“华为杯”第十五届中国研究生

数学建模竞赛

学　　校		华东师范大学
参赛队号		**18102690044**
队员姓名	**1.**	黄可蒙
	2.	吴柏威
	3.	董天文

“华为杯”第十五届中国研究生

数学建模竞赛

题 目 多无人机对组网雷达的协同干扰

摘 要：

在军事中，组网雷达系统具有较强的抗干扰能力，利用多无人机协同干扰，可以对其产生理想的干扰效果。本文就多无人机在协同干扰中的应用进行研究。

针对问题一，研究无人机的可能所在位置节点后发现，这些节点必须位于五个雷达与虚假目标的连上，这些连线与 2000m、2500m 的平面截得 20 个五棱柱，每一个五棱柱的棱需要有至少三架无人机干扰。无人机做匀速直线运动，结合速度、等时到达、直线飞行等约束，以无人机数量最少为目标函数建立优化模型。采用禁忌搜索算法遍历 2000m 到 2500m 的空间，得到无人机的最少架数为 **31 架**。其中，29 架无人机负责干扰编号 2~4 的雷达，另外 2 架无人机分别负责干扰 1 号雷达和 5 号雷达。利用 31 架无人机航迹对虚假目标点进行重构，与附件 1 给出的虚假目标点进行对比，验证了模型的可靠性。

针对问题二，分析无人机机动飞行的特点，基于一架无人机最多产生 7 个虚假目标的要求，综合考虑最大转弯半径、最大加速度、无人机速度范围等约束条件，建立以使用无人机架数最少为目标函数的优化模型，求解得到实现附件 1 的虚假目标轨迹的最优结果为 **3 架**无人机，干扰的雷达编号分别为 1 号、3 号和 4 号。其余 6 架无人机在 5 分钟飞行时间的限制下进行轨迹搜索，得到 3 条虚假

轨迹。结合几何关系推算后，得到另一条虚假航迹。最终得到结论：利用 9 架无人机机动飞行，完成附件 1 要求的虚假航迹的同时至多还可产生**4 条**虚假的航迹。对模型进行了重构检验与运动合理性分析，验证了模型的可靠性。

针对问题三，针对组网雷达系统能被干扰的情况，分析无人机航迹维持策略，建立了问题二约束条件下的协同飞行轨迹补偿模型。一条航线上最多允许三个不连续的空缺目标点。问题二生成的 5 条航线中有 3 条航线可以被调度产生 9 个空缺目标点，调度提供给新航迹，加上自身的 3 个空缺目标点共形成 12 个目标点。在剩余的 100s 时间内还可以形成 10 个目标点，一共有 22 个目标点作为新航线备选。搜索这些点中连续的 20 个点组成完整轨迹，求解得到一条虚假航迹。加上问题二的 4 条虚假航迹，9 架无人机组成的编队在 5 分钟内，完成附件 1 要求的虚假航迹的同时，至多还可产生 **5 条**虚假的航迹。

最后对模型的鲁棒性和灵敏度进行了分析，验证了模型的可靠性，并对模型进行了推广。本文基于机理分析的无人机协同调度策略能够为电子对抗领域提供一定的参考。

关键词：多无人机协同　禁忌搜索算法　路径规划　雷达干扰

目录

一 问题重述

1.1 问题背景

组网雷达系统是应用两部或两部以上空间位置互相分离而覆盖范围互相重叠的雷达的观测或判断来实施搜索、跟踪和识别目标的系统，如何对组网雷达实施行之有效的干扰，是当今电子对抗界面临的一个重大问题。

为了能对组网雷达实施有效干扰，现在可利用多架无人机对组网雷达协同干扰。无人机搭载的干扰设备对接收到的雷达信号进行相应处理后转发回对应的雷达，雷达接收到转发回的干扰信号形成目标航迹点信息，传输至组网雷达信息融合中心。由于多无人机的协同飞行，因此在融合中心就会出现多部雷达在统一坐标系的同一空间位置上检测到目标信号，基于一定的融合规则就会判断为一个合理的目标航迹点，多个连续的合理目标航迹点就形成了目标航迹，即实现了一条虚假航迹。通过协同控制无人机的飞行航迹，可在敌方的组网雷达系统中形成一条或多条欺骗干扰航迹，迫使敌方加强空情处置，达到欺骗目的。

某组网雷达系统由 5 部雷达组成，雷达最大作用距离均为 150km，地理位置坐标也相应给出。雷达将检测到的回波信号经过处理后形成航迹点状态信息传输到融合中心，如果至少有 3 部雷达同一时刻解算出的目标空间位置是相同的，融合中心就将其确定为一个合理的航迹点，20 个连续的经融合中心确认的航迹点形成的合理航迹，将被组网雷达系统视为一条真实的目标航迹。

无人机的参数如下：飞行速度 120km/h~180km/h，飞行高度 2000m~2500m，最大加速度不超过 10 m/s^2，安全间距不小于 100m，同一时刻一架无人机只能干扰一部雷达，但可在该部雷达接收机终端（雷达屏幕上）产生多个目标点，这些目标点均位于雷达与无人机连线以及延长线上，距雷达距离超过 150 km 的假目标信息直接被雷达系统删除；同一时刻多架无人机可以干扰同一部雷达。雷达同一时刻接收的多个目标点的状态信息均同时传送到信息融合中心。每架无人机不同时刻可干扰不同雷达。同一条航迹不同时刻的航迹点，可以由组网雷达系统中不同的三部雷达检测确定。

1.2 待解决问题

针对不同的需求，本文依次解决以下无人机协同干扰的几个问题。

问题一

前提：附件 1 给出了一条拟产生的虚假目标航迹数据，该虚假航迹数据包含 20 个时刻的虚假目标位置坐标信息，时间间隔为 10 秒。

条件：限定每架无人机在该空域均做匀速直线运动，航向、航速和飞行高度可在允许范围内根据需要确定。

目标: 以最少数量的无人机实现附件 1 要求的虚假目标航迹。

问题二

前提：对雷达实施有源假目标欺骗干扰时，干扰设备可同时转发多个假目标信息（本赛题限定每一架无人机同一时刻至多产生 7 个假目标信息），但它们均存在于雷达与无人机连线以及延长线上，延迟（或导前）的时间可根据实际需要确定。

条件：该组网雷达系统的每一部雷达的数据更新率为 10 秒，为控制方便无人机尽可能少做转弯、爬升、俯冲等机动动作，转弯半径不小于 250m。

目标: 讨论由 9 架无人机组成的编队在 5 分钟内，完成附件 1 要求的虚假航迹的同时，至多还可产生出多少条虚假的航迹。

问题三

前提：当组网雷达系统中的某部雷达受到压制干扰或其它因素的干扰时，可能在某些时刻无法正常获取回波信号，此时组网雷达系统信息融合中心可以采用下面的航迹维持策略：若之前与受干扰的雷达联合检测到目标的另 2 部雷达没有受到干扰，正常检测到回波信号，那么在融合中心就对这两部雷达检测的目标航迹点信息进行同源检验 ，若通过亦视为是合理的目标航迹点；若一条航迹中这类航迹点的个数不超过 3 个时（该航迹的其余航迹点仍需通过前面规定的“同源检验”），该航迹就被继续保留。

条件：该组网雷达系统的每一部雷达的数据更新率仍为 10 秒。

目标：针对上述航迹维持策略，协同无人机编队的飞行，有可能产生更多的虚假航迹。重新讨论由 9 架无人机组成的编队在 5 分钟内，完成附件 1 要求的虚假航迹的同时，至多还可产生出多少条虚假的航迹。

二 基本假设和符号说明

2.1 基本假设

为了便于问题的研究，对题目中某些条件进行简化及合理的假设。

- 针对问题一：

假设 1：所有无人机与干扰设备的参数完全相同，忽略制造工艺产生的性能影响，飞行途中不存在控制异常状态，不考虑天气、风速等因素影响。

假设 2：无人机为质点，忽略无人机大小，不考虑无人机爬升与下降。

假设 3：无人机的干扰信号全部成功被雷达接收，组网雷达系统不存在其他因素干扰，且每个雷达能接受来自雷达上方 150km 内所有方位的信号。

假设 4：无人机能做标准的匀速直线运动，忽略微小波动与曲线位移。

假设 5：无人机任务开始不开启干扰设备，到达指定目的点控制开启干扰设备且无延迟。

假设 6：允许雷达在对无人机的显程测量距离上有 1%[1]的测量误差（IEEE 标准），也即在 1%范围内，无人机的相对直线的微小偏离近似作为直线考虑（由于无人机速度为 120km/h~180km/h，10s 的测量会带来 3.33 ~ 5.00 米的误差，为了方便计算，将最大误差计为 5 米）。

- 针对问题二：

假设 7：无人机可以随时以大于 250 米的转弯半径转弯，转弯半径可以实时变化，不考虑其工程实现难度。

假设 8：无人机转发回对应雷达的假目标信息能及时获取，不考虑延迟。

- 针对问题三：

假设 9：某雷达受到干扰后，其余雷达能及时做出相应，不存在延迟。

假设 10：在一个时间点，至少有一台雷达正常工作，即不存在同一个时间 5 台雷达都受到干扰而无法正常获取回波信号的情况。

2.2 符号说明

符号	符号说明
P_i	2000 米与 2500 米平面截连线得到的第 i 个五棱柱
P_{ij}	第 i 个五棱柱的第 j 条棱
$E_{P_{ij}}$	0-1 变量，在 P_{ij} 上存在无人机则为 1，否则为 0
φ	误差参数
$r(t)$	无人机在 t 时刻的转弯半径
$\alpha(t)$	无人机在 t 时刻的爬升或俯冲角度
a_τ	无人机切向加速度
a_n	无人机法向加速度
$a(t)$	无人机在 t 时刻的加速度
T_i	第 i 个虚假目标
V_i	第 i 架无人机

三 问题 1：匀速直线飞行协同干扰模型

3.1 问题分析

从题中给出的雷达方位与问题 1 给出的拟产生的虚假目标航迹数据，做出五个雷达的位置与虚假目标航迹如图 1 所示。

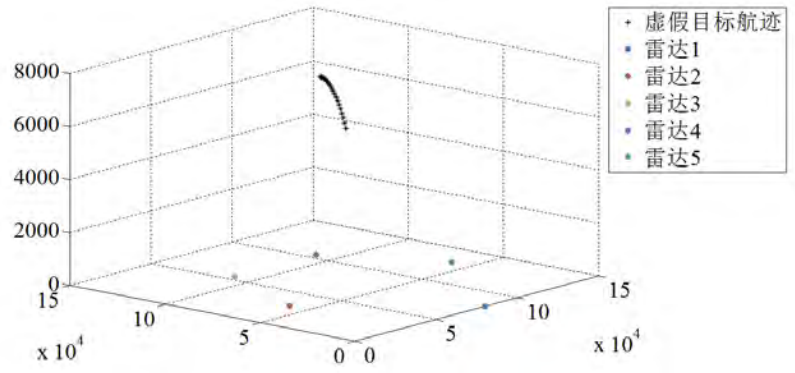


图 1　5 个雷达位置与虚假目标轨迹示意图

限定每架无人机在该空域均做匀速直线运动，要求以最少数量的无人机实现上图中的虚假目标航迹。通过分析，此题可以分为三步来解答：

第一步：分析无人机与雷达之间的关系以及无人机飞行的高度限制我们可以得到，无人机的飞行区域为以虚假目标点与五个雷达连线被 2000 米平面与 2500 米平面所截得的五棱柱的五条棱上（图示在模型建立给出）。而无人机飞行的速度不足以从一条棱跨越至另一条棱，因此每台无人机在飞行过程中只能干扰处于所在棱上的雷达。通过给出的 20 个虚假目标坐标，我们可以推算出所有无人机可能的飞行路径。

第二步：根据无人机飞行的速度与已知测量时间（10s）与无人机飞行安全性原则（两架无人机间距不小于 100m），我们可以对路径进行筛选，即如果距离

不足，就采取多架无人机分层飞行的方式。为了尽可能减少无人机数量，在筛选路径时，尽可能减少分层数量，同时在分层时尽可能寻找等距的线段（2000-2500m 以步长为 1m 遍历搜索）。

第三步：根据一系列约束条件与无人机架数最少作为目标函数，建立优化模型。利用智能搜索算法遍历搜索所有可能的飞行路径，得出最优结果。最后根据我们的模型进行虚假目标重构与路径模拟，将重构的目标与题目给的虚假目标坐标进行对比来验证模型的可靠性。

3.2 模型建立

3.2.1 无人机飞行方案概述

根据分析，给出无人机的合理位置示意图如图 2 所示。图中，T 为虚假目标点之一，R_1、R_2、R_3、R_4、R_5为五个雷达，F_1与F_2分别是位于 2000m 和 2500m 的平面。连接T 与$R_1 \sim R_5$，与平面F_1分别交于$A_1 \sim A_5$构成五边形$A_1A_2A_3A_4A_5$，与平面F_2分别交于$B_1 \sim B_5$构成五边形$B_1B_2B_3B_4B_5$，无人机可能在A_1B_1、A_2B_2、A_3B_3、A_3B_3、A_4B_4、A_5B_5五条棱上的任意位置。

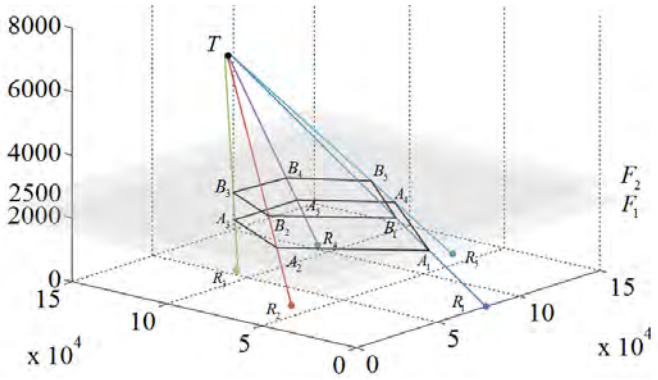


图 2　无人机合理位置示意

无人机的可能飞行方案为从一条棱上飞到另外一条棱上（非同一个五棱柱）。无人机的飞行速度为 120km/h 至 180km/h，10s 的时间无人机能飞行的距离为 333m 至 500m，而经过计算最短的棱间距离B_2B_3为 32629m，因此一架无人机至始至终只负责干扰一台雷达。无人机只能在对应的棱上飞行，不能跨棱飞行。为了更好体现无人机的飞行情况，图 3 的俯视图（无人机从实线位置飞向虚线位置）中给出了两种无人机飞行方案。根据我们的论证，①至②的飞行方式是允许的，①至③的飞行方案不被允许。

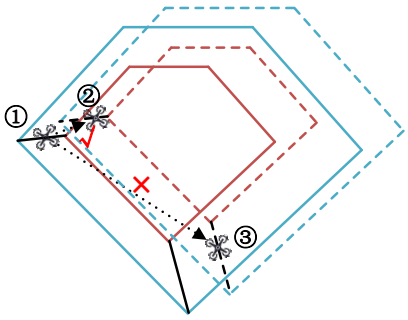
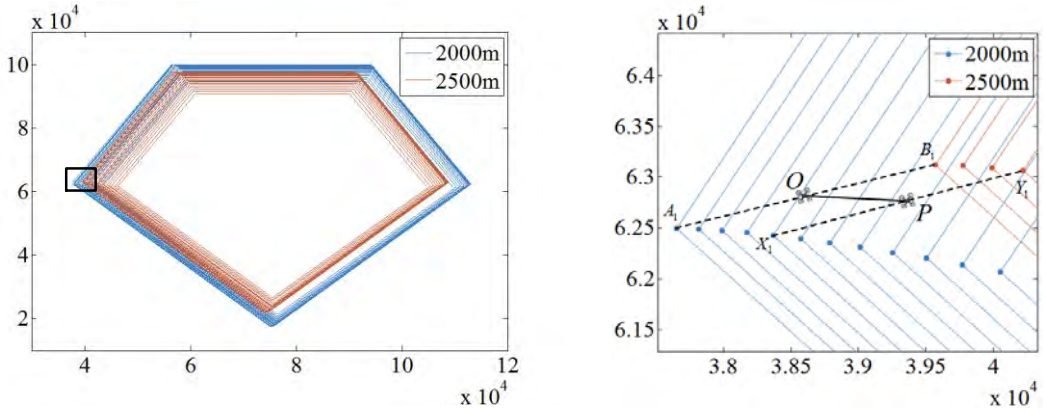


图 3　无人机合理飞行路线示意

根据附件 1 给出的 20 个虚假目标坐标，我们可以做出 20 个五棱柱。为了便于展示，图 4（a）给出的俯视图为未连接棱的双五边形结构。图 4（b）展示了图 4（a）中黑框部分的放大情况。其中 A_1B_1 是对应于第一个虚假目标的无人机可能所在位置，X_1Y_1 是对应于第五个虚假目标的无人机可能所在位置。O 点在棱 A_1B_1 上，P 点在棱 X_1Y_1 上，且 O、P 所在位置的高度相等。OP 展示了该机干扰一个雷达的一次飞行。



（a） （b）

图 4　无人机飞行方案示意图

（a）所有路径示意图 （b）a 中黑框区域放大图

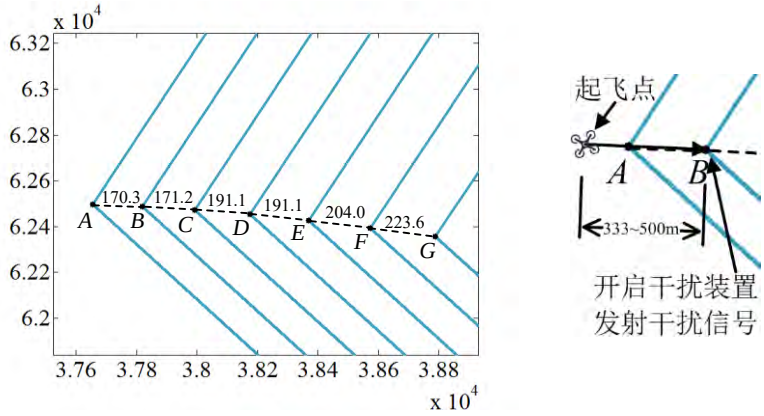
3.2.2 匀速直线飞行优化模型建立

- 存在性约束

根据上一小节的分析，我们一共得到了 $20\times5=100$ 条棱，对于每一个五边形来说，同一时间至少有三条棱上存在无人机才能正确形成虚假目标。设第 i 个虚假目标对应的五棱柱为 P_i，P_i 的棱为 $P_{ij}(j=1,2,3,4,5)$，$E_{P_{ij}}$ 是 $0-1$ 分布的变量，$E_{P_{ij}}=0$ 表示第 i 个五棱柱的第 j 条棱上不存在无人机，反之 $E_{P_{ij}}=1$ 表示该棱上存在无人机。该约束条件可以写成以下形式：

$$\sum_{j=1}^{5} E_{P_{ij}} \geq 3 \quad i=1,2,...,20 \tag{1}$$

- 距离约束



（a） （b）

图 5　距离约束

（a）飞行距离（b）B 点干扰示意

该问基于的条件是无人机做匀速直线运动，由于虚假目标的坐标间隔十秒获取，我们可以根据无人机的速度约束（120km/h~180km/h）计算得到无人机在该间隔内移动的距离为 333m~500m。我们测量了一个雷达在无人机飞行高度为2000m 的相邻五边形点的距离如图 5（a）所示。

由无人机的飞行速度得，一架无人机如果从 A 点出发，经过 10 秒后，只能满足停留在 C 点（ A 到 C 的距离在 333m~500m 区间内）。A、B 点较为特殊，无论哪条棱上的距离都不满足大于 333m 的条件。只有一种可能做到在 A、B 点有三架以上的无人机：无人机从外部飞至 A、B 点，一开始不开启干扰装置，到达 A、B 点开启干扰装置（如图 5（b）所示）。同时，由于无人机间有距离大于 100m 的条件（为了简化搜索，只考虑在工作的无人机间的距离），即需要增加如下约束：

$$\begin{cases} \sqrt{\left(p_{(i+n)j}-p_{ij}\right)^2+\left(H_{i+\lambda}-H_i\right)^2}\geq 100 \\ 2000\leq H\leq 2500 \\ 1\leq i\leq 20, 1\leq j\leq 5 \\ n\geq 1, \lambda\geq 1 \end{cases} \tag{2}$$

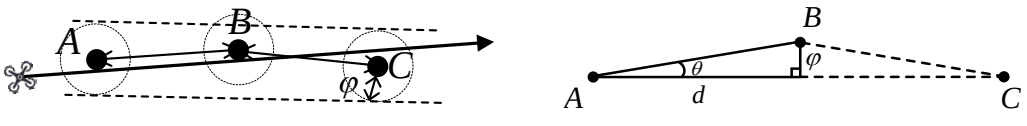
等距离约束的描述如下：

$$\begin{cases} 333\leq p_{(i+n)j}-p_{ij}\leq 500 \\ 333\leq p_{(i+n+m)j}-p_{(i+n)j}\leq 500 \\ \left(p_{(i+n+m)j}-p_{(i+n)j}\right)-\left(p_{(i+n)j}-p_{ij}\right)\leq \varphi \\ 3\leq i\leq 20, 1\leq j\leq 5 \\ n\geq 1, m\geq 1 \end{cases} \tag{3}$$

对于（2）、（3）式，p_{ij}、$p_{(i+n)j}$、$p_{(i+n+m)j}$ 为某相同高度下五边形 i、$i+n$、$i+n+m$ 的第 j 个端点，两两差值为两点间的距离，φ 为雷达误差，由 IEEE 标准得，本题雷达误差允许范围为 $\varphi=5\text{m}$，H_i 为第 i 架无人机所在高度（注：以上约束只针对无人机在这些端点上的情况）。

- 直线约束

无人机做直线运动，因此，如果一架无人机能通过多个五边形端点完成干扰任务，这些端点应该在误差 φ 的允许范围内近似在同一直线上。如图 6（a）所示，在满足距离约束的前提下，如果 B 点所成的误差环在 A、C 误差环切线所形成的区域内，则可以近似认为这几点可以由一架无人机匀速直线飞过形成干扰。为了方便计算，我们将距离转换为斜率表示（如图 6（b）所示）。



（a） （b）

图 6 无人机飞行直线约束

（a）误差环示意（b）距离-斜率转换示意图

考虑最大误差 $\varphi=5$，$AB=333$，$\theta=\arcsin\left(\frac{5}{333}\right)\approx 0.02^\circ$。

也即在直线简化后：如果存在 AB 相对于 AC 所成的角度值在 0.02° 以内的 B

点，可以近似认为 A、B、C 在同一直线上。该约束条件的公式表示为：

$$\theta=\arccos\left(\frac{\left(p_{(i+n+m)j}-p_{(i+n)j}\right)^2+\left(p_{(i+n)j}-p_{ij}\right)^2-\left(p_{(i+n+m)j}-p_{ij}\right)^2}{2\left(p_{(i+n+m)j}-p_{(i+n)j}\right)\left(p_{(i+n)j}-p_{ij}\right)}\right)\le 0.02 \qquad (4)$$

- 优化模型

题中所描述的目标是在满足以上所有约束条件的前提下，尽可能用少的无人机完成欺骗任务，设无人机的数量为 N，也即目标函数与约束条件如下：

$$\text{Obj: } \min(N)$$

$$\text{s.t.}\begin{cases}\sum_{j=1}^{5}E_{p_{ij}}\ge 3 \quad i=1,2,...,20\\ 333\le p_{(i+n)j}-p_{ij}\le 500\\ 333\le p_{(i+n+m)j}-p_{(i+n)j}\le 500\\ \left(p_{(i+n+m)j}-p_{(i+n)j}\right)-\left(p_{(i+n)j}-p_{ij}\right)\le\varphi\\ \sqrt{\left(p_{(i+n)j}-p_{ij}\right)^2+\left(H_{i+\lambda}-H_i\right)^2}\ge 100\\ \arccos\left(\frac{\left(p_{(i+n+m)j}-p_{(i+n)j}\right)^2+\left(p_{(i+n)j}-p_{ij}\right)^2-\left(p_{(i+n+m)j}-p_{ij}\right)^2}{2\left(p_{(i+n+m)j}-p_{(i+n)j}\right)\left(p_{(i+n)j}-p_{ij}\right)}\right)\le 0.02\\ 2000\le H_i\le 2500\\ 1\le i\le 20, 1\le j\le 5\\ n\ge 1, m\ge 1, \lambda\ge 1\end{cases} \qquad (5)$$

3.3 TS 算法模型求解

由于约束条件复杂，搜索范围较广（20 个五棱柱棱所组成的空间），遍历搜索很有可能陷入局部最优或迂回搜索，本文采用禁忌搜索（Tabu Search 或 Taboo Search，简称 TS）算法来获得全局最优情况。

- TS 算法[1]

TS 算法在局部搜索的过程中引进了贪心选择机制，并利用禁忌表修改邻域，通过构造的候选邻域来控制解得选择和接受过程。在搜索的过程中，TS 算法从上一步计算解的候选邻域里选择一个最好的解，即使这个解比上一步得到的解还差，也接受它，同时修改禁忌表，以避免该解在禁忌期限内再次被选择。

简单来说，算法有三步:

STEP 1 给定一个禁忌表 $H=\Phi$,并选定一个初始解 X_{now}；

STEP 2 如果满足停止规则，则停止计算，输出结果；否则，在 X_{now} 的领域 $N(H,X_{now})$ 中选出满足禁忌要求的候选集 $Can_{N(X_{now})}$.在 $Can_{N(X_{now})}$ 中选择一个评价值最佳的解 X_{next}， $X_{now}:=X_{next}$；

STEP 3 更新禁忌表 H，重复 STEP 2。

具体算法流程图如图 7 所示。

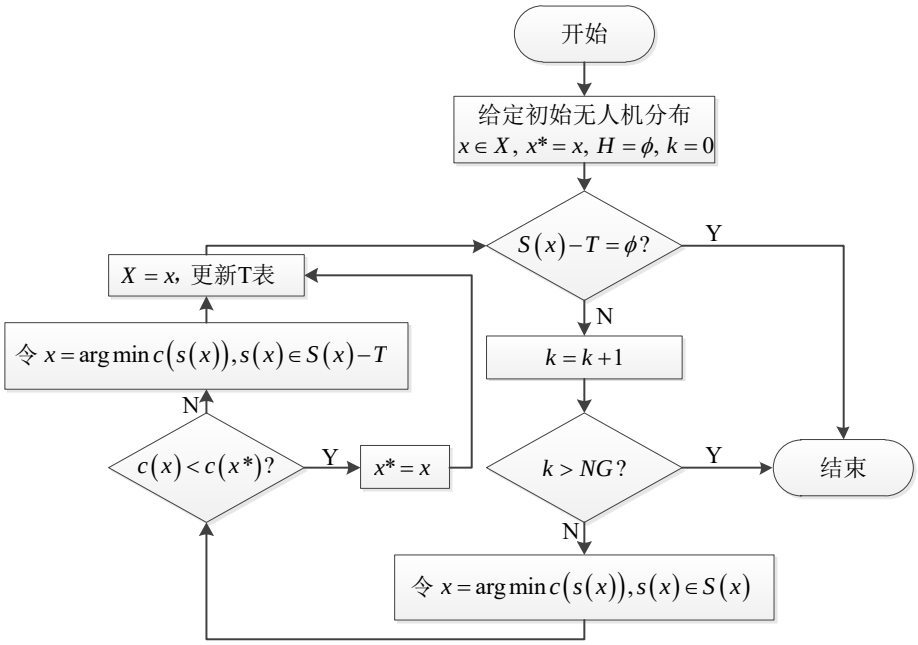


图 7　TS 算法流程图

利用 TS 算法对 2000m～2500m 的空间进行遍历搜索，最终得到的需要的无人机数量为 31 架（很遗憾的是，考虑所有约束后，遍历搜索中没有发现一架无人机在同一个高度能经过三个以上的干扰点，一共有 60 个干扰点，每架无人机最多负责干扰两个节点，其中有两架无人机分别只干扰一个节点），经过我们的分析，这是在水平面匀速直线运动时的最优结果。表 1 给出了部分无人机飞行的结果（详见附件 2）。

表 1 无人机调度情况

无人机编号	（出发）虚假目标编号	（到达）虚假目标编号	干扰雷达编号	出发坐标（m）		到达坐标（m）		飞行高度（m）
				x	y	x	y	z
1	1	20	4	91972	98257	90996	91997	2346
2	1	19	2	39117	62974	44983	61703	2382
3	1	16	3	56743	97547	59865	93635	2489
…	…	…	…	…	…	…	…	…
29	10	11	2	41578	62681	41902	62602	2436
30	10	12	3	57839	98868	58362	98013	2000
31	10	13	4	92071	96285	92001	95622	2464

为了更好地展示结果，我们将 31 架无人机航线路径绘制在同一空间下，如图 8 所示。

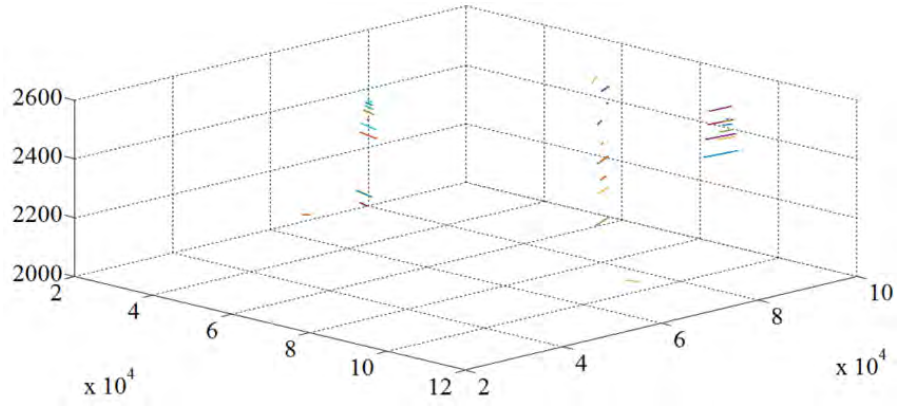


图 8　无人机航线示意图

由图 8 中可以较为明显地看出，无人机主要干扰编号为 2、3、4 号的雷达，只有两架无人机分别干扰了 1 号与 5 号雷达一次，进而形成虚假目标点。

3.4 模型检验与结果分析

根据我们得到的结果与每架无人机每一个时刻干扰的雷达编号，我们可以反推重构出 31 架无人机飞行形成的 20 个虚假目标位置。然后将重构目标点与题中给出的 20 个虚假目标位置进行对比，进而检验模型的可靠性。本文的模型重构虚拟目标轨迹的结果如图 9 所示（下方为五个雷达，上方交点为轨迹点）。

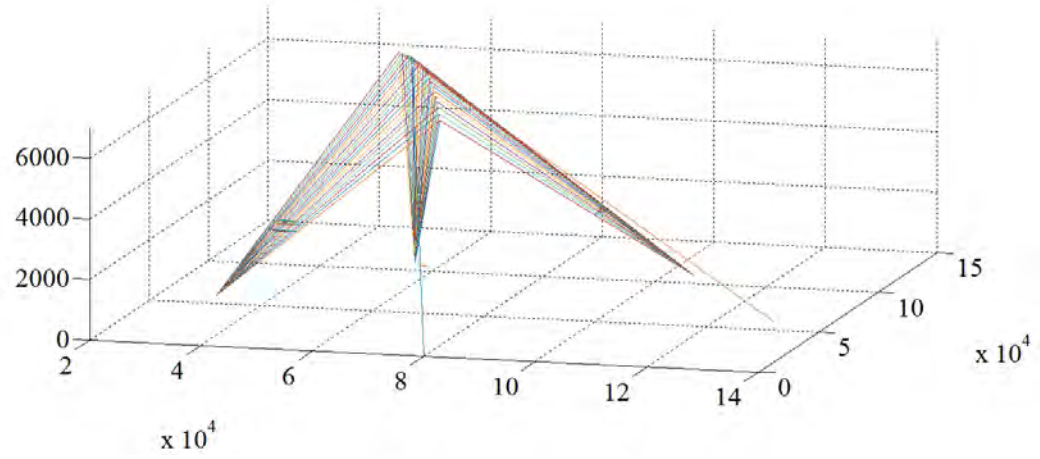
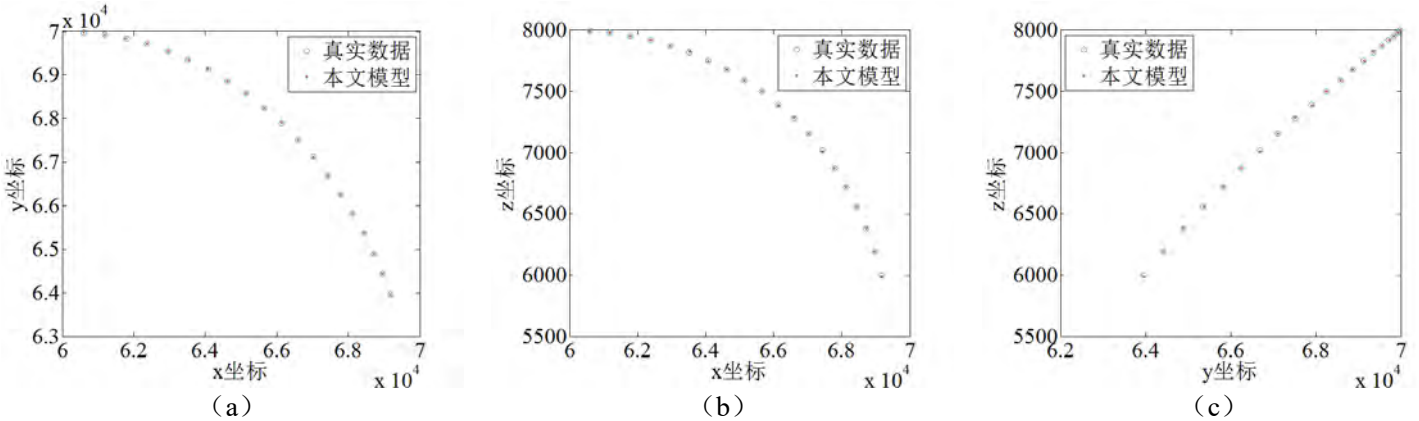


图 9　虚假目标点重构

为了更好地验证本文模型，我们将题目给出的虚假目标与本文模型生成的虚假坐标进行三个二维维度（x-y 投影，x-z 投影，y-z 投影）的对比，结果如图 10（a）、（b）、（c）所示。



（a）　（b）　（c）

图 10　模型检验

（a）x-y 投影检验（b）x-z 投影检验（c）y-z 投影检验

经验证，我们发现本文模型重构的坐标点与真实数据非常接近，误差在合理范围之内，轨迹具有连贯性，这证明了我们模型的可靠性。

四 问题 2：多假目标-机动飞行协同干扰模型

4.1 问题分析

由该问的要求可知，相比于第一问，第二问附加两个条件：

- 无人机可机动飞行
- 每架无人机能产生至多 7 个假目标信息

无人机机动飞行无非增加了几个约束条件，去除了第一问匀速直线运动的约束，但是仍然具有约束，如：无人机尽可能少做转弯、爬升、俯冲等机动动作，转弯半径不小于 250m，最大加速度不超过 $10\,m/s^2$ 等。在下文模型构建时，重点考虑这些约束条件带来的影响。

题中已经限定无人机数量为 9 架，为了获得尽可能多的虚假航迹，我们用尽量少的无人机数量达成目标，剩余的无人机架数越多，自由度越多，虚假航迹也就越多。因此，我们可以建立新约束条件下的优化模型，推算出每架无人机的运动轨迹。在得到无人机飞行情况后，只需将所有无人机与 5 台雷达连线，找出所有连线及它们延长线（不超过 150km）的交点即为所有可能的假目标点，搜索这些点中可能存在的合理路径便能找到其余虚假航迹。

4.2 模型建立

4.2.1 机动飞行优化模型建立

设一台无人机从 0 时刻至 t 时刻的节点序列 $P=\{v_1,v_2,..,v_t\}$，所形成的三维空间 $\mathbb{R}^3$ 下的参数曲线为 $c(t)$，在实际过程中，t 是 $\mathbb{R}^3$ 上连续变化的参数。令无人机每个时间点所在的的方位三维坐标为 (x,y,z)。

- 转弯半径约束

无人机的转弯半径与其横向加速度成反比[3]，其三维环境下的数学表达式为：

$$r(t)=\frac{\left|\dot{c}(t)\right|^3}{\left|\dot{c}(t)\times\ddot{c}(t)\right|} \tag{6}$$

由于惯性作用，无人机在改变飞行方向时，需要一定的空间与时间，因此，为了无人机能正常完成转弯动作，题中给出了无人机最小转弯半径 $r(t)_{\min}=250$ m，$r(t)$ 需要满足 $r(t)\geq r(t)_{\min}$。

- 爬升、俯冲角度约束

无人机的最大爬升（俯冲）角度受多种因素影响，如：转向速率、飞行高度、无人机翼面结构等。过大的爬升角度容易导致无人机失速，因此题目要求尽可能少做爬升、俯冲等机动动作，减少失速的可能与控制难度。三维环境下，t 时刻无人机的爬升（俯冲）角度 $\alpha(t)$ 的数学表达式如下：

$$\alpha(t)=\tan^{-1}\left(\frac{\dot{z}(t)}{\sqrt{\dot{x}(t)^2+\dot{y}(t)^2}}\right) \tag{7}$$

相关研究[4]表明，较为安全且容易控制的爬升俯冲角度为与水平面夹角在 $\pm10^{\circ}$ 之间，设无人机飞行最大的爬升角度为 $\alpha(t)_{\max}$，最大的俯冲角度为 $\alpha(t)_{\min}$，$\alpha(t)$ 需要满足条件 $\alpha(t)_{\min}\leq\alpha(t)\leq\alpha(t)_{\max}$。

无人机爬升或者俯冲的目的有两种：

1）微调两点间路程，这与转弯的目的基本一致，如图 11（a）所示；

2）跨层飞行，飞至不同水平面的某节点，可以选择直线或曲线飞行，如图 11（b）所示。

（a）　　（b）

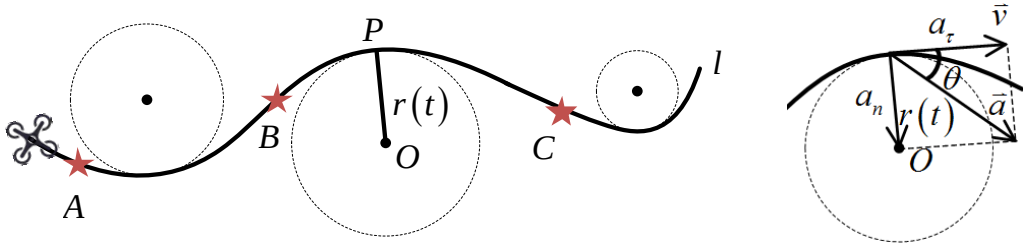
图 11　无人机爬升（俯冲）飞行示意图（x-z 轴或 y-z 轴方向投影）

（a）微调路程飞行示意图（b）跨层飞行示意图

● 加速度约束

无人机作曲线运动时，由假设可得，飞行半径在满足不小于 250m 的前提下可以实时变化。图 12（a）的 l 曲线为一种较为极端的假设飞行路径，一架无人机尝试利用曲线飞行策略通过三个干扰目标点 A、B、C。

我们可以将飞行路径 l 看作许多曲率半径不同的圆弧段组成。曲率中心为 O，曲率为曲率半径的倒数 $1/r(t)$。



（a）　　（b）

图 12　无人机曲线飞行示意图

（a）曲线飞行方式（b）加速度合成

由图 12（b）的加速度合成可以得到，切向加速度 $a_\tau = dv/dt$，法向加速度 $a_n = v^2/r(t)$，总加速度大小表达式为：

$$a = \sqrt{a_\tau^2 + a_n^2} = \sqrt{\left(\frac{dv}{dt}\right)^2 + \left(\frac{v^2}{r(t)}\right)^2} \tag{8}$$

题目规定最大加速度 $a_{\max} = 10m/s^2$，因而曲线飞行过程的加速度约束为 $a \le a_{\max}$。

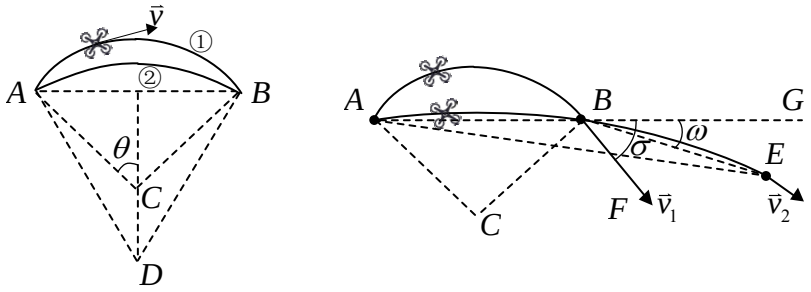
● 距离约束

与问题 1 类似，无人机需要满足同时到达某个节点，且数量不少于 3 架，该位置才能被识别为虚假目标。考虑此问无人机的速度可变，且速度范围为 120km/h 至 180km/h，当只能做匀速直线运动时，10 秒在水平方向上的移动距离在 333m 至 500m 区间内。而做曲线运动时，水平距离的飞行区间变大。

如图 13（a）所示，当无人机以最小速度 $v = 33.3\,\text{m/s}$，曲率半径为 250m 飞行时，AB 水平距离最小。

无人机飞行 10s，弧 AB=333m，曲率半径 AC=250m，根据角速度公式求得 $\theta = 38.196^\circ$，进而求得 AB=309m。当曲率半径逐渐变大时（如 AC 变为 AD）相

同的弧 AC 对应的线段 AC 距离变长。因而，水平方向移动等效距离变为 309m 至 500m。



（a）　　（b）

图 13　无人机曲线飞行距离约束

（a）飞行距离区间（b）飞行角度区间

除此之外，由于在误差范围内，需要尽可能通过多的节点，我们讨论在飞行过程当中的角度极限情况。如图 13（b）所示，A、B、E 为三个干扰节点，弧 AB 为线段 AB 上的弧，弧 AE 为线段 AE 上的弧，弧 AB 的曲率半径小于弧 AE 。无人机到达 B 点时有一个切向速度 $\vec{v}_1$，设该切向速度方向为 BF， BF 与 BG 所成的夹角为 σ， BE 与 BG 所成的夹角为 ω。弧 AB 、 AE 对应的曲率半径为 $r(t_1)$、$r(t_2)$，线段 AB 、 AE 对应的长度为 d_1 、 d_2，由几何关系可以得到：

$$\sigma = \arccos\left(\sqrt{1-\left(\frac{d_1^2}{2\cdot d_1\cdot r(t_1)}\right)^2}\right) \tag{9}$$

$$\omega = \arccos\left(\sqrt{1-\left(\frac{d_2^2}{2\cdot d_2\cdot r(t_2)}\right)^2}\right) \tag{10}$$

如果 $\sigma > \omega$ 则必定能找到一个较大的曲率半径使弧 AE 通过 A、B、C 三个节点。与直线运动的分析同理， $d_1 = p_{(i+n)j} - p_{ij}$， $d_2 = p_{(i+n+m)j} - p_{(i+n)j}$。

根据以上分析，当无人机机动飞行时，需要满足的距离约束条件如下所示：

$$\begin{cases} \dfrac{d}{m-n} \le p_{(i+n)j} - p_{ij} \le 500 \\ \dfrac{d}{m-n} \le p_{(i+n+m)j} - p_{(i+n)j} \le 500 \\ \left(p_{(i+n+m)j} - p_{(i+n)j}\right) - \left(p_{(i+n)j} - p_{ij}\right) \le \varphi \\ \dfrac{p_{(i+n)j} - p_{ij}}{r(t_1)} > \dfrac{p_{(i+n+m)j} - p_{(i+n)j}}{r(t_2)} \\ 3 \le i \le 20, 1 \le j \le 5 \\ n \ge 1, m \ge 1 \end{cases} \tag{11}$$

- 优化模型

本问要求我们使用 9 架无人机组成的编队在 5 分钟内，完成附件 1 要求的虚假航迹。同时，考虑一架无人机至多产生 7 个假目标信息，需要考虑除附件 1 给出的航迹外还能产生多少虚假航迹。我们的目标函数依旧是在满足约束的前提

下使无人机的架数最少。给出简单解释如下：

如果能用 8 架无人机机动飞行产生附件 1 的虚假目标航迹，剩余 1 架无人机能够有更多的自由度，从而产生更多虚假目标航迹。

综合以上约束条件，我们可以得到本问的优化模型为：

$$\text{Obj: } \min(N)$$

$$\text{s.t.}\begin{cases} \dfrac{|\dot{c}(t)|^3}{|\dot{c}(t)\times\ddot{c}(t)|}\geq 250 \\ -10\leq \tan^{-1}\left(\dfrac{\dot{z}(t)}{\sqrt{\dot{x}(t)^2+\dot{y}(t)^2}}\right)\leq 10 \\ \sqrt{\left(\dfrac{dv}{dt}\right)^2+\left(\dfrac{v^2}{r(t)}\right)^2}\leq 10 \\ 309\leq p_{(i+n)j}-p_{ij}\leq 500 \\ 309\leq p_{(i+n+m)j}-p_{(i+n)j}\leq 500 \\ \left(p_{(i+n+m)j}-p_{(i+n)j}\right)-\left(p_{(i+n)j}-p_{ij}\right)\leq \varphi \\ \dfrac{p_{(i+n)j}-p_{ij}}{r(t_1)}>\dfrac{p_{(i+n+m)j}-p_{(i+n)j}}{r(t_2)} \\ 1\leq i\leq 20, 1\leq j\leq 5, n\geq 1, m\geq 1 \end{cases} \tag{12}$$

4.2.2 搜索更多虚假航迹模型

上文的优化模型可以帮助我们找到最少的无人机来完成附件 1 给出的虚假目标点航迹。本问限制无人机的飞行总时间为 5 分钟（即 300s），而我们已经花费 200s 来实现附件 1 的虚假轨迹，因而要产生的其余虚假轨迹，在时间点上有一部分必定与附件 1 的时间节点重合。

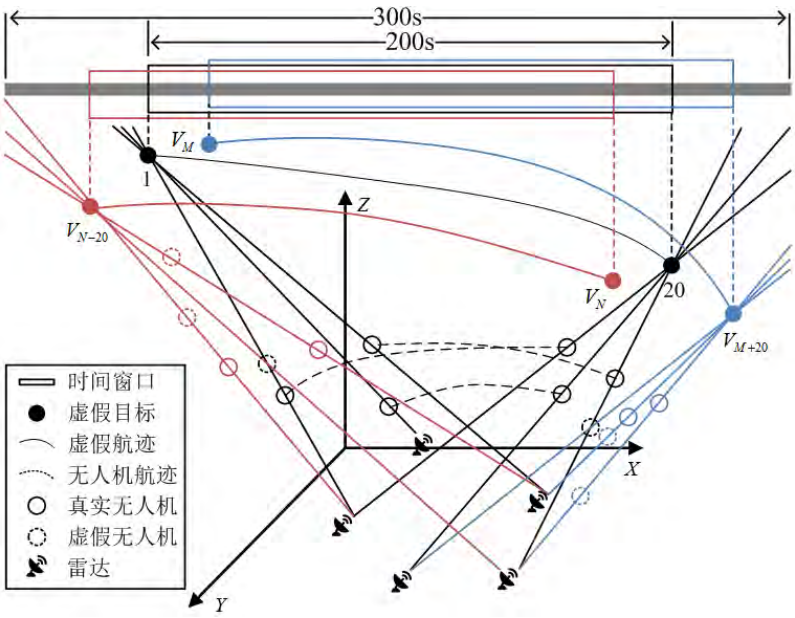


图 14　虚假航线构建规则

在这段时间内，无人机的飞行路径依旧满足上文分析的沿五棱柱的棱飞行，

200s 时间窗口之外的时间（示意图见图 14 中的不与黑框重合的红色框与蓝色框部分），无人机可以利用其余无人机制造的虚假无人机（1 台无人机制造的虚假目标上限为 7 个）目标构造新的虚假航迹。

需要特别注意的是，雷达与任何虚假无人机的连线及延长线不能和题给 200s 内的任何无人机（包括真实与虚假）产生新的虚假目标，这是因为连线已经脱离原有的平面，不能再产生新的交点（由几何关系很容易得到该结论）。因此，只能使用在该时间窗口以外的另外 2 架无人机或其制造的虚假机来完成新的虚假目标构建（如图 14 中V_{N-20}与V_{M+20}的构造示意）。

据上述分析，我们可以建立以重合时间节点尽可能多为目标函数建立虚假航迹搜索的优化模型。设从第t_m时刻开始在时间节点上重合，从t_n时刻开始脱离时间节点重合的状态，除了上文中所有需要满足的无人机约束之外，还需要满足$t_n - t_m \le 200$的约束。同时，我们还可以通过无人机制造的虚假目标尝试构建更多虚假航迹点。

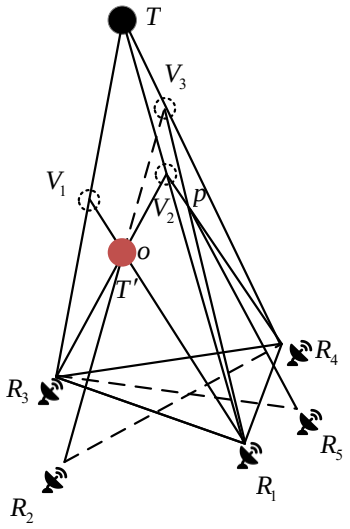


图 15 几何关系示意图

如图 15 所示，假设某一时刻的两架无人机或无人机制造的虚假目标点位于V_1、V_2处，T为这三个目标点所制造的虚假航迹目标，我们可以通过一定的方法构造虚假航迹点T'。构造方法如下：连接雷达R_1与虚假目标点V_1，连接雷达R_3与虚假目标点V_2，交于点o，连接雷达R_2与点o并延长，由几何关系可以得到，必定存在一个在R_4T连线及其延长线上的虚假目标点V_3位于R_2o的延长线上（ΔTR_2R_4的投影为R_2R_4），我们能够很轻易在TR_4上找到虚假目标点V_3，它与V_1、V_2共同形成了虚假航迹点T'。

需要特别注意的是：当形成虚假航迹点T'后，在其他两个面上不可能再存在新的虚假航迹点T''。设连线R_4V_2与R_1V_3相交得到点p，一个直观的解释是，与雷达R_5的连线R_5p的延长线不可能与直线R_3T产生任何交点（由几何关系易得），因而最多存在干扰两台雷达的情况，也就不可能再增加新的虚假航迹点。

以上便是使得额外虚假航迹最大化的优化模型思路。

4.3 模型求解

由于无人机机动飞行的自由度大大高于无人机匀速直线飞行的自由度，用第一问我们采用的 TS 算法进行求解非常耗时，且有很大可能陷入局部最优。为了

便于模型的求解，我们将空间的点离散化后进行搜索。

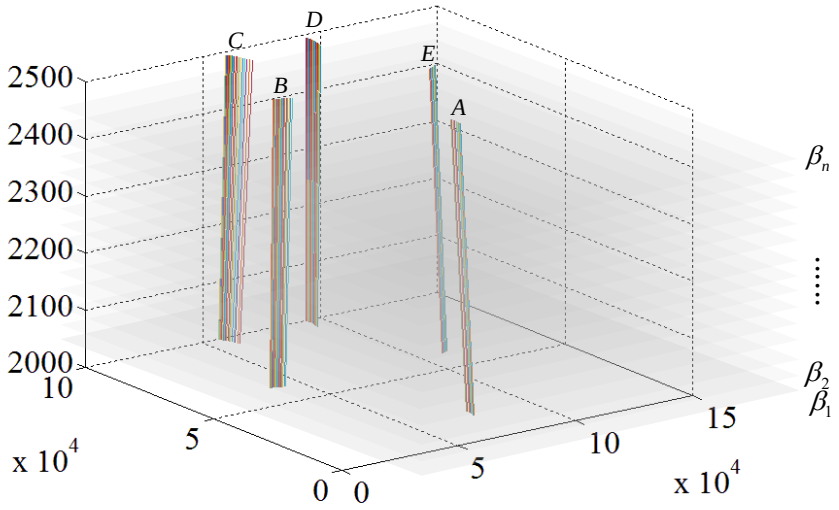


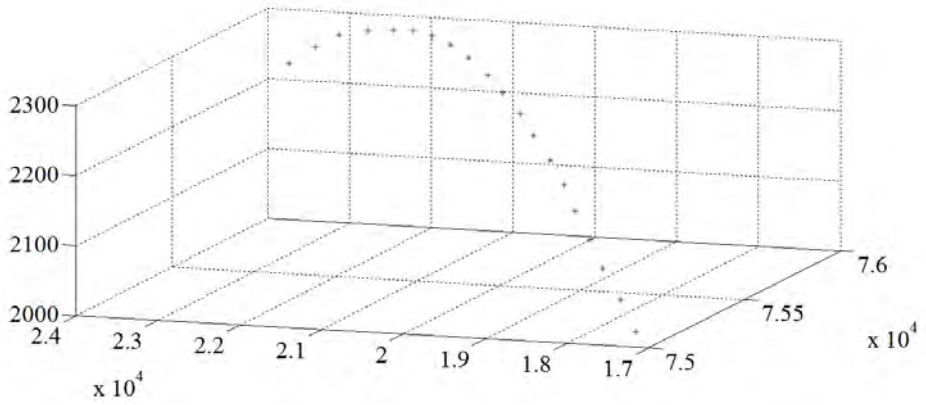
图 16　解空间离散化

如图 15 所示，机动飞行的无人机所在节点位置的所有解空间为 2000m 到 2500m 平面所夹的空间 $\mathbb{R}^3$ 中，簇 A 到簇 E 共 100 条线段。我们用有限个平面 $\beta_1,\beta_2,...,\beta_n$ 截 100 段线段簇，将求解空间数量降为100n 个离散点，在这些离散点内寻找最优的干扰节点组合。最终我们搜索得到的最优结果如表 2 所示。

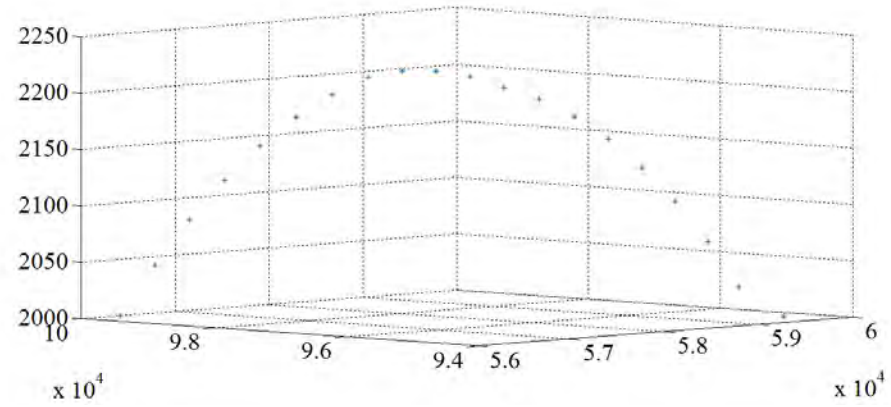
表 2 问题 2 结果展示

无人机编号	形成的虚假目标编号	干扰雷达编号
1	1-20	1
2	1-20	3
3	1-20	4
4	空余	
5	空余	
6	空余	
7	空余	
8	空余	
9	空余	

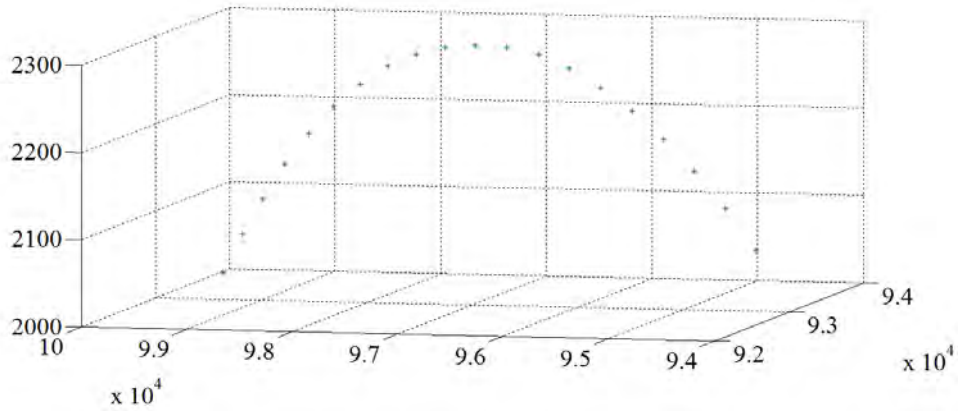
根据我们的搜索结果，3 架无人机即可产生附件 1 中所示的虚假目标，编号为 1、2、3 号的无人机分别干扰编号为 1、3、4 号的雷达。图 17（a）、（b）、（c）分别展示了编号为 1~3 的无人机机动飞行轨迹。



（a）1 号无人机，干扰 1 号雷达



（b）2 号无人机，干扰 3 号雷达



（c）3 号无人机，干扰 4 号雷达

图 17　无人机机动飞行轨迹示意图

在额外的虚假航迹求解中，根据我们的模型建立过程分析，求解出，空余的 6 架无人机两两组合能形成三条虚假航迹，而图 15 的几何关系能帮助我们再寻找到一条虚假航迹。因此，加上附件 1 的虚假航迹我们一共得到了 5 条虚假航迹。9 架无人机飞行路径与航迹的结果展示如图 18 所示。

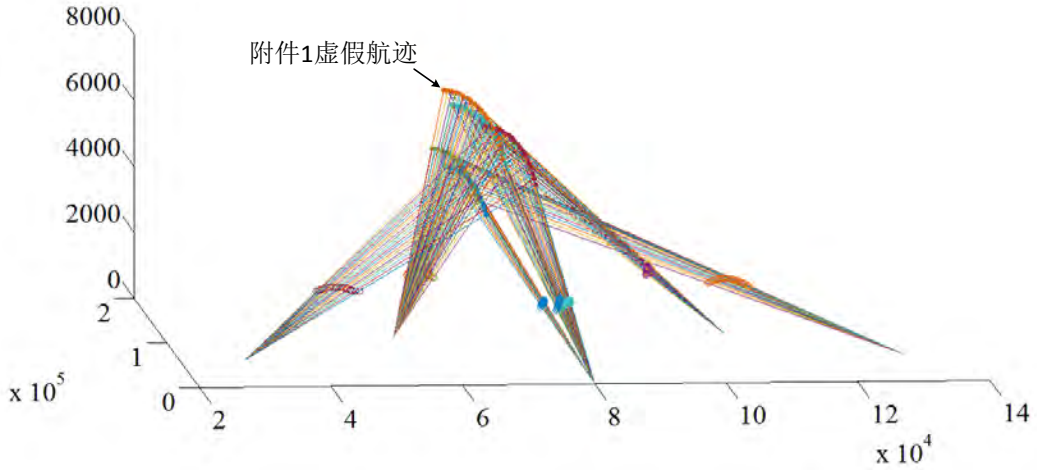


图 18　问题 2 最终结果：9 架无人机飞行路径与 5 条虚假航迹

4.4 模型检验与结果分析

根据我们得到的结果与每架无人机每一个时刻干扰的雷达编号，反推重构出3架无人机飞行形成的20个虚假目标位置。然后将重构目标点与题中给出的20个虚假目标位置进行对比，进而检验模型。

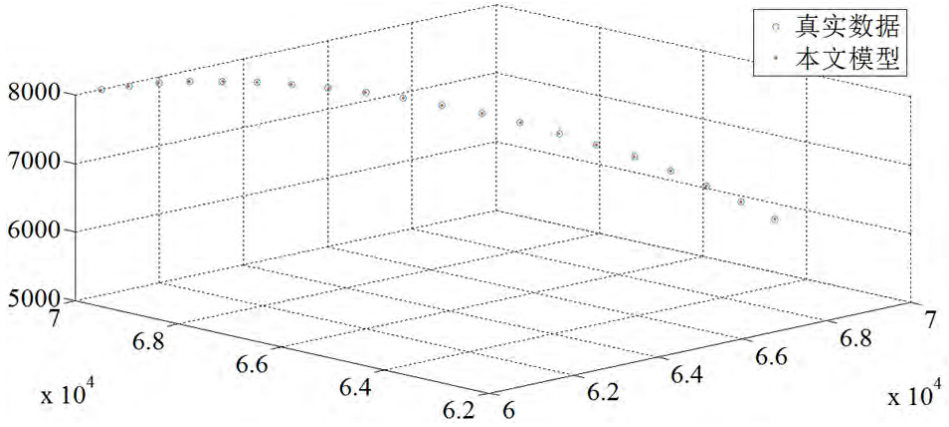


图 19　虚假目标重构检验

经验证，我们发现本文模型重构的坐标点与真实数据非常接近，误差在合理范围之内，同时，无人机轨迹具有连贯性，我们给出9架无人机的相关参数，进一步验证了轨迹的合理性，如表3所示，所有参数都在题目给定的合理参数区间内，也即通过参数检验。

表 3　9架无人机部分参数检验

无人机编号	最小转弯半径（m）	最大爬升（俯冲）角度（°）	最大加速度（m/s^2）	速度区间（km/h）
1	371.37	4.35	6.32	127.53-174.41
2	288.45	8.71	8.24	125.47-171.56
3	304.72	6.95	7.45	128.41-169.63
4	405.38	3.97	4.69	137.14-168.15
5	291.47	4.13	4.57	131.52-173.08
6	356.82	7.37	7.41	135.12-176.90
7	462.54	5.73	6.25	127.62-174.14
8	394.23	7.28	8.56	152.51-172.63
9	296.70	6.42	3.04	141.15-173.26

经过上述的检验，我们进一步证明了问题2模型的合理性与可靠性。

五 问题 3：雷达系统存在干扰下的协同干扰模型

5.1 问题分析

本问在第二问的基础上还需要考虑雷达系统可能存在压制干扰或其它因素的干扰导致在某些时刻无法正常获取回波信号的情况。

这实际上是在问题 2 的基础上给了更多的无人机飞行空间，特别是条件：若一条航迹中航迹点的个数不超过 3 个时（通过前面规定的“同源检验”），该航迹就被继续保留，可以采用更少的无人机完成轨迹，从而留出更多无人机进行额外虚假航迹的产生。

那么根据我们问题 2 的额外虚假航迹讨论思路，其中有两个雷达作用的虚假航几点有可能被保留。我们在搜索更多虚假航迹时，主要分析这种双雷达作用下的虚假航迹点带来的影响。

5.2 模型建立

问题 2 的模型主要目标是优化航迹，与其不同，本问的模型目标是在问题 2 的基础上优化无人机调度方案。

航迹被保留的前提条件是前后都有完整的航迹目标点，不能连续跳过两个以上的不足三个无人机的飞行节点，为了更好地理解本问的附加条件，将其用用图 20 表示如下。

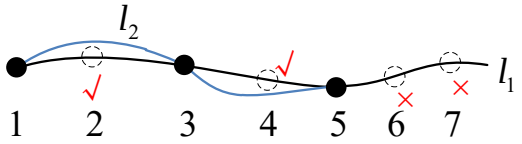


图 20 雷达允许被干扰时的飞行规则示意图

如果没有问题 3 的附加条件，无人机 3 在问题二的条件下只能采用l_1路径的飞行方式。问题 3 给的条件是：若一条航迹中航迹点的个数不超过 3 个时（通过前面规定的“同源检验”），该航迹就被继续保留。即如果符合无人机飞行基本条件的约束（如转弯半径允许、加速度在允许范围内等），则无人机可以跳过类似于产生航迹节点 2 与航迹节点 4 的节点（前提是有至少两个其他雷达能在 2 时刻以及 4 时刻被干扰），这使得无人机的航线自由度变得更高，能为节省无人机的架数提供可能性。

类似于图 20 中的连续跳过第 6 个航迹节点与第 7 个航迹节点两个节点的飞行方式不被允许（无法正常保留航迹信息）。

同时，根据题目要求若一条航迹中这类航迹点的个数不超过 3 个时，该航迹就被继续保留，则一条航迹中的这类点不允许超过三个，否则无法正常保留航迹。

为了更直观给出本问针对无人机协同飞行的轨迹保留情况，我们给出如图 21 的描述。

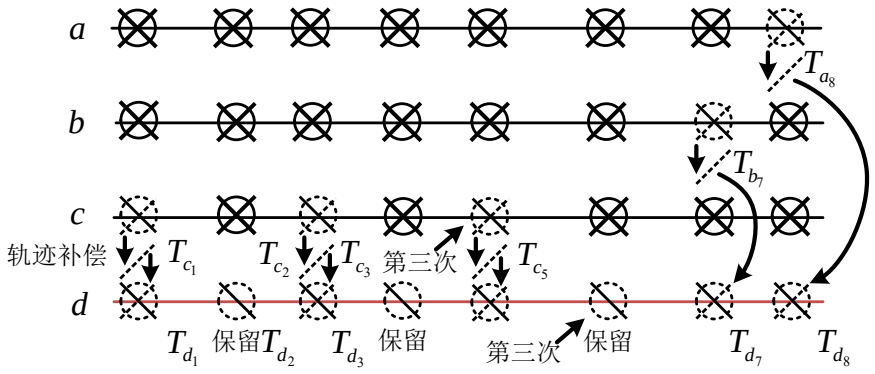


图 21 协同飞行轨迹补偿策略

对图 21 的协同飞行轨迹补偿策略进行解释如下：

取问题二求得的 5 条轨迹（包括附件 1 产生的虚假航迹）中的任意三条轨迹 a 、b、c 对其进行说明，d 轨迹为问题 3 新增的轨迹。

图 20 展示了我们的基本飞行策略，每条轨迹上能够有不连续的轨迹空缺，且一条轨迹上该类点的数量不超过三个。在图 21 示意图中，虚假目标点 T_{c_1} 空缺后下一个同一轨迹上的虚假目标点 T_{c_2} 不允许空缺，否则无法形成完整的 c 轨迹。对于新增轨迹 d 而言，每个虚假目标点上已经有两条交线，需要用其余轨迹的目标点补偿 d 上的目标点完成整个航线。补偿的方案有多种，在本示例中，我们只给出一种补偿调度策略：

Step1：T_{c_1} 取出一条航迹补偿 T_{d_1}，c 航线空缺目标点数量为 1。

Step2：T_{c_2} 不进行调度，此时 T_{d_2} 有两条交线，被暂时保留，c 航线空缺目标点数量为 1，d 航线空缺目标点数量为 1。

Step3：T_{c_3} 取出一条航迹补偿 T_{d_3}，c 航线空缺目标点数量为 2，d 航线空缺目标点数量为 1。

Step4：T_{c_4} 不进行调度，此时 T_{d_4} 有两条交线，被暂时保留，c 航线空缺目标点数量为 2，d 航线空缺目标点数量为 2。

Step5：T_{c_5} 取出一条航迹补偿 T_{d_5}，c 航线空缺目标点数量为 3，d 航线空缺目标点数量为 2，c 航线空缺目标已达到三个，不能再提供更多交线补偿。

Step6：此时 T_{d_6} 有两条交线，被暂时保留，d 航线空缺目标点数量为 3，不能再有其余空缺。

Step7：任务转移至 b，T_{b_7} 取出一条航迹补偿 T_{d_7}，b 航线空缺目标点数量为 1。

Step8：由于此时 c、d 航线已经用完三次空缺机会，b 航线不能有连续两个空缺目标点，补偿任务转移至 a 航线，T_{a_8} 取出一条航迹补偿 T_{d_8}，a 航线空缺目标点数量为 1，b 航线空缺目标点数量为 1。

Step9：重复执行 Step7 与 Step8，直到问题 2 的五条虚假航线被完全调度结束，此时 d 航线存在的目标点数量为：3（空缺三次）+3×3（每条其余航线补偿三次）=12 个（5 条航线中只有 3 条能被调度）。

Step10：d 航线已经产生 12 个目标点，剩余的时间还有100 s，一共产生 22 个目标点，在这些点中搜索连续的 20 个点组成完整轨迹。

5.3 模型求解与分析

利用上述的模型，结合问题 2 有关的约束，对解空间进行遍历搜索，最后我们找到的新增虚假航线如图 22 中的 * 所示。

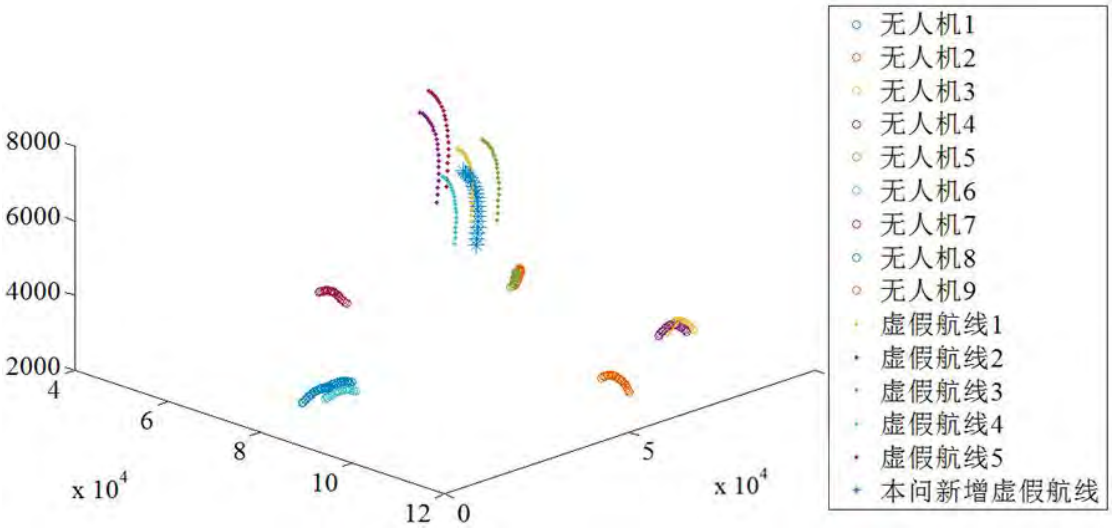


图 22 问题三新增虚假航线示意图

加上问题 2 我们发现的航迹与附件 1 给出的一条虚假航迹，在本问中我们利用 9 架无人机机动飞行一共能产生 6 条虚假航迹。

5.4 模型检验与结果分析

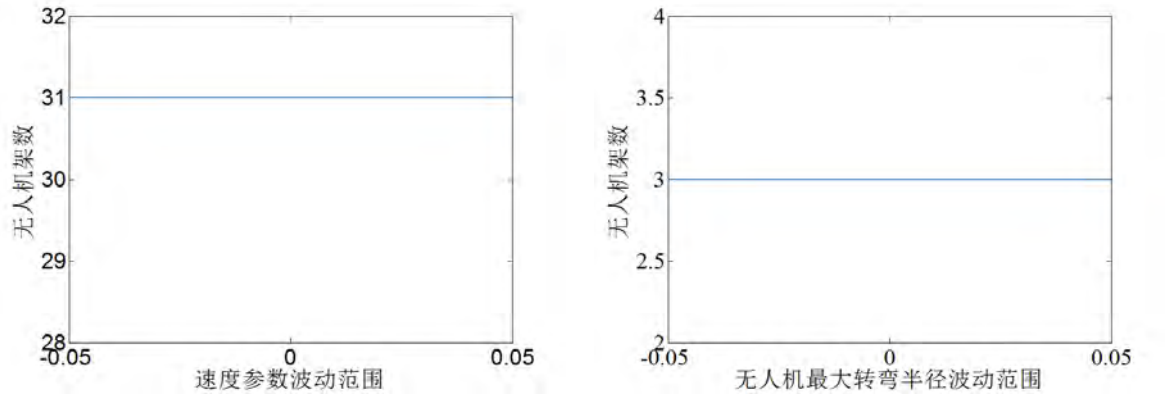
经过检验，该轨迹符合运动规律，没有明显的误差，模型结果可靠。

六 模型评价与推广

6.1 鲁棒性分析

本文的前提中给出了相关的参数大小，如：无人机的飞行速度控制在120km/h~180km/h，飞行高度控制在2000m~2500m，最大加速度不超过 $10\,m/s^2$。由于安全等因素的考虑，无人机间距需控制在 100 m 以上，无人机机动飞行机转弯半径不小于 250m 等。给予每个参数 95%的置信区间，本节取了无人机速度参数约束与无人机转弯半径约束，分别对问题 1 的模型鲁棒性与问题 2 的模型鲁棒性进行分析。

图 18（a）给出了无人机速度约束取±5%时的问题 1 结果波动情况，图 18（b）给出了无人机转弯半径约束取±5%时的问题 2 结果波动情况。



（a） （b）

图 23 模型鲁棒性分析（a）问题 1 模型鲁棒性（b）问题 2 模型鲁棒性

6.2 灵敏度分析

在本文中，我们假设与引用了参考文献中的参数，如引用参考文献[1]中的雷达误差，参考文献[4]中的无人机爬升角度限制等。

对于这些参数，我们对其进行灵敏度分析来观察当这些因素参数值与我们假定的因素不一致时对模型结果的影响。为了观察结果，我们代表性的选取了雷达误差和无人机爬升角度限制分别对问题 1 与问题 2 的模型结果进行分析。

6.2.1 问题 1 模型灵敏度

当雷达误差参数发生变化（不考虑是否符合科学性）时，模型 1 的结果变化如图 16 所示。

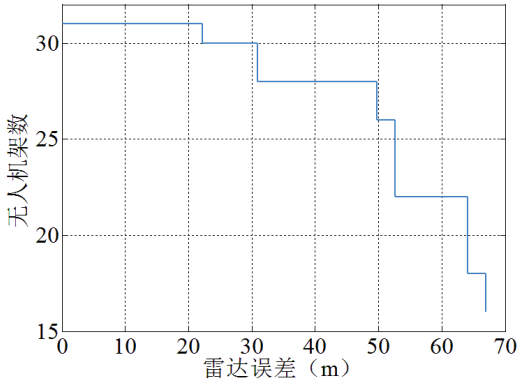


图 24 问题 1 模型灵敏度

雷达误差(m)	无人机架数
0	31
22.2024	30
30.9441	28
49.7214	26
52.6025	22
64.0858	18
66.9272	16

当雷达误差大于 22.2m 时，会对模型结果造成一定影响，无人机架数减少。

6.2.2 问题 2 模型灵敏度

当无人机爬升角度参数发生影响时（不考虑是否符合科学性）时，模型 1 的结果变化如图 17 所示。

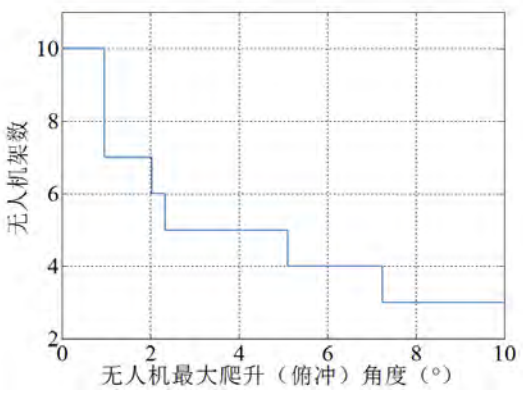


图 25 问题 2 模型灵敏度

爬升（俯冲）最大角度	无人机架数
10	3
8.715	3
7.246	4
5.107	5
2.341	6
2.047	7
0.963	8
0	10

在图 17 中，我们可以很直观地发现，无人机的爬升（俯冲）角度参数限制对本问题的影响较大，在不允许无人机做爬升与俯冲的机动动作的极限情况下（允许无人机以大于 250m 的半径转弯），一共需要 14 架无人机完成附件 1 的虚假航迹。只有当允许爬升角度大于 8.7 度时，需要的无人机架数才减少为 7 架。

6.3 模型优点

根据模型建立与求解过程中对模型以及结果的分析，总结模型的优点如下：

- **模型从问题机理出发建立模型，引入 TS 算法进行求解，避免了 NP 难问题：**在问题 1 的模型建立时充分分析了模型机理，将本来复杂的问题简化，通过逆向思维寻找无人机飞行路径规律建立模型，采用 TS 算法求解模型，避免陷入局部最优解，使得模型结果更加可靠。
- **模型重构虚假航迹，验证了模型的可靠性：**在问题 1 与问题 2 中，利用无人机真实航线重构虚假航线，与附件 1 坐标进行对比，在允许误差范围内验证了模型的可靠性。
- **模型通过鲁棒性检验与灵敏度分析：**我们对模型进行了鲁棒性检验与灵敏度分析，发现模型具有较好的鲁棒性。

6.4 模型缺点

根据模型建立与求解过程中对模型以及结果的分析，总结模型的缺点如下：

- **模型的时间代价较大：**虽然采取了较高效的智能算法，但是由于采用了空间搜索的方法，如问题 2 对空间的 100*500 个点进行遍历，以及问题 2 额外虚假航迹求解时对所有虚假目标 100*500*7 个点进行遍历，求解花费的时间代价依旧较大。

6.5 模型推广

本文从机理分析出发，对多无人机对组网雷达的协同干扰问题进行了细致的研究，所采用的模型能给电子对抗领域一定的启发。

本文采用的算法与解空间离散化的思想在图搜索与遍历、大数据分析等领域具有较好的前景。

参考文献

[1] http://www.FindPdf.ir. IEEE Standard for Ultrawideband Radar Definitions - Corrigendum 1[C]// IEEE 1672-2006/cor. IEEE, 2012:1-5.

[2] Glover F, Laguna M. Tabu Search[J]. General Information, 1997, 106(2):221-225.

[3] Kozłowski M, Marciakkozłowska J. Radius, velocity and acceleration of the space-time[J]. Nuovo Cimento B Serie, 2001, 116(7).

[4] Yang I Y, Lee B H, Chang B H. Design Parameter Analysis of a Solar-Powered, Potential Energy-Storing, Long Endurance UAV[J]. 한국항공우주학회지, 2011, 39(10):927-934.

[5] 杨俭, 汤亚波等. 假目标欺骗干扰无人机空域协同干扰策略研究[J]. 航空科学技术, 2017(5):37-41.

[6] 李珂, 王正平. 多架无人机的协同攻击航路规划[J]. 航空计算技术, 2006, 36(5):98-101.

[7] Nikolos I K, Valavanis K P, Tsourveloudis N C, et al. Evolutionary algorithm based offline/online path planner for UAV navigation.[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 2003, 33(6):898-912.

[8] 王庆贺. 基于最优化理论的三维多无人机协同任务规划技术研究[D]. 解放军信息工程大学, 2017.

[9] Berni J, Zarcotejada P J, Suarez L, et al. Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle.[J]. IEEE Transactions on Geoscience & Remote Sensing, 2009, 47(3):722-738.

[10] Luo S G. Development Status and Trend of Electronic Warfare Unmanned Aerial Vehicle[J]. Shipboard Electronic Countermeasure, 2009.

[11] 沈文亮, 张卓鸿. 无人机在电子对抗中的应用研究[J]. 舰船电子对抗, 2013, 36(6):14-18.

[12] 赵铭, 盛怀洁, 王春雷. 基于突防的电子对抗无人机航路规划研究[C]// 2006 中国无人机大会. 2006.

[13] 王威, 许鹏, 张多林. 混合优化策略在巡航导弹多航迹规划中的应用[J]. 电光与控制, 2008, 15(4):66-69.

[14] Judd K, Mclain T. Spline based path planning for unmanned air vehicles[C]// 2001.

[15] 鞠明, 陈永光. 空袭行动中雷达干扰无人机作战运用初探[J]. 飞航导弹, 2009(8):21-25.

附录

源码部分过长，所以只展示部分求解代码

问题 1 部分代码

```
bool judge(float d1,int k){
        if ((d1 > k*333.33&&d1 < k * 500)){
            return true;
        }
    return false;
}

int fun1(){
    ofstream out("road.txt");
    ifstream myin("in.txt");
    ofstream out2("pos2500.txt");
    //    vector<int> staI, staJ;
    int se[20][21] = { 0 };
    float x[5] = { 80000, 30000, 55000, 105000, 130000 };
    float y[5] = { 0, 60000, 110000, 110000, 60000 };
    float px[20] = { 60600, 61197, 61790, 62377, 62955, 63523, 64078, 64618, 65141, 65646, 66131,
66594, 67026, 67426, 67796, 68134, 68442, 68719, 68966, 69184 };
    float py[20] = { 69982, 69928, 69838, 69713, 69553, 69359, 69131, 68870, 68577, 68253, 67900,
67518, 67116, 66697, 66263, 65817, 65361, 64897, 64429, 63957 };
    float pz[20] = { 7995, 7980, 7955, 7920, 7875, 7820, 7755, 7680, 7595, 7500, 7395, 7280, 7155,
7020, 6875, 6720, 6555, 6380, 6195, 6000 };
    float h[2] = { 2000, 2500 };
    float ux[20][5];
    float uy[20][5];
    int lv[20][5] = { 0 };
    float height = 2499;
    int k = 9;
    int lal = 10;
    int use[20] = { 0 };
        int count = 0;
            for (int j = 0; j < 50; j++){
                height = 2000 + j * 10;
                for (int j = 0; j < 20; j++){
                    for (int i = 0; i < 5; i++){
                        ux[j][i] = (height - pz[j])*(px[j] - x[i]) / (pz[j] - 0) + px[j];
                        uy[j][i] = (height - pz[j])*(py[j] - y[i]) / (pz[j] - 0) + py[j];
                    }
                }
                for (int i = 0; i < 5; i++){
```

```
                    float detad = sqrt(pow(ux[k][i] - ux[lal][i], 2) + pow(uy[k][i] - uy[lal][i],
2));
                    if (judge(detad, lal - k)){
                        if (i + 1 == 2){
                            cout << k + 1 << "__" << lal + 1 << "    " << i + 1 << "     " << ux[k][i]
<< "---" << uy[k][i] << "      " << ux[lal][i] << "---" << uy[lal][i] << "***************" << height
<< endl;;
                        }
                        // cout << k  << "    " << lal  << "    " << i  << endl;// "     " <<
ux[k][i] << "---" << uy[k][i] << "      " << ux[lal][i] << "---" << uy[lal][i] << "***************"
<< height << endl;;
                    }
                }
            }
    return 0;
}

int fun12(){
    int se[20][21] = { 0 };
    float x[5] = { 80000, 30000, 55000, 105000, 130000 };
    float y[5] = { 0, 60000, 110000, 110000, 60000 };
    float px[20] = { 60600, 61197, 61790, 62377, 62955, 63523, 64078, 64618, 65141, 65646, 66131,
66594, 67026, 67426, 67796, 68134, 68442, 68719, 68966, 69184 };
    float py[20] = { 69982, 69928, 69838, 69713, 69553, 69359, 69131, 68870, 68577, 68253, 67900,
67518, 67116, 66697, 66263, 65817, 65361, 64897, 64429, 63957 };
    float pz[20] = { 7995, 7980, 7955, 7920, 7875, 7820, 7755, 7680, 7595, 7500, 7395, 7280, 7155,
7020, 6875, 6720, 6555, 6380, 6195, 6000 };
    float h[2] = { 2000, 2500 };
    float ux[20][5];
    float uy[20][5];
    int lv[20][5] = { 0 };
    int w = 100;
    float hx[20], hy[20];
    int mmm = 0;
    float xx, yy;
    for (int t = 0; t < 50; t++){
        float height = 2000 + t*10;
        int use[20] = { 0 };
        for (int j = 0; j < 20; j++){
            for (int i = 0; i < 5; i++){
                ux[j][i] = (height - pz[j])*(px[j] - x[i]) / (pz[j] - 0) + px[j];
                uy[j][i] = (height - pz[j])*(py[j] - y[i]) / (pz[j] - 0) + py[j];
            }
        }
```

```
        int count = 0;
        for (int j = 0; j < 20; j++){
            if (use[j] == 0){
                count++;
                use[j] == 1;
                cout << j << ' ';
                int k = j;
                for (int lal = j + 1; lal < 20; lal++){
                    if (use[lal] == 0){
                        int tt = 0;
                        for (int i = 0; i < 5; i++){
                            float detad = sqrt(pow(ux[k][i] - ux[lal][i], 2) + pow(uy[k][i] -
uy[lal][i], 2));
                            if (judge(detad,lal-k)){
                                tt++;
             //                     cout << ux[k][i] << "---" << uy[k][i] << "     " << ux[lal][i]
<< "---" << uy[lal][i];
                            }
                            if (tt > 3){
                                k = lal;
                                use[lal] = 1;
                                cout << k << ' ';
                                if (j == 9){
                                }
                                break;
                            }
                        }
                    }
                }
                cout << endl;
            }
        }
        if (count < w){
            w = count;
            if (w == 2)
                cout << height << ' ' << endl;
        }
    }
        return 0;
}
```

问题 2 与问题 3 部分代码

```
bool jud3d(float vx1, float vy1, float vz1, float vx2, float vy2, float vz2, float len1, float len2, int k1, int k2){
    float cs1 = (vx1*vx2 + vy1*vy2 + vz1*vz2) / (len1*len2);//(sqrt(pow(vx1, 2) + pow(vy1, 2) + pow(vz1, 2))*sqrt(pow(vx2, 2) + pow(vy2, 2) + pow(vz2, 2)));
    float cs2 = sqrt(1 - pow(len1/ k1 / 500,2));
    if (cs2 < cs1&&len2>k2*309.182&&len2 < 500 * k2){
        return true;
    }
    return false;
}

int quesion2(){
    ofstream out("road.txt");
    float ox3 = 31.2 / (4.4 + 2.0 / 3.0);
    float oy3 = 35.2 - (4.4*ox3);
    float oz3 = 0;
    ox3 *= 10000;
    oy3 *= 10000;
    oz3 *= 10000;
 //   cout << asin(50 / 309) << endl;
    int se[20][21] = { 0 };
    float x[5] = { 80000, 30000, 55000, 105000, 130000 };
    float y[5] = { 0, 60000, 110000, 110000, 60000 };
    float px[20] = { 60600, 61197, 61790, 62377, 62955, 63523, 64078, 64618, 65141, 65646, 66131, 66594, 67026, 67426, 67796, 68134, 68442, 68719, 68966, 69184 };
    float py[20] = { 69982, 69928, 69838, 69713, 69553, 69359, 69131, 68870, 68577, 68253, 67900, 67518, 67116, 66697, 66263, 65817, 65361, 64897, 64429, 63957 };
    float pz[20] = { 7995, 7980, 7955, 7920, 7875, 7820, 7755, 7680, 7595, 7500, 7395, 7280, 7155, 7020, 6875, 6720, 6555, 6380, 6195, 6000 };
    float ux1[5][20];
    float uy1[5][20];
    float ux2[5][20];
    float uy2[5][20];
    float rate =0.9;
    int sq = 0;
    for (int i = 0; i < 20; i++){
   //     float hr = pz[i]/7995;
        float Px4 = px[i];
        float Py4 = py[i];
        float Pz4 = pz[i];
        //pz[i] *= rate;
        px[i] = x[sq] + rate*(px[i] - x[sq]);
        py[i] = y[sq] + rate*(py[i] - y[sq]);
```

```
        pz[i] *= rate;
    //    float cz = (-(pz[i] - 0)*(ox3 - px[i]) - pz[i] * (px[i] - x[2])) / ((pz[i] - 0)*(ox3
- Px4) / (oz3 - Pz4) - (px[i] - x[2]));
        float cz = pz[i]*Pz4 * (ox3 - x[2]) / (pz[i] * (ox3 - Px4) + Pz4*(px[i] - x[2]));
        float cx = (cz - pz[i])*(px[i] - x[2]) / (pz[i] - 0) + px[i];
        float cy = (cz - pz[i])*(py[i] - y[2]) / (pz[i] - 0) + py[i];
  //      float dz = (-(cz - 0)*(x[2] - cx) - cz * (cx - x[0])) / ((cz - 0)*(x[2] - Px4) / (0
- Pz4) - (cz - x[0]));
        float dz = cz*Pz4*(x[2] - x[0]) / (cz*(x[2] - Px4) +Pz4*( cx - x[0]));
        float dx = (dz - cz)*(cx - x[0]) / (cz - 0) + cx;
        float dy = (dz - cz)*(cy - y[0]) / (cz - 0) + cy;
        float bz = cz*Pz4*(x[3] - x[1]) / (cz*(x[3] - Px4) + Pz4*(cx - x[1]));
        float bx = (bz - cz)*(cx - x[1]) / (cz - 0) + cx;
        float by = (bz - cz)*(cy - y[1]) / (cz - 0) + cy;
        float kz = pz[i] * bz*(x[0] - x[3]) / ((x[0] - bx)*pz[i] + bz*(px[i] - x[3]));
        float kx = (kz - bz)*(bx - x[0]) / (bz - 0) + bx;
        float ky = (kz - bz)*(by - y[0]) / (bz - 0) + by;
   //     px[i] = bx; py[i] = by; pz[i] = bz;
  //      px[i] = dx; py[i] = dy; pz[i] = dz;
  //      px[i] = cx; py[i] = cy; pz[i] = cz;
        px[i] = kx; py[i] = ky; pz[i] = kz;
        cout << px[i] << "  " << py[i] << "  " << pz[i] << endl;
    }
    for (int j = 0; j < 20; j++){
        for (int i = 0; i < 5; i++){
            ux1[i][j] = (2000 - pz[j])*(px[j] - x[i]) / (pz[j] - 0) + px[j];
            uy1[i][j] = (2000 - pz[j])*(py[j] - y[i]) / (pz[j] - 0) + py[j];
        }
    }
    for (int j = 0; j < 20; j++){
        for (int i = 0; i < 5; i++){
            ux2[i][j] = (2500 - pz[j])*(px[j] - x[i]) / (pz[j] - 0) + px[j];
            uy2[i][j] = (2500 - pz[j])*(py[j] - y[i]) / (pz[j] - 0) + py[j];
        }
    }
    float ax[5][20][100], ay[5][20][100], az[5][20][100];
    for (int i = 0; i < 5; i++){
        for (int j = 0; j < 20; j++){
            float dx = (ux2[i][j] - ux1[i][j]) / 100;
            float dy = (uy2[i][j] - uy1[i][j]) / 100;
            for (int k = 0; k < 100; k++){
                az[i][j][k] = 2000 + 5 * k;
                ax[i][j][k] = ux1[i][j] + dx*k;
                ay[i][j][k] = uy1[i][j] + dy*k;
```

```
            }
        }
    }
    for (int i = 4; i <5; i++){
        for (int j = 0; j < 1; j++){
            for (int h = 0; h < 100; h++){
                for (int j2 = j + 1; j2 < 19; j2++){
                    for (int h2 = 0; h2 < 100; h2++){
                        float dx1 = ax[i][j2][h2] - ax[i][j][h];
                        float dy1 = ay[i][j2][h2] - ay[i][j][h];
                        float dz1 = az[i][j2][h2] - az[i][j][h];
                        float leng1 = sqrt(pow(dx1, 2) + pow(dy1, 2) + pow(dz1, 2));
                        int k = j2 - j;
                        if (leng1 < k*309.182 || leng1>k * 500){
                            continue;
                        }
                        int h2p = h2;
                        for (int j3 = j2 + 1; j3 < 20; j3++){
                            bool jp3 = false;
                            for (int h3 = 0; h3 < 100; h3++){
                     //           if (j3 == 17) break;
                              //   if (j3 == 15) break;
                                float dx2 = ax[i][j3][h3] - ax[i][j2][h2];
                                float dy2 = ay[i][j3][h3] - ay[i][j2][h2];
                                float dz2 = az[i][j3][h3] - az[i][j2][h2];
                                float leng2 = sqrt(pow(dx2, 2) + pow(dy2, 2) + pow(dz2, 2));
                                bool tj = jud3d(dx1, dy1, dz1, dx2, dy2, dz2, leng1, leng2, k,
j3 - j2);
                                if (tj){
                                    bool tj = jud3d(dx1, dy1, dz1, dx2, dy2, dz2, leng1, leng2,
k, j3 - j2);
                                    cout << i << ":    " << j << " ( " << ax[i][j][h] << ", "
<< ay[i][j][h] << ", " << az[i][j][h] << " )" << "  "
                                        << j2 << " ( " << ax[i][j2][ h2] << ", " << ay[i][j2][h2]
<< ", " << az[i][j2][h2] << " )" << "  "
                                        << j3 << " ( " << ax[i][j3][h3] << ", " << ay[i][j3][h3]
<< ", " << az[i][j3][h3] << " )" << endl;
                                    j2 = j3;
                                    h2 = h3;
                                    //    if ()
                                    if (j3 == 19){
                                        cout << "nice" << endl;
                                    }
                                    break;
```

```
                                        //              j2 = j3;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    cout << "adsfd" << endl;
    return 0;
}
```

Matlab 结果绘图代码

```
hold on
scatter3(n1(:,1),n1(:,2),n1(:,3))
scatter3(n2(:,1),n2(:,2),n2(:,3))
scatter3(n3(:,1),n3(:,2),n3(:,3))
scatter3(n4(:,1),n4(:,2),n4(:,3))
scatter3(n5(:,1),n5(:,2),n5(:,3))
scatter3(n6(:,1),n6(:,2),n6(:,3))
scatter3(n7(:,1),n7(:,2),n7(:,3))
scatter3(n8(:,1),n8(:,2),n8(:,3))
scatter3(n9(:,1),n9(:,2),n9(:,3))

for i=1:60
 plot3(gj1(i,[1,4]),gj1(i,[2,5]),gj1(i,[3,6]))
end

for i=1:60
 plot3(gj2(i,[1,4]),gj2(i,[2,5]),gj2(i,[3,6]))
end

for i=1:60
 plot3(gj3(i,[1,4]),gj3(i,[2,5]),gj3(i,[3,6]))
end

for i=1:60
 plot3(gj4(i,[1,4]),gj4(i,[2,5]),gj4(i,[3,6]))
end

hold on
scatter3(n1(:,1),n1(:,2),n1(:,3))
scatter3(n2(:,1),n2(:,2),n2(:,3))
scatter3(n3(:,1),n3(:,2),n3(:,3))

scatter3(n6(:,1),n6(:,2),n6(:,3))
scatter3(n7(:,1),n7(:,2),n7(:,3))
scatter3(n8(:,1),n8(:,2),n8(:,3))
scatter3(n9(:,1),n9(:,2),n9(:,3))

for i=1:60
 plot3(gj1(i,[1,4]),gj1(i,[2,5]),gj1(i,[3,6]))
end

hold on
scatter3(n1(:,1),n1(:,2),n1(:,3))
```

```
scatter3(n2(:,1),n2(:,2),n2(:,3))
scatter3(n3(:,1),n3(:,2),n3(:,3))
scatter3(n4(:,1),n4(:,2),n4(:,3))
scatter3(n5(:,1),n5(:,2),n5(:,3))
scatter3(n6(:,1),n6(:,2),n6(:,3))
scatter3(n7(:,1),n7(:,2),n7(:,3))
scatter3(n8(:,1),n8(:,2),n8(:,3))
scatter3(n9(:,1),n9(:,2),n9(:,3))
for i=1:60
 plot3(gj1(i,[1,4]),gj1(i,[2,5]),gj1(i,[3,6]))
end

for i=1:60
 plot3(gj2(i,[1,4]),gj2(i,[2,5]),gj2(i,[3,6]))
end

for i=1:60
 plot3(gj3(i,[1,4]),gj3(i,[2,5]),gj3(i,[3,6]))
end

for i=1:60
 plot3(gj4(i,[1,4]),gj4(i,[2,5]),gj4(i,[3,6]))
end

scatter3(oo(:,1),oo(:,2),oo(:,3),300,'.')
scatter3(oo1(:,1),oo1(:,2),oo1(:,3),300,'.')
scatter3(oo2(:,1),oo2(:,2),oo2(:,3),300,'.')
scatter3(oo3(:,1),oo3(:,2),oo3(:,3),300,'.')
scatter3(oo4(:,1),oo4(:,2),oo4(:,3),300,'.')

hold on
scatter3(n1(:,1),n1(:,2),n1(:,3))
scatter3(n2(:,1),n2(:,2),n2(:,3))
scatter3(n3(:,1),n3(:,2),n3(:,3))
scatter3(n4(:,1),n4(:,2),n4(:,3))
scatter3(n5(:,1),n5(:,2),n5(:,3))
scatter3(n6(:,1),n6(:,2),n6(:,3))
scatter3(n7(:,1),n7(:,2),n7(:,3))
scatter3(n8(:,1),n8(:,2),n8(:,3))
scatter3(n9(:,1),n9(:,2),n9(:,3))

scatter3(oo(:,1),oo(:,2),oo(:,3),50,'.')
scatter3(oo1(:,1),oo1(:,2),oo1(:,3),50,'.')
scatter3(oo2(:,1),oo2(:,2),oo2(:,3),50,'.')
```

```
scatter3(oo3(:,1),oo3(:,2),oo3(:,3),50,'.')
scatter3(oo4(:,1),oo4(:,2),oo4(:,3),50,'.')
scatter3(oo5(:,1),oo5(:,2),oo5(:,3),300,'*')
```