

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 东南大学

---

参赛队号 20102860223

---

1.刘德平

---

队员姓名 2.王国标

---

3.郑晓义

---

# 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

无人机集群协同对抗

#### 摘 要：

本文针对非线性约束条件下红蓝双方无人机集群协同对抗的最优规划问题，结合贪婪队形、非线性规划、内点法、蒙特卡洛方法和全联立正交配置有限元法，构建了无人机集群协同对抗推演模型。

针对问题一，构建了基于蒙特卡洛法的蓝方突防概率推演模型。我们综合考虑了基于贪婪队形的红方协作拦截策略，和基于内点法的蓝方突防求解模型。首先分析红方在集群约束下的贪婪队形和拦截策略，然后我们基于内点法联合 MATLAB 和 AMPL 编程求解蓝方突防的非线性规划解，最后基于蒙特卡洛法对整个蓝方突防过程进行推演，得到突防概率图。通过推演，我们提出了区域 P 和 Q 两块突防成功率最高的区域作为所求区域，并对蓝方的最优突防策略进行分析。

针对问题二，构建了基于正则化目标函数的蓝方突防策略推演模型。我们考虑通过改进第一问的模型进行求解，首先针对所求时间最短突防策略，基于正则化思想改进优化的目标函数。然后在固定的通道宽度下对新的约束和目标函数进行攻防策略推演。针对所求带宽下限，对每一轮推演使用的带宽变化量进行渐进松弛，不断逼近所求下限。获得的带宽下限  $M_{\min}$  为 91km，并对不同条件下的蓝方最短时间突防策略进行分析。

针对问题三，构建了基于全联立正交配置有限元法的非线性求解优化模型。综合考虑多重非线性约束条件和红方多波次释放无人机的复杂环境问题，我们通过全联立正交配置有限元法将问题中的所有状态量离散化，提高非线性优化模型的推演效率。然后在固定的通道宽度下对红蓝双方的攻防过程进行推演。针对所求带宽上限，对每一轮推演使用的带宽变化量进行渐进紧缩，不断逼近所求上限。获得的带宽上限  $M_{\max}$  为 39km，并对不同条件下的红方最优拦截策略进行分析。

针对问题四，构建了基于 NLP 的红蓝双方攻防策略推演模型。综合考虑前述的多重非线性约束和红蓝双方无人机集群博弈的复杂环境，我们通过前述建立的红蓝双方攻防策略推演模型，对红蓝无人机集群的协同突防和协同拦截状态进行推演，最后对不同条件下的红方最优拦截策略和蓝方最优突防策略进行分析。

整体来说，红蓝双方的最优攻防策略均以红蓝双方无人机集群的距离为评价标准，针对蓝方突防，尽量选择远离红方无人机的方向突防；针对红方拦截，尽量选择朝向蓝方无人机的移动方向拦截、

**关键词：**非线性约束 最优规划 集群协同 蒙特卡洛方法 全联立正交配置有限元法

## 目 录

摘 要:	1
1. 问题重述	4
2. 问题假设及符号说明	6
2.1 问题假设	6
2.2 符号说明	6
3. 问题一模型的建立与求解	7
3.1 针对问题一的分析	7
3.2 攻防区域的数值化	8
3.3 基于贪婪队形的红方协作拦截策略	9
3.3.1 基于队形的红方无人机集群有效拦截面扩展	9
3.3.2 基于贪婪队形的集群协作拦截	10
3.4 基于内点法的蓝方突防求解模型	11
3.4.1 基于非线性规划 NLP 模型	11
3.4.2 基于最小化性能指标的最优控制指标转化	12
3.4.3 基于内点法的非线性最优规划模型	12
3.5 基于蒙特卡洛法的蓝方突防概率模型	13
3.5.1 基于蒙特卡洛法的概率推演	13
3.5.2 蓝方突防概率建模	14
3.6 蓝方突防概率结果与最优突防策略分析	14
3.6.1 针对蓝方突防概率结果	14
3.6.2 针对蓝方最优突防策略	16
4. 问题二模型的建立与求解	18
4.1 针对问题二的分析	18
4.2 基于 L2 正则化思想的目标函数优化	19
4.3 基于正则化目标函数的蓝方突防策略推演模型	20
4.4 最小带宽与蓝方突防时间最短策略分析	21
4.4.1 针对最小带宽	21
4.4.2 蓝方突防时间最短策略	22
5. 问题三模型的建立与求解	25
5.1 针对问题三的分析	25
5.2 基于全联立正交配置有限元法的非线性求解优化模型	25
5.3 最大带宽与红方最优拦截策略分析	26
5.3.1 针对最大带宽	26

5.3.1 针对红方最优拦截策略 .....	26
6. 问题四模型的建立与求解.....	31
6.1 针对问题四的分析 .....	31
6.2 基于 NLP 的红蓝双方攻防策略推演模型.....	31
6.3 红方最优拦截策略与蓝方最优突防策略分析 .....	32
7. 模型的评价.....	36
7.1 模型的优点 .....	36
7.2 模型的缺点 .....	36
8. 参考文献.....	37
附  录 .....	38

## 1. 问题重述

人工智能技术的不断发展催生了无人战车、无人机、无人艇等新型作战力量，同时也推动着作战方式从单体对抗的智能化到群体协同对抗的智能化方向发展。在复杂的战场环境下，通过多架次无人机集群的协同探测、协同进攻、协同拦截和协同躲避，能有效提高任务的完成效果。

本题即研究在一定战场环境约束和飞行状态约束条件下，攻守无人机集群的最优协同突防和协同拦截策略问题。

攻防过程中，红蓝双方无人机需要满足的约束如表 1 所示。

表 1 红蓝双方集群攻防约束表

对象		参数	约束
对抗区域		攻击纵深 $BC(AD)$	$L = 50km$
		通道宽度 $AB(DC)$	$M$ 为限定约束
红无人机集群	运载机	速度 $V_H$	$V_H = 300m/s$ ，方向依转弯半径改变
		最小转弯半径 $R_H$	$R_H = 1000m$
	无人机	速度 $V_P$	$V_P = 200m/s$ ，方向依转弯半径改变
		最小转弯半径 $R_P$	$R_P = 350m$
蓝无人机集群		速度 $V_E$	$V_E = 250m/s$ ，方向依转弯半径改变
		最小转弯半径 $R_E$	$R_E = 500m$
飞行边界		红无人机集群	无边界限制
		蓝无人机集群	不能越过 $AD$ 和 $BC$
编队	红无人机		(1) 每个集群 5 架无人机初始位置近似分布在一个圆周上，任意相邻 2 架无人机的间距相同；
			(2) 任意 2 架无人机的间距大于 $30m$ ；
			(3) 每一架无人机与本集群中至少 2 架无人机的距离不超过 $200m$ 。
	红运载机		(1) 与所属无人机集群中至少一架无人机的距离不超过 $10km$ ；
			(2) 与任何一架无人机的距离需大于 $100m$ ；
			(3) 与蓝方无人机距离大于 $5km$ ；
攻防结果判定		蓝方突防成功	$360s$ 内越过边界，且不被拦截
		红方拦截成功	蓝方突防无人机与红方至少 2 架无人机的距离均小于 $300m$
		蓝无人机	任意两架无人机的间距大于 $30m$

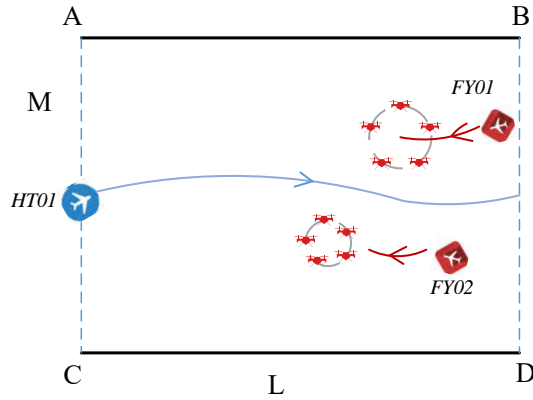


图 1 红蓝双方无人机集群协同突防示意图

如图 1 所示，红蓝双方在平面矩形区域  $ABCD$  内进行无人机集群的协同对抗，蓝色方为进攻方，进攻出发位置根据需求变化，进攻目标位置为边界  $CD$ ；红色方为防守方，从边界  $CD$  出发对蓝方进行拦截。

在满足表 1 所示约束的情况下，针对以下问题对红蓝双方在攻防区域内的攻防策略进行建模分析：

**问题 1.** 对抗伊始，在约束条件下，针对设定：

(A) 红方 2 个无人机集群分别以边界  $CD$  上的  $G_1$  和  $G_2$  为中心的圆周队形进入攻防区域，圆周半径为  $100m$ ， $DG_1 = 20km$ ， $G_1G_2 = 30km$ ， $CG_2 = 20km$ ，即通道带宽  $M = 70km$ ；

(B) 蓝方无人机可从矩形区域  $ABCD$  内任意位置突防。

分析蓝方总能采用合适的策略躲避拦截，成功突防的进入位置，并讨论相应的最优突防策略。

**问题 2.** 对抗伊始，在约束条件下，针对设定：

(A) 蓝方无人机边界  $AB$  的中心点进入攻防区域；

(B) 红方 2 个无人机集群分别以  $G_1$  和  $G_2$  为中心的圆周队形进入攻防区域，圆周半径为  $100m$ ， $G_1$  和  $G_2$  位于边界  $CD$  上，具体位置未知。

分析通道带宽  $M$  的下限  $M_{min}$  的存在性，满足当  $M$  比  $M_{min}$  大时，蓝方无人机一定能突防成功，并给出该情形下蓝方无人机时间最短的突防策略。

**问题 3.** 在约束条件下，通道带宽  $M = 70km$ ，针对设定：

(A) 红方每架运载机分两波次共发射 10 架无人机，发射的无人机集群初始队形满足表 1 中的编队约束，运载机与圆周中心的距离为  $2km$ ，集群队形可调整；

对抗伊始，

(B) 蓝方无人机从边界  $AB$  中心进入；

(C) 红方两架运载机分别从边界  $CD$  上  $G_1$  和  $G_2$  进入，同时发射第一波次的无人机集群，运载机和无人机集群中心具体位置未知；

(D) 运载机在保证与第一波次集群满足约束条件下，发射第二波次无人机集群。

讨论红方的最优拦截策略，包括：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。

分析通道带宽  $M$  的上限  $M_{max}$  的存在性，满足当  $M$  小于  $M_{max}$  时，红方无人机一定能拦截成功。

**问题 4.** 对抗伊始，在约束条件下，针对设定：

(A) 蓝方突防集群的 3 架无人机从边界  $AB$  一侧开始突防，通道带宽  $M = 100km$ ；

(B) 红方 5 架运载机在约束下从边界  $CD$  一侧开始拦截，同时发射第一波次的无人机集群，运载机和无人机集群中心的距离为  $2km$ ，分两波次发射无人机集群；

(C) 蓝方希望尽可能多的无人机突防成功；

(D) 红方希望尽可能多的拦截蓝方无人机。

讨论红方最优拦截策略和蓝方最优突防策略，包括：红方运载机初始位置、红方运载机发射的第一个波次的无人机集群中心位置、红方运载机发射第二波次无人机集群的时刻和位置、第二波次发射的无人机集群中心位置、两个波次无人机数量以及蓝方突防无人机初始位置。

## 2. 问题假设及符号说明

### 2.1 问题假设

1. 假设红、蓝双方的运动只考虑对抗平面上的运动，不考虑俯仰、滚转运动，飞行速度大小不变，方向可根据策略需要改变；

2. 假设运载机在满足约束条件下瞬间布设好无人集群为设定队形，攻防时在约束下转变队形；

3. 假设红、蓝双方无人机的体积和质量分布不影响运动和对抗结果判定，即无人机可视为质点进行分析；

4. 假设红、蓝双方无人机集群在初始进入区域时即探测到对方位置；

5. 假设红、蓝双方攻防结果只有蓝方突防成功和红方拦截成功两种判定情况；

6. 假设红方无人机拦截成功后，被拦截的蓝方无人机退出攻防，实施拦截动作的红方无人机位置不变继续参加攻防。

### 2.2 符号说明

符号	含义
$P_i(E_i)$	红(蓝)方第 $i$ 个无人机
$H_i$	红方第 $i$ 个运载机
$V_{Pi}(V_{Ei})$	红(蓝)方第 $i$ 个无人机速度
$V_{Hi}$	红方第 $i$ 个运载机速度
$R_{Pi}(R_{Ei})$	红(蓝)方第 $i$ 个无人机转弯半径
$R_{Hi}$	红方第 $i$ 个运载机转弯半径
$t_{Ei}$	蓝方第 $i$ 个无人机的突防时间
$l_{Ei}$	蓝方第 $i$ 个无人机的突防路径长度
$S_{Pk}$	红方第 $k$ 个集群有效拦截面积
$P_k^*$	红方第 $k$ 个集群队形的尖端点无人机
$D$	红方集群有效拦截面的最大宽度
$M$	通道宽度(突防带宽)
$L$	攻击纵深
$N$	蒙特卡洛试验次数

$\alpha(E)$	蓝方无人机突防成功率
$\ \cdot\ $	算式 $\cdot$ 的范数

### 3. 问题一模型的建立与求解

#### 3.1 针对问题一的分析

在无人机协同对抗的环境中，无人机单体可以相互通信、控制，避免碰撞的同时还可以构成集群，进行一定的编队组合，从而最大化躲避、拦截等决策的收益。满足表 1 中编队约束的队形示意图如图 2 所示：

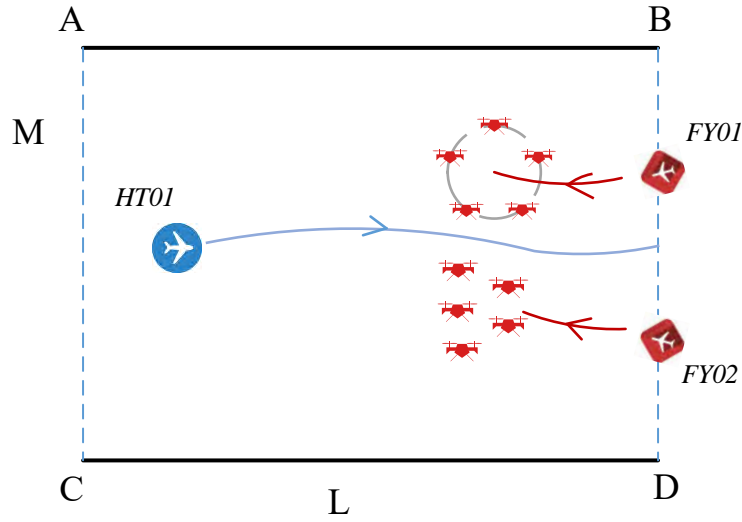


图 2 攻防过程中的队形示意图

问题一固定了整个对抗平面的攻击纵深  $L$ ，通道宽度  $M$ ，红方运载机释放无人机的位置点  $G_1$  和  $G_2$ ，要求我们设计模型讨论满足约束时蓝方无人机从对抗平面内的哪些位置进入时，总能成功突防，并且讨论蓝方无人机的最优突防策略。

在这种情景下，蓝方总能找到成功突防的策略进行突防，可以转化为计算在不同位置进入时蓝方突防的成功率  $\alpha(E)$ 。为了找到蓝方从攻防区域的何处进入才能使蓝方突防成功率  $\alpha(E)$  最高，针对多约束条件下的动态博弈策略问题，我们参考“有限元”的思想，构建基于蒙特卡洛的蓝方突防概率模型，解决非线性约束的动态规划问题，利用模拟出的不同位置下蓝方突防成功率对最优突防位置点进行分析。

针对每一个位置点进入后的红蓝攻防动态决策过程，参考以下约束进行非线性动态规划：



$$\begin{aligned}
s.t. \quad & V_P = 200m/s, V_H = 300m/s, V_E = 250m/s, \\
& R_P \geq 300m, R_H \geq 1000m, R_E \geq 500m, \\
& M = 70000m, t_E < 360s, \\
& i, j = 1, 2, 3, 4, 5, j \neq i, m = 2, 3, 4, I = 1, 2, \dots, ki, \\
& \|P_{km} - P_{k(m-1)}\| = \|P_{km} - P_{k(m+1)}\|, \|P_I - P_J\| \geq 30m, \\
& \|H_k - P_I\| > 100m, \|H_k - E_1\| > 5000m, \\
& \text{if } \|H_k - P_{ki}\| > 10000m, \\
& \text{then error,} \\
& \text{if } \|P_{ki} - P_{kj}\| > 200m \text{ or } \|P_{ki} - P_{kj_1}\| > 200m, \\
& \text{then error}
\end{aligned} \tag{1}$$

其中， $ki$  为红方第  $k$  个集群内的第  $i$  个无人机， $kj_1$  为红方第  $k$  个集群内除第  $i$  个无人机外的某一个标号为  $j_1$  的无人机

蓝方的优化目标函数为：

$$\max d_{E,P} - l_E = \|E_1 - P_k^*\| - l_E, k = 1, 2 \tag{2}$$

红方的优化目标函数为：

$$\min d_{E,P} = \|P_k^* - E_1\|, k = 1, 2 \tag{3}$$

根据这些目标和约束，我们可以对动态过程进行模拟，得到使得蓝方突防成功率最高的区域。

### 3.2 攻防区域的数值化

蓝方在某一确定位置进入攻防区域后，可以视为一种分阶段递进的动态决策过程。首先红方基于探测到的蓝方位置进行拦截决策，在飞向蓝方进行拦截的过程中通过变更队形以达到最大化拦截面积的效果，这可以提高动态对抗中红方拦截的成功率。而蓝方可以通过对红方这一决策的探测进行有针对性的动态避障。我们可以利用数值方法对红蓝双方的动态对抗过程进行规划模拟。建立二维直角坐标系对整个区域进行数值化分析的步骤如下：

Step1. 选取坐标原点，建立二维平面直角坐标系。

Step2. 依据空间尺度关系和运动约束构建目标物的动态模型。

Step3. 对目标物的动态模型进行递进规划，完成数值模拟。

根据上述过程，我们可以对整个攻防区域  $ABCD$  进行数值化。

如图 3 所示，以点  $B$  为原点，以边界  $BC$  为  $x$  轴，以边界  $BA$  为  $y$  轴建立平面直角坐标系，可以假设红方运载机 FY01 及其发射集群内的无人机以  $45^\circ$  的夹角进入对抗区域，运载机 FY02 及其发射集群内无人机的进入方向与 FY01 集群的方向关于边界  $CD$  的中垂线对称。

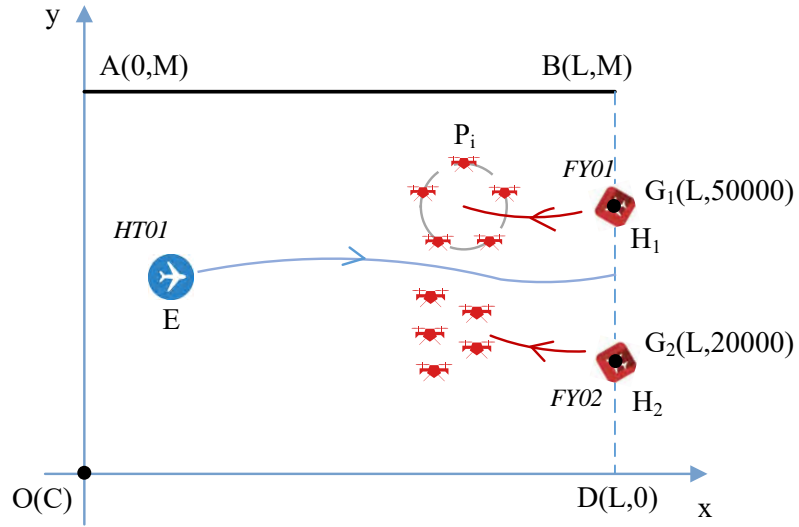


图 3 攻防区域  $ABCD$  的数值化

红方无人机集群的运动状态模型可以用  $P$  表示，红方运载机的运动状态模型可以用  $H$  表示，蓝方无人机的运动状态模型可以用  $E$  表示：

$$P = (V_p, R_p, t_p)^T \quad (4)$$

$$H = (V_H, R_H, t_H)^T \quad (5)$$

$$E = (V_E, R_E, t_E)^T \quad (6)$$

针对各集群的状态量可以进行递进的数值模拟。

### 3.3 基于贪婪队形的红方协作拦截策略

#### 3.3.1 基于队形的红方无人机集群有效拦截面扩展

初始时刻，红方运载机在释放出无人机集群后，无人机集群首先形成一个半径为  $100m$  的圆形队形，此时红方集群需要实时探测红蓝方之间的距离  $d_{E,P}$ ，进行拦截决策。此时红蓝双方的位置示意图如图 4 所示：

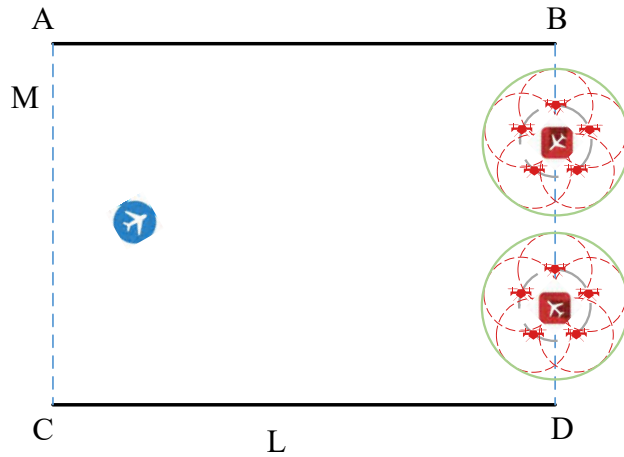


图 4 红蓝双方位置示意图

由于红方集群形成了某种拓扑结构，不容易对距离  $d_{E,P}$  进行表示，但双方的相对位置已知，并且对每一个红方无人机来说，其周围半径  $300m$  的圆形区域是蓝方无人机的危险区域，所以我们在红方队形拓扑结构的基础上，以每架红无人机为圆心，以  $300m$  为半径，可以得到约束下的红方无人机集群对蓝方无人机的有效拦截面，以初始时刻的圆周队形为例，如图 5 中所示，这样可以有效表示出红方的拦截效果。

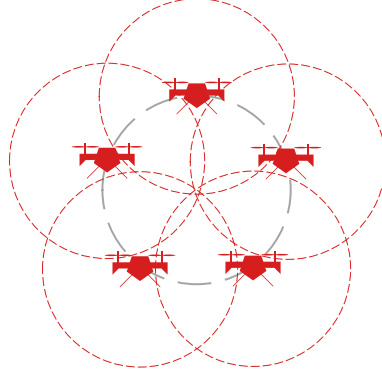


图 5 红方集群的有效拦截面

在红方有效拦截面基础上进行扩展，使用合适的外接几何结构可以得到方便描述集群外轮廓的尖端点  $P_k^*$ ，在所有的尖端点中，我们选取与蓝方无人机方向更接近的尖端点，代入距离  $d_{E,P}$  作为的近似。这样的选取方法既能保证红方拦截方向的局部最优，又能简化动态规划过程。

此时可以定义距离  $d_{E,P}$ ：

$$d_{E,P} = \|E_1 - P_k^*\|_2, k = 1, 2 \quad (7)$$

### 3.3.2 基于贪婪队形的集群协作拦截

按上小节方法选取合适的红蓝方距离  $d_{E,P}$  后，红方集群将对优化函数(3)进行优化以靠近蓝方无人机，实施拦截。同时，在靠近红方无人机还会不断变更队形来提高拦截概率，主要体现在通过变换队形保持较大的有效拦截面积  $S_{pk}$ 。为了达到最大的拦截概率，我们基于的等边三角形、圆形和正五边形的内外接关系设计了下面几种“贪婪”队形，如图 6 所示：

在图 6 的队形中， $P_k^*$  为其中的一个尖端点。红方无人机均以“贪婪条件”，即最大约束条件编组队形，相邻两架无人机的距离  $d_0$  都为  $200m$ ，这样可以在约束条件下尽可能的扩大有效拦截面。以每一种队形都在运动约束下以最宽的拦截距离向蓝方靠近。队形(A)为圆形，有效拦截面扩展后的最大宽度  $D_A$  为绿色扩展圆的直径，为：

$$D_A = \frac{\sqrt{50+10\sqrt{5}}}{5} \cdot d_0 \approx 875.28m \quad (8)$$

同理可以得到，队形(B)的有效拦截面扩展后的最大宽度  $D_B$  为：

$$D_B = \left(\frac{\sqrt{3}}{2} \cdot d_0 + 300\right) \cdot 2 \approx 946.41m \quad (9)$$

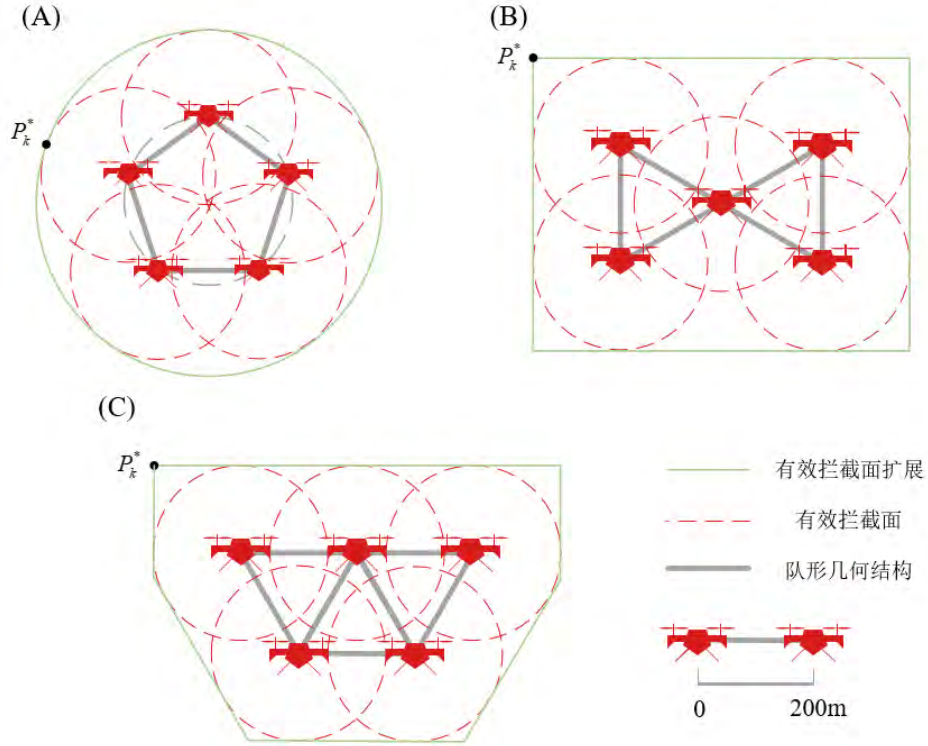


图 6 红方集群的贪婪队形示意图

队形(C)的有效拦截面扩展后的最大宽度  $D_c$  为:

$$D_c = 1000m \quad (10)$$

红色无人机群在进行拦截的过程中,保持以“贪婪”队形的最大拦截宽度对蓝方实施拦截,形成基于“贪婪”队形的红方协作拦截策略。

### 3.4 基于内点法的蓝方突防求解模型

针对题目中蓝方无人机突防的约束条件具有较强非线性,甚至体现非凸性及病态性,无人机集群协同合作运动以及不同集群间的对抗规划任务中,碰撞躲避约束条件与拦截条件的非线性耦合进一步增加了求解难度。

AMPL (A Mathematical Programming Language) 是一种描述并求解大规模复杂数学问题的建模语言,支持世界上大部分的求解器,主要的特点是对优化问题的数学表达式的简化,使得简明地可读地定义优化问题成为可能。在对蓝方突防过程进行非线性的动态规划时,我们基于 AMPL 的自动微分功能,实现求解最优突防策略的精确解。

#### 3.4.1 基于非线性规划 NLP 模型

蓝方无人机进入攻防区域后,在约束(1)中的运动约束、边界约束和对抗约束下,不断接近目标边界,优化函数为公式(2)。对于每一个考虑的场景设置相应的性能指标函数  $J$ ,完整的无人机集群协同对抗运动规划任务可描述为一下最优控制问题<sup>[1]</sup>:

优化目标:

$$\min J, \quad (11)$$

$$J = \Phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), t) dt$$

约束条件为(1)。上式中给出的复合型最小化性能指标  $J$  由末值型性能指标  $\Phi(x(t_0), t_0, x(t_f), t_f)$  和积分型性能指标  $\int_{t_0}^{t_f} L(x(t), t) dt$  这两部分构成。其中，末值型性能指标用于描述任务对终止时刻状态的优化需求，积分型性能指标用于表征任务对整个时域上某项指标累积程度的优化需求。这里需要蓝方无人机的突防尽早完成，所以设置  $\Phi = x_i(t_f) - x_i(t_0)$

### 3.4.2 基于最小化性能指标的最优控制指标转化

复合型性能指标式  $J$  中包含的积分项不利于最优控制问题的统一求解。为此，我们可以通过引入新状态变量的办法将积分型性能指标转化为末值型性能指标<sup>[2]</sup>。对于一般的复合型性能指标泛函  $J = \Phi(x(t_0), t_0, x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), t) dt$ ，可以引入一个过渡状态变量  $trans(t)$ ，要求其满足：

$$\frac{dtrans(t)}{dt} = L(t), t \in [t_0, t_f] \quad (12)$$

$$trans(t_0) = 0 \quad (13)$$

因此， $trans(t) = \int_{t_0}^t L(\tau) d\tau$ ，因此复合型性能指标式  $J$  转换为纯粹的末值型性能指标式  $J = \Phi(x(t_0), t_0, x(t_f), t_f) + trans(t_f)$ ，在转化的过程中，衍生出的微分方程(12)补入系统动态方程的约束条件(1)中，作为系统动态特性的一部分；而代数关系式(13)补入边值约束中。

### 3.4.3 基于内点法的非线性最优规划模型

通过上述建立的方便进行数值求解的蓝方突防策略最优控制问题，我们通过 MATLAB 和 AMPL 结合联合编程，基于内点法(Interior Point Method)，可以对该最优控制问题进行最优规划求解，即 AMPL – Interior Point Optimum Programming Technology, AMPL-IPOPT 方法。该方法的求解步骤框图如图 7 所示：

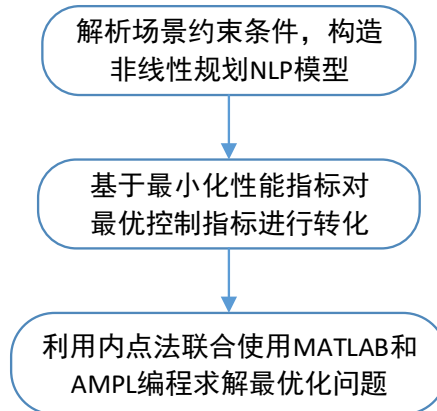


图 7 AMPL-IPOPT 求解方法

### 3.5 基于蒙特卡洛法的蓝方突防概率模型

基于前两小节建立的红蓝双方攻防模型，我们可以针对蓝方固定的进入位置下，红蓝双方的动态攻防决策过程进行数值模拟，从而得到当前位置下蓝方获得突防成功的概率。但是题目中需要分析蓝方在区域内哪个位置进入总能够突防成功，因此我们需要分析蓝方在整个区域内的突防成功概率分布图。我们参考蒙特卡洛方法进行推演评估。

#### 3.5.1 基于蒙特卡洛法的概率推演

蒙特卡洛方法是一种统计试验方法，其基本思想是，为了求解物理、数学以及工程技术上的特定问题，建立一个随机概率模型，使该模型的参数等于问题的解，再通过对模型或过程进行采样，产生随机数模拟模型中的概率分布，并进行多次试验计算所求参数的统计特征，最后给出所求解的近似值<sup>[3,4]</sup>。应用蒙特卡洛法模拟计算的流程如图 8 所示：

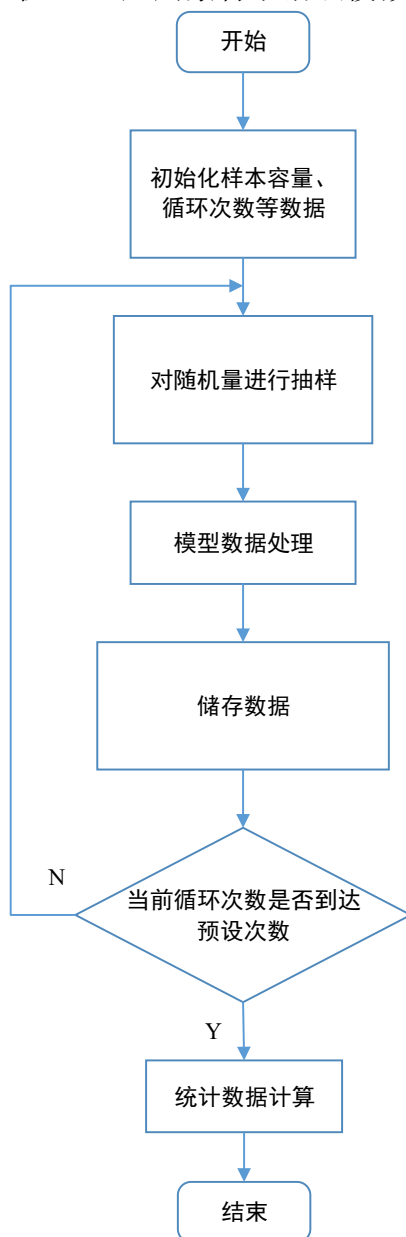


图 8 蒙特卡洛模拟方法流程图

利用蒙特卡洛方法可以逼真地描述具有随机性质的实物特点及物理实验过程，我们在

这里使用蒙特卡洛方法在攻防区域  $ABCD$  中随机洒点，模拟蓝方无人机在不同位置进入攻防区域的情况下，突防成功的概率。工程上，在统计试验中，可以用一定的置信度  $\delta$ ，精度  $\varepsilon$  作为选择试验次数的依据。在给定  $\delta$  和  $\varepsilon$  和真实概率  $P$ ，利用随机变量  $P$  具有渐进正态分布的统计特性<sup>[4]</sup>，在试验次数  $N$  足够大时，有：

$$N = \frac{4P(1-P)}{\varepsilon^2} \quad (14)$$

由上式可知，当取  $N = 5000$  时， $\varepsilon \leq 0.014$ 。

### 3.5.2 蓝方突防概率建模

基于蒙特卡洛法的蓝方突防概率建模的结构框图如图 9 所示：

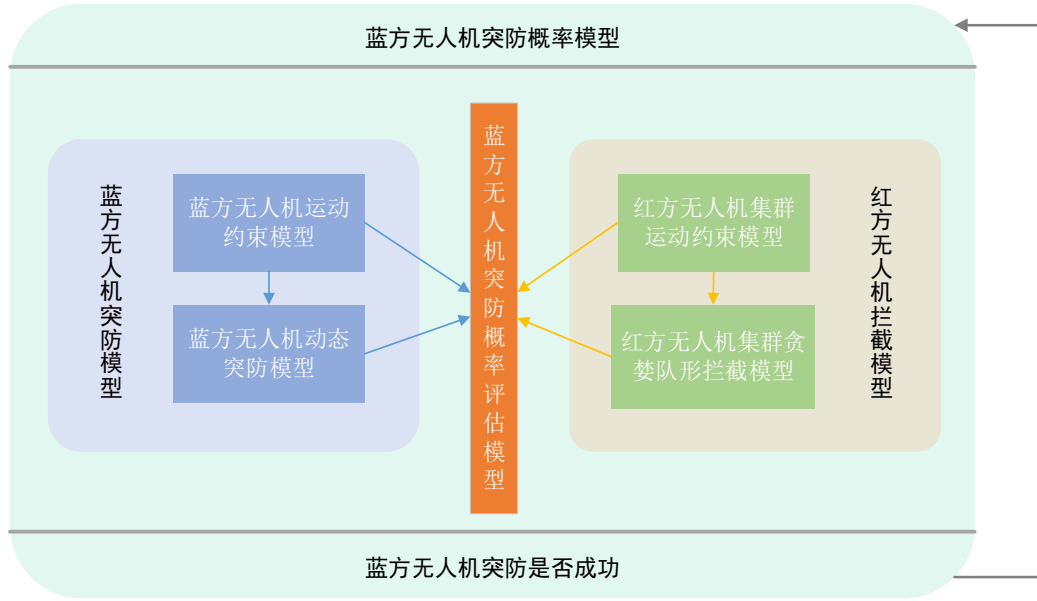


图 9 基于蒙特卡洛法的蓝方突防概率模型结构框图

首先根据蒙特卡洛法对蓝方进入攻防区域的位置进行随机初始化，然后针对每个位置点进行如图 9 所示的攻防推演，在 3.3 和 3.4 节的基础上，可以通过数值推演递归出每个位置点红蓝双方的攻防决策，并得到蓝方最终突防成功与否的结果。对这些初始进入点的结果进行统计，当“撒点”次数足够多时，可以分析得到蓝方突防概率模型，由此可以分析出蓝方在攻防区域内不同区域出发的突防成功率，进而选出蓝方总能突防成功的区域，并对最优决策进行讨论。

## 3.6 蓝方突防概率结果与最优突防策略分析

### 3.6.1 针对蓝方突防概率结果

采用蒙特卡洛随机洒点的方法，结合非线性动态规划得到了蓝方无人机突防情况散点图，如图 10 所示，其中，红色点代表蓝方无人机突防成功的点，蓝点代表蓝方无人机突防失败的点。



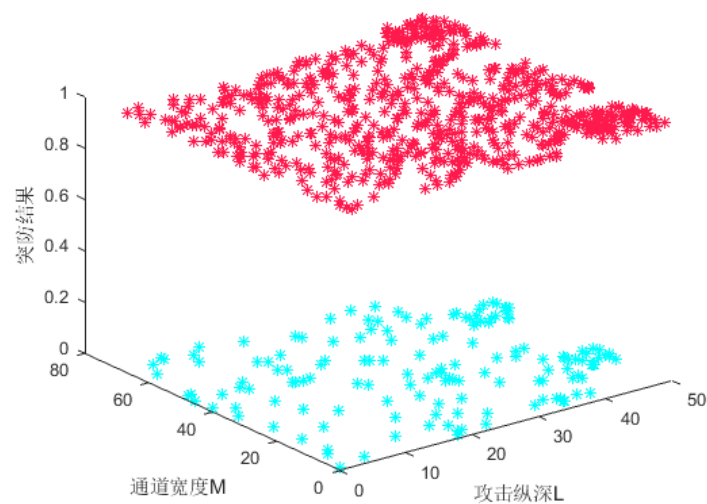


图 10 蓝方无人机突防情况散点图

根据蓝方无人机突防情况散点图，我们绘制了蓝方无人机从不同位置进入的情况下的三维突防概率图和二维突防概率地形图，如图 11，图 12 所示：

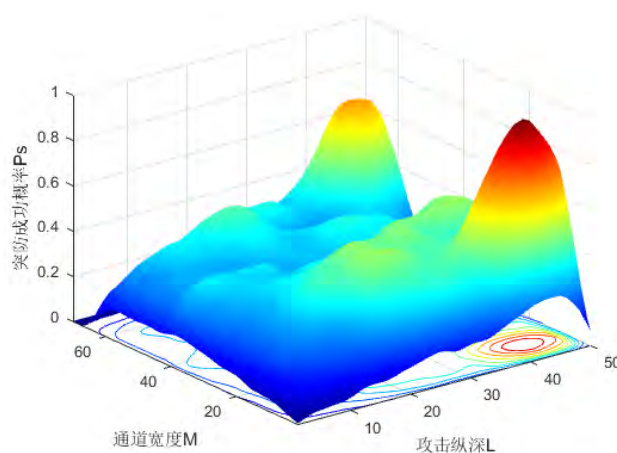


图 11 蓝方无人机三维突防概率图

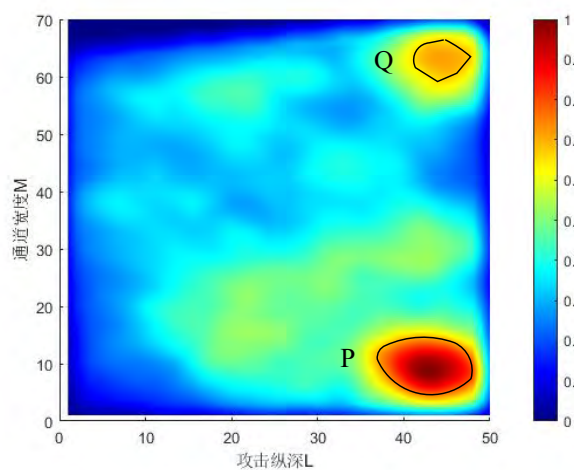


图 12 蓝方无人机二维突防概率地形图



图 12 中的区域 P, Q 即为问题需要求解的蓝方进入位置区域。

### 3.6.2 针对蓝方最优突防策略

从推演中,可以看出,红方无人机机群先采取圆形队形向前突进,再利用拦截宽带最大的队形进行拦截。蓝方无人机为避免面对红方两个无人机机群的围堵,尽量不从 G1G2 通道突围,而选择向 AD (BC) 边机动突围,对应的红方无人机机群 G1 (G2) 也朝着 AD (BC) 边移动围堵,并尽可能接近蓝方无人机。以蓝方无人机与红方无人机机群的尖端点之间的相对距离作为评价标准,红方无人机机群 G1 (G2) 朝着蓝方无人机方向移动,而相应的另一红方无人机机群 G2 (G1) 也向着蓝色无人机方向移动,辅助协防,避免蓝方无人机从红方两无人机机群中间位置实现突防,即均朝着使目标函数减小的方向移动,这时,蓝方无人机要朝着避开红方无人机机群 G1 (G2) 的方向机动,即朝着使目标函数增大的方向移动,实现突防。但由于红方无人机机群中间距离相对较大,蓝方无人机也有从中间通道突防的机会,如图 13 所示:

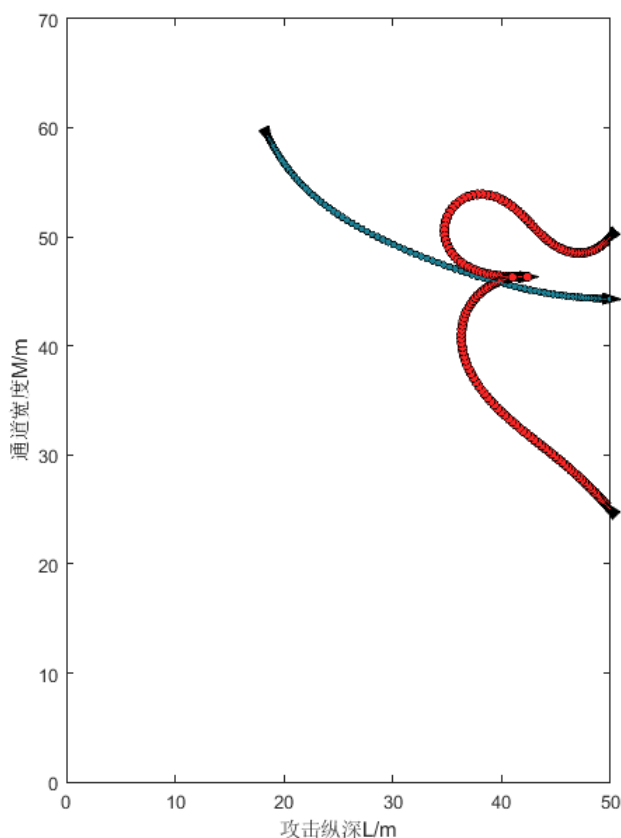


图 13 蓝方无人机从中间位置实现成功突防的推演图

针对此问中,由于蓝方无人机的起始位置未知,我们采用两类蓝方无人机的起始位置进行了策略讨论。

(1) 图 12 对应的蓝方无人机位于 G1G2 通道内任意位置起飞的红蓝双方示意图,由于红方无人机机群中间距离相对较大,有从中心突围的可能性,关于策略的表述见表 2。

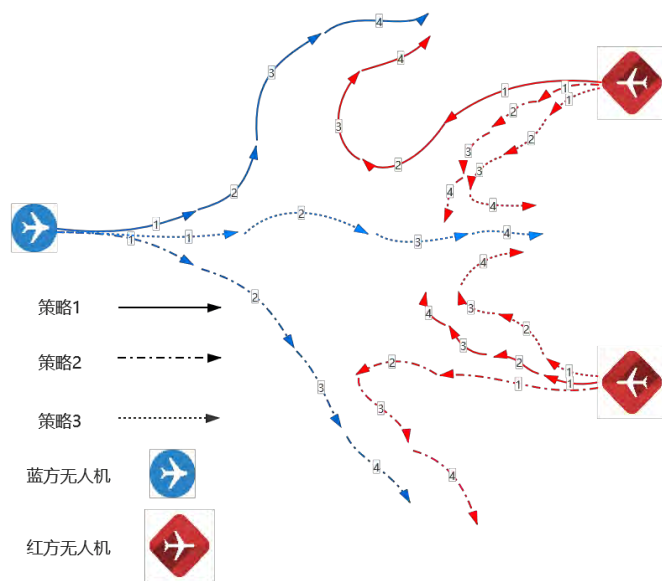


图 14 蓝方无人机位于 G1G2 通道中间任意位置起飞红蓝双方策略示意图

表 2 图 14 对应的策略表述

主要信息	蓝方无人机 E1	红方无人机机群 G1	红方无人机机群 G2
起始位置	G1G2 通道内任意位置	G1	G2
评价标准	尽量大	尽量小	尽量小
策略 1	大致向 AD 边躲避 G1 的方向移动	大致向 AD 边向着 E1 的方向移动	大致向 AD 边向着 E1 的方向移动
策略 2	大致向 BC 边躲避 G2 的方向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的方向移动
策略 3	沿中心线机动	向 E1 方向合围	向 E1 方向合围
策略 1 目的	躲避突防	拦截 E1	辅助协防
策略 2 目的	躲避突防	辅助协防	拦截 E1
策略 3 目的	躲避突防	合围 E1	合围 E1

(2) 图 15 对应的蓝方无人机位于 DG1 通道内任意位置起飞的红蓝双方示意图，关于策略的表述见表 3。

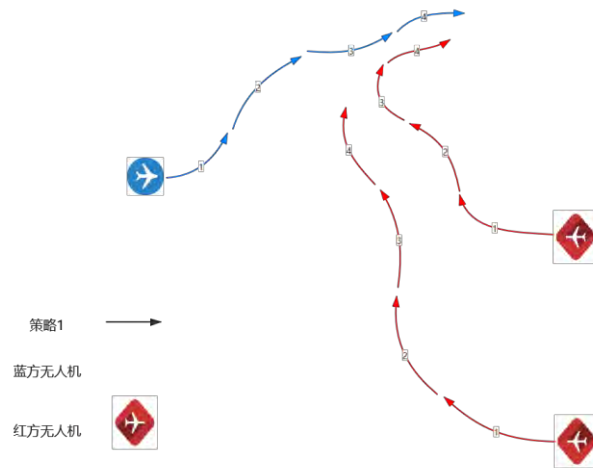


图 15 蓝方无人机位于 DG1 通道中间任意位置起飞红蓝双方策略示意图

表 3 图 15 对应的策略表述

主要信息	蓝方无人机 E1	红方无人机机群 G1	红方无人机机群 G2
起始位置	DG2 通道内任意位置	G1	G2
评价标准	尽量大	尽量小	尽量小
策略 1	大致向 AD 边躲避 G1 的方向移动	大致向 AD 边向着 E1 的方向移动	大致向 AD 边向着 E1 的方向移动
策略 1 目的	躲避突防	拦截 E1	辅助协防

同理，如果蓝方无人机位于 CG2 通道中间任意位置，则红蓝双方无人机所有方向与图 9 中的方向相反，此时红方无人机机群 G1 辅助协防，G2 围堵 E1。

## 4. 问题二模型的建立与求解

### 4.1 针对问题二的分析

问题二与问题一的约束条件不同，将蓝方无人机的进入位置固定，但需要对红方集群的进入位置和对抗平面的通道宽度进行确定和讨论，还要对突防用时最短的策略进行讨论。

根据问题一的推演分析，若取边界 AB 的中心点  $m$  (即蓝方无人机的初始进入点)，边界 CD 的中心点  $n$ ，则连线  $MN$  将整个对抗平面  $ABCD$  分割成了对称的部分。根据问题一得到的蓝方突防概率图，如图 16 所示，如果红方无人机集群以问题一中的点  $G_1$  和  $G_2$  (坐标位置与问题一中一致)作为初始位置，当蓝方无人机从点  $m$  进入平面时，突防成功的概率  $\alpha(E)$  约为 20%。

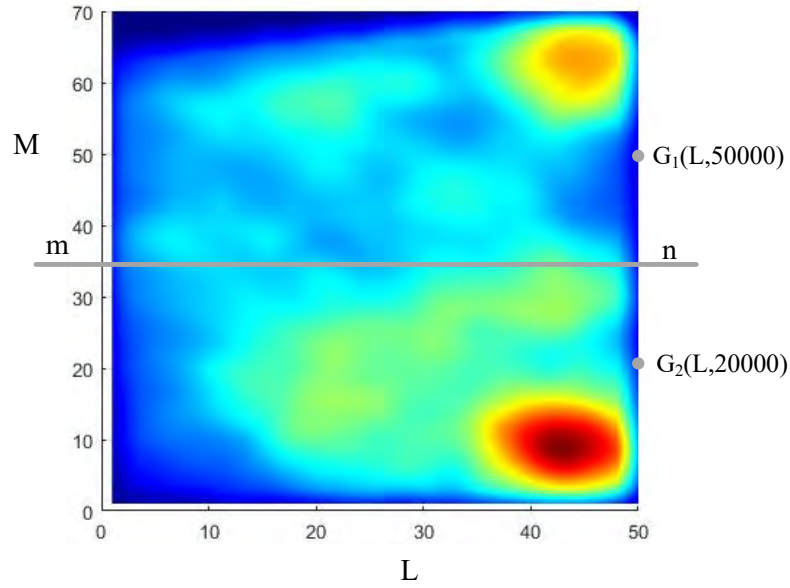


图 16 蓝方突防概率图

由于蒙特卡洛法是基于随机试验的方法，受随机数取数影响会造成一定的不可信情况，但从图中仍可以分析出，蓝方无人机突防成功的概率图具有一定的对称性，与红方飞机的入射点具有一定的关系。这是因为由于通道宽度  $M$  不是足够宽的，因此必定会影响到红蓝双方在攻防过程中的运动轨迹。由于蓝方无法穿过边界  $AD$  和  $BC$ ，所以蓝方飞行器的飞行轨迹就会与红方的入射点和入射方向有较大的关联。

另一方面，因此我们在针对变通道约束下蓝方非线性规划问题时，关注到用时最短的优化目标，通过改进优化函数对红蓝双方的攻防策略进行推演。

#### 4.2 基于 $L_2$ 正则化思想的目标函数优化

本问题中需要建模并分析能够使蓝方无人机突防时间最短的最优策略，而问题一中我们使用的红蓝双方的优化目标函数(2)(3)都是关注距离的，所以得到最优策略是路径长度最优的。但根据本题的速度约束，红蓝双方无人机和运载机的速度大小不变，那么在同一通道宽度下，突防轨迹长度  $l_E$  与突防时间  $t_E$  可以认为满足关系：

$$l_E = l(t_E) \quad (15)$$

在机器学习中，正则化方法是一类通过在损失函数里加优化项，来改善损失函数的效果，提高优化效率，获得更好优化结果的方法。常用的有  $L_1$  范数和  $L_2$  范数，其中  $L_1$  范数是在损失函数中添加优化项的线性项，这样使求解的过程中，系数矩阵出现更多的 0 项，使系数矩阵变得更加稀疏，实现对特征进行提取的效果； $L_2$  范数是在损失函数中添加优化项的 2 次项，这样使求解的过程中，系数矩阵不会出现 0 项，但会导致部分元素的值变小。若原损失函数：

$$L = L(\theta) \quad (16)$$

则加上正则项的损失函数可以表示为：

$$L = L(\theta) + \lambda \sum_i^n \theta_i^2 \quad (17)$$

其中  $\lambda$  控制正则项的大小，较大的取值将较大程度约束模型复杂度，反之亦然。

$L_2$  正则项通常对稀疏的有尖峰的权重向量施加大的惩罚，而偏好于均匀的参数。所以  $L_2$  正则项加入之后，权重的绝对值大小就会整体倾向于减少，尤其不会出现特别大的值（比如噪声），这种衰减可以视作对权值的一种惩罚。

对于本题中通道宽度变更，并且更需要关注突防时间更短的策略，所以我们考虑对蓝方无人机突防目标函数进行优化，增加关于突防时间的正则项：

$$\max d_{E,P} - (1-\lambda)l_E - \lambda t_E = \|E_1 - P_k^*\| - (1-\lambda)l_E - \lambda t_E, k=1,2 \quad (18)$$

同时，约束(1)中删去关于通道宽度  $M$  的约束，再基于蒙特卡洛方法对蓝方突防策略进行推演。

#### 4.3 基于正则化目标函数的蓝方突防策略推演模型

基于上一小节建立的约束和正则化目标函数，我们可以对固定通道带宽下的蓝方突防策略进行推演。针对通道带宽的下限问题，我们借鉴二分法中的逐次逼近思想，在取不同的通道宽度变化量  $\Delta M$  进行推演时，先取较大的变化量值，当发现蓝方无人机的突防概率降低梯度较快时，将变化量的步长按二分思想减少，渐进逼近蓝方总能突防成功的通道带宽下限  $M_{\min}$ ，流程如图 17 所示。

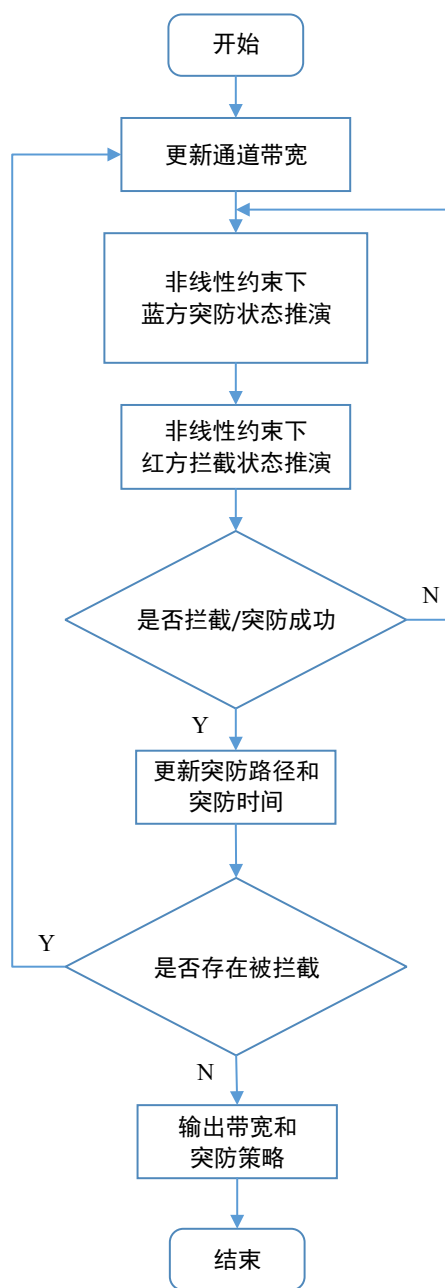


图 17 基于正则化目标函数的蓝方突防策略推演模型流程图

#### 4.4 最小带宽与蓝方突防时间最短策略分析

##### 4.4.1 针对最小带宽

通过渐进松弛的方式得到通道宽度与蓝方无人机突防成功率的潜在模型，通过模型收敛得到最小通道宽度为 91km，如图 18 所示。

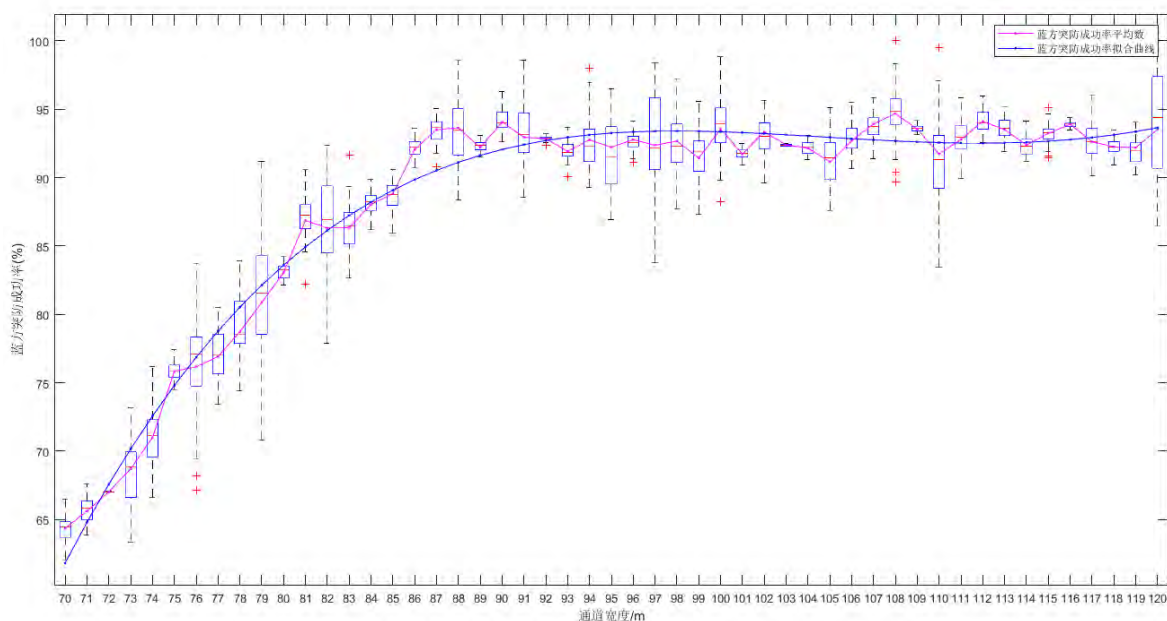


图 18 最小通道宽度确定模型迭代图

#### 4.4.2 蓝方突防时间最短策略

此问中，蓝方无人机位置确定在 **AB** 中心点上，而红方无人机机群的具体位置不固定，本题以红方无人机的相对位置讨论了两种蓝方突防策略类型，根据蓝方无人机的位置可以确定红方无人机机群的起飞位置要处在距离蓝方无人机所在中心线等距的两侧，分别设定为距 **C (D)**  $M/8$ 、 $3M/8$  的位置上，整体的突防策略参照前文的分析，此处略有不同的地方是，当红方无人机机群设定为距 **C (D)**  $M/8$  的位置上起飞时，由于红方两无人机机群中心距离增大，蓝方无人机也有从中间突围成功的可能性，由图 19(a)使用带宽为 120Km 时，验证了这种蓝方无人机从中间位置机动时成功的可行性，所以又增加了蓝方无人机从中间突防的策略，相应的图 19(b)使用带宽为 80Km 时，另外，结合对称性验证了蓝方无人机向两侧机动的可行性，蓝方最短时间策略分析如图 20，图 21 所示。

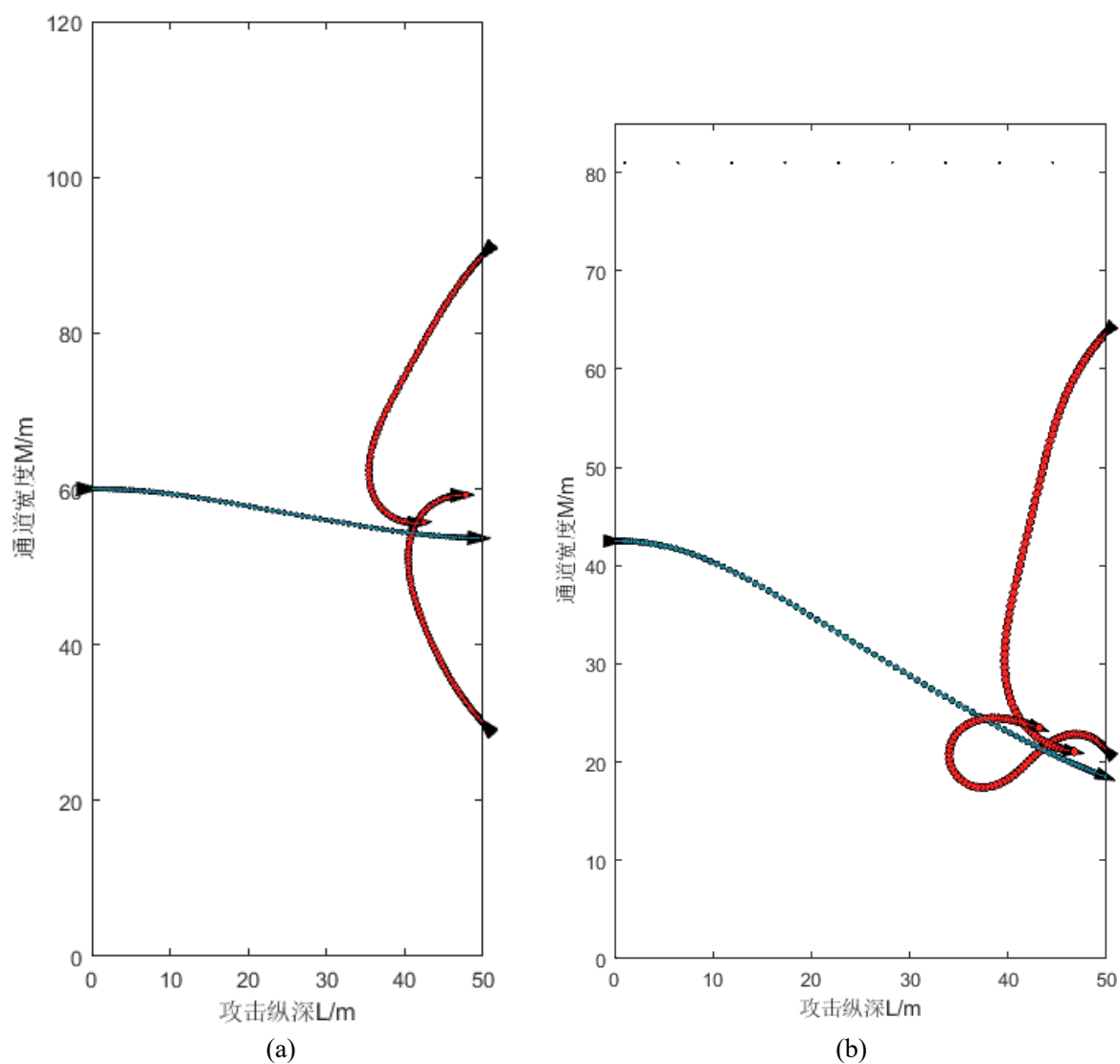


图 19(a) 通道宽度为 120Km 时蓝方无人机中间位置突防成功可行性示意图  
 (b) 通道宽度为 120Km 时蓝方无人机两侧突防成功可行性示意图

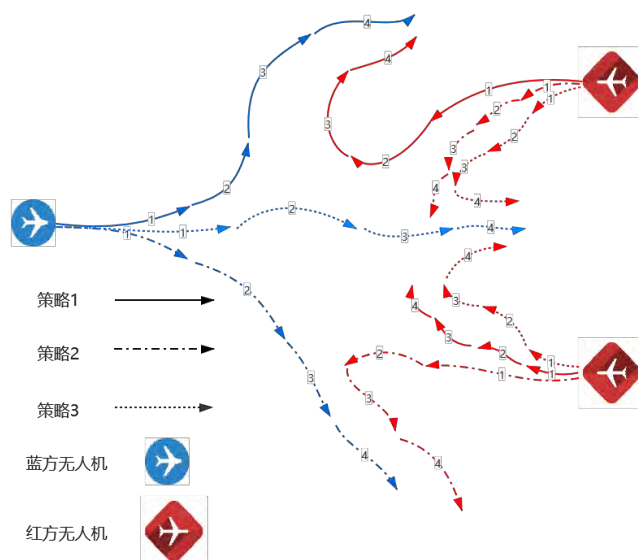


图 20 红方无人机机群起飞位置距 C (D) M/8 时, 红蓝双方策略示意图



图 20 对应红方无人机机群起飞位置距 C (D) M/8 时，红蓝双方策略示意图，由于红方两无人机机群中心距离增大，蓝方无人机也有从中间突围成功的可能性，所以有策略 3 的存在，关于策略的表述见表 4。

表 4 图 20 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机机群 G1	红方无人机机群 G2
起始位置	AB 中心点	距 D 点 M/8 处	距 C 点 M/8 处
评价标准	尽量大	尽量小	尽量小
策略 1	大致向 AD 边躲避 G1 的方向移动	大致先向中心移动再向 E1 的方向移动	大致向 AD 边向着 E1 的方向移动
策略 2	大致向 BC 边躲避 G2 的方向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的方向移动
策略 3	从中间机动	向中心向着 E1 的方向合围	向中心向着 E1 的方向合围
策略 1 目的	躲避突防	拦截 E1	辅助协防
策略 2 目的	躲避突防	辅助协防	拦截 E1
策略 3 目的	躲避突防	合围 E1	合围 E1

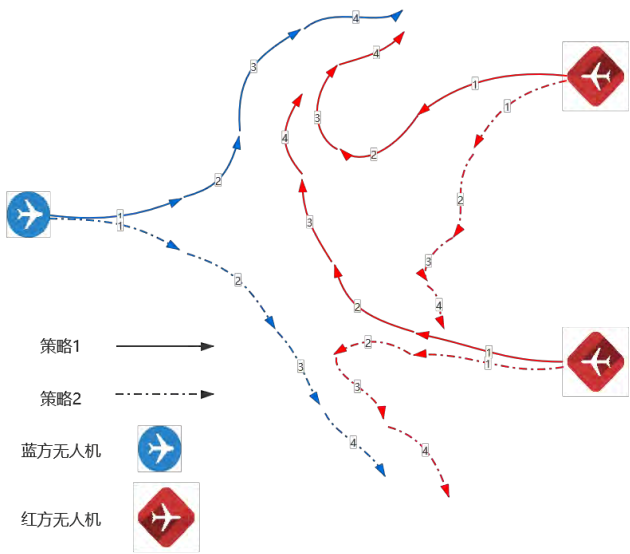


图 21 红方无人机机群起飞位置距 C (D) 3M/8 时，红蓝双方策略示意图

图 21 对应红方无人机机群起飞位置距 C (D) 3M/8 时，红蓝双方策略示意图，关于策略的表述见表 5。

表 5 图 21 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机机群 G1	红方无人机机群 G2
起始位置	AB 中心点上	距 D 点 3M/8 处	距 C 点 3M/8 处
策略 1	大致向 AD 边躲避 G1 的方向移动	大致先向中心移动再向 E1 的方向移动	大致向 AD 边向着 E1 的方向移动
策略 2	大致向 BC 边躲避 G2 的方向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的方向移动
策略 1 目的	躲避突防	拦截 E1	辅助协防
策略 2 目的	躲避突防	辅助协防	拦截 E1

## 5. 问题三模型的建立与求解

### 5.1 针对问题三的分析

在前两问的问题中，蓝方无人机始终只有一架进行突防，没有形成集群进行协同突防，而红方运载机则始终只发射一波无人机，因此发射时双方无人机的相对位置会对红方的拦截效果产生影响。

在本题中，红方运载机可携带 10 架无人机，并且可以根据蓝方无人机的相对位置和本方无人机的拦截情况进行 2 波次的投放，这使得蓝方无人机的突防难度大大提高。同时，我们在建模时需要考虑前后两波红方无人机的投放时间差和投放距离。这使得约束的非线性更强，同时在进行优化策略推演的过程中计算量更大。因此我们考虑简化推演的计算量。

同时，针对通道带宽的上限问题，我们借鉴与问题二中类似的渐进紧缩思想，在取不同的通道宽度变化量  $\Delta M$  进行推演时，先取较大的变化量值，当发现蓝方无人机的突防概率降低梯度较快时，将变化量的步长按二分思想减少，渐进逼近蓝方无法突防成功的通道带宽上限  $M_{\max}$ 。

### 5.2 基于全联立正交配置有限元法的非线性求解优化模型

在本题中，由于红方运载机可以释放 2 波次的无人机，需要考虑的约束非线性较强，且计算量较大，我们考虑对该约束下的规划问题进行分解简化，提高运算效率。

正交配置有限元方法的形式类似隐式龙格-库塔方法，将最优控制问题中的所有状态全部同时离散化，在  $N_f$  段等长的有限元  $[\bar{t}_{i-1}, \bar{t}_i]$ ，采用拉格朗日插值函数形式对各变量予以描述：

$$z(t) = \sum_{j=0}^K l_j(\tau) z_{i,j} \quad (19)$$

$$t = \bar{t}_{i-1} + (\bar{t}_i - \bar{t}_{i-1}) \cdot \tau, \tau \in [0,1] \quad (20)$$

其中， $K$  为分段插值的阶次， $l_j(\tau)$  为微分变量  $z(t)$  的拉格朗日基函数， $z_{i,j}$  为微分变量的离散配置点。可以得到以下分段 NLP 问题：

$$\begin{aligned}
& \min J(t_f) \\
s.t. \quad & \sum_{k=0}^K \left( \frac{d(\prod_{m=0, \neq j}^K \frac{\tau - \tau_m}{\tau_j - \tau_m})}{d\tau} \Big|_{\tau = \tau_k} \cdot z_{i,k} \right) = (\bar{t}_i - \bar{t}_{i-1}) \cdot F(z_{i,j}) \\
& G(z_{i,j}) \leq 0, \\
& \bar{t}_i - \bar{t}_{i-1} = \frac{t_f - t_0}{N_{fe}}, \bar{t}_0 = t_0, \bar{t}_{N_{fe}} = t_f, \\
& i = 1, \dots, N_{fe}, j = 0, \dots, K
\end{aligned} \tag{21}$$

### 5.3 最大带宽与红方最优拦截策略分析

#### 5.3.1 针对最大带宽

通过渐进紧缩的方式得到通道宽度与蓝方无人机突防成功率的潜在模型，通过模型收敛得到最大通道宽度为 39Km，如图 22 所示。

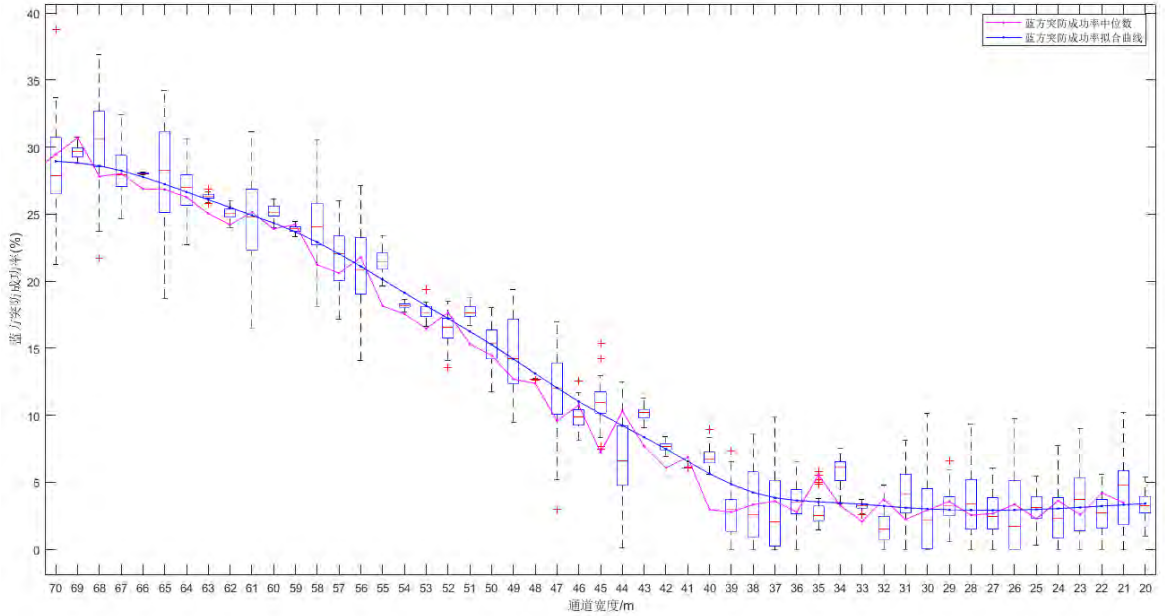


图 22 最大通道宽度确定模型示意图

#### 5.3.1 针对红方最优拦截策略

类似于问题二，蓝方无人机位置固定在 AB 中心点上，而红方两无人机机群的具体位置不固定，这里依旧采用问题二中位置的选择讨论攻防策略，策略整体参考 3.6 节的分析，不同之处是，另外红方运载机会分两个批次发射无人机，具体的发射位置信息根据当时战场的情况决定，发射时间为红方运载机到达与第一批次无人机群距离限制最大边界值的时刻，再者，当红方无人机机群设定为距 C (D) M/8 的位置上起飞时，由于红方两无人机机群中心距离增大，蓝方无人机也有从中间突围成功的可能性，由图 23(a)验证了蓝方无人机从中间飞出的可能性。另外，由图 23(b)结合对称性，验证了蓝方无人机向两侧机动的可行性，具体的红蓝双方攻防策略，如图 24-28 所示。

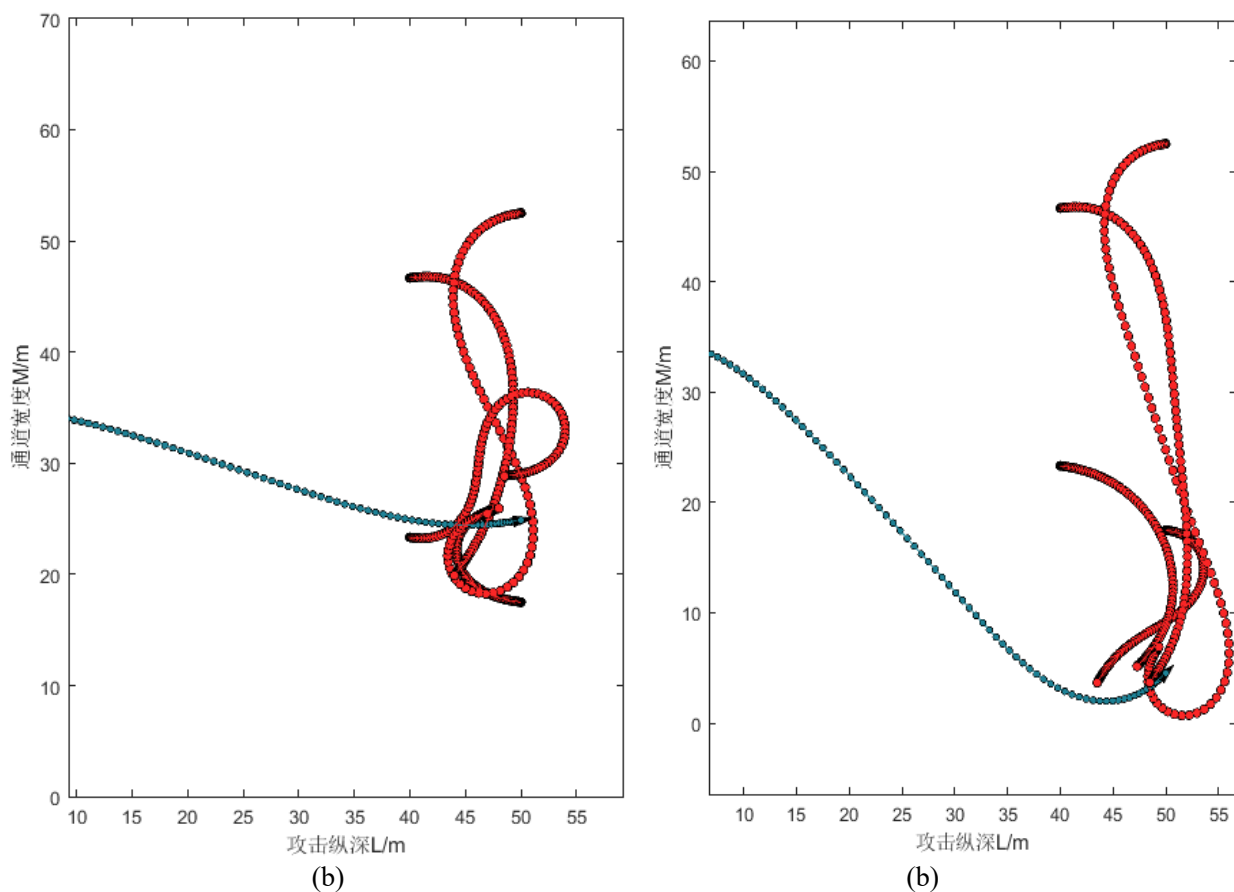


图 23(a) 蓝方无人机从中间位置成功突防的可行性示意图  
(b) 蓝方无人机从两侧方向成功突防的可行性示意图

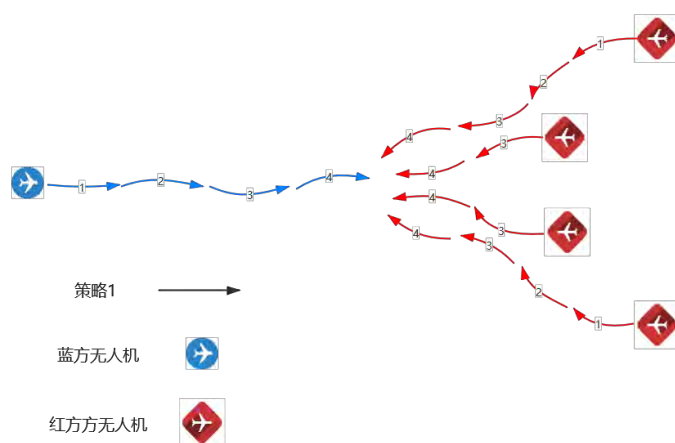


图 24 红方无人机机群起飞位置距 C (D) M/8 时, 红蓝双方策略示意图 (1)

图 24 对应红方无人机机群起飞位置距 C (D) M/8 时, 红蓝双方策略示意图, 由于红方无人机群中间距离非常大, 所以蓝方无人机有从中间突围的可能性, 也包含了红方无人机第二批次发射的位置信息, 以及相应的策略描述, 关于策略的表述见表 6。

表 6 图 24 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机机群 G1	红方无人机机群 G2	红方无人机机群 G1'	红方无人机机群 G2'
起始位置	AB 中心点上	距 D 点 M/8 处	距 C 点 M/8 处	G1 斜下方	G2 斜上方
评价标准	尽量大	尽量小	尽量小	尽量小	尽量小
策略 1	沿中心线机动	向中心向着 E1 的方向合围	向中心向着 E1 的方向合围	向中心向着 E1 的方向合围	向中心向着 E1 的方向合围
策略 1 目的	躲避突防	合围 E1	合围 E1	合围 E1	合围 E1

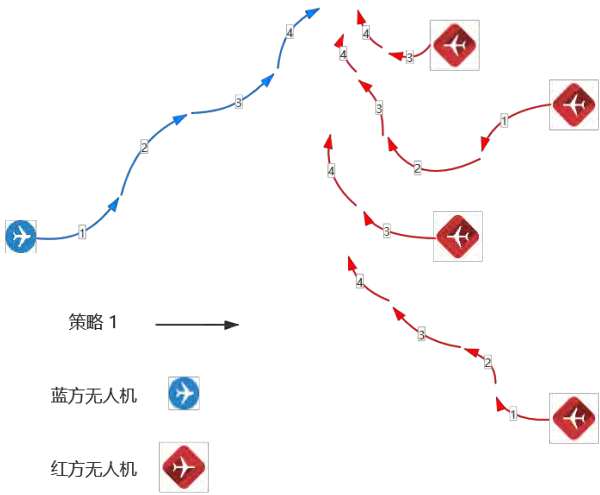


图 25 红方无人机机群起飞位置距 C (D) M/8 时，红蓝双方策略示意图 (2)

图 25 对应红方无人机机群起飞位置距 C (D) M/8 时，蓝方无人机向 AD 侧机动的红蓝双方策略示意图，也包含了红方无人机第二批次发射的位置信息，以及相应的策略描述，关于策略的表述见表 7。

表 7 图 25 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机机群 G1	红方无人机机群 G2	红方无人机机群 G1'	红方无人机机群 G2'
起始位置	AB 中心点上	距 D 点 M/8 处	距 C 点 M/8 处	G1 斜上方	G2 斜上方
评价标准	尽量大	尽量小	尽量小	尽量小	尽量小
策略 1	大致向 AD 边躲避 G1 的方向移动	大致向中心线移动，再向 E1 的方向移动	大致向 AD 边向着 E1 的方向移动	大致向 AD 边向着 E1 的方向移动	大致向 AD 边向着 E1 的方向移动
策略 1 目的	躲避突防	拦截 E1	辅助协防	拦截 E1	拦截 E1

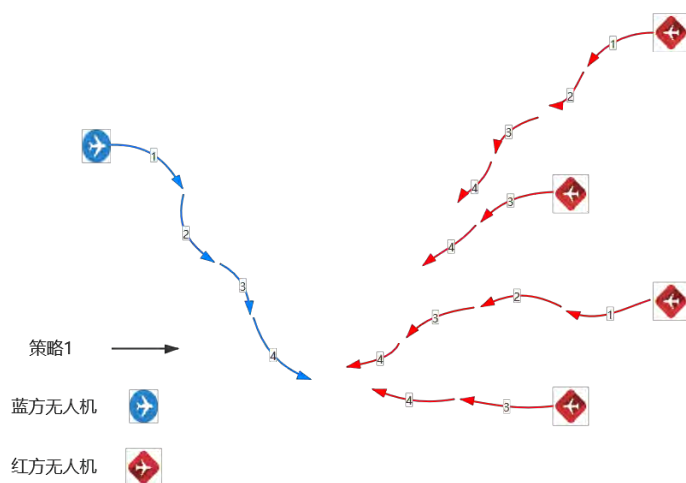


图 26 红方无人机机群起飞位置距 C (D)  $M/8$  时，红蓝双方策略示意图 (3)

图 26 对应红方无人机机群起飞位置距 C (D)  $M/8$  时，蓝方无人机向 BC 侧机动的红蓝双方策略示意图，也包含了红方无人机第二批次发射的位置信息，以及相应的策略描述，关于策略的表述见表 8。

表 8 图 26 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机 机群 G1	红方无人机 机群 G2	红方无人机 机群 G1'	红方无人机 机群 G2'
起始位置	AB 中心点上	距 D 点 $M/8$ 处	距 C 点 $M/8$ 处	G1 斜下方	G2 斜下方
评价标准	尽量大	尽量小	尽量小	尽量小	尽量小
策略 1	大致向 BC 边 躲避 G2 的方 向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的 方向移动	大致向 BC 边向着 E1 的 方向移动
策略 1 目的	躲避突防	辅助协防	拦截 E1	拦截 E1	拦截 E1

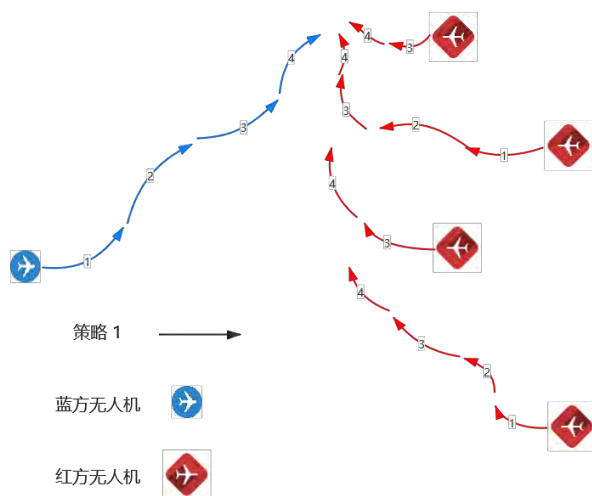


图 27 红方无人机机群起飞位置距 C (D)  $3M/8$  时，红蓝双方策略示意图 (1)

图 27 对应红方无人机机群起飞位置距 C（D）3M/8 时，蓝方无人机向 AD 侧机动的红蓝双方策略示意图，也包含了红方无人机第二批次发射的位置信息，以及相应的策略描述，关于策略的表述见表 9。

表 9 图 27 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机 机群 G1	红方无人机 机群 G2	红方无人机 机群 G1’	红方无人机 机群 G2’
起始位置	AB 中心点上	距 D 点 3M/8 处	距 C 点 3M/8 处	G1 斜上方	G2 斜上方
评价标准	尽量大	尽量小	尽量小	尽量小	尽量小
策略 1	大致向 AD 边 躲避 G1 的方 向移动	大致向 AD 边向着 E1 的方向移动	大致向 AD 边向着 E1 的方向移动	大致向 AD 边向着 E1 的 方向移动	大致向 AD 边向着 E1 的 方向移动
策略 1 目的	躲避突防	拦截 E1	辅助协防	拦截 E1	拦截/协防

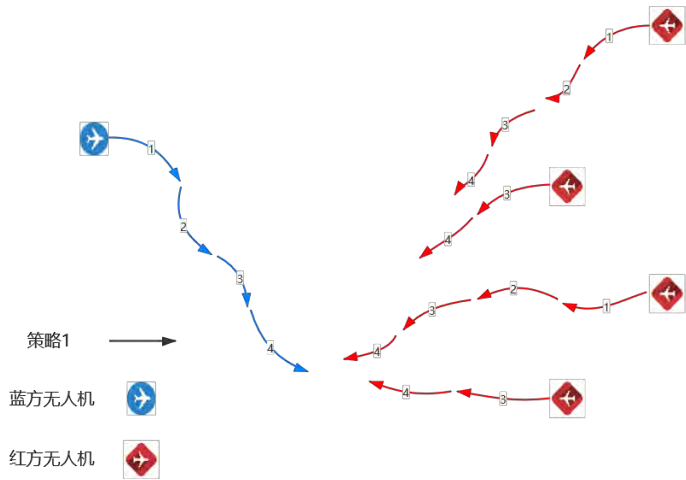


图 28 红方无人机机群起飞位置距 C（D）3M/8 时，红蓝双方策略示意图（2）

图 28 对应红方无人机机群起飞位置距 C（D）3M/8 时，蓝方无人机向 BC 侧机动的红蓝双方策略示意图，也包含了红方无人机第二批次发射的位置信息，以及相应的策略描述，关于策略的表述见表 10。

表 10 图 28 策略示意图表述

主要信息	蓝方无人机 E1	红方无人机 机群 G1	红方无人机 机群 G2	红方无人机 机群 G1’	红方无人机 机群 G2’
起始位置	AB 中心点上	距 D 点 3M/8 处	距 C 点 3M/8 处	G1 斜下方	G2 斜下方
评价标准/ 相对距离	躲避/尽量大	接近/尽量小	接近/尽量 小	接近/尽量小	接近/尽量小
策略 1	大致向 BC 边 躲避 G2 的方 向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的方向移动	大致向 BC 边向着 E1 的 方向移动	大致向 BC 边向着 E1 的 方向移动
策略 1 目的	躲避突防	辅助协防	拦截 E1	围堵/协防	拦截 E1

## 6. 问题四模型的建立与求解

### 6.1 针对问题四的分析

在前三问的问题中，始终只有一架蓝方无人机进行突防，没有形成机群进行协同突防，第三问中，红方运载机携带 10 架无人机分批次发射无人机，加大了拦截效果。

在本题中，红方运载机数量增加为 5 架，依旧携带 10 架无人机，并且可以根据蓝方无人机的相对位置和本方无人机的拦截情况进行第二次的发射，这就使得蓝方无人机的突防难度变得比第三问更大。相应的本题在建模时两波次红方无人机的投放时间、投放位置随着红方无人机数量的增加使得非线性更强，同时在进行优化策略推演的过程中计算量将变得比第三问更大。同时，蓝方无人机增加为三架，可以形成机群进行协同突防，使得突防能力变强，因此需要简化约束条件，以降低计算量。

同时，问题四还规定了整个对抗平面的攻击纵深  $L$ ，通道宽度  $M$ ，但没有规定三架蓝方无人机、五架红方运载机的相应的在  $AB$ 、 $CD$  边的具体起飞位置，因此需要讨论红蓝双方可能存在的相对起飞位置，并考虑在此约束下的非线性规划问题，另外，需要关注目标函数的改进，蓝方尽可能多的无人机突防成功，而红方尽可能多的拦截蓝方无人机，通过对目标函数的改进对红蓝双方的攻防策略进行推演。

### 6.2 基于 NLP 的红蓝双方攻防策略推演模型

红、蓝两方无人机在平面空间进行对抗的过程中，蓝方无人机首先要发起进攻，也就是选择有利于突防成功的策略，而对于红方无人机集群来说，面对蓝方的突防，首先需要采取有利于拦截蓝方无人机的策略，整个过程中红蓝双方的无人机实时对对方的决策进行认知和回应，形成了一种在约束条件下的动态博弈的过程<sup>[5,6]</sup>。

基于问题一、二、三建立的约束和正则化目标函数，对非线性约束下的最优化问题采用简化的全联立正交有限元-内点法进行数值计算，我们可以对固定通道带宽下的红蓝双方拦截和突防策略进行推演，结构框如图 29 所示。

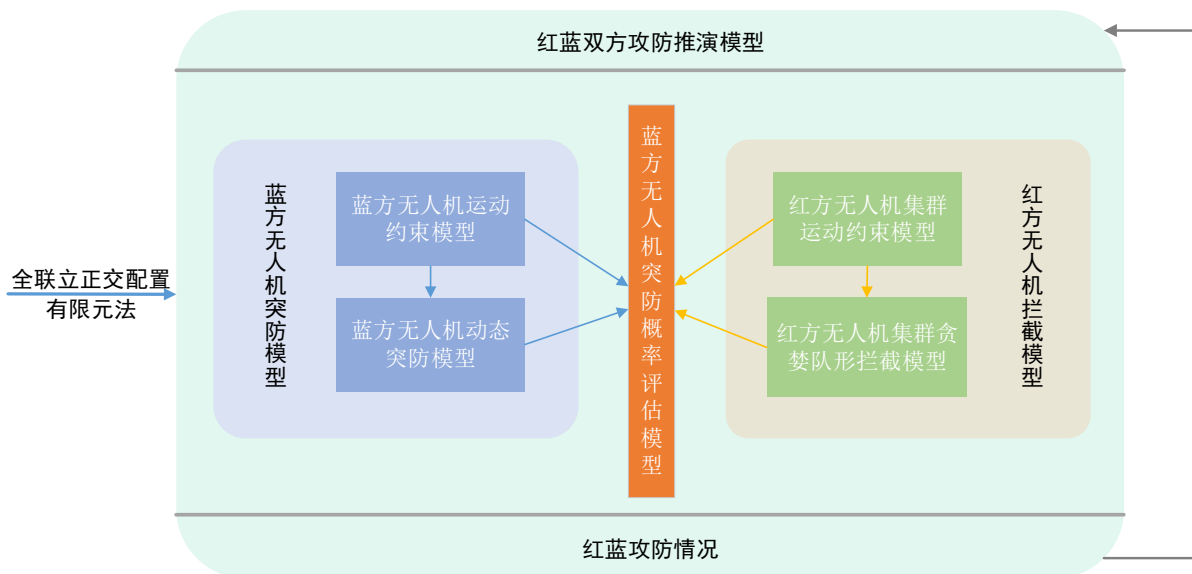


图 29 基于 NLP 的红蓝双方攻防策略推演模型



### 6.3 红方最优拦截策略与蓝方最优突防策略分析

此问中，蓝方无人机和红方无人机机群具体出发位置未知，此处以双方相对出发位置提出具体的攻防策略，具体策略可参考前述小节的描述。不同之处是，蓝方无人机数量变为 3 架，红方运载机增加为 5 架，会分两个波次发射无人机，具体的发射位置信息根据当时战场的情况决定，发射时间为红方运载机到达与第一波次无人机群距离限制最大边界值的时刻，蓝方为尽可能的实现多架无人机的成功突防，三架不同的蓝方无人机分别向 AD、BC、以及中心线位置机动，图 30 验证了这种飞行方策略的可行性。具体的攻防策略，如图 31 所示。

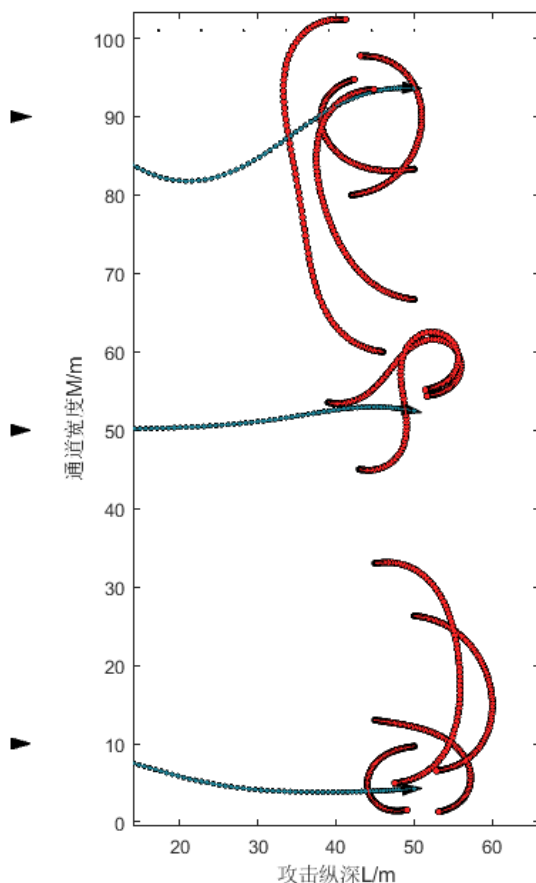


图 30 蓝方无人机分别向三个方向突围时可行性示意图

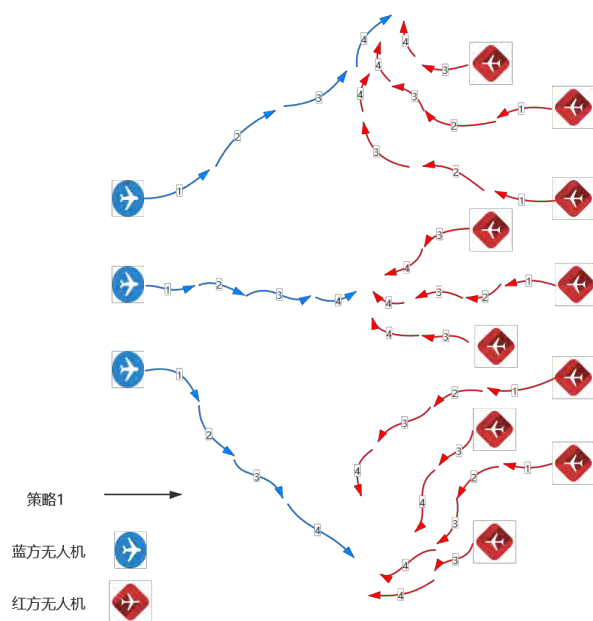


图 31 蓝方无人机分别向三个方向突围时，红蓝双方策略示意图

图 31 对应蓝方三架无人机向三个方向突围的红蓝双方策略示意图，也包含了第二批无人机群的发射位置信息，具体的红蓝双方的位置信息及策略描述见下表 11。

表 11 图 31 策略示意图表述

无人机类型	起始位置	策略 1	策略 1 目的
蓝方 E1	大约 2M/3	大致向 AD 方向机动	躲避突防
蓝方 E2	M/2	沿中心位置机动	躲避突防
蓝方 E3	大约 M/3	大致向 BC 方向机动	躲避突防
红方机群 G1	大约 5M/6	大致向 E1 方向机动	拦截 E1
红方机群 G2	大约 2M/3	大致向 E1 方向机动	拦截 E1
红方机群 G3	大约 M/2	大致向 E2 方向机动	拦截 E2
红方机群 G4	大约 M/3	大致向 E3 方向机动	拦截 E3
红方机群 G5	大约 M/6	大致向 E3 方向机动	拦截 E3
红方机群 G1'	G1 斜上方	大致向 E1 方向机动	拦截 E1
红方机群 G2'	G2 斜下方	大致向 E2 方向机动	拦截 E2
红方机群 G3'	G3 斜下方	大致向 E2 方向机动	拦截 E2
红方机群 G4'	G4 斜下方	大致向 E3 方向机动	拦截 E3
红方机群 G5'	G5 斜下方	大致向 E3 方向机动	拦截 E3

由于时间关系，没来得及做另外两种情况的仿真，此处提供了红蓝双方的另外两种示意图，结合前文实验结果发现，蓝方无人机向两侧机动，会有更大成功的可能性，也给下文的两种策略提供了可行性分析。

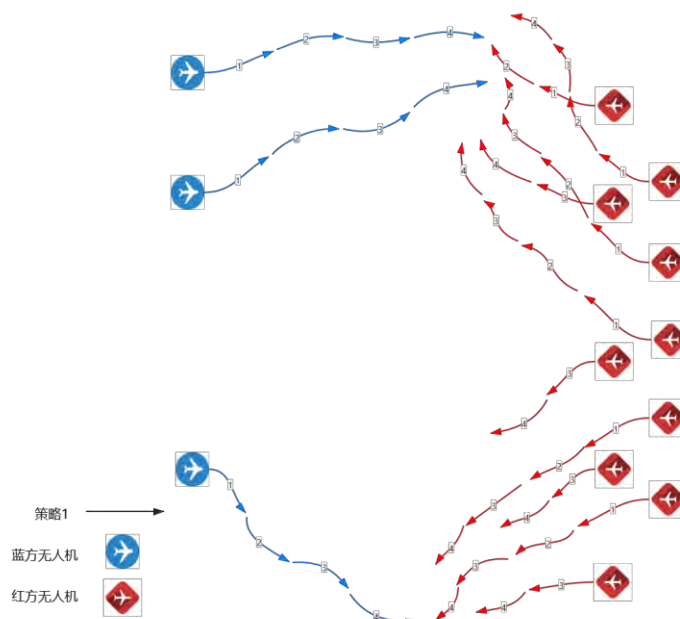


图 32 蓝方无人机采取 2-1 两边突围时，红蓝双方策略示意图

图 32 对应蓝方三架无人机采取 2-1 向两边方向突围的红蓝双方策略示意图，也包含了第二批次无人机群的发射位置信息，具体的红蓝双方的位置信息及策略描述见下表 12。

表 12 图 32 策略示意图表述

无人机类型	起始位置	策略 1	策略 1 目的
蓝方 E1	靠近 3M/4	大致向 AD 方向机动	躲避突防
蓝方 E2	大约 5M/8	大致向 AD 方向机动	躲避突防
蓝方 E3	大约 3M/16	大致向 BC 方向机动	躲避突防
红方机群 G1	大约 5M/8	大致向 E1 方向机动	拦截 E1
红方机群 G2	大约 M/2	大致向 E2 方向机动	拦截 E2
红方机群 G3	大约 3M/8	大致向 E2 方向机动	拦截 E2
红方机群 G4	大约 M/4	大致向 E3 方向机动	拦截 E3
红方机群 G5	大约 M/8	大致向 E3 方向机动	拦截 E3
红方机群 G1'	G1 斜上方	大致向 E1 方向机动	拦截 E1
红方机群 G2'	G2 斜上方	大致向 E2 方向机动	拦截 E2
红方机群 G3'	G3 斜下方	大致向中心方向机动	辅助协防
红方机群 G4'	G4 斜下方	大致向 E3 方向机动	拦截 E3
红方机群 G5'	G5 斜下方	大致向 E3 方向机动	拦截 E3

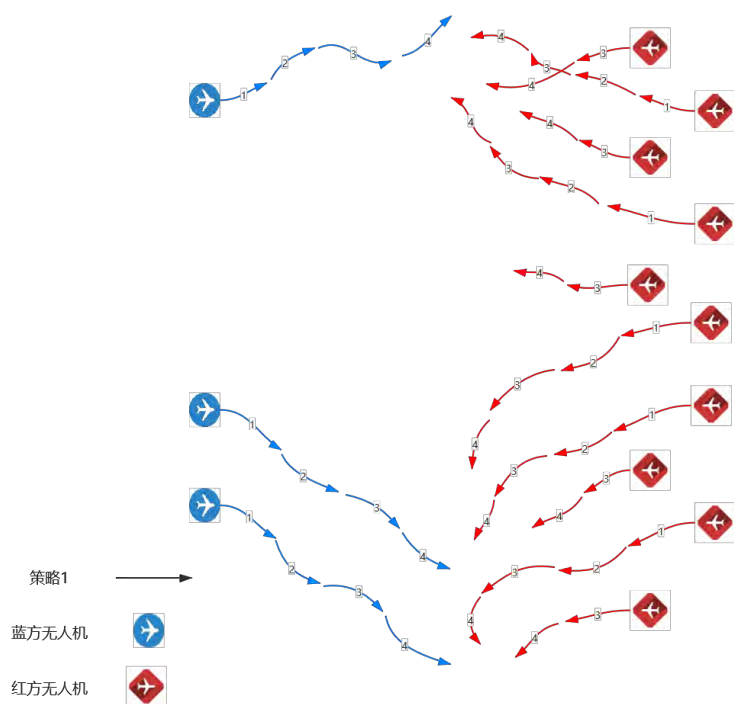


图 33 蓝方无人机采取 1-2 两边突围时，红蓝双方策略示意图

图 33 对应蓝方三架无人机采取 1-2 向两边方向突围的红蓝双方策略示意图，也包含了第二批次无人机群的发射位置信息，具体的红蓝双方的位置信息及策略描述见下表 13。

表 13 图 33 策略示意图表述

无人机类型	起始位置	策略 1	策略 1 目的
蓝方 E1	靠近 13M/16	大致向 AD 方向机动	躲避突防
蓝方 E2	大约 3M/8	大致向 BC 方向机动	躲避突防
蓝方 E3	大约 M/4	大致向 BC 方向机动	躲避突防
红方机群 G1	大约 7M/8	大致向 E1 方向机动	拦截 E1
红方机群 G2	大约 3M/4	大致向 E1 方向机动	拦截 E1
红方机群 G3	大约 5M/8	大致向 E2 方向机动	拦截 E2
红方机群 G4	大约 M/2	大致向 E2 方向机动	拦截 E2
红方机群 G5	大约 3M/8	大致向 E3 方向机动	拦截 E3
红方机群 G1'	G1 斜上方	大致向 E1 方向机动	拦截 E1
红方机群 G2'	G2 斜上方	大致向 E1 方向机动	拦截 E1
红方机群 G3'	G3 斜上方	大致向中心方向机动	辅助协防
红方机群 G4'	G4 斜下方	大致向 E2 方向机动	拦截 E2
红方机群 G5'	G5 斜下方	大致向 E3 方向机动	拦截 E3

## 7. 模型的评价

### 7.1 模型的优点

1. 以蓝方无人机动态避障、红方无人机协作拦截及各无人机之间安全与拦截裕度建立解析形式约束条件，构建了统一的模型框架求解所考虑的场景下的无人机集群协同对抗的最优策略。

2. 针对红方无人机设计基于贪婪队形的集群协作拦截，当双方距离较远时可以采用“圆形”队形，保持直线突进距离最小；当接近蓝方无人机时，切换至“五环”队形，实现布防面积最大化，从而有效增加拦截机率。

3. 针对蓝方无人机设计非线性突防策略推演模型，基于 AMPL-IPOPT，采用“全联立正交配置有限元方法”及“内点算法”对非线性规划问题进行求解，大大降低了算法的复杂度，提高了实时性能。

### 7.2 模型的缺点

1. 模型所考虑到的红方变换队形相对单一，一定程度上影响红方拦截的成功率。

2. 针对问题二，推演蓝方无人机成功突防区域时，采用的是基于 Monte-Carlo 随机采样算法，运算负荷较大，可以进一步考虑基于数理推演简化的机理模型。

## 8. 参考文献

- [1] 李柏. 复杂约束下自动驾驶车辆运动规划的计算最优控制方法研究[D]. 浙江大学, 2018.
- [2] 赫孝良, 葛照强. 最优化与最优控制[M]. 西安: 西安交通大学出版社, 2009.
- [3] 李明辉. 基于蒙特卡洛法的水下对抗效能推演评估研究[D]. 中国舰船研究院, 2014.
- [4] 李明辉, 闵绍荣, 谢红胜. 基于蒙特卡洛法的舰艇规避鱼雷效能推演评估[J]. 舰船电子工程, 2014, 34(02):136-140+170.
- [5] P. D. Moral, A. Douee, A. Jasra. Sequential Monte Carlo Samplers[J]. Royal statistical Society, 2006, 68(3): 411-436.
- [6] Lowe R, Wu Y, Tamar A, et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments[C]. neural information processing systems, 2017: 6379-6390.
- [7] 孔寒雪. 反舰导弹攻防对抗技术研究[D]. 北京理工大学, 2016.
- [8] Zhu X . An Optimal Control View of Adversarial Machine Learning[J]. 2018.
- [9] Green P L, Moore R E, Jackson R J, et al. Increasing the efficiency of Sequential Monte Carlo samplers through the use of approximately optimal L-kernels[J]. arXiv preprint arXiv:2004.12838, 2020..

## 附 录

```
1. %% 《无人机集群协同对抗》实现源码
2. %% main 主函数
3. clc;clear; close all;
4. % 基础参数设置 注:为方便计算与作图, 本文采用 Km,Km/s,s 作为距离,速度,时间的度量单位
5. global params_ bandwidth
6. % 基于贪婪队形的红方协作拦截策略 队形(A)/(B)/(C)
7. Formation_Red = 0;          % (A):0 / (B):1 / (C):2
8. if Formation_Red == 0
9.     params_.maxwidth = 0.87528;
10. elseif Formation_Red == 1
11.     params_.maxwidth = 0.94641;
12. elseif Formation_Red == 2
13.     params_.maxwidth = 1.0000;
14. end
15. params_.vE_max = 0.25;
16. params_.vP_max = 0.20;
17. params_.vT_max = 0.30;
18. params_.a_max = 0;
19. params_.phy_max = 0;
20. params_.w_max = 0;
21. params_.x_min = 0;
22. params_.x_max = 50;
23. params_.y_min = 0;
24. params_.y_max = 70;
25. params_.x_scale = params_.x_max - params_.x_min;
26. params_.y_scale = params_.y_max - params_.y_min;
27. params_.tf = 1;
28. params_.num_nodes_x = 1500;
29. params_.num_nodes_y = 1500;
30. params_.num_nodes_t = 200;
31. params_.resolution_t = params_.tf / (params_.num_nodes_t - 1);
32. params_.resolution_x = params_.x_scale / (params_.num_nodes_x - 1);
33. params_.resolution_y = params_.y_scale / (params_.num_nodes_y - 1);
34. params_.multiplier_for_H = 2.0;
35. params_.weight_for_time = 5.0;
36. params_.max_cycle = 1000;
37. params_.n_large_value = 10000000;
38. params_.Rcos_ang = params_.radius .* cos(linspace(0, 2 * pi, 30));
39. params_.Rsin_ang = params_.radius .* sin(linspace(0, 2 * pi, 30));
40. params_.Nfe = 100;
41. params_.norm_L_threshold = 1;
```

```

42. params_.L_threshold = params_.norm_L_threshold;
43. params_.alpha = 0.5;
44. params_.Iter_max = 10;
45. bandwidth = 100;
46. num_cases = 5000;
47. success_flag = zeros(1, num_cases);
48. cpu_time = zeros(1, num_cases);
49. xv_library = 50*rand(1,num_cases);
50. yv_library = 70*rand(1,num_cases);
51. theta0_library = -1.6+(3.2)*rand(1,num_cases);
52. yvtf_library = abs(50*rand(1,num_cases));
53. thetatf_library = -1.6+(3.2)*rand(1,num_cases);
54.
55. for case_id = 1:num_cases
56.     clc;
57.     str=['Iterations = ' num2str(case_id)];
58.     disp(str);
59.     str1=['Success flag = ' num2str(success_flag)];
60.     disp(str1);
61.     case_str = ['BenchmarkCases\' , num2str(1)];
62.     global boundary_configs_ obstacles_ original_obstacle_layers_
63.     load(case_str);
64.     tic;
65.     WriteBoundaryValuesAndBasicParams(Nv, Nobs);
66.     [init_x, init_y, init_theta, init_v, init_a, init_phy, init_w] = GenerateInitialGuess(Nv);
67.     iter = 0; fail_flag = 0;
68.     while (~IsCollisionFree(init_x, init_y, init_theta))
69.         if (iter > params_.Iter_max)
70.             fail_flag = 1;
71.             break;
72.         end
73.         WriteObstaclesForReducedNLP(init_x, init_y, init_theta, Nv);
74.         !ampl rr.run
75.         load opti_flag.txt
76.         if (~opti_flag)
77.             params_.L_threshold = params_.L_threshold * params_.alpha;
78.         else
79.             [init_x, init_y, init_theta, init_phy] = UpdateInitialGuess(Nv);
80.             params_.L_threshold = params_.norm_L_threshold;
81.         end
82.         iter = iter + 1;
83.     end
84.     cpu_time(case_id) = toc;

```



```

85.     if (~fail_flag)
86.         success_flag(case_id) = 1;
87.         GenerateVideo_task04(init_x, init_y, init_theta, init_phy);
88.     else
89.         success_flag(case_id) = 0;
90.     end
91. end

```

```

1. %% WriteObstaclesForReducedNLP.m
2. function num_constraints= WriteObstaclesForReducedNLP( x, y, theta, Nv)
3. global params_
4. Nfe= params_. Nfe;
5. index= ones( Nfe, Nv- 1, Nv- 1).*- 99;
6. num_conflict_vehicles= ones( Nfe, Nv- 1).*- 99;
7. num_v2v_constraints= 0;
8. for ii= 1: Nfe
9.     for jj= 1:( Nv- 1)
10.        counter= 0;
11.        for kk=( jj+ 1): Nv
12.            if (~ IsConfig1FarFromConfig2([ x( jj, ii), y( jj, ii), theta( jj, ii)], [ x
( kk, ii), y( kk, ii), theta( kk, ii)]))
13.                counter= counter+ 1;
14.                num_v2v_constraints= num_v2v_constraints+ 1;
15.                index( ii, jj, counter)= kk;
16.            end
17.        end
18.        num_conflict_vehicles( ii, jj)= counter;
19.    end
20. end
21.
22. delete( 'IND_V2V');
23. fid= fopen( 'IND_V2V', 'w');
24. for ii= 1: Nfe
25.     for jj= 1:( Nv- 1)
26.         for kk= 1:( Nv- 1)
27.             fprintf( fid, '%g %g %g %g\r\n', ii, jj, kk, index( ii, jj, kk));
28.         end
29.     end
30. end
31. fclose( fid);
32.
33. delete( 'NUM_IND_V2V');

```

```

34. fid= fopen( 'NUM_IND_V2V', 'w');
35. for ii= 1: Nfe
36.     for jj= 1:( Nv- 1)
37.         fprintf( fid, '%g %g %g\r\n', ii, jj, num_conflict_vehicles( ii, jj));
38.     end
39. end
40. fclose( fid);
41.
42. global obstacles_
43. Nobs= size( obstacles_, 2);
44. Precision= length( obstacles_{ 1, 1});
45. index_list= round( linspace( 1, Precision, Nfe));
46. delete( 'OBS_XY');
47. fid= fopen( 'OBS_XY', 'w');
48. for ii= 1: Nobs
49.     elem= obstacles_{ 1, ii};
50.     for jj= 1: Nfe
51.         frame= elem{ 1, index_list( jj)};
52.         fprintf( fid, '%g %g 1 %f\r\n', ii, jj, frame. x);
53.         fprintf( fid, '%g %g 2 %f\r\n', ii, jj, frame. y);
54.         fprintf( fid, '%g %g 3 %f\r\n', ii, jj, frame. radius);
55.     end
56. end
57. fclose( fid);
58.
59. index= ones( Nfe, Nv, Nobs).*- 99;
60. num_conflict_obstacles= ones( Nfe, Nv).*- 99;
61. num_v2o_constraints= 0;
62. for ii= 1: Nfe
63.     for jj= 1: Nv
64.         counter= 0;
65.         for kk= 1: Nobs
66.             obs= obstacles_{ 1, kk};
67.             frame= obs{ 1, index_list( ii)};
68.             if (~ IsConfigFarFromObstacle([ x( jj, ii), y( jj, ii), theta( jj, ii)], frame. x, frame. y, frame. radius))
69.                 counter= counter+ 1;
70.                 num_v2o_constraints= num_v2o_constraints+ 1;
71.                 index( ii, jj, counter)= kk;
72.             end
73.         end
74.         num_conflict_obstacles( ii, jj)= counter;
75.     end
76. end

```

```

77.
78. delete( 'IND_V20');
79. fid= fopen( 'IND_V20', 'w');
80. for ii= 1: Nfe
81.     for jj= 1: Nv
82.         for kk= 1: Nobs
83.             fprintf( fid, '%g %g %g %g\r\n', ii, jj, kk, index( ii, jj, kk));
84.         end
85.     end
86. end
87. fclose( fid);
88.
89. delete( 'NUM_IND_V20');
90. fid= fopen( 'NUM_IND_V20', 'w');
91. for ii= 1: Nfe
92.     for jj= 1: Nv
93.         fprintf( fid, '%g %g %g\r\n', ii, jj, num_conflict_obstacles( ii, jj));
94.     end
95. end
96. fclose( fid);
97.
98. num_constraints= 4* num_v2v_constraints+ 2* num_v2o_constraints;
99. end
100.
101. function is_distant= IsConfig1FarFromConfig2( vec1, vec2)
102. global params_
103. xr=[ vec1( 1), vec2( 1)]+ params_. r2p* cos([ vec1( 3), vec2( 3)]);
104. yr=[ vec1( 2), vec2( 2)]+ params_. r2p* sin([ vec1( 3), vec2( 3)]);
105. xf=[ vec1( 1), vec2( 1)]+ params_. f2p* cos([ vec1( 3), vec2( 3)]);
106. yf=[ vec1( 2), vec2( 2)]+ params_. f2p* sin([ vec1( 3), vec2( 3)]);
107. P11=[ xr( 1), yr( 1)];
108. P12=[ xf( 1), yf( 1)];
109. P21=[ xr( 2), yr( 2)];
110. P22=[ xf( 2), yf( 2)];
111. distance= min([ norm( P11- P21), norm( P11- P22), norm( P12- P21), norm( P12- P22)])-
    2* params_. radius;
112.
113. if ( distance> params_. L_threshold)
114.     is_distant= 1;
115. else
116.     is_distant= 0;
117. end
118. end
119.

```

```

120. function is_distant= IsConfigFarFromObstacle( vec, obs_x, obs_y, obs_radius)
121. global params_
122. xr= vec( 1)+ params_. r2p* cos( vec( 3));
123. yr= vec( 2)+ params_. r2p* sin( vec( 3));
124. xf= vec( 1)+ params_. f2p* cos( vec( 3));
125. yf= vec( 2)+ params_. f2p* sin( vec( 3));
126. P1=[ xr, yr];
127. P2=[ xf, yf];
128. P0=[ obs_x, obs_y];
129. distance= min([ norm( P1- P0), norm( P2- P0)])- params_. radius- abs( obs_radius);
130. if ( distance> params_. L_threshold)
131.     is_distant= 1;
132. else
133.     is_distant= 0;
134. end
135. end

```

```

1. function WriteBoundaryValuesAndBasicParams(Nv, Nobs)
2. global params_ boundary_configs_
3.
4. warning off
5. delete('BV');
6. fid = fopen('BV', 'w');
7. for ii = 1 : size(boundary_configs_,2)
8.     theta0 = boundary_configs_{1,ii}.theta0;
9.     thetatf = boundary_configs_{1,ii}.thetatf;
10.    while (thetatf - theta0 > pi + 0.01)
11.        thetatf = thetatf - 2 * pi;
12.    end
13.    while (thetatf - theta0 < -pi - 0.01)
14.        thetatf = thetatf + 2 * pi;
15.    end
16.    fprintf(fid, '%g 1 %f\r\n', ii, boundary_configs_{1,ii}.x0);
17.    fprintf(fid, '%g 2 %f\r\n', ii, boundary_configs_{1,ii}.y0);
18.    fprintf(fid, '%g 3 %f\r\n', ii, boundary_configs_{1,ii}.theta0);
19.    fprintf(fid, '%g 4 %f\r\n', ii, boundary_configs_{1,ii}.xtf);
20.    fprintf(fid, '%g 5 %f\r\n', ii, boundary_configs_{1,ii}.ytf);
21.    fprintf(fid, '%g 6 %f\r\n', ii, thetatf);
22. end
23. fclose(fid);
24.
25. delete('BP');
26. fid = fopen('BP', 'w');

```

```

27. fprintf(fid, '1 %f\r\n', params_.wheelbase);
28. fprintf(fid, '2 %f\r\n', params_.radius);
29. fprintf(fid, '3 %f\r\n', params_.r2p);
30. fprintf(fid, '4 %f\r\n', params_.f2p);
31.
32. fprintf(fid, '5 %f\r\n', params_.v_max);
33. fprintf(fid, '6 %f\r\n', params_.a_max);
34. fprintf(fid, '7 %f\r\n', params_.phy_max);
35. fprintf(fid, '8 %f\r\n', params_.w_max);
36.
37. fprintf(fid, '9 %f\r\n', params_.tf);
38. fprintf(fid, '10 %g\r\n', Nv);
39. fprintf(fid, '11 %g\r\n', params_.Nfe);
40. fprintf(fid, '12 %g\r\n', Nobs);
41. fclose(fid);
42. end

```

```

1. %% UpdateInitialGuess.m
2. function [init_x, init_y, init_theta, init_phy] = UpdateInitialGuess(Nv)
3. global params_
4.
5. load x.txt; load y.txt; load theta.txt; load phy.txt;
6. init_x = reshape(x', params_.Nfe, Nv)';
7. init_y = reshape(y', params_.Nfe, Nv)';
8. init_theta = reshape(theta', params_.Nfe, Nv)';
9. init_phy = reshape(phy', params_.Nfe, Nv)';
10. end

```

```

1. %% UpdateObstacleLayers
2. function UpdateObstacleLayers(x, y, theta)
3. global original_obstacle_layers_ vehicle_geometrics_ xyt_graph_search_
4. xr = x + vehicle_geometrics_.r2p * cos(theta);
5. yr = y + vehicle_geometrics_.r2p * sin(theta);
6. xf = x + vehicle_geometrics_.f2p * cos(theta);
7. yf = y + vehicle_geometrics_.f2p * sin(theta);
8.
9. for ii = 1 : length(x)
10.     x = xr(ii) + xyt_graph_search_.Rcos_ang;
11.     y = yr(ii) + xyt_graph_search_.Rsin_ang;
12.     x = [x, xf(ii) + xyt_graph_search_.Rcos_ang];
13.     y = [y, yf(ii) + xyt_graph_search_.Rsin_ang];

```

```

14.    original_obstacle_layers_{1,ii}.x = [original_obstacle_layers_{1,ii}.x, x];
15.    original_obstacle_layers_{1,ii}.y = [original_obstacle_layers_{1,ii}.y, y];
16. end
17. end

```

```

1. NLP.mod
2. param BP{i in {1..12}};
3. param wheelbase == BP[1];
4. param FourSqR == 4 * ((BP[2])^2);
5. param R == BP[2];
6. param r2p == BP[3];
7. param f2p == BP[4];
8. param vmax == BP[5];
9. param amax == BP[6];
10. param phymax == BP[7];
11. param wmax == BP[8];
12. param tf == BP[9];
13. param Nv == BP[10];
14. param Nfe == BP[11];
15. param Nobs == BP[12];
16. param hi = tf / Nfe;
17.
18. param BV{i in {1..Nv}, j in {1..6}};
19. param IND_V2V{i in {1..Nfe}, j in {1..(Nv-1)}, k in {1..(Nv-1)}};
20. param NUM_IND_V2V{i in {1..Nfe}, j in {1..(Nv-1)}};
21. param OBS_XY{i in {1..Nobs}, j in {1..Nfe}, k in {1..3}};
22. param IND_V2O{i in {1..Nfe}, j in {1..Nv}, k in {1..Nobs}};
23. param NUM_IND_V2O{i in {1..Nfe}, j in {1..Nv}};
24.
25. var x{i in {1..Nv}, j in {1..Nfe}};
26. var y{i in {1..Nv}, j in {1..Nfe}};
27. var xf{i in {1..Nv}, j in {1..Nfe}};
28. var yf{i in {1..Nv}, j in {1..Nfe}};
29. var xr{i in {1..Nv}, j in {1..Nfe}};
30. var yr{i in {1..Nv}, j in {1..Nfe}};
31. var theta{i in {1..Nv}, j in {1..Nfe}};
32. var v{i in {1..Nv}, j in {1..Nfe}};
33. var a{i in {1..Nv}, j in {1..Nfe}};
34. var phy{i in {1..Nv}, j in {1..Nfe}};
35. var w{i in {1..Nv}, j in {1..Nfe}};
36.
37. minimize cost_function:
38. hi * sum{i in {1..Nv}, j in {1..Nfe}}(d[i,j]^2 + t[i,j]^2);

```

```

39.
40. s.t. Bonds_x {k in {1..Nv}, i in {1..Nfe}}:
41. 0 <= x[k,i] <= 50;
42. s.t. Bonds_y {k in {1..Nv}, i in {1..Nfe}}:
43. 0 <= y[k,i] <= 70;
44. s.t. Bonds_vE {k in {1..Nv}, i in {1..Nfe}}:
45. vE[k,i] = vE_max;
46. s.t. Bonds_vP {k in {1..Nv}, i in {1..Nfe}}:
47. vP[k,i] = vP_max;
48. s.t. Bonds_vT {k in {1..Nv}, i in {1..Nfe}}:
49. vT[k,i] = vT_max;
50. s.t. Bonds_RP {k in {1..Nv}, i in {1..Nfe}}:
51. RP[k,i] >= 0.3;
52. s.t. Bonds_RH {k in {1..Nv}, i in {1..Nfe}}:
53. RH[k,i] >= 1.0;
54. s.t. Bonds_RE {k in {1..Nv}, i in {1..Nfe}}:
55. RE[k,i] >= 0.5;
56. s.t. Bonds_M {k in {1..Nv}, i in {1..Nfe}}:
57. M[k,i] >= 70;
58. s.t. Bonds_tE {k in {1..Nv}, i in {1..Nfe}}:
59. tE[k,i] >= 360;
60. s.t. Bonds_Pkm {k in {1..Nv}, i in {1..Nfe}}:
61. abs(Pkm[k,i]-Pkm[k,i-1])^2 = abs(Pkm[k,i]-Pkm[k,i+1])^2 ;
62. s.t. Bonds_PI {k in {1..Nv}, i in {1..Nfe}}:
63. PI[k,i] >= 0.03
64. s.t. Bonds_HP {k in {1..Nv}, i in {1..Nfe}}:
65. HP[k,i] >= 0.1
66. s.t. Bonds_HE {k in {1..Nv}, i in {1..Nfe}}:
67. HE[k,i] >= 5
68. s.t. Bonds_Pkij {k in {1..Nv}, i in {1..Nfe}}:
69. Pkij[k,i] >= 0.2
70.
71. s.t. DIFF_dxdt {k in {1..Nv}, i in {2..Nfe}}:
72. x[k,i] = x[k,i-1] + hi * v[k,i] * cos(theta[k,i]);
73. s.t. DIFF_dydt {k in {1..Nv}, i in {2..Nfe}}:
74. y[k,i] = y[k,i-1] + hi * v[k,i] * sin(theta[k,i]);
75. s.t. DIFF_dvdt {k in {1..Nv}, i in {2..Nfe}}:
76. v[k,i] = v[k,i-1] + hi * a[k,i];
77. s.t. DIFF_dthetadt {k in {1..Nv}, i in {2..Nfe}}:
78. theta[k,i] = theta[k,i-1] + hi * tan(phy[k,i]) * v[k,i] / wheelbase;
79. s.t. DIFF_dphydt {k in {1..Nv}, i in {2..Nfe}}:
80. phy[k,i] = phy[k,i-1] + hi * w[k,i];
81.
82. s.t. RELATIONSHIP_XF {k in {1..Nv}, i in {1..Nfe}}:

```

```

83. xf[k,i] = x[k,i] + f2p * cos(theta[k,i]);
84. s.t. RELATIONSHIP_YF {k in {1..Nv}, i in {1..Nfe}}:
85. yf[k,i] = y[k,i] + f2p * sin(theta[k,i]);
86. s.t. RELATIONSHIP_XR {k in {1..Nv}, i in {1..Nfe}}:
87. xr[k,i] = x[k,i] + r2p * cos(theta[k,i]);
88. s.t. RELATIONSHIP_YR {k in {1..Nv}, i in {1..Nfe}}:
89. yr[k,i] = y[k,i] + r2p * sin(theta[k,i]);
90.
91.
92. ### Collision Avoidance Type 1 ###
93. s.t. ObsToFrontDics {i in {1..Nfe}, j in {1..Nv}, k in {1..NUM_IND_V20[i,j]}}:
94. (xf[j,i] - OBS_XY[IND_V20[i,j,k],i,1])^2 + (yf[j,i] - OBS_XY[IND_V20[i,j,k],i,2])^2 >=
    (R + OBS_XY[IND_V20[i,j,k],i,3])^2;
95.
96. s.t. ObsToRearDics {i in {1..Nfe}, j in {1..Nv}, k in {1..NUM_IND_V20[i,j]}}:
97. (xr[j,i] - OBS_XY[IND_V20[i,j,k],i,1])^2 + (yr[j,i] - OBS_XY[IND_V20[i,j,k],i,2])^2 >=
    (R + OBS_XY[IND_V20[i,j,k],i,3])^2;
98.
99. ### Collision Avoidance Type 2 ###
100. s.t. VehicleItoJff {i in {1..Nfe}, j in {1..(Nv-1)}, k in {1..NUM_IND_V2V[i,j]}}:
101. (xf[j,i] - xf[IND_V2V[i,j,k],i])^2 + (yf[j,i] - yf[IND_V2V[i,j,k],i])^2 >= FourSqR;
102.
103. s.t. VehicleItoJrr {i in {1..Nfe}, j in {1..(Nv-1)}, k in {1..NUM_IND_V2V[i,j]}}:
104. (xr[j,i] - xr[IND_V2V[i,j,k],i])^2 + (yr[j,i] - yr[IND_V2V[i,j,k],i])^2 >= FourSqR;
105.
106. s.t. VehicleItoJfr {i in {1..Nfe}, j in {1..(Nv-1)}, k in {1..NUM_IND_V2V[i,j]}}:
107. (xf[j,i] - xr[IND_V2V[i,j,k],i])^2 + (yf[j,i] - yr[IND_V2V[i,j,k],i])^2 >= FourSqR;
108.
109. s.t. VehicleItoJrf {i in {1..Nfe}, j in {1..(Nv-1)}, k in {1..NUM_IND_V2V[i,j]}}:
110. (xr[j,i] - xf[IND_V2V[i,j,k],i])^2 + (yr[j,i] - yf[IND_V2V[i,j,k],i])^2 >= FourSqR;
111.
112.
113. s.t. EQ_init_x {m in {1..Nv}}:
114. x[m,1] = BV[m,1];
115. s.t. EQ_init_y {m in {1..Nv}}:
116. y[m,1] = BV[m,2];
117. s.t. EQ_init_theta {m in {1..Nv}}:
118. theta[m,1] = BV[m,3];
119. s.t. EQ_init_v {m in {1..Nv}}:
120. v[m,1] = 0.25;
121. # v[m,1] = 0;
122. s.t. EQ_init_a {m in {1..Nv}}:
123. a[m,1] = 0;
124. s.t. EQ_init_phy {m in {1..Nv}}:

```



```

125. phy[m,1] = 0;
126. s.t. EQ_init_w {m in {1..Nv}}:
127. w[m,1] = 0;
128. s.t. EQ_end_x {m in {1..Nv}}:
129. x[m,Nfe] = BV[m,4];
130. s.t. EQ_end_y {m in {1..Nv}}:
131. y[m,Nfe] = BV[m,5];
132. s.t. EQ_end_theta {m in {1..Nv}}:
133. tan(theta[m,Nfe]) = tan(BV[m,6]);
134. #theta[m,Nfe] = BV[m,6];
135.
136. s.t. EQ_end_v {m in {1..Nv}}:
137. v[m,Nfe] = 0;
138. # v[m,Nfe] = 0.25;
139. s.t. EQ_end_a {m in {1..Nv}}:
140. a[m,Nfe] = 0;
141. s.t. EQ_end_phy {m in {1..Nv}}:
142. phy[m,Nfe] = 0;
143. s.t. EQ_end_w {m in {1..Nv}}:
144. w[m,Nfe] = 0;
145.
146. data;
147. param: BV := include BV;
148. param: BP := include BP;
149. param: IND_V2V := include IND_V2V;
150. param: NUM_IND_V2V := include NUM_IND_V2V;
151. param: OBS_XY := include OBS_XY;
152. param: IND_V20 := include IND_V20;
153. param: NUM_IND_V20 := include NUM_IND_V20;

```

```

1. runNLP.run
2. reset;
3. model NLP.mod;
4. include IG.INIVAL;
5. option solver ipopt;
6.
7. shell ("del " & ("opti_flag.txt"));
8. printf "0">>("opti_flag.txt");
9. close ("opti_flag.txt");
10.
11. solve;
12.
13. shell ("del " & ("opti_flag.txt"));

```

```

14. if match (solve_message, "Optimal") > 0 then {
15.     printf "1">>("opti_flag.txt");
16. } else{
17.     printf "0">>("opti_flag.txt");
18. }
19. close ("opti_flag.txt");
20.
21. shell ("del " & ("x.txt"));
22. shell ("del " & ("y.txt"));
23. shell ("del " & ("theta.txt"));
24. shell ("del " & ("phy.txt"));
25.
26. for {i in {1..Nv}} {
27.     for {j in {1..Nfe}} {
28.         printf "%5.4f;", x[i,j]>>("x.txt");
29.         printf "%5.4f;", y[i,j]>>("y.txt");
30.         printf "%5.4f;", theta[i,j]>>("theta.txt");
31.         printf "%5.4f;", phy[i,j]>>("phy.txt");
32.     }
33. }
34. close ("x.txt");
35. close ("y.txt");
36. close ("theta.txt");
37. close ("phy.txt");
38.
39. if match (solve_message, "Optimal") > 0 then {
40.     shell ("del " & ("IG.INIVAL"));
41.     for {k in {1..Nv}} {
42.         for {i in {1..Nfe}} {
43.             printf "let x[%d,%d]:=5.3f;\n", k, i, x[k,i]>>("IG.INIVAL");
44.             printf "let y[%d,%d]:=5.3f;\n", k, i, y[k,i]>>("IG.INIVAL");
45.             printf "let xr[%d,%d]:=5.3f;\n", k, i, xr[k,i]>>("IG.INIVAL");
46.             printf "let yr[%d,%d]:=5.3f;\n", k, i, yr[k,i]>>("IG.INIVAL");
47.             printf "let xf[%d,%d]:=5.3f;\n", k, i, xf[k,i]>>("IG.INIVAL");
48.             printf "let yf[%d,%d]:=5.3f;\n", k, i, yf[k,i]>>("IG.INIVAL");
49.             printf "let theta[%d,%d]:=5.3f;\n", k, i, theta[k,i]>>("IG.INIVAL");
50.             printf "let v[%d,%d]:=5.3f;\n", k, i, v[k,i]>>("IG.INIVAL");
51.             printf "let a[%d,%d]:=5.3f;\n", k, i, a[k,i]>>("IG.INIVAL");
52.             printf "let phy[%d,%d]:=5.3f;\n", k, i, phy[k,i]>>("IG.INIVAL");
53.             printf "let w[%d,%d]:=5.3f;\n", k, i, w[k,i]>>("IG.INIVAL");
54.         }
55.     }
56. }

```