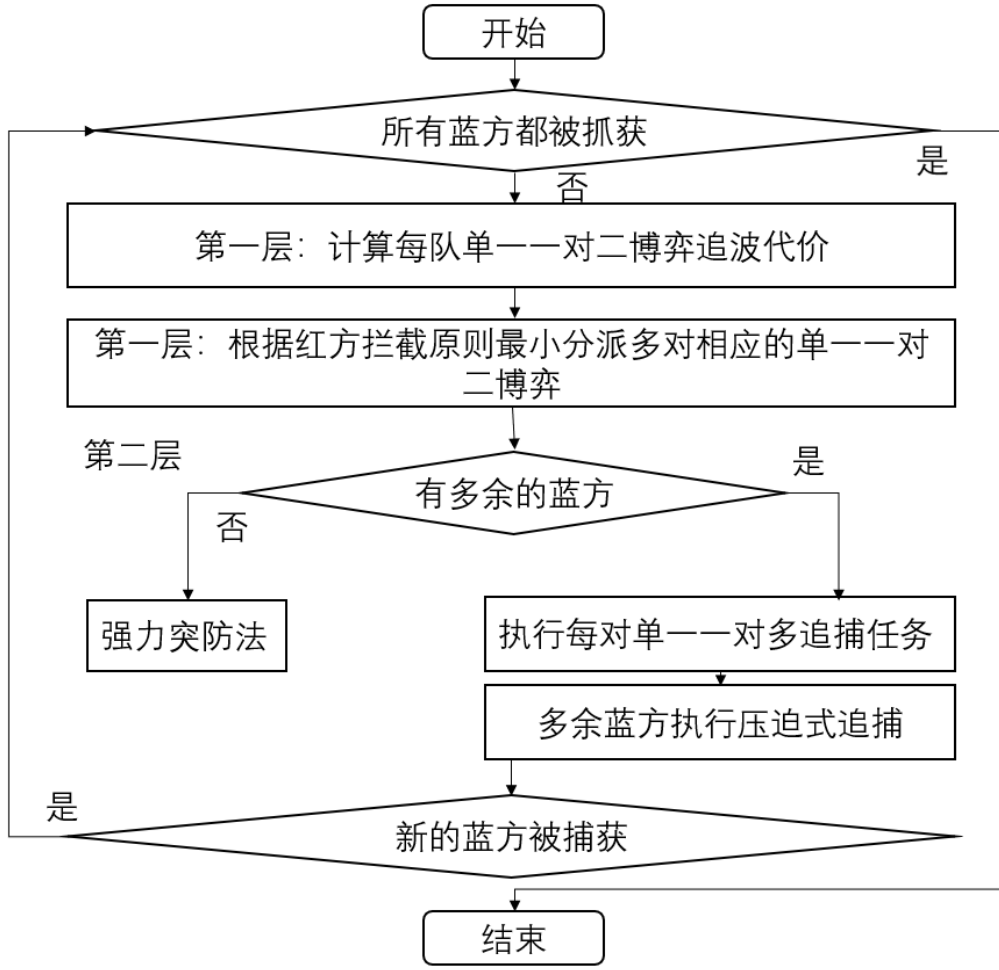


图 6-3 分层算法流程图

本问题提出的分层追捕算法分为三个阶段，包括基于红方拦截原则追捕代价计算阶段、红方运载机分配阶段、执行阶段。

追捕代价计算阶段：

在 5 个红方运载机和 3 个蓝方无人机的多对多追捕逃逸系统中，一共存在 15 对一对二追捕博弈的可能，本阶段就是要计算出每队单一双人追捕博弈的追捕轨迹并计算其追捕代价。

$$x_e = v_e \cos \theta_e, y_e = v_e \sin \theta_e \quad (32)$$

$$x_p = v_p \cos \theta_p, y_p = v_p \sin \theta_p \quad (33)$$

上面公式表示有 2 个追捕者和 1 个逃跑者的追捕动态方程。其中 θ 表示对应无人机的运动方向角度。初始时刻的红方和蓝方的最佳运动方向 θ_p^* 、 θ_e^* 。根据参考论文，计算得到公式，其中

$$\sqrt{V_x^2(x, y) + V_y^2(x, y)} = \frac{1}{v_p - v_e} \quad (34)$$

$$\cos\theta_p^* = \cos\theta_e^* = -\frac{x_0}{\sqrt{V_x^2(x,y) + V_y^2(x,y)}} \quad (35)$$

$$\sin\theta_p^* = \sin\theta_e^* = -\frac{y_0}{\sqrt{V_x^2(x,y) + V_y^2(x,y)}} \quad (36)$$

任务分配阶段:

系统的任务是在合理分配对应任务使预期追捕时间最小。求解该问题可以转化为线性阻塞分派问题，可以基于网络流理论的多项式时间算法来进行解决。分配阶段用追捕代价来设置图 6-4 所示的偶图的连接权重。

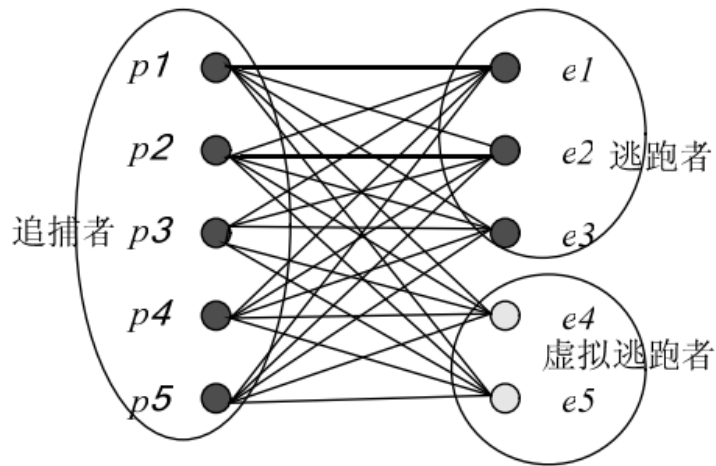


图 6-4 偶图的连接权重

由于问题中蓝方逃跑者为 3 架无人机，红方追捕者运载机为 5 架，可以为每个蓝方分配至少一个红方运载机负责追捕，因此每个蓝方的被拦截时间都可以被估算出来（如图 6-5）。

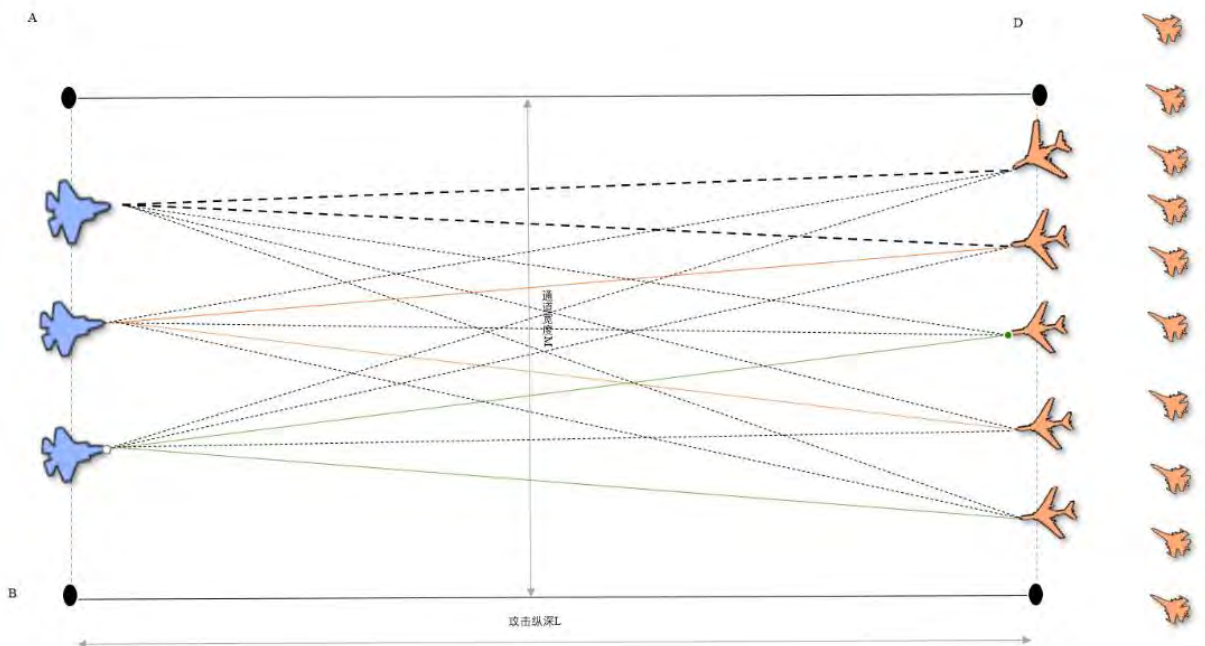


图 6-5 基于问题四红蓝双方分派图

为了使预计的总追捕时间最小，可以采用下面的任务分配算法：

表 3 任务分配算法

任务分配算法
1. 移除偶图中代价最高的边
2. 检查是不是所有的顶点都至少有一条边相连（即图中不包含独立的顶点）
3. if 是，返回 步骤 1
4. if 否，则把连接到这条边上的追捕者和逃避者配对并且从图中移除它们（连同它们所连接的边）
5. 遍历剩下的子图

在不变换策略的情况下，此分配算法可以获得最优的分配结果。

执行阶段：

第二阶段的任务分配过程是在追捕开始之前执行，也就是说忽略一个在多对多追捕博弈中很重要的追捕者之间的协作，即红方无人机（集群）之间会进行协作。关于此问题会在 6.2.4 中进行具体分析。

值得注意的是在执行过程中，最开始红方追捕者的数量大于蓝方逃跑者的数量，此时执行压迫式追捕方法，从而体现红方无人机之间的协作。随着蓝方无人机被捕获，红方追捕者数量上的优势越来越大，蓝方越容易被拦截。

6.2.3 红蓝双方最优策略分析

根据前面三个问题的讨论，可以总结目前已有的红蓝双方的最优策略：

对于蓝方而言：寻找是否存在某个 i 满足 $\frac{r_{i,i+1}}{(d_i+d_{i+1})} > \gamma$ 或者 $\theta_{i,i+1} > \arccos(1 + 2a^2)$ 。

若是，它选择红方无人机（集群） p_i 和 p_{i+1} 采用问题一中近似直线突破模型进行逃逸。若非，则采用问题一中的欺骗性突破模型进行逃逸。

对于红方无人机（集群）而言，被选择的红方无人机（集群）的策略是首先保证蓝方处于自己的捕获区内，然后执行问题一中相应的追捕策略，而其他红方无人机（集群）只需要跟随着蓝方无人机运动即可。

根据题意，红方存在速度上的劣势，因此对于红方无人机（集群）来说想要拦截成功是一件比较困难的事情。在定性微分对策的意义下，多人追逃问题所关注的是追逃双方能够确保赢得对策的条件与相应的策略。上一小节已经将红方 5 架运载体根据分层追捕算法分为一对二或一对多追捕逃逸问题，本小节将会进一步讨论分析一对多追捕逃逸问题。

对于红方而言：

一般来说，要捕获到速度更快的逃跑者，如下三个条件需要得到满足：

一是要有一定多的追捕者；二是这些追捕者的初始位置必须围绕在逃跑者的周围；三是追捕者所形成的包围圈是可以采取一系列措施缩小的。

基于参考论文对一对多追捕逃逸问题数学理论上的分析，可以计算出满足上述包围条件所需追捕者的最小数目为 4,其中 $\lceil \cdot \rceil$ 为取整符号，因此在围捕阶段红方无人机（集群）可以采用 5.2.1 节所分析的阵型进行围捕。

$$n_{min} \geq \left\lceil \frac{2\pi}{(\arccos(1 - 2\gamma^2))} \right\rceil = \left\lceil \frac{\pi}{(\arcsin(\gamma))} \right\rceil = [3.38] = 4$$

因此红方可以采取通过初始阶段释放第一波无人机（集群），在一段时间内，尽量保证 $\theta_{i,i+1} > \arccos(1 + 2a^2)$ 恒成立，同时红方运载机在满足约束条件下，以最优策略运动到对应蓝方无人机的周围，准备开始释放第二波无人机（如图 6-6 中的红方运载机起始区）。当红方运载机释放第二波无人机的同时，第一波无人机开始变换阵型，使得包围时扫描面积最大。如图 6-6 中的围捕区。当红方无人机对目标蓝方无人机实现围捕后，基于参考论文中的定理 5.4，满足过渡条件并实施抓捕。

对于蓝方而言：

根据题目假设，蓝方作为优势逃跑者可以自由地在平面上运动，当蓝方被红方逼迫至包围状态时，它可以在满足突破时间360s内不断地更改其突破口，这样红方无人机（集群）要想拦截蓝方，则比较困难。可以理解为，对于红方无人机（集群）而言，并不能一直保证 $\theta_{i,i+1} > \arccos(1 + 2a^2)$ 是恒成立的。即使包围条件在任何时间被满足，相对距离 d_i 也会出现不变的情况，此时逃跑者只需要保持和追捕者相反的角度运动就可以维持在一种近似“平衡”的状态。

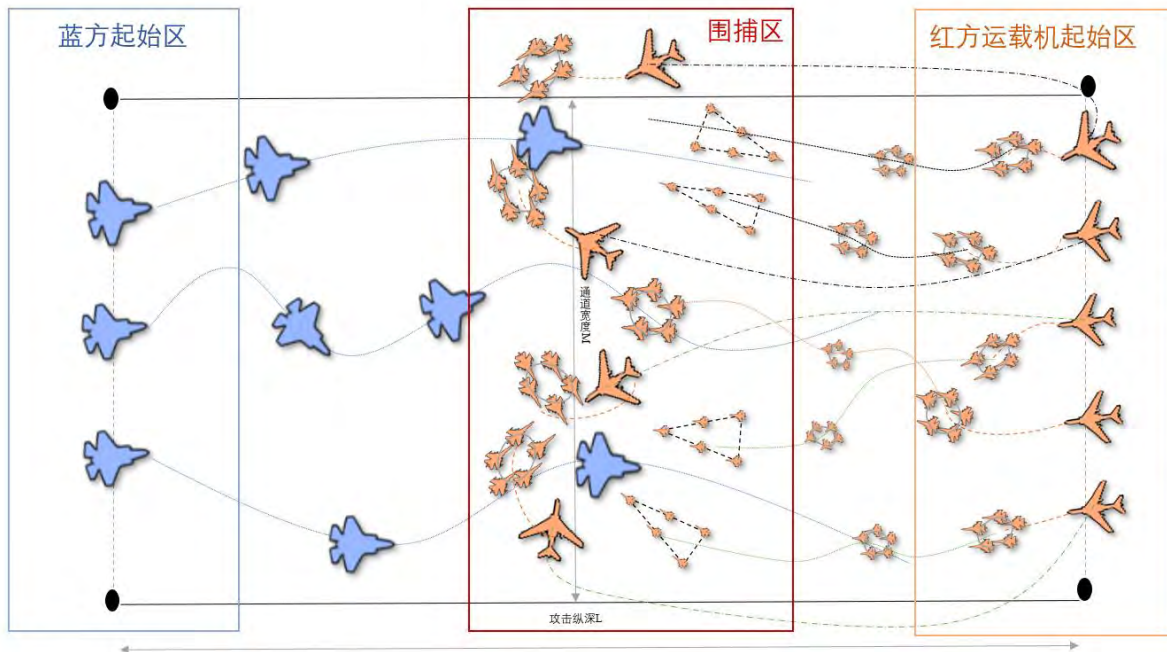


图 6-6 红蓝方采取最优策略理想轨迹图

6.2.4 总结红方最优拦截策略和蓝方最优突防策略

根据前面小节的分析与讨论，当红方运载机数量增加至 5 架，蓝方无人机数量增加至三架时，在满足题目约束条件下，

红方最优拦截策略：

结合参考论文中推导的围困条件和过渡条件，红方最优拦截策略称为“分配-包围-围困-收缩-抓捕”。

- (1) 分配：根据分层追捕算法将红方 5 架运载机合理的分配，将多对多的追捕博弈问题转化为一对多的追捕博弈问题，保证最优化的拦截蓝方。
- (2) 包围：最快的保证红方无人机满足追捕者包围的三个条件。
- (3) 围困：红方无人机通过协同合作改变整个无人机扫描范围大小，分散包围蓝方无人机以满足围困条件。
- (4) 收缩：在围困满足的条件下，基于论文中的过渡条件定理，使用合适的非劣策略来尽可能的收缩包围圈。
- (5) 抓捕：红方无人机径直朝向包围圈的中心运动就可以确保拦截到蓝方无人机。

蓝方最优突防策略：

寻找是否存在某个 i 满足 $\frac{r_{i,i+1}}{(d_i+d_{i+1})} > \gamma$ 或者 $\theta_{i,i+1} > \arccos(1 + 2a^2)$ 。若是，它选择红方无人机（集群） p_i 和 p_{i+1} 采用问题一中近似直线突破模型进行逃逸。若非，尽可能的保证不处于包围状态，当蓝方被红方逼迫至包围状态时，它可以在满足突破时间 360s 内不断地更改其突破口。没有时间限制时，当蓝方无人机距离红方无人机的相对距离 d_i 出现不变时，保持和追捕者相反的角度运动就可以维持在一种近似平衡的状态。

参考文献

- [1] 吴其昌,张洪波,基于生存型微分对策的航天器追逃策略及数值求解[J],控制与信息技术,2019,(04):39-43.
- [2] 刘胜男,多对多卫星拦截的任务规划与轨道优化[D],哈尔滨工业大学,2018.
- [3] 祝海,基于微分对策的航天器轨道追逃最优控制策略[D],国防科技大学,2017.
- [4] 李龙跃,刘付显,史向峰,王菊,导弹追逃博弈微分对策建模与求解[J],系统工程理论与实践,2016,36(08):2161-2168.
- [5] 查文中,单个优势逃跑者的多人定性微分对策研究[D],北京理工大学,2016.
- [6] 孙传鹏,基于博弈论的拦截制导问题研究[D],哈尔滨工业大学,2014.
- [7] Walrand J, Polak E, Chung H. Harbor Attack: A Pursuit-Evasion Game, Forty-Ninth Annual Allerton Conference[C]. Allerton, UIUC, Illions, USA, 2011, 1584-1591.
- [8] 方宝富,潘启树,洪炳镭,丁磊,蔡则苏,多追捕者-单-逃跑者追逃问题实现成功捕获的约束条件[J],机器人,2012,34(03):282-291.
- [9] Isaacs R. Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization[M]. New York: John Wiley and Sons, 1965.

附件

```
%%solve problem1

for i=1:1000

    for j=1:1400

        [x1,y1,r1]=solve_apolo(i,j,xp,yp,0.8);

        [x2,y2,r2]=solve_apolo(i,j,xp,yp-600,0.8);

        [y11,y12]=solve_pointy(1000,x1,y1,r1);%圆在 x=1000 时的两个交点

        [y21,y22]=solve_pointy(1000,x2,y2,r2);

        r1r2=r1+r2;

        o1o2=sqrt((x2-x1)^2+(y2-y1)^2);

        [x111,x112]=solve_pointx(0,x1,y1,r1);%圆 1 在 y=0 时的两个交点

        [x121,x122]=solve_pointx(1400,x1,y1,r1);%圆 1 在 y=1400 时的两个交点

        [x211,x212]=solve_pointx(0,x2,y2,r2);%圆 2 在 y=0 时的两个交点

        [x221,x222]=solve_pointx(1400,x2,y2,r2);%圆 2 在 x=1400 时的两个交点

        flag=0;

        if(r1r2<o1o2)%相离

            %先判断与两边是否都存在交点

            if( ((y1<0)&(y1+r1>1400))|((y1<1400)&(y1>0)&(y1+r1>1400)&(y1-r1<0))|((y1>1400)&(y1-r1<0)))

                if(((x112<=1000)&(x122<=1000)))

                    flag=1;

                end

            end

        end

    end

end
```



```

        if(((y2<0)&(y2+r2>1400))|((y2<1400)&(y2>0)&(y2+r2>1400)&(y2-
r2<0))|((y2>1400)&(y2-r2<0)))

            if(((x212<=1000)&(x222<=1000)))%最左边的交点小于 50km 一定不能逃脱

                flag=1;

            end

        end

        if(flag==1)

            grid(j,i)=255;

        end

    end

    flag=0;

    if(o1o2==r1r2)%相切

        q=r1/r2;%计算切点

        xq=(x1+q*x2)/(1+q);

        yq=(y1+q*y2)/(1+q);

        if( ((y1<0)&(y1+r1>1400))|((y1<1400)&(y1>0)&(y1+r1>1400)&(y1-
r1<0))|((y1>1400)&(y1-r1<0)))

            if(((x112<=1000)&(x122<=1000)))

                flag=1;

            end

        end

        if(((y2<0)&&(y2+r2>1400))|((y2<1400)&(y2>0)&(y2+r2>1400)&(y2-
r2<0))|((y2>1400)&(y2-r2<0)))

            if(((x212<=1000)&&(x222<=1000)))%最左边的交点小于 50km 一定不能逃脱

                flag=1;

            end

        end
    end

```

```

        end

    end

    if(flag==1)

        grid(j,i)=255;

    end

    if(y1>y2)

        if((y1+r1>=1400)&(y2-r2<=0)&(xq<1000))%两边相交

            if((x122<=1000)&(x212<=1000))

                grid(j,i)=255;

            end

        end

    else

        if((y1-r1<=0)&(y2+r2>=1400)&(xq<1000))%两边相交

            if((x222<=1000)&&(x112<=1000))

                grid(j,i)=255;

            end

        end

    end

end

if(r1r2>o1o2)%相交

    if(y1>y2)

        if((y1+r1>=1400)&(y2-r2<=0)&(y12<y21))%两边相交

```

```

        if((x122<=1000)&(x212<=1000))

            grid(j,i)=255;

        end

    end

elseif(y1<=y2)

    if((y1-r1<=0)&(y2+r2>=1400)&(y22<y11))%两边相交

        if((x222<=1000)&(x112<=1000))

            grid(j,i)=255;

        end

    end

end

end

end

end %for j

end %for i

imshow(grid,[0,255]);

hold on;

axis on

axis equal ;

set(gca,'xtick',0:100:1000)

set(gca,'ytick',0:100:1400)

set(gca,'YDir','reverse')%对 Y 方向反转

hold on

%%problem3

```

```

load grid;

for i=1:1000

    for j=1:1400

        for yp=994:1006

            for xp=994:1000

xe=0;ye=35000;

gx1=50000;gy1=30000;

gx2=50000;gy2=40000;

a=pi/4;%发射角度

px1=gx1-2000*sin(a);

py1=gy1+2000*cos(a);

px3=gx2-2000*sin(a);

py3=gy2-2000*cos(a);


[x1,y1,r1]=solve_apolo(xe,ye,gx1,gy1,300/250);

[x2,y2,r2]=solve_apolo(xe,ye,gx2,gy2,300/250);%G1 G2 初始阿波罗圆

xo1=x1-sqrt(r1^2-(35000-y1)^2)%jiaodian

angle=atan2(5000,50000-xo1)

a=tan(angle)

b=atan(0.0667)

cos(angle)

sin(angle)

syms t;

```

```

gx11=gx1-300*cos(angle)*t;

gy11=gy1+300*sin(angle)*t;

xe1=xe+250*t;

yel=yel;

t=solve((gx11-xe1)^2+(gy11-yel)^2==2470^2);

[px1,py1,pr1]=solve_apolo(xe,ye,px1,py1,0.8);

[px3,py3,pr3]=solve_apolo(xe,ye,px3,py3,0.8);%G1 G2 初始阿波罗圆

xo2=px1-sqrt(pr1^2-(35000-py1)^2)%jiaodian

[y11,y12]=solve_pointy(1000,x1,y1,r1);%圆 1 在 x=1000 时的两个交点

[y21,y22]=solve_pointy(1000,x2,y2,r2);

r1r2=r1+r2;

o1o2=sqrt((x2-x1)^2+(y2-y1)^2);

[x111,x112]=solve_pointx(0,x1,y1,r1);%圆 1 在 y=0 时的两个交点

[x121,x122]=solve_pointx(1400,x1,y1,r1);%圆 1 在 y=1400 时的两个交点

[x211,x212]=solve_pointx(0,x2,y2,r2);%圆 2 在 y=0 时的两个交点

[x221,x222]=solve_pointx(1400,x2,y2,r2);%圆 2 在 x=1400 时的两个交点

flag=0;

if(r1r2<o1o2)%相离

    先判断与两边是否都存在交点

    if( ((y1<0)&(y1+r1>1400))|((y1<1400)&(y1>0)&(y1+r1>1400)&(y1-
r1<0))|((y1>1400)&(y1-r1<0)))

        if(((x112<=1000)&(x122<=1000)))

```

```

        flag=1;

    end

end

    if(((y2<0)&(y2+r2>1400))|((y2<1400)&(y2>0)&(y2+r2>1400)&(y2-
r2<0))|((y2>1400)&(y2-r2<0)))

        if(((x212<=1000)&(x222<=1000)))%最左边的交点小于 50km 一定不能逃脱

            flag=1;

        end

    end

    if(flag==1)

        grid(i,j)=255;

    end

end

flag=0;

if(o1o2==r1r2)%相切

    q=r1/r2;%计算切点

    xq=(x1+q*x2)/(1+q);

    yq=(y1+q*y2)/(1+q);

    if( ((y1<0)&(y1+r1>1400))|((y1<1400)&(y1>0)&(y1+r1>1400)&(y1-
r1<0))|((y1>1400)&(y1-r1<0)))

        if(((x112<=1000)&(x122<=1000)))

            flag=1;

        end

    end

end

```

```

        if(((y2<0)&&(y2+r2>1400))|((y2<1400)&(y2>0)&(y2+r2>1400)&(y2-
        r2<0))|((y2>1400)&(y2-r2<0)))

            if(((x212<=1000)&&(x222<=1000)))%最左边的交点小于 50km 一定不能逃
            脱

                flag=1;

            end

        end

    end

    if(flag==1)

        grid(i,j)=255;

    end

    if(y1>y2)

        if((y1+r1>=1400)&(y2-r2<=0)&(xq<1000))%两边相交

            if((x122<=1000)&(x212<=1000))

                grid(i,j)=255;

            end

        end

    end

    else

        if((y1-r1<=0)&(y2+r2>=1400)&(xq<1000))%两边相交

            if((x222<=1000)&&(x112<=1000))

                grid(i,j)=255;

            end

        end

    end

end

```

```

end

if(r1r2>o1o2)%相交

    if(y1>y2)

        if((y1+r1>=1400)&(y2-r2<=0)&(y12<y21))%两边相交

            if((x122<=1000)&(x212<=1000))

                grid(i,j)=255;

            end

        end

    end

    elseif(y1<=y2)

        if((y1-r1<=0)&(y2+r2>=1400)&(y22<y11))%两边相交

            if((x222<=1000)&(x112<=1000))

                grid(i,j)=255;

            end

        end

    end

end

end

end

end

end for j

end for i

imshow(grid,[0,255]);

hold on;

```