



# “华为杯”第十四届中国研究生 数学建模竞赛

上海理工大学

学 校

10252101

参赛队号

1.王绍彬

队员姓名

2.皮特尔

3.梁康

参赛密码 \_\_\_\_\_

(由组委会填写)



## “华为杯”第十四届中国研究生 数学建模竞赛

题 目                    多波次导弹发射中的规划问题

---

### 摘                    要：

本文对多波次导弹发射规划问题进行了研究,对导弹发射车辆的行车路由选择,转载位置的合理部署,新发射车辆的替换选择,敌方破坏道路节点位置预测等问题建立了相应的数学问题进行了研究。

针对问题一,对于整体暴露时间最短目标,分析了所有车辆暴露的总时长的主要构成为所有车辆的路程行驶时间。同时,针对行车路线,使用[节点,到达时间,离开时间]有序三元组序列表示行车路线计划作为问题决策变量。针对网络中的单行道路以及主干道路通行给出了单车道相向行驶与同向行驶的行车约束,并结合齐射任务约束,转载地域时间等相关约束信息,建立了问题一的理论优化模型。针对该理论模型的求解,设计了一种三阶段的求解策略。三个优化阶段分别从宏观的整体行车任务节点选择,逐步细化至单一路线路由节点的选择,直至微观层面对路线在具体节点的起止停留行为以及行车时间细节进行细致优化。在阶段一中,忽略单行道行车约束等细节,只考虑行程最短,使用网路信息以及装置完成任务行车必须经过的发射点与中转点,建立了0-1规划整数模型,求得使得总体行程路线最短的发射节点与中转节点选择,得到最短任务完成路线的行程加权总长度为4216.853千米。在阶段二,根据阶段一求解得到的最短路线行程任务节点组合,以线路冲突系数最小为方向,确定每辆任务车辆的具体行车路线节点。最后在阶段三中,依据每辆车的具体行车路线,以最大屏蔽保护时间最小暴露时间为方向,在保证发射任务的同时,初始化每辆车的出发时间以及在中转点驻留的时间,并根据行车在单行道上发生的超车与相遇冲突,进一步调

整车辆行程时间。采用 matlab 编程实现阶段二与阶段三的优化策略，最终得到问题一中 24 辆车辆的总曝光时间为 141.6 小时，8459.4 分钟，所有车辆的行车路线时刻安排信息存于指定附件 Excel 文件中。

针对问题二，相较问题 1 的模型，从 J25、J34、J36、J42、J49 节点选择两个新中转区域以协助完成任务减少曝光时间。据此建立 0-1 整数规划模型，采用 Lingo11.0 编程求解本问最终最优的新增节点位置为 J25、J34，新增中转节点后的最优行程总长为 3920.6 千米，比问题 1 结果 4216.4 千米减少了 295.8 千米，整体暴露时间减少了约 9.86 小时。

针对问题三，相较问题 1 的模型，在第二阶段减少 3 辆 C 型车，并从 J04、J06、J08、J13、J14、J15 节点选择新增车辆，建立 0-1 整数规划模型，采用 Lingo11.0 编程求解本问最终最优的新增 C 型装置的初始位置为 J13，J14，J15，这三个节点出发替换原有装置后的最优行程总长为 4087 千米，比为破坏前提高了 4216.4 千米减少了 129.4 千米，整体暴露时间减少了约 4.3 小时。

针对问题四，假设敌军对我方道路节点情况清晰，通过破坏中转节点使其失去功能造成我军任务行车路线大幅增加进而增高暴露时间。以被 3 个中转节点被破坏后的最小总体行程路线最大为目标，建立 0-1 整数规划模型，采用 Lingo11.0 软件编程求解寻找 3 个中转点，求解得到被破坏后使得整体行车路径增大最大的中转节点为 Z01，Z04，Z06，这三个节点被破坏后整体加权最小行程为 5385 千米，比为破坏前提高了 4216.4 千米提高了，整体暴露时间增加约 33.4 小时。

针对问题五，在考虑了最长曝光时间约束的情况下，重构了问题 1 模型，并在三阶段求解的第二阶段利用最长曝光时间约束限制过长路径的生成，使得 3 类车辆路线长度与自身速度比例得到相对平衡。问题五模型最终求解结果为 24 辆车辆的总暴露时间为 145.6 小时，8736 分钟。

**关键词：**多波次任务 多阶段优化 最优节点组合 0-1 整数规划

## 1 问题重述

随着导弹武器系统的不断发展，导弹在未来作战中将发挥越来越重要的作用，导弹作战将是未来战场的主要作战样式之一。

为了提高导弹部队的生存能力和机动能力，常规导弹大都使用车载发射装置，平时在待机地域隐蔽待机，在接受发射任务后，各车载发射装置从待机地域携带导弹沿道路机动到各自指定发射点位实施发射。每台发射装置只能载弹一枚，实施多波次发射时，完成了上一波次发射任务的车载发射装置需要立即机动到转载地域（用于将导弹吊装到发射装置的专门区域）装弹，完成装弹的发射装置再机动至下一波次指定的发射点位实施发射。连续两波次发射时，每个发射点位使用不超过一次。

某部参与作战行动的车载发射装置共有 24 台，依据发射装置的不同大致分为 A、B、C 三类，其中 A、B、C 三类发射装置的数量分别为 6 台、6 台、12 台，执行任务前平均部署在 2 个待机地域（D1，D2）。所属作战区域内有 6 个转载地域（Z01~ Z06）、60 个发射点位（F01~ F60），每一发射点位只能容纳 1 台发射装置。各转载地域最多容纳 2 台发射装置，但不能同时作业，单台转载作业需时 10 分钟。各转载地域弹种类型和数量满足需求。相关道路情况如图 1-1 所示（道路节点 J01~J62），相关要素的坐标数据如附件 1 所示。图 1-1 中主干道路（图中红线）是双车道，可以双车通行；其他道路（图中蓝线）均是单车道，只能在各道路节点处会车。A、B、C 三类发射装置在主干道路上的平均行驶速度分别是 70 公里/小时、60 公里/小时、50 公里/小时，在其他道路上的平均行驶速度分别是 45 公里/小时、35 公里/小时、30 公里/小时。

部队接受发射任务后，需要为每台车载发射装置规划每个波次的发射点位及机动路线，要求整体暴露时间（所有发射装置的暴露时间之和）最短。本问题中的“暴露时间”是指各车载发射装置从待机地域出发时刻至第二波次发射时刻为止的时间，其中发射装置位于转载地域内的时间不计入暴露时间内。暂不考虑发射装置在发射点位必要的技术准备时间和发射后发射装置的撤收时间。

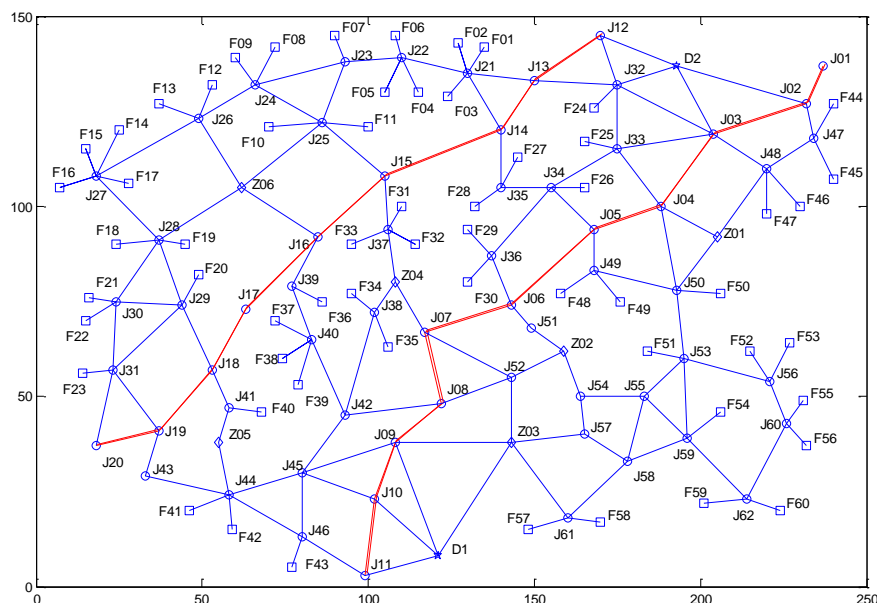


图 1-1 作战区域道路示意图

请你们团队结合实际，建立数学模型研究下列问题：

**问题一：**该部接受到实施两个波次的齐射任务（齐射是指同一波次的导弹同一时刻发射），每个波次各发射 24 枚导弹。给出具体发射点位分配及机动路线方案，使得完成两个波次发射任务的整体暴露时间最短。

**问题二：**转载地域的合理布设是问题的“瓶颈”之一。除已布设的 6 个转载地域外，可选择在道路节点 J25、J34、J36、J42、J49 附近临时增设 2 个转载地域（坐标就取相应节点的坐标）。应该如何布设临时转载地域，使得完成两个波次发射任务的整体暴露时间最短。

**问题三：**新增 3 台 C 类发射装置用于第二波次发射。这 3 台发射装置可事先选择节点 J04、J06、J08、J13、J14、J15 附近隐蔽待机（坐标就取相应节点的坐标），即这 3 台发射装置装弹后从待机地域机动到隐蔽待机点的时间不计入暴露时间内。每一隐蔽待机点至多容纳 2 台发射装置。待第一波次导弹发射后，这 3 台发射装置机动至发射点位参与第二波次的齐射，同时被替代的 3 台 C 类发射装置完成第一波次齐射后择机返回待机地域（返回时间不计入暴露时间）。转载地域仍为事先布设的 6 个的前提下，应该如何选择隐蔽待机点，使得完成两个波次发射任务的整体暴露时间最短。

**问题四：**道路节点受到攻击破坏会延迟甚至阻碍发射装置按时到达指定发射点位。请结合图 1-1 路网特点，考虑攻防双方的对抗博弈，建立合理的评价指标，量化分析该路网最可能受到敌方攻击破坏的 3 个道路节点。

**问题五：**在机动方案的拟制中，既要考虑整体暴露时间尽可能短，也要规避敌方的侦察和打击，采用适当分散机动的策略，同时还要缩短单台发射装置的最长暴露时间。综合考虑这些因素，重新讨论问题（1）。

## 2 符号说明和基本假设

## 2.1 符号说明

$R_i$ : 第  $i$  台车的行车路线计划  $R$   
 $V_{ij}$ : 第  $i$  台车的机动路线中的第  $j$  个节点;  
 $A_{ij}$ : 为第  $i$  台车到达节点  $V_{ij}$  的时间, 显然  $A_{i,1}=0$ ;  
 $L_{ij}$ : 为第  $i$  台车离开节点  $V_{ij}$  的时间;  
 $T$ : 暴露总时间;  
 $T_1$ : 行车总时间;  
 $T_2$ : 站点停留总时间;  
 $T_3$ : 中转点给予屏蔽掩护的总时间;

## 2.2 基本假设

**假设一:** 因各发射点位仅能容纳一台车, 故而每辆车在发射后不会在发射点位停留, 立刻撤离。此时, 在第二波次开始时刻为各车从第一波次发射点为离开的时刻, 此时即第一波次齐射时刻。  
**假设二:** 所有车在路上均保持匀速行驶, 速度为当前路段最高速度, 若两辆车在单车道上同向行驶且在路上追及, 则将其中一车的出发时间延迟, 使二车会车点恰好在节点处而非在路上。(同向行驶过程中, 每台车仅在节点处停车, 不会在道路中停车)。

# 3 问题一分析、建模与求解

## 3.1 问题一分析

### 3.1.1 暴露时间构成

通过对题意的分析, 装载车辆暴露的时间可以表示为

$$T_{\text{暴露总时间}} = \underbrace{T_{\text{行车总时间}}}_{\text{主要}} + T_{\text{站点驻留总时间}} - T_{\text{掩体屏蔽时间}}$$

通过粗略计算与分析可知, 所有**导弹装载车辆暴露时间**的主要构成来自于**行车时间**, 并且所有车辆在行车进程中产生的暴露时间是无法规避和缩短的, 所以对**车辆路径总耗时的优化**是本问题的核心任务。

### 3.1.2 不同速率车辆

根据题意, 不同车辆以及在不同的路况中, 具有不同的车速, 具体列出如下表 1。关于对车辆的使用的选择, 若总行程一定, **充分使用高速车辆能进一步缩短车辆暴露时间**。

表 1 车辆速度信息

车辆类型	速度	
	主干道路	其它道路
A 类(6 辆)	70 公里/小时	45 公里/小时
B 类(6 辆)	60 公里/小时	35 公里/小时
C 类(12 辆)	50 公里/小时	30 公里/小时

### 3.1.3 道路信息与驻点停留

问题中，车辆在主干道形式除了速度不同，还有通行功能的差异。在主干道允许双向通行，而其他道路只能单向形式。其他道路限定只能单向通行，同时本文限定其他单向单行线路不允许超车。

在单行道上，当出现快车遇上慢车，或者遇上车辆相遇时，可采取以如下策略：

**策略 1：**单车道发生相遇时，可调整其中一车的发车时间避开相遇发生；

**策略 2：**单车道发生相遇时，可另其中一车在相遇道路节点驻留等待；

**策略 3：**单车道发生超车时，可调整其中一车的发车时间避开超车；

**策略 4：**单车道发生超车时，可另快车尾随慢车，至下一个节点再超车。

### 3.1.4 单波次任务

考虑单波次的导弹投掷任务中，所有车辆的总暴露时间即所有车辆的行车时间总和得到

$$\text{Min } T_{\text{暴露总时间}} = T_{\text{行车总时间}} = \sum T_{\text{车辆行车时间}} = \sum \frac{D_{\text{车辆路程}}}{V_{\text{车辆速度}}}$$

因为车辆仅需选择

$$T_{\text{车辆A出发时间}} = T_{\text{最长车辆行程耗时}} - T_{\text{车辆A行程耗时}}$$

对每辆车出发时机的选择，在没有其他路况的因素影响下，可以完美的为每辆车屏蔽除了车辆行程以外的行车时间。

### 3.1.5 多波次任务与中转区域

在多波次的任务重中，所有车辆完成一次齐射后需要进入转载区域，转载区域类似单波次任务的出发地，能够为所车辆提供有限的暴露屏蔽。所以在多波次的任务规划中，转载区域的位置选择以及在转载区域停留对缩减暴露总时间具有总要作用。

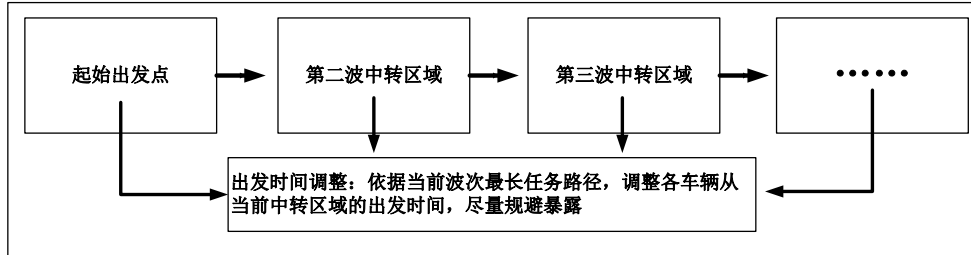


图 1 中转区域出发时间调整示意图

借鉴单波次任务起始时间的选择方式，多波次任务中车辆在中转区域依据执行该波次任务的车辆最长任务时间调整其滞留时间，减少暴露行车时间。

### 3.1.6 问题一求解思路

3.1.1~3.1.5 中对本问题的各种因素的分析，现对本文对该问题的求解思路进行介绍与概括，下图显示思路的结构图。

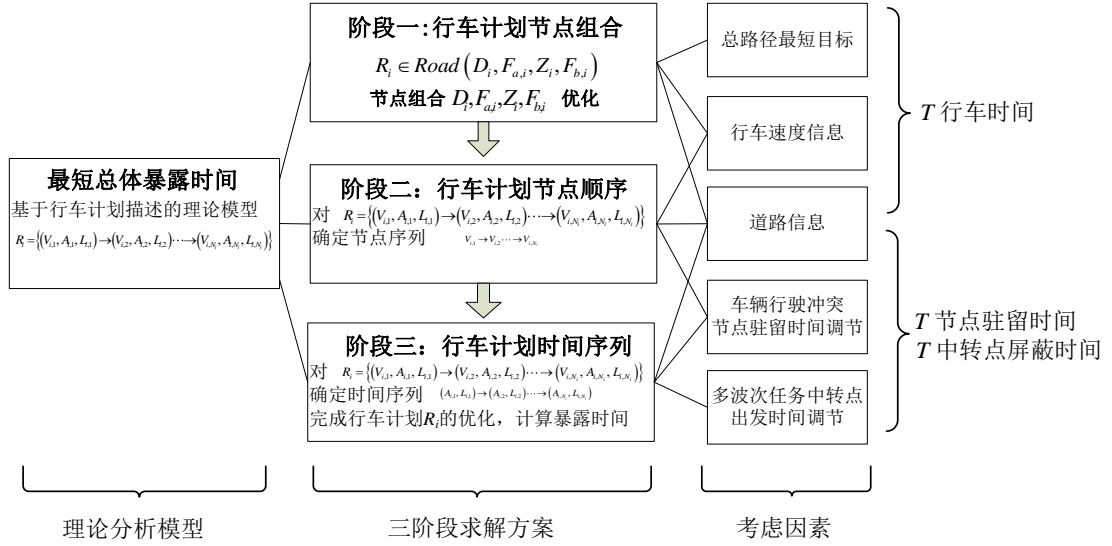


图 2 中转区域出发时间调整示意图

如上图所示，本文拟采用基于行车路线计划描述的理论模型对问题一进行建模分析，并设计了三个阶段的求解方案，对问题行车总路径最小等因素直到车辆行程细节如果中转点调节，行车冲突等细节信息，对问题从宏观的到微观进行全面的剖析与求解。下文 3.2 将对理论分析与构建进行梳理，3.3 将对三阶段求解方案的相关模型与算法进行阐述说明，3.4 将给出本问求解结果。

## 3.2 问题一理论模型建立

### 3.2.1 决策变量：行车路线计划 $R$

本问题要求对所有车辆的详细机动路线与时刻进程进行规划，我们用有序三元序列  $R_i$  表示第  $i$  台车的行车路线计划：

$$R_i = \{(V_{i,1}, A_{i,1}, L_{i,1}) \rightarrow (V_{i,2}, A_{i,2}, L_{i,2}) \cdots \rightarrow (V_{i,j}, A_{i,j}, L_{i,j}) \cdots \rightarrow (V_{i,N_i}, A_{i,N_i}, L_{i,N_i})\}$$

其中  $(V_{i,j}, A_{i,j}, L_{i,j})$  表示车辆  $i$  在其第  $j$  个目的点位的行动信息，其中具体意义如下表

表 2 行车计划  $R$  中三元素含义

元素	意义
$V_{i,j}$	表示第 $i$ 台车的机动路线中的第 $j$ 个节点;
$A_{i,j}$	表示 $i$ 台车到达节点 $V_{i,j}$ 的时间，显然 $A_{i,1}=0$ ;
$L_{i,j}$	表示 $i$ 台车离开节点 $V_{i,j}$ 的时间

$V_{i,j}$  的范围包括 2 个待机地域 ( $D1, D2$ )、6 个转载地域 ( $Z01 \sim Z06$ )、62 个道路节点 ( $J01 \sim J62$ )、60 个发射点位 ( $F01 \sim F60$ )； $N_i$ ：表示  $i$  台车工经过  $N_i$  个节点

### 3.2.2 目标函数

所有车的总体暴露时间  $T$  包括以下几个部分：

$$T = T_1 + T_2 - T_3 \quad (3-1)$$

式中， $T_1$  为线路行驶时间表示所有车在线路上的时间之和， $T_2$  为节点驻留时间表示所有车在出发后经所有节点的驻留时间之和， $T_3$  为转载地域掩护时间表示所



有（ $M$  台）车在六个转载地域（Z01~Z06）驻留被掩护不暴露的时间之和。

#### (1)线路行驶时间 $T_1$

线路行驶时间计算如下

$$T_1 = \sum_{i=1}^M \sum_{j=1}^{N_i-1} \frac{d_i(V_{i,j}, V_{i,j+1})}{v_{i,j}} \quad (3-2)$$

式中， $d_i(V_{i,j}, V_{i,j+1})$  为第  $i$  台车的第  $j$  段路（第  $j$  个节点至第  $j+1$  个节点之间）的长度； $v_{i,j}$  为第  $i$  台车在第  $j$  段路（第  $j$  个节点至第  $j+1$  个节点之间）的行驶速度，对每一台车来说，速度仅有两种取值，主干道路和其它道路两种行驶速度，具体速度信息由车辆和道路状况决定。

#### (2)节点驻留时间

由于除了主干道可以双向通行，单向车道的道路中无法让异向车辆相遇通过，导致行车过程中在节点驻留产生额外的暴露时间

$$T_2 = \sum_{i=1}^M \sum_{j=2}^{N_i} (L_{i,j} - A_{i,j}) \quad (3-3)$$

#### (3)转载区域屏蔽的时间

$$T_3 = \sum_{i=1}^{N_Z} T_{3,i} \quad (3-4)$$

式中， $N_Z$  为转载地域数，本题中取  $N_Z=6$ ； $T_{3,i}$  表示第  $i(i=1, \dots, N_Z)$  个转载地域中所有（ $M$  台）车的不暴露的时间之和，计算如下：

$$T_{3,i} = \sum_{j=1}^{N_{U1,i}} (E_{1,i,j} - S_{1,i,j}) + 2 \sum_{j=1}^{N_{U2,i}} (E_{2,i,j} - S_{2,i,j}) \quad (3-5)$$

式中， $N_{U1,i}$  为当第  $i$  个转载地域中仅有 1 台车时的时间闭区间集合  $U_{1,i}$  的子集数， $N_{U2,i}$  为当第  $i$  个转载地域中有 2 台及以上车时的时间闭区间集合  $U_{2,i}$  的子集数；集合  $U_{1,i}$  与  $U_{2,i}$  分别如下：

$$U_{1,i} = [E_{1,i,1}, S_{1,i,1}] \cup \dots \cup [E_{1,i,j}, S_{1,i,j}] \cup \dots \cup [E_{1,i,N_{U1,i}}, S_{1,i,N_{U1,i}}] \quad (3-6)$$

$$U_{2,i} = [E_{2,i,1}, S_{2,i,1}] \cup \dots \cup [E_{2,i,j}, S_{2,i,j}] \cup \dots \cup [E_{2,i,N_{U2,i}}, S_{2,i,N_{U2,i}}] \quad (3-7)$$

式中， $E$ 、 $S$  分别为各闭区间的起止时间。

#### (4)综合目标函数

结合问题一要求，目标为完成两个波次发射任务的整体暴露时间最短，则综合目标函数可表示如下

$$\min T = \sum_{i=1}^M \sum_{j=1}^{N_i-1} \frac{d_i(V_{i,j}, V_{i,j+1})}{v_{i,j}} + \sum_{i=1}^M \sum_{j=2}^{N_i} (L_{i,j} - A_{i,j}) - \sum_{i=1}^{N_Z} T_{3,i} \quad (3-8)$$

其中当所有车辆的路由决策信息

$$R_i = \{(V_{i,1}, A_{i,1}, L_{i,1}) \rightarrow (V_{i,2}, A_{i,2}, L_{i,2}) \cdots \rightarrow (V_{i,j}, A_{i,j}, L_{i,j}) \cdots \rightarrow (V_{i,N_i}, A_{i,N_i}, L_{i,N_i})\}$$

确定时，目标函数值中相应的未知变量可进一步求出并确定目标函数值。

### 3.2.3 约束分析

### (1) 行车路线计划集合 $Road(D_a, F_a, Z_a, F_b)$

在两个波次的齐射任务中，必须从待机地域（记为  $D_a$ ）转出，然后经过若干节点（可以为 4 中类型的任意节点），到达第 1 波次齐射任务的发射点位（记为  $F_a$ ）；然后返回转载地域（记为  $Z_a$ ）重新装弹，再经过若干节点（可以为 4 中类型的任意节点），到达第 2 波次齐射任务的发射点位（记为  $F_b$ ）。可以看出，整个过程中，必须顺次包含 4 个节点，即  $D \rightarrow F_a \rightarrow Z \rightarrow F_b$ 。记第顺次经过  $D \rightarrow F_a \rightarrow Z \rightarrow F_b$  的所有路线的集合为

$$Road(D, F_a, Z, F_b) = \{ \forall R | (D, A_D, L_D) \rightarrow \dots \rightarrow (F_a, A_{Fa}, L_{Fa}) \rightarrow \dots \rightarrow (Z, A_Z, L_Z) \rightarrow \dots \rightarrow (F_b, A_{Fb}, L_{Fb}) \} \quad (3-9)$$

那么对于第  $i$  台车辆的行车路线计划  $R_i$  中由  $D_{a,i}$  出发顺次经过  $F_{a,i}, Z_{a,i}, F_{b,i}$  两个发射点和一个中转点位，那么其行车计划  $R_i$  有

$$s.t. \quad R_i \in Road(D_{a,i}, F_{a,i}, Z_{a,i}, F_{b,i}) \quad \forall i = 1, \dots, M \quad (3-10)$$

任意中转区域点可以车辆重复选择，但是每个发射点位只能使用一次，即对于任意行车计划  $R_i$  的发射点位有

$$s.t. \quad F_{a,i} \neq F_{b,i} \neq F_{a,j} \neq F_{b,j}, \quad \forall i \neq j, \quad i, j = 1, 2, \dots, M,$$

### (2) 相邻节点连接约束

每台车的线路方案中，相邻节点必须相连接，即

$$s.t. \quad Adj(V_{i,j}, V_{i,j+1}) = 1 \quad \forall i = 1, \dots, M; j = 1, \dots, N_i - 1 \quad (3-11)$$

式中， $Adj$  为所有节点的邻接矩阵，两个节点直接连接则相应元素为 1，否则为 0。

### (3) 时间顺序约束

该约束包含两部分，其一为，对同一节点，到达时刻要早于离开时刻，即

$$s.t. \quad A_{i,j} \leq L_{i,j} \quad \forall i = 1, \dots, M; j = 1, \dots, N_i \quad (3-12)$$

其二为，对相邻节点，下一节点的到达时刻等于上一节点的离开时刻加道路时间，即

$$s.t. \quad A_{i,j+1} = L_{i,j} + \frac{d_i(V_{i,j}, V_{i,j+1})}{v_{i,j}} \quad \forall i = 1, \dots, M; j = 1, \dots, N_i - 1 \quad (3-13)$$

### (4) 转载地域时间约束

每台车在转载地域进行转载作业  $T_Z = 10\text{min}$ ，即

$$s.t. \quad L_{i,N_Z} \geq A_{i,N_Z} + T_Z \quad \forall i = 1, \dots, M \quad (3-14)$$

### (5) 齐射任务约束

齐射任务所有车的发射时间应相同，约束如下：

$$\begin{aligned} s.t. \quad L_{1,Fa} &= \dots = L_{i,Fa} = \dots = L_{M,Fa} \\ L_{1,Fb} &= \dots = L_{i,Fb} = \dots = L_{M,Fb} \end{aligned} \quad (3-15)$$

### (6) 单车道相向行驶约束

任意两车不能在单车道相向行驶。假设第  $i$ 、 $m$  台车的线路方案均含有同一段单车道，但方向相反；两台车在该段的时间区间分别为  $(L_{i,j}, A_{i,j+1})$  和  $(L_{m,n}, A_{m,n+1})$ 。以第  $i$  台车先发车为例，此时两车在时间轴上需满足图 3 情形。

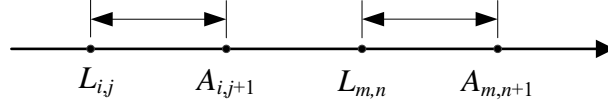


图 3 时间轴分布

该情况约束如下：

$$\begin{aligned}
 s.t. \quad & A_{i,j+1} \leq L_{m,n} \quad OR \quad A_{m,n+1} \leq L_{i,j} \\
 & \forall V_{i,j} = V_{m,n+1} \text{ 且 } V_{i,j+1} = V_{m,n} \text{ 且 } W_{i,j} = W_{m,n} = 1 \\
 & \forall i = 1, \dots, M; m = 1, \dots, M; i \neq m
 \end{aligned} \tag{3-16}$$

式中， $W_{i,j}$ 表示第  $i$  台车的第  $j$  段路的权重，1 表示单车道，2 表示双车道。

### (7) 单车道同向行驶约束

任意两车不能在单车道追及。对任意同向行驶的两车，假设第  $i$ 、 $m$  台车的线路方案均含有同一段单车道，且方向相同；两台车在该段的时间区间分别为  $(L_{i,j}, A_{i,j+1})$  和  $(L_{m,n}, A_{m,n+1})$ 。以第  $i$  台车先发车为例，此时车辆在时间轴上需满足图 3-2 情形。

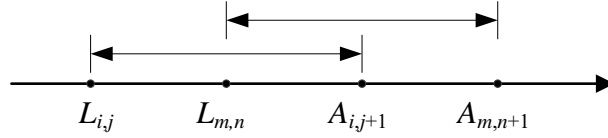


图 4 时间轴分布

该情况约束如下：

$$\begin{aligned}
 s.t. \quad & L_{i,j} \leq L_{m,n} \text{ 且 } A_{i,j+1} \leq A_{m,n+1} \quad OR \quad L_{m,n} \leq L_{i,j} \text{ 且 } A_{m,n+1} \leq A_{i,j+1} \\
 & \forall V_{i,j} = V_{m,n} \text{ 且 } V_{i,j+1} = V_{m,n+1} \text{ 且 } W_{i,j} = W_{m,n} = 1 \\
 & \forall i = 1, \dots, M; m = 1, \dots, M; i \neq m
 \end{aligned} \tag{3-17}$$

### 3.2.4 问题一模型

根据上述 3.2.1~3.2.3 的分析，对问题 1 求最少暴露时间的理论综合模型可以建立如下

$$\min T = \sum_{i=1}^M \sum_{j=1}^{N_i-1} \frac{d_i(V_{i,j}, V_{i,j+1})}{v_{i,j}} + \sum_{i=1}^M \sum_{j=2}^{N_i} (L_{i,j} - A_{i,j})_2 - \sum_{i=1}^{N_Z} T_{3,i}$$

$$\begin{cases}
L_{1,Fa} = \dots = L_{i,Fa} = \dots = L_{M,Fa} \\
L_{1,Fb} = \dots = L_{i,Fb} = \dots = L_{M,Fb} \\
L_{i,N_Z} \geq A_{i,N_Z} + T_Z \quad \forall i=1, \dots, M \\
A_{i,j+1} = L_{i,j} + \frac{d_{i,j}}{v_{i,j}} \quad \forall i=1, \dots, M; j=1, \dots, N_i - 1 \\
A_{i,j} \leq L_{i,j} \quad \forall i=1, \dots, M; j=1, \dots, N_i \\
Adj(V_{i,j}, V_{i,j+1}) = 1 \quad \forall i=1, \dots, M; j=1, \dots, N_i - 1 \\
(A_{i,j+1} - L_{i,j})v_{i,j} \geq (A_{i,j+1} - L_{m,n})v_{m,n} \\
A_{i,j+1} \leq L_{m,n} \\
\forall V_{i,j} = V_{m,n+1} \text{ 且 } V_{i,j+1} = V_{m,n} \text{ 且 } W_{i,j} = W_{m,n} = 1 \text{ 且 } L_{i,j} \leq L_{m,n} \\
\forall i=1, \dots, M; m=1, \dots, M; i \neq m \\
R_i(D_i, F_{a,i}, Z_i, F_{b,i}) \in Road(D_i, F_{a,i}, Z_i, F_{b,i}), i=1, 2, \dots, M \\
F_{a,i} \neq F_{b,i} \neq F_{a,j} \neq F_{b,j}, \forall i \neq j, \quad i, j=1, 2, \dots, M,
\end{cases}$$

**决策变量：**第  $i$  辆车的行车计划  $R_i(D_i, F_{a,i}, Z_i, F_{b,i}) \in Road(D_i, F_{a,i}, Z_i, F_{b,i})$  其中  $D_i, F_{a,i}, Z_i, F_{b,i}$  分别为第  $i$  辆车的出发点  $D_i$ ，第一波射击点位  $F_{a,i}$ ，中转区域  $Z_i$  以及第二波射击点位  $F_{b,i}$ ， $i=1, 2, \dots, M$ 。

### 3.3 问题 1 求解

#### 3.3.1 问题 1 三阶段求解方案

根据 3.2 中对问题理论模型的建立与分析过程，可发现直接对该问题进行规划求解十分困难，本文考虑将本文问题的求解拆分成 3 个阶段进行分析求解，同时该阶段也相应的对每条路径规划。下图显示了这三个阶段的流程图示。

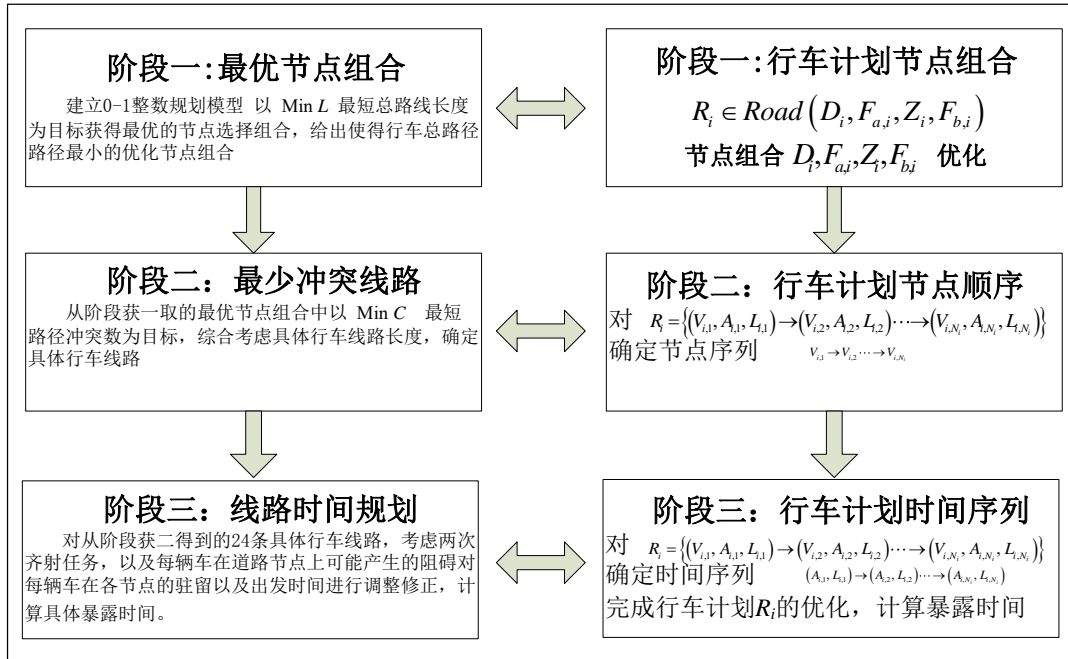


图 5.三阶段求解流程图

首先，在求解的第一个阶段，先忽略道路上可能产生的各种行车冲突以及出发与到达时间的相关约束，针对 24 辆车可以采用的最短路线建立一个 0-1 整数规划模型，仅考虑所有行车路线的获取行车路线总长度最小的节点组合。

在此基础上，在第二阶段通过从已有的节点组合里，依据：

原则 1：使得冲突数最少，便于下一阶段对行程路线的时间规划；

原则 2：使得单一路线尽可能长，在总行程路线确定的情况下，充分利用高速车辆减少暴露时间；

最后，在第三阶段，依据两次齐射时间初始化每条线路的出发时间以及在中转点滞留时间，根据道路中发生的相遇与追车冲突进一步调整。

### 3.3.2 阶段一：最短路线节点组优化

根据前文 3.1.1 的分析，所有车在线路上的时间之和  $T_1$  越小越接近目标，车在道路上的速度恒定，则所有线路长度之和越小，整体暴露时间就越小，越接近目标。对于任意一辆车的行车路线计划，行车计划  $R_i(D_i, F_{a,i}, Z_i, F_{b,i}) \in Road(D_i, F_{a,i}, Z_i, F_{b,i})$  必须其出发点  $D_i$ ，第一波射击点位  $F_{a,i}$ ，中转区域  $Z_i$  以及第二波射击点位  $F_{b,i}$  确定。所有路线必须完成导弹齐射任务需要行经特定数量的发射点，所以对发射点位置以及中转点位置的选择将是影响车辆行车线路长度的主要因素，车辆的出发点已经确定，所以在此阶段，对所有车辆的发射位置和中转点可能位置进行建立 0-1 整数规划模型进行路径最小的寻优。

阶段一对最短线路阶段组合的 0-1 整数优化模型可以建立如下：

#### (1) 决策变量

下表 3 给出了所有 0-1 决策变量。

表 3 最短路线节点组合 0-1 整数规划

0-1 变量	解释
$F_{D1a,i}$	1 代表选择第 $i$ 个发射点位作为从 D1 出发车辆的第 1 波次齐射任务的发射点位 $F_a$ ；0 代表不选择； $N_F = 60$ ，为发射点位总数， $i = 1, \dots, N_F$
$F_{D1b,i}$	1 代表选择第 $i$ 个发射点位作为从 D1 出发车辆的第 2 波次齐射任务的发射点位 $F_b$ ；0 代表不选择， $i = 1, \dots, N_F$ ；
$F_{D2a,i}$	1 代表选择第 $i$ 个发射点位作为从 D2 出发车辆的第 1 波次齐射任务的发射点位 $F_a$ ；0 代表不选择； $i = 1, \dots, N_F$
$F_{D2b,i}$	1 代表选择第 $i$ 个发射点位作为从 D2 出发车辆的第 2 波次齐射任务的发射点位 $F_b$ ；0 代表不选择， $i = 1, \dots, N_F$ ；
$Z_{D1a,i,j}$	1 代表第 $i$ 个发射点位与第 $j$ 个转载地域同时选为，从 D1 出发车辆的第 1 波次齐射任务的发射点位 $F_a$ 与该点位去往的转载地域 $Z_a$ ；0 代表不同时被选择； $N_Z = 6$ ，为转载地域总数； $i = 1, \dots, N_F, j = 1, \dots, N_Z$
$Z_{D1b,i,j}$	1 代表第 $i$ 个发射点位与第 $j$ 个转载地域同时选为，从 D1 出发车辆的第 2 波次齐射任务的发射点位 $F_b$ 与去往该点位的转载地域 $Z_a$ ；0 代表不同时被选择； $i = 1, \dots, N_F, j = 1, \dots, N_Z$

$Z_{D2a,i,j}$	1 代表第 $i$ 个发射点位与第 $j$ 个转载地域同时选为, 从 D2 出发车辆的第 1 波次齐射任务的发射点位 $F_a$ 与该点位去往的转载地域 $Z_a$ ; 0 代表不同时被选择, $i=1, \dots, N_F, j=1, \dots, N_Z$ ;
$Z_{D2b,i,j}$	1 代表第 $i$ 个发射点位与第 $j$ 个转载地域同时选为, 从 D2 出发车辆的第 2 波次齐射任务的发射点位 $F_b$ 与去往该点位的转载地域 $Z_a$ ; 0 代表不同时被选择, $i=1, \dots, N_F, j=1, \dots, N_Z$ 。

## (2) 目标函数

目标为使所有线路长度之和最小, 所有线路长度之和为:

$$d_{tol} = \sum_{i=1}^{N_F} F_{D1a,i} d_{D1,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1a,i} Z_{D1a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1b,i} Z_{D1b,i,j} d_{FZ,i,j} \\ + \sum_{i=1}^{N_F} F_{D2a,i} d_{D2,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2a,i} Z_{D2a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2b,i} Z_{D2b,i,j} d_{FZ,i,j} \quad (3-18)$$

式中,  $d_{D1,i}$ 、 $d_{D2,i}$ 、 $d_{FZ,i,j}$  分别表示 D1 与第  $i$  个发射点位的最短距离、D2 与第  $i$  个发射点位的最短距离、第  $i$  个发射点位与第  $j$  个转载地域的最短距离。该最短距离可利用使用 **Dijkstra** 算法求得。

Dijkstra 算法中计算的距离计算按照由于车辆在单车道和双车道行驶速度不同, 做如下近似处理: 分别求得 A、B、C 三类发射装置在双车道与单车道上行行驶速度比, 然后根据三类车数量比重进行加权平均, 得到双车道与单车道行驶速度比的平均值为:

$$\frac{70}{45} \times \frac{6}{24} + \frac{60}{35} \times \frac{6}{24} + \frac{50}{30} \times \frac{12}{24} \approx 1.65 \quad (3-19)$$

将双车道长度除以 1.65, 换算为单车道长度后进行线路寻优。

## (3) 约束

①车辆约束: 以下数量均为  $M/2=12$ :

$$s.t. \quad \sum_{i=1}^{N_F} F_{D1a,i} = \sum_{i=1}^{N_F} F_{D1b,i} = \sum_{i=1}^{N_F} F_{D2a,i} = \sum_{i=1}^{N_F} F_{D2b,i} = \frac{M}{2} \quad (3-20)$$

②发射点位不重复约束: 每个发射点位最多使用 1 次。

约束如下:

$$s.t. \quad F_{D1a,i} + F_{D1b,i} + F_{D2a,i} + F_{D2b,i} \leq 1 \quad \forall i=1, \dots, N_F \quad (3-21)$$

③转载地域与发射点位相对应约束

$$s.t. \quad \begin{cases} \sum_{j=1}^{N_Z} Z_{D1a,i,j} = F_{D1a,i} \\ \sum_{j=1}^{N_Z} Z_{D1b,i,j} = F_{D1b,i} \\ \sum_{j=1}^{N_Z} Z_{D2a,i,j} = F_{D2a,i} \\ \sum_{j=1}^{N_Z} Z_{D2b,i,j} = F_{D2b,i} \end{cases} \quad \forall i=1, \dots, N_F \quad (3-22)$$

④转载地域进入车辆与出发车辆相对应约束

$$s.t. \begin{cases} \sum_{i=1}^{N_F} Z_{D1a,i,j} = \sum_{i=1}^{N_F} Z_{D1b,i,j} \\ \sum_{i=1}^{N_F} Z_{D2a,i,j} = \sum_{i=1}^{N_F} Z_{D2b,i,j} \end{cases} \quad \forall j=1, \dots, N_Z \quad (3-23)$$

求解该 0-1 规划，可直接得到  $M$  条线路中分别顺次对应的  $D_a$ 、 $F_a$ 、 $Z_a$ 、 $F_b$  四个点的可选组合范围，并且在这个组合范围内选择的路径总长度都是相同的。

### 3.3.3 阶段二：最少冲突数路径选择

在阶段一中，得到  $M$  条线路中分别顺次对应的  $D_a$ 、 $F_a$ 、 $Z_a$ 、 $F_b$  四个点的可选组合范围，但并没有确定具体的 1-1 对应的路线。通过确定 1-1 对应的路线，可以确定行车计划  $R$  车辆的具体的行驶走向。

在阶段二中，需要对  $M$  条线路的具体路线进行确定，对于 24 辆车的出发点已经确定，阶段一所提供的最小路径长度组合内，为每条路线选择合适的发射点和中转点  $F_a$ 、 $Z_a$ 、 $F_b$  即可确认一种具有确定路径走向的行驶方案。

定义行车计划  $R_i$  的单行道冲突系数  $C(R_i)$  为其该行车计划的冲突系数，

$$\min C(R_i) = \text{冲突路段总数}$$

在阶段二中，可通过简单计算随机组合，并筛选一定数量冲突系数数  $C(R_i)$  较低的备选方案，也可设计专门算法用于对可选方案进行优化筛选，限于篇幅与时间，本文就采用手工加计算的简便获得相应的解。

### 3.3.4 阶段三：线路时间规划

经过阶段二对具体线路的筛选，可以得到线路冲突较少的具体行车计划，但对于每一个行车计划，之前的两个阶段都没有对行车过程可能产生的追击以及相遇状况的发生进行处理，并且没有对每个线路的出发时间，以及在每个节点的驻留时间进行设置。

#### (1) 线路分配

为了使整体暴露时间最短，较长的线路应分配给速度快的车，较短的线路应分配给速度慢的车。基于此原则，为每一台车分配一个线路。

#### (2) 初始化线路各节点到达与出发时间

将整体线路分为两段，称第 1 波次齐射时刻及其之前的时间为第 1 波次，称第 1 波次齐射时刻至第 2 波次齐射时刻的时间为第 2 波次。

##### ①第 1 波次

第  $i$  ( $i=1, \dots, M$ ) 台车的 1 个节点的离开时刻为：

$$L_{i,1} = 0, i=1, \dots, M \quad (3-24)$$

第  $i$  ( $i=1, \dots, M$ ) 台车的第  $j+1$  个节点的到达时刻为：

$$A_{i,j+1} = L_{i,j} + \frac{d_{i,j}}{v_{i,j}}, i=1, \dots, M; j=1, \dots, N_{Fa,i} - 1 \quad (3-25)$$

第  $i$  ( $i=1, \dots, M$ ) 台车的第  $j$  个节点的离开时刻为：

$$L_{i,j} = A_{i,j}, i=1, \dots, M; j=2, \dots, N_{Fa,i} \quad (3-26)$$

$N_{Fa,i}$  为节点  $F_{a,i}$  及其之前的节点总数。

利用上式迭代求解，可以得到第  $i (i=1, \dots, M)$  台车在  $F_{a,i}$  及其之前所有节点的到达时刻与离开时刻。由于节点  $F_{a,i}$  处为齐射，根据约束(4)中“齐射任务所有车的发射时间应相同”的约束式(3-15)，因此  $L_{i,Fa}$  的时间应选为所有值中的最大者，新的值为：

$$L_{i,Fa} = \max \{L_{i,Fa}, i=1, \dots, M\}, i=1, \dots, M \quad (3-27)$$

每台车在节点  $F_{a,i}$  处的等待时间为：

$$T_{i,Fa0} = L_{i,Fa} - A_{i,Fa}, i=1, \dots, M \quad (3-28)$$

## ②第 2 波次

对第  $i (i=1, \dots, M)$  台车的第  $j (j=N_{Fa,i}+1, \dots, N_{Z,i})$  个点，即转载地域  $Z_{a,i}$  及其之前的点：

第  $i (i=1, \dots, M)$  台车的第  $j+1$  个节点的到达时刻为：

$$A_{i,j+1} = L_{i,j} + \frac{d_{i,j}}{v_{i,j}}, i=1, \dots, M; j = N_{Fa,i}, \dots, N_i - 1 \quad (3-29)$$

第  $i (i=1, \dots, M)$  台车的第  $j$  个节点的离开时刻为：

$$L_{i,j} = A_{i,j}, i=1, \dots, M; j = N_{Fa,i} + 1, \dots, N_i, j \neq N_{Z,i} \quad (3-30)$$

由于第  $N_{Z,i}$  个点为转载地域，要转载作业  $T_Z=10\text{min}$ ，因此有

$$L_{i,N_Z} = A_{i,N_Z} + T_Z, i=1, \dots, M \quad (3-31)$$

同样，第 2 波次齐射时也应进行修正， $L_{i,N}$  的时间应选为所有值中的最大者，新的值为：

$$L_{i,N} = \max \{L_{i,N}, i=1, \dots, M\}, i=1, \dots, M \quad (3-32)$$

每台车在节点  $F_{b,i}$  处的等待时间为：

$$T_{i,N0} = L_{i,N} - A_{i,N}, i=1, \dots, M \quad (3-33)$$

至此，得到不考虑单车道行驶约束的线路方案：

$$R_i = \{(V_{i,j}, A_{i,j}, L_{i,j}), j=1, \dots, N_i\}, i=1, \dots, M。$$

## (3) 修正单行道约束的相遇超车问题

在得到初始化后的行车时间规划里，存在着在单行道上的相遇与赶超问题。需要根据以下表中所描述的情形进行修正。

表 5 行车冲突调整手段

	情形	调整方式
第一波齐射前	相遇	由出发时间更早的车辆提早 $k$ 分钟出发， $k$ 的值为正好使得两车在相遇弧的终点相遇；
	超车	由速度更慢的提早 $k$ 分钟出发， $k$ 的值为正好使得两车在超车弧相遇弧的终点相遇；
第一波齐射后到达中转点	相遇	由在中转点驻留时间更长的车辆在相遇弧前置节点等待，知道相遇车辆到达该节点
	超车	快车尾随慢车到达这段弧的终点，在终点进行超车
第二波齐射前到达中转	相遇	由出发时间更早的车辆提早 $k$ 分钟从出发， $k$ 的值为正好使得两车在相遇弧的终点相遇；
	超车	由速度更慢的提早 $k$ 分钟出发， $k$ 的值为正好使得两车在超车弧相



点		遇遇弧的终点相遇；
---	--	-----------

#### (4) 线路时间规划算法流程

上述对行车线路时间的规划，可以总结成如下图所示算法的流程

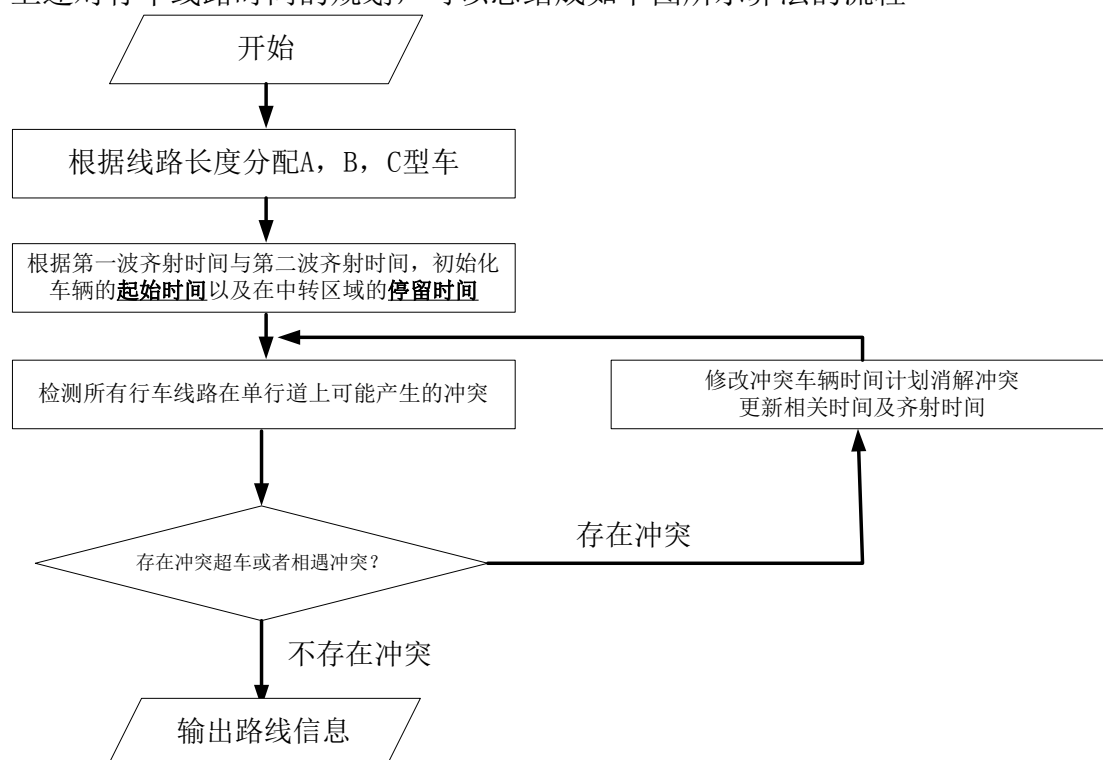
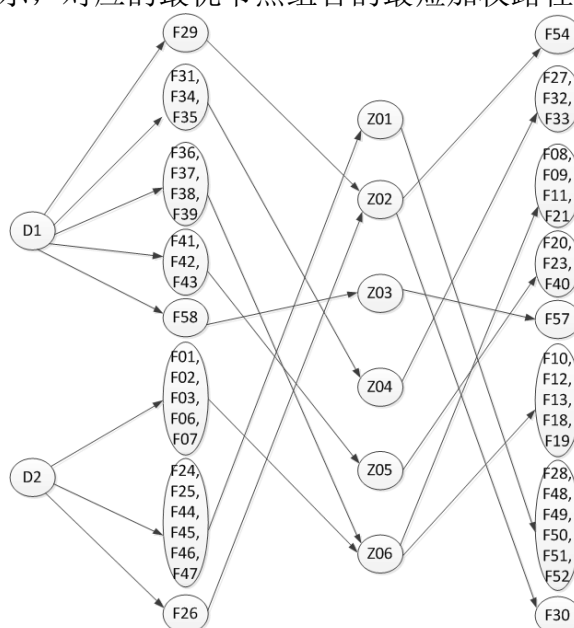


图 6 单向道车辆冲突消除流程

### 3.1 问题 1 求解结果

采用 lingo11.0 编程实现阶段 1 里的 0-1 整数规划模型，求解最终的最优节点组合具体如下图所示，对应的最优节点组合的最短加权路径为 4216.4 千米



采用 matlab 对阶段 2 和阶段 3 的任务进行编程求解，最终得到行车所有车辆的行车路线计划信息存于指定附件 xls 中，最终最短总暴露时间约为 8459.4 分

钟，合计约 140.99 小时。

## 四 问题二、建模与求解

### 4.1 问题二分析

问题二相对问题一，要求在道路节点 J25、J34、J36、J42、J49 附近临时增设 2 个转载地域。由问题一对暴露总时间的分析

$$T_{\text{暴露总时间}} = \underbrace{T_{\text{行车总时间}}}_{\text{主要}} + T_{\text{站点驻留总时间}} - T_{\text{掩体屏蔽时间}}$$

可知，暴露总时间的最主要的影响因素是行车总时间，后期的调整对暴露总时间的影响相对较小。在本问题中，增加转载区域的选择对行车总时间的影响相对较大，故在本问中主要考虑行车总时间的影响。那么，问题一中求解阶段一中，以最小行程总路径为目标的最优节点组合选择的 0-1 模型可以较好地修改移植到本问中。

### 4.2 问题二模型

根据问题一求解极端的 0-1 整数规划模型，针对问题的背景，建立如下 0-1 整数规划模型

#### (1) 决策变量

问题 1 中的表 3 的决策变量依然是本文的决策变量，此外新增了决策变量

表 4 决策变量

0-1 变量	解释
$M_X$	1 代表地域 X 被选为转载地域，0 表示未被选择 $X \in \{J25, J34, J36, J42, J49\}$
$Z_{D1a,i,X}$	1 代表第 i 个发射点位与转载地域 X 同时选为，从 D1 出发车辆的第 1 波次齐射任务的发射点位 $F_a$ 与该点位去往的转载地域 $Z_a$ ；0 代表不同时被选择； $N_Z = 6$ ，为转载地域总数； $X \in \{J25, J34, J36, J42, J49\}, i = 1, \dots, N_F$
$Z_{D1b,i,X}$	1 代表第 i 个发射点位与转载地域 X 同时选为，从 D1 出发车辆的第 2 波次齐射任务的发射点位 $F_b$ 与去往该点位的转载地域 $Z_a$ ；0 代表不同时被选择； $X \in \{J25, J34, J36, J42, J49\}, i = 1, \dots, N_F$
$Z_{D2a,i,X}$	1 代表第 i 个发射点位与转载地域 X 同时选为，从 D2 出发车辆的第 1 波次齐射任务的发射点位 $F_a$ 与该点位去往的转载地域 $Z_a$ ；0 代表不同时被选择； $X \in \{J25, J34, J36, J42, J49\}, i = 1, \dots, N_F$
$Z_{D2b,i,j}$	1 代表第 i 个发射点位与转载地域 X 同时选为，从 D2 出发车辆的第 2 波次齐射任务的发射点位 $F_b$ 与去往该点位的转载地域 $Z_a$ ；0 代表不同时被选择 $X \in \{J25, J34, J36, J42, J49\}, i = 1, \dots, N_F$ 。

#### (2) 目标函数

问题 2 中，最小路线目标函数表示为

$$\begin{aligned} & \min d_1 + d_2 \\ d_1 = & \sum_{i=1}^{N_F} F_{D1a,i} d_{D1,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1a,i} Z_{D1a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1b,i} Z_{D1b,i,j} d_{FZ,i,j} \\ & + \sum_{i=1}^{N_F} F_{D2a,i} d_{D2,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2a,i} Z_{D2a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2b,i} Z_{D2b,i,j} d_{FZ,i,j} \end{aligned}$$

$$d_2 = \sum_{i=1}^{N_F} \sum_{X \in \{J25, J34, J36, J42, J49\}} F_{D1a,i} Z_{D1a,i,X} d_{FZ,i,X} + \sum_{i=1}^{N_F} \sum_{X \in \{J25, J34, J36, J42, J49\}} F_{D1b,i} Z_{D1b,i,X} d_{FZ,i,X}$$

其中  $d_1$  表示选择原来中转区域作为节点产生的线路距离总和, 其中  $d_2$  表示选择新增中转区域作为节点产生的线路距离总和,  $d_{FZ,i,X}$  表示节点  $i$  到中转区域  $X$  的最短距离, 有 dijkstra 算法计算得到。

### (3) 约束

转载地域与发射点位相对应约束:

$$s.t. \begin{cases} \sum_{X \in \{J25, J34, J36, J42, J49\}} Z_{D1a,i,X} + \sum_{j=1}^{N_Z} Z_{D1a,i,j} = F_{D1a,i} \\ \sum_{X \in \{J25, J34, J36, J42, J49\}} Z_{D1b,i,X} + \sum_{j=1}^{N_Z} Z_{D1b,i,j} = F_{D1b,i} \\ \sum_{X \in \{J25, J34, J36, J42, J49\}} Z_{D2a,i,X} + \sum_{j=1}^{N_Z} Z_{D2a,i,j} = F_{D2a,i} \\ \sum_{X \in \{J25, J34, J36, J42, J49\}} Z_{D2b,i,X} + \sum_{j=1}^{N_Z} Z_{D2b,i,j} = F_{D2b,i} \end{cases} \quad \forall i = 1, \dots, N_F$$

转载地域进入车辆与出发车辆相对应约束

$$s.t. \begin{cases} \sum_{i=1}^{N_F} Z_{D1a,i,j} = \sum_{i=1}^{N_F} Z_{D1b,i,j} \\ \sum_{i=1}^{N_F} Z_{D2a,i,j} = \sum_{i=1}^{N_F} Z_{D2b,i,j} \end{cases} \quad \forall j \in \{1, \dots, N_Z\} \cup \{J25, J34, J36, J42, J49\}$$

车辆数目约束:

$$\sum_{i=1}^{N_F} F_{D1a,i} = \sum_{i=1}^{N_F} F_{D1b,i} = \sum_{i=1}^{N_F} F_{D2a,i} = \sum_{i=1}^{N_F} F_{D2b,i} = \frac{M}{2}$$

发射点位不重复约束:

$$F_{D1a,i} + F_{D1b,i} + F_{D2a,i} + F_{D2b,i} \leq 1 \quad \forall i = 1, \dots, N_F$$

新增转载区域个数 2 的约束

$$M_{J25} + M_{J34} + M_{J36} + M_{J42} + M_{J49} = 2$$

转载区域的选择约束

$$\begin{cases} Z_{D1a,i,X} \leq M_X \\ Z_{D2a,i,X} \leq M_X \\ Z_{D1b,i,X} \leq M_X \\ Z_{D2b,i,X} \leq M_X \end{cases}, i = 1, 2, \dots, N_Z, X \in \{J25, J34, J36, J42, J49\}$$

### 4.3 问题二求解与求解结果

采用 Lingo11.0 编程求解本问最终最优的新增节点位置为 J25、J34, 新增中转节点后的最优行程总长为 3920.6 千米, 比问题 1 结果 4216.4 千米减少了 295.8 千米, 整体暴露时间减少了约 9.86 小时。

## 五 问题三、建模与求解

### 5.1 问题三分析

问题三相对问题一，新增 3 台 C 类发射装置用于第二波次发射。这 3 台发射装置可事先选择节点 J04、J06、J08、J13、J14、J15 附近隐蔽待机。同样是对节点的选择变动，在本文中仍可以使用最小行程总路径为目标的最优节点组合选择的 0-1 模型来对事先结点进行选择

### 5.2. 问题三模型

#### (1)决策变量

问题 1 中的表 3 的决策变量依然是本文的决策变量，此外新增了决策变量  
表 4 决策变量

0-1 变量	解释
$F_{Xb,i}^{k_1}$	1 代表第 $i$ 个发射点位否被新增的车辆装 k1 使用，0 表示未被使用 $X \in \{J04, J06, J08, J13, J14, J15\}, i = 1, \dots, N_F$
$F_{Xb,i}^{k_2}$	1 代表第 $i$ 个发射点位否被新增的车辆装 k2 使用，0 表示未被使用 $X \in \{J04, J06, J08, J13, J14, J15\}, i = 1, \dots, N_F$
$F_{Xb,i}^{k_3}$	1 代表第 $i$ 个发射点位否被新增的车辆装 k3 使用 0 表示未被使用 $X \in \{J04, J06, J08, J13, J14, J15\}, i = 1, \dots, N_F$

表示发射点  $i$  是否被新增的车辆装置 k1, k2, k3 选取。

#### (2)目标函数

$$\begin{aligned}
 d_{tol} = & \sum_{i=1}^{N_F} F_{D1a,i} d_{D1,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1a,i} Z_{D1a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1b,i} Z_{D1b,i,j} d_{FZ,i,j} \\
 & + \sum_{i=1}^{N_F} F_{D2a,i} d_{D2,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2a,i} Z_{D2a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2b,i} Z_{D2b,i,j} d_{FZ,i,j} \\
 & + \sum_{i=1}^{N_F} \sum_{X \in \{J04, J06, J08, J13, J14, J15\}} (F_{Xb,i}^{k_1} d_{X,i} + F_{Xb,i}^{k_2} d_{X,i} + F_{Xb,i}^{k_3} d_{X,i})
 \end{aligned}$$

目标函数中的新增项  $\sum_{i=1}^{N_F} \sum_{X \in \{J04, J06, J08, J13, J14, J15\}} F_{Xb,i}^{k_1} d_{X,i}$  表示新增节点所产生暴露路程，其中  $d_{X,i}$  表示节点  $X$  到发射点  $i$  的最短距离由 Dijkstra 算法计算得到。

#### (3)约束

##### 车辆约束：

第一波发射点选取数量不变，发射点选取数量约束修减少了，替换为

$$s.t. \quad \sum_{i=1}^{N_F} F_{D1a,i} = \sum_{i=1}^{N_F} F_{D2a,i} = \frac{M}{2}$$

$$\sum_{i=1}^{N_F} F_{D1b,i} + \sum_{i=1}^{N_F} F_{D2b,i} = \frac{M}{2} - 3$$

$$\sum_{i=1}^{N_F} \sum_{X \in \{J04, J06, J08, J13, J14, J15\}} F_{Xb,i}^{k_1} = 3$$

##### 节点前后选择约束：

由于新增车辆替换了缘由节点，所以对同一个中转点而言，第一波选择的发

射点数目多于第二波选择的中转点数目，但不超过 3 个

$$s.t. \begin{cases} 3 \geq \sum_{i=1}^{N_F} Z_{D1a,i,j} - \sum_{i=1}^{N_F} Z_{D1b,i,j} \geq 0 \\ 3 \geq \sum_{i=1}^{N_F} Z_{D2a,i,j} - \sum_{i=1}^{N_F} Z_{D2b,i,j} \geq 0 \end{cases} \quad \forall j = 1, \dots, N_Z$$

射点位不重复约束：

发射点不重复，加入了新车辆后约束原修正为

$$s.t. \quad F_{Xb,i}^{k_1} + F_{Xb,i}^{k_2} + F_{Xb,i}^{k_3} + F_{D1a,i} + F_{D1b,i} + F_{D2a,i} + F_{D2b,i} \leq 1$$

其中  $\forall i = 1, \dots, N_F, \forall X \in \{J04, J06, J08, J13, J14, J15\}$

转载地域与发射点位相对应约束：

$$s.t. \begin{cases} \sum_{j=1}^{N_Z} Z_{D1a,i,j} = F_{D1a,i} \\ \sum_{j=1}^{N_Z} Z_{D1b,i,j} = F_{D1b,i} \\ \sum_{j=1}^{N_Z} Z_{D2a,i,j} = F_{D2a,i} \\ \sum_{j=1}^{N_Z} Z_{D2b,i,j} = F_{D2b,i} \end{cases} \quad \forall i = 1, \dots, N_F$$

### 5.3 问题三求解

对于问题二可以沿用问题一的三阶段求解方案，第二阶段与第三阶段的求解过程不变，而阶段一的求解模型可以采用如下两种方式，采用 Lingo11.0 编程求解本问最终最优的新增节点位置为 J13, J14, J15，这三个节点出发替换原有装置后的最优行程总长为 4087 千米，比为破坏前提高了 4216.4 千米减少了 317.4 千米，整体暴露时间减少了约 4.3 小时。

## 六 问题四、建模与求解

### 6.1 问题四分析

在问题 4 中要求考虑攻防双方的对抗博弈下，该网路中最优可能被地方破坏的网络节点。本文考虑地方对我方中转区域进行打击，被打击的中转节点将不再提供中转弹药提供以及隐蔽功能。

假设地方了解我方中转区域进的所有位置，将会选择打击后使得我方车辆暴露时间最大的节点进行打击，本文以此作为目标进行建模研究。

### 6.2 问题四模型

#### (1) 决策变量

问题 1 中的表 3 的决策变量依然是本文的决策变量，此外新增了决策变量

表 4 决策变量

0-1 变量	解释
$Z_i$	1 代表第 i 个中间节点被摧毁，0 代表不被摧毁， $i=1,2,3,4,5,6$

## (2)目标函数

在问题四中，希望找到使得打击后暴露时间最大的节点，令  $\min d_{tol}$  表示打击后的所有行车路线总路程，那敌方需要打击 3 个节点使得被打击后的行车总路程最大，那么目标函数可以表示如下

$$\max \min d_{tol}$$

其中  $d_{tol}$  表示被打击后的行车路线综合。

$$\begin{aligned} d_{tol} = & \sum_{i=1}^{N_F} F_{D1a,i} d_{D1,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1a,i} Z_{D1a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D1b,i} Z_{D1b,i,j} d_{FZ,i,j} \\ & + \sum_{i=1}^{N_F} F_{D2a,i} d_{D2,i} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2a,i} Z_{D2a,i,j} d_{FZ,i,j} + \sum_{i=1}^{N_F} \sum_{j=1}^{N_Z} F_{D2b,i} Z_{D2b,i,j} d_{FZ,i,j} \end{aligned}$$

## (3)约束

转载地域与发射点位相对应约束：

$$s.t. \quad \begin{cases} \sum_{j=1}^{N_Z} Z_{D1a,i,j} = F_{D1a,i} \\ \sum_{j=1}^{N_Z} Z_{D1b,i,j} = F_{D1b,i} \\ \sum_{j=1}^{N_Z} Z_{D2a,i,j} = F_{D2a,i} \\ \sum_{j=1}^{N_Z} Z_{D2b,i,j} = F_{D2b,i} \end{cases} \quad \forall i = 1, \dots, N_F$$

车辆约束：

$$s.t. \quad \sum_{i=1}^{N_F} F_{D1a,i} = \sum_{i=1}^{N_F} F_{D1b,i} = \sum_{i=1}^{N_F} F_{D2a,i} = \sum_{i=1}^{N_F} F_{D2b,i} = \frac{M}{2}$$

转载区域摧毁约束：

$$s.t. \quad \begin{cases} \sum_{i=1}^{N_F} Z_{D1a,i,j} = \sum_{i=1}^{N_F} Z_{D1b,i,j} \\ \sum_{i=1}^{N_F} Z_{D2a,i,j} = \sum_{i=1}^{N_F} Z_{D2b,i,j} \end{cases} \quad \forall j = 1, \dots, N_Z$$

射点位不重复约束：

$$F_{D1a,i} + F_{D1b,i} + F_{D2a,i} + F_{D2b,i} \leq 1, \forall i = 1, \dots, N_F$$

中转区域损毁数约束：

$$\sum_{i=1}^6 Z_i = 3$$

中转区域损毁约束：

$$\begin{cases} Z_{D1a,i,j} \leq 1 - Z_i \\ Z_{D1b,i,j} \leq 1 - Z_i \\ Z_{D2a,i,j} \leq 1 - Z_i \\ Z_{D2b,i,j} \leq 1 - Z_i \end{cases}, i = 1, 2, \dots, N_F, j = 1, 2, \dots, N_Z$$

### 6.3 问题四结果

利用 Lingo11.0 软件编程求解寻找逐个寻找 3 个中转点，求解得到被破坏后使得整体行车路径增大最大的中转节点为 Z01, Z04, Z06，这三个节点被破坏后整体加权最小行程为 5385 千米，比破坏前提高了 4216.4 千米提高了，整体暴露时间增加约 33.4 小时。

## 七 问题五分析、建模与求解

### 7.1 问题五分析

在问题 5 中，要求对单辆车辆暴露时间的长度进行限制，所以问题 5 可以很好地沿用问题 1 的理论模型与三阶段求解方法，并结合问题 5 限制单车暴露时常的背景对模型进行改进。

### 7.2 问题五模型

在问题 5 中，基本背景与问题 1 相同，可以沿用问题 1 的模型即

$$\min T = \sum_{i=1}^M \sum_{j=1}^{N_i-1} \frac{d_i(V_{i,j}, V_{i,j+1})}{v_{i,j}} + \sum_{i=1}^M \sum_{j=2}^{N_i} (L_{i,j} - A_{i,j}) - \sum_{i=1}^{N_Z} T_{3,i}$$

$$.t. \begin{cases} L_{1,Fa} = \dots = L_{i,Fa} = \dots = L_{M,Fa} \\ L_{1,Fb} = \dots = L_{i,Fb} = \dots = L_{M,Fb} \\ L_{i,N_Z} \geq A_{i,N_Z} + T_Z \quad \forall i = 1, \dots, M \\ A_{i,j+1} = L_{i,j} + \frac{d_{i,j}}{v_{i,j}} \quad \forall i = 1, \dots, M; j = 1, \dots, N_i - 1 \\ A_{i,j} \leq L_{i,j} \quad \forall i = 1, \dots, M; j = 1, \dots, N_i \\ Adj(V_{i,j}, V_{i,j+1}) = 1 \quad \forall i = 1, \dots, M; j = 1, \dots, N_i - 1 \\ (A_{i,j+1} - L_{i,j})v_{i,j} \geq (A_{i,j+1} - L_{m,n})v_{m,n} \\ A_{i,j+1} \leq L_{m,n} \\ \forall V_{i,j} = V_{m,n+1} \text{ 且 } V_{i,j+1} = V_{m,n} \text{ 且 } W_{i,j} = W_{m,n} = 1 \text{ 且 } L_{i,j} \leq L_{m,n} \\ \forall i = 1, \dots, M; m = 1, \dots, M; i \neq m \\ R_i(D_i, F_{a,i}, Z_i, F_{b,i}) \in Road(D_i, F_{a,i}, Z_i, F_{b,i}), i = 1, 2, \dots, M \\ F_{a,i} \neq F_{b,i} \neq F_{a,j} \neq F_{b,j}, \forall i \neq j, i, j = 1, 2, \dots, M, \end{cases}$$

相较问题一，在问题五中我们加入车辆最长暴露时间限制约束

$$L(R_i(D_i, F_{a,i}, Z_i, F_{b,i})) \leq C$$

其中  $L(R_i(D_i, F_{a,i}, Z_i, F_{b,i}))$  表示第  $i$  辆车的行车路线计划  $R_i$  的暴露时间长度， $C$  表示车辆行驶的最长暴露时间。

### 7.3 问题五求解

根据问题一的三阶段求解过程，可以知道第一阶段的求解结果不影响道路的长度。在第二阶段，对暴露路线的生成时，在问题一种并不考虑生成路径的暴露时间，本文中限制产生的行车计划的路程行驶时间不能超过一定限度  $C$ ，故在第五问中，求解阶段二即是一个在最长暴露时间长度的约束下，寻找最少冲突系数

的的行车计划生成。

$$\begin{aligned} \min C(R_i) &= \text{冲突路段总数} \\ \min C(R_i) &= \text{冲突路段总数} \\ s.t \quad L(R_i) &< C \quad R_i \in \text{Road}(D_i, F_{a,i}, Z_i, F_{b,i}) \end{aligned}$$

## 7.4 问题 5 求解结果

根据 7.3 改进阶段 2 的行车路线生成过程，限制最长暴露时间 9 小时，进一步对问题一进行优化求解，使得 3 类车辆路线长度与自身速度比例得到相对平衡。最终，问题五模型求解结果为 24 辆车辆的总暴露时间为 145.6 小时，8736 分钟。

## 八、模型的评价与展望

### 8.1 模型的评价

**优点：**针对问题 1 到问题 5，本文在问题 1 建立起具有一般性的模型，能够较好的适应这类问题的各种变化情形，如问题 2~4。针对暴露时间最小目标，本文所提的三阶段优化方案，从宏观到微观逐步对目标进行优化求解。对简单赞的问题可直接用阶段一的 0-1 整数规划问题进行分析求解，对复杂细节的线路规划，可以使用三个阶段的求解进一步求出每一条路线计划的详细信息，模型具有通用性。

**不足：**对于问题本文提出的三阶段求解方案，限于比赛时间限制，对于优解的质量并没有一个较好的分析，下一步可以考虑引入人工智能方法对本问题在求解阶段二产生的解进行进一步优化求解。

### 8.2 模型的展望

本文所建立的理论模型具有良好的拓展性，三阶段的求解方法易推广到更多波次任务下的情形，在更多波次的任务规划中，逐波对路径时间进行优化。

## 参考文献

- [1] 姜启源, 谢金星, 叶俊. 数学模型[J]. 2003.
- [2] 季青梅, 辛文芳. 多波次导弹火力打击任务研究[J]. 信息技术与信息化, 2017 (1): 122-128.

附录:

```
function [d,path]=dijkstra(D,s)
```

```
%Dijkstra 最短路算法 Matlab 程序用于  
求从起始点 s 到其它各点的最短路
```

```
%D 为赋权邻接矩阵
```

```
%d 为 s 到其它各点最短路径的长度
```

```
%DD 记载了最短路径生成树
```

```
tic;
```

```
[m,n]=size(D);
```

```
d=inf.*ones(1,m);
```

```
d(1,s)=0;
```

```
dd=zeros(1,m);
```

```
dd(1,s)=1;
```

```
y=s;
```

```
for iii=1:m
```

```
    path{iii,1}(1)=iii;
```

```
end
```



```

while length(find(dd==1))<m
    for i=1:m
        if dd(i)==0
            d(i)=min(d(i),d(y)+D(y,i));
            if d(i)==d(y)+D(y,i)

path{i}=path{y};path{i}(length(path{y,1})
+1)=i;

                end
            end
        end
        ddd=inf;
        for i=1:m
            if dd(i)==0&& d(i)<ddd

ddd=d(i);                %新生成的
最小边

                end
            end
            yy=find(d==ddd);

                for i=1:length(yy)
                    y=yy(1,i);
                    if dd(1,y)~=1
                        dd(1,y)=1;break
                    end
                end
            end

end

toc

function
[pathaim,dist]=dijkstra2(D,s,aim)

%Dijkstra 最短路算法 Matlab 程序用于
求从起始点 s 到其它各点的最短路
%D 为赋权邻接矩阵
%d 为 s 到其它各点最短路径的长度
%DD 记载了最短路径生成树
tic;
[m,n]=size(D);
d=inf.*ones(1,m);

clear all
clc
load Distance_12

RouteD1 = [
    99  4  124
    111  7  90
    112  7  110

```

```

d(1,s)=0;
dd=zeros(1,m);
dd(1,s)=1;
y=s;

for iii=1:m
    path{iii,1}(1)=iii;
end

while length(find(dd==1))<m
    for i=1:m
        if dd(i)==0
            d(i)=min(d(i),d(y)+D(y,i));
            if d(i)==d(y)+D(y,i)

path{i}=path{y};path{i}(length(path{y,1})
+1)=i;

                end
            end
        end
        ddd=inf;
        for i=1:m
            if dd(i)==0&& d(i)<ddd

ddd=d(i);                %新生成的
最小边

                end
            end
            yy=find(d==ddd);

                for i=1:length(yy)
                    y=yy(1,i);
                    if dd(1,y)~=1
                        dd(1,y)=1;break
                    end
                end
            end

pathaim=path{aim};
dist=d(aim);

end

toc

113  7  93
106  8  88
107  8  91
108  8  89
109  8  92
101  6  102
104  6  103.
105  6  97
127  5  128 ] ;

```

```

D1_Car_type = [ 2 1 1 1 3 3 2 3 1 1 2 1 ];

RouteD2 = [
    71 8 80
    72 8 81
    73 8 82
    76 8 83
    77 8 79
    114 3 118
    115 3 119
    116 3 120
    117 3 121
    94 3 122
    95 3 123
    96 3 98 ];

D2_Car_type = [ 2 2 3 3 3 1 1 1 1 1 1 2 ];

D_list1 = zeros( 12,4 );
R_list1 = zeros( 12,30 );
D_list2 = zeros( 12,4 );
R_list2 = zeros( 12,30 );

for i = 1:12

    [r11,d11] = dijkstra2(Distance2, 1,
    RouteD1(i,1));
    [r12,d12] = dijkstra2(Distance2,
    RouteD1(i,1), RouteD1(i,2));
    [r13,d13] = dijkstra2(Distance2,
    RouteD1(i,2), RouteD1(i,3));
    D_list1(i,:) = [ d11 , d12 , d13 ,
    d11+d12+d13 ];
    temp = [ r11(1:end-1) , r12(1:end-1) ,
    r13 ];
    R_list1(i,1:length(temp)) = temp;

    [r21,d21] = dijkstra2(Distance2, 2,
    RouteD2(i,1));
    [r22,d22] = dijkstra2(Distance2,
    RouteD2(i,1), RouteD2(i,2));
    [r23,d23] = dijkstra2(Distance2,
    RouteD2(i,2), RouteD2(i,3));
    D_list2(i,:) = [ d21 , d22 , d23 ,
    d21+d22+d23 ];
    temp = [ r21(1:end-1), r22(1:end-1),
    r23 ];
    R_list2(i,1:length(temp)) = temp;

end

%%%%

V_type = [ 30,50 ; 35,60; 45,70; ];
A_list1 = zeros( 12,30 ); %%%%% D1
出发时刻表

A_list2 = zeros( 12,30 ); %%%%% D2
出发时刻表
L_list1 = zeros( 12,30 ); %%%%% D1
离开时刻表
L_list2 = zeros( 12,30 ); %%%%% D2
离开时刻表
High_way = [ 9:19,20:28 ]; %%%% 高
速路段

TFa = zeros(12,2); %%%% Fa
到达初始时刻表
TFb = zeros(12,2); %%%% Fb
到达初始时刻表
Stay_time1 = zeros(12,2);
Stay_time2 = zeros(12,2);

for i = 1:12
    Index = 1;
    while 1
        if R_list1(i,Index+1) == 0
            break
        end
        Dist = Distance( R_list1(i,Index) ,
        R_list1(i,Index+1) );
        if
            sum( ismember( [R_list1(i,Index) ,
            R_list1(i,Index+1)] , High_way ) ) == 2
                temp = A_list1(i,Index) +
                Dist / V_type( D1_Car_type(i), 2 )*60;
                A_list1(i,Index+1) = temp ;
                L_list1(i,Index+1) = temp ;
            else
                temp = A_list1(i,Index) +
                Dist / V_type( D1_Car_type(i), 1 )*60;
                A_list1(i,Index+1) = temp ;
                L_list1(i,Index+1) = temp ;
            end

            if R_list1(i,Index) ==
            RouteD1(i,1)
                TFa(i,1) = A_list1(i,Index);
            end

            if R_list1(i,Index) ==
            RouteD1(i,2)
                L_list1(i,Index) =
                L_list1(i,Index) +10;
                A_list1(i,Index+1) =
                A_list1(i,Index+1) +10;
                L_list1(i,Index+1) =
                L_list1(i,Index+1) +10;
            end

            Index = Index+1;
        end
    end
end

```

```

end

%%%%%%%%%%%%%%
for i = 1:12
    Index = 1;
    while 1
        if R_list2(i,Index+1) == 0
            break
        end
        Dist = Distance( R_list2(i,Index) ,
R_list2(i,Index+1) );
        if
            sum( ismember( [R_list2(i,Index) ,
R_list2(i,Index+1)] , High_way ) ) == 2
                temp = A_list2(i,Index) +
Dist / V_type( D2_Car_type(i), 2 )*60;
                A_list2(i,Index+1) = temp ;
                L_list2(i,Index+1) = temp ;
            else
                temp = A_list2(i,Index) +
Dist / V_type( D2_Car_type(i), 1 )*60;
                A_list2(i,Index+1) = temp ;
                L_list2(i,Index+1) = temp ;
            end

            if R_list2(i,Index) ==
RouteD2(i,1)
                TFa(i,2) = A_list2(i,Index);
            end
            if R_list2(i,Index) ==
RouteD2(i,2)
                L_list2(i,Index+1) =
L_list2(i,Index+1) +10;
                A_list2(i,Index+1) =
A_list2(i,Index+1) +10;
                L_list2(i,Index+1) =
L_list2(i,Index+1) +10;
            end

            Index = Index+1;
        end
    end

end

[ N_break2, ~ ] = size(BreakMat2_inedx);

for i = 1:N_break2

    S_temp =
intersect( Are_line_Fb{BreakMat2_inedx(
i,1),1}( List_Len(BreakMat2_inedx(i,1),1)
:end) , ...

Are_line_Fb{BreakMat2_inedx(i,2),1}( Li
st_Len(BreakMat2_inedx(i,2),1):end) );

```

```

%%% F1 发射后的点

Index1 = [];
Index2 = [];

for K = 1:length(S_temp)
    temp1 =
find( Are_line_Fb{BreakMat2_inedx(i,1),
1}( List_Len(BreakMat2_inedx(i,1),1):end)
== S_temp(K) );
    temp2 =
find( Are_line_Fb{BreakMat2_inedx(i,2),
1}( List_Len(BreakMat2_inedx(i,2),1):end)
== S_temp(K) );
    Index1 = [ Index1 ,temp1 ];
    Index2 = [ Index2 ,temp2 ];
end

if
List_Type( BreakMat2_inedx(i,1) ) >
List_Type( BreakMat2_inedx(i,2) )
    Qua1 =
List_L( BreakMat2_inedx(i,1),
List_Len(BreakMat2_inedx(i,1),1) +
Index1 );
    Qua2 =
List_A( BreakMat2_inedx(i,1),
List_Len(BreakMat2_inedx(i,1),1) +
Index1+1 );
    Qub1 =
List_L( BreakMat2_inedx(i,2),
List_Len(BreakMat2_inedx(i,2),1) +
Index1 );
    Qub2 =
List_A( BreakMat2_inedx(i,2),
List_Len(BreakMat2_inedx(i,2),1) +
Index1+1 );

    Time = Qua1>Qub1 & Qua2<
Qub2;
    disp(sum( Time))

% Q_min =
min( Index1(Qua1>Qub1 & Qua2<
Qub2) );
% Q_min_index =
(Index1==Q_min);
%
%
List_Put2( BreakMat2_inedx(i,1),1 ) =
Qua2(Q_min_index) -
Qub2(Q_min_index);
%
%
List_A( BreakMat2_inedx(i,1),
List_Len(BreakMat2_inedx(i,1),1)+Q_mi
n:List_Len(BreakMat2_inedx(i,1),3) ) = ...

```

```

%
List_A( BreakMat2_inedx(i,1),
List_Len(BreakMat2_inedx(i,1),1)+Q_min:List_Len(BreakMat2_inedx(i,1),3) )
+Qua2(Q_min_index) -
Qub2(Q_min_index);

%
List_Put2( BreakMat2_inedx(i,1),1 ) =
sum( (Qub2 - Qua2).*(Qua1>Qub1 & Qua2<
Qub2).*Q );
%
%
List_L( BreakMat2_inedx(i,1),
List_Len(i,3):end ) = List_L( i,
List_Len(i,3):end ) + Z_Stay(i);
%
List_A( i,
List_Len(i,3)+1:end ) = List_A( i,
List_Len(i,3)+1:end ) + Z_Stay(i);

%
List_Put( BreakMat2_inedx(i,2),1 ) =
max( (Qub2 - Qua2).*(Qua1>Qub1 &
Qua2< Qub2) );

%
List_Put( BreakMat2_inedx(i,1),1 ) =
max( (Qub2 - Qua2).*(Qua1>Qub1 &
Qua2< Qub2) );

end

end

%%%%%%%%%% 阶段 2 调整

T_end = zeros(24,1);
for i = 1:24
    T_end( i,1 )=List_L( i,
List_Len(i,2) );
end

List_TFb = max(T_end);
Z_Stay = max(T_end) - T_end;

for i = 1:24
    List_L( i, List_Len(i,3):end ) =
List_L( i, List_Len(i,3):end ) + Z_Stay(i);
    List_A( i, List_Len(i,3)+1:end ) =
List_A( i, List_Len(i,3)+1:end ) +
Z_Stay(i);
end

List_Put2 = zeros(24,1);

```

```

BreakMat1 = zeros(24,24);
BreakMat2 = zeros(24,24);

for i = 1:24
    for j = i+1:24
        S1 =
intersect( Are_line_Fb{i,1}( List_Len(i,1):
end)
Are_line_Fb{j,1}( List_Len(j,1):end) );
        %%% F1 发射后的点
        S2 =
intersect( Are_line_Fb{i,1}( List_Len(i,1):
end),
-Are_line_Fb{j,1}( List_Len(j,1):end) );
        if ~isempty( S1 ) && Car_Type(i)
~= Car_Type(j)
            BreakMat1(i,j) =
length(S1);
        end
        if ~isempty( S2 )
            length(S2)
            BreakMat2(i,j) =
length(S2);
        end
    end
end

[Break1,Break2] = find( BreakMat1~=0 );
BreakMat1_inedx = [Break1,Break2];
[Break1,Break2] = find( BreakMat2~=0 );
BreakMat2_inedx = [Break1,Break2];

[ N_break1, ~ ] = size(BreakMat1_inedx);

i = 8
for i = 1:N_break1

    S_temp =
intersect( Are_line_Fb{BreakMat1_inedx(
i,1),1}( List_Len(BreakMat1_inedx(i,1),1)
:end) , ...

Are_line_Fb{BreakMat1_inedx(i,2),1}( Li
st_Len(BreakMat1_inedx(i,2),1):end) );
    %%% F1 发射后的点

    Index1 = [];
    Index2 = [];

    for K = 1:length(S_temp)
        temp1 =
find( Are_line_Fb{BreakMat1_inedx(i,1),
1}( List_Len(BreakMat1_inedx(i,1),1):end)

```

```

== S_temp(K) );
    temp2 =
find( Are_line_Fb{BreakMat1_inedx(i,2),
1}( List_Len(BreakMat1_inedx(i,2),1):end)
== S_temp(K) );
    Index1 = [ Index1 ,temp1 ];
    Index2 = [ Index2 ,temp2 ];
end

if
List_Type( BreakMat1_inedx(i,1) ) >
List_Type( BreakMat1_inedx(i,2) )
    Qua1 =
List_L( BreakMat1_inedx(i,1),
List_Len(BreakMat1_inedx(i,1),1) +
Index1 );
    Qua2 =
List_A( BreakMat1_inedx(i,1),
List_Len(BreakMat1_inedx(i,1),1) +
Index1+1 );
    Qub1 =
List_L( BreakMat1_inedx(i,2),
List_Len(BreakMat1_inedx(i,2),1) +
Index2 );
    Qub2 =
List_A( BreakMat1_inedx(i,2),
List_Len(BreakMat1_inedx(i,2),1) +
Index2+1 );

    Time = Qua1>Qub1 & Qua2<
Qub2;
    disp(i)
    disp(sum( Time))

%
% Q_min =
min( Index1(Qua1>Qub1 & Qua2<
Qub2) );
%
% Q_min_index =
(Index1==Q_min);
%
%
List_Put2( BreakMat1_inedx(i,1),1 ) =
Qua2(Q_min_index) -
Qub2(Q_min_index) ;
%
%
List_A( BreakMat1_inedx(i,1),
List_Len(BreakMat1_inedx(i,1),1)+Q_min:
List_Len(BreakMat1_inedx(i,1),3) ) = ...
%
List_A( BreakMat1_inedx(i,1),
List_Len(BreakMat1_inedx(i,1),1)+Q_min:
List_Len(BreakMat1_inedx(i,1),3) )
+Qua2(Q_min_index) -
Qub2(Q_min_index);
%

```

```

List_Put2( BreakMat1_inedx(i,1),1 ) =
sum( (Qub2 ).*(Qua1>Qub1 & Qua2<
Qub2).*Q );
%
%
List_L( BreakMat1_inedx(i,1),
List_Len(i,3):end ) = List_L( i,
List_Len(i,3) :end ) + Z_Stay(i);
%
List_A( i,
List_Len(i,3)+1:end ) = List_A( i,
List_Len(i,3)+1:end ) + Z_Stay(i);

%
List_Put( BreakMat1_inedx(i,2) ,1 ) =
max( (Qub2 - Qua2).*(Qua1>Qub1 &
Qua2< Qub2) );

else
    Qub1 =
List_L( BreakMat1_inedx(i,1), Index1 );
    Qub2 =
List_A( BreakMat1_inedx(i,1),
Index1+1 );
    Qua1 =
List_L( BreakMat1_inedx(i,2), Index2 );
    Qua2 =
List_A( BreakMat1_inedx(i,2),
Index2+1 );

    Time = Qua1>Qub1 & Qua2<
Qub2;
    disp(i)
    disp(sum( Time))

%
List_Put( BreakMat1_inedx(i,1) ,1 ) =
max( (Qub2 - Qua2).*(Qua1>Qub1 &
Qua2< Qub2) );

end

end

Route_plan2
Time_com1

List_R = [R_list1;R_list2];
List_A = [ A_list1; A_list2 ];
List_L = [ L_list1; L_list2 ];

Mat = zeros( 24,90 );
Mat( :, 1:3:88 ) = List_R;

```

```

Mat( :, 2:3:89 ) = List_A;
Mat( :, 3:3:90 ) = List_L;

D1_Car_type = [ 2 1 1 1 3 3 2 3 1 1 2 1 ];
D2_Car_type = [ 2 2 3 3 3 1 1 1 1 1 1 2 ];

Mat_Car1 = zeros( 24,90 );
Mat_Car2 = zeros( 24,90 );

Mat_Car1(1:3,:) = Mat( [5,6,8] , : );
Mat_Car1(4:6,:) = Mat( [15,16,17],: );
Mat_Car1(7:9,:) = Mat( [1,7,11] , : );
Mat_Car1(10:12,:) = Mat( [13,14,24] , : );
Mat_Car1(13:18,:) =
Mat( [2,3,4,9,10,12] , : );
Mat_Car1(19:24,:) =
Mat( [18,19,20,21,22,23] , : );

% xlswrite( '无初始时间.xlsx', Mat )

Ts = max(max(TFa)) - TFa;
for i = 1:12
    A_list1(i,2:end) = A_list1(i,2:end)
+Ts(i,1);
    A_list2(i,2:end) = A_list2(i,2:end)
+Ts(i,2);
    L_list1(i,:) = L_list1(i,) +Ts(i,1);
    L_list2(i,:) = L_list2(i,) +Ts(i,2);
end

List_R = [R_list1;R_list2];
List_A = [ A_list1; A_list2 ];
List_L = [ L_list1; L_list2 ];
List_RouteD = [ RouteD1; RouteD2];
List_Len = zeros(24,3);

for i = 1:24
    List_Len(i,1) = find( List_R(i,:) ==
List_RouteD(i,1) );
    temp = find( List_R(i,:) ==
List_RouteD(i,1) );
    List_Len(i,2) = sum( List_R(i,:)~=0 );
    temp2 = find( List_R(i,:) ==
List_RouteD(i,2) );
    List_Len(i,3) = temp2(temp2>temp);

```

```

end
T_end = zeros(24,1);
for i = 1:24
    T_end( i,1 )=List_L( i,
List_Len(i,2) );
end

List_TFb = max(T_end);
Z_Stay = max(T_end) - T_end;

for i = 1:24
    List_L( i, List_Len(i,3):end ) =
List_L( i, List_Len(i,3) :end ) + Z_Stay(i);
    List_A( i, List_Len(i,3)+1:end ) =
List_A( i, List_Len(i,3)+1:end ) +
Z_Stay(i);
end

for i = 1:24
    List_L( i, List_Len(i,2)+1:end ) = 0;
    List_A( i, List_Len(i,2)+1:end ) = 0;
end

max(List_A')

Mat2 = zeros( 24,90 );
Mat2( :, 1:3:88 ) = List_R;
Mat2( :, 2:3:89 ) = List_A;
Mat2( :, 3:3:90 ) = List_L;

% xlswrite( '有初始时间.xls', Mat )

Mat_Car2(1:3,:) = Mat2( [5,6,8] , : );
Mat_Car2(4:6,:) = Mat2( [15,16,17],: );
Mat_Car2(7:9,:) = Mat2( [1,7,11] , : );
Mat_Car2(10:12,:) =
Mat2( [13,14,24] , : );
Mat_Car2(13:18,:) =
Mat2( [2,3,4,9,10,12] , : );
Mat_Car2(19:24,:) =
Mat2( [18,19,20,21,22,23] , : );

```

