中国科学院大学

University of Chinese Academy of Sciences

硕士学位论文

基于列生成算法的复杂武器目标分配求解研究

作者姓名：李冠达

指导教师：戴彧虹 研究员

中国科学院数学与系统科学研究院

学位类别：理学硕士

学科专业：计算数学

培养单位：中国科学院数学与系统科学研究院

2024 年 6 月

Research on Column Generation Algorithm for Complex Weapon Target Assignment Problem

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Natural Science

in Computational Mathematics

By

Li Guanda

Supervisor: Professor Dai Yuhong

Academy of Mathematics and Systems Science

June, 2024

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

自 1958 年被 Manne 提出开始，武器目标分配问题就成为了运筹优化领域的一个经典问题，它的基本形式是将 m 个武器分配给 n 个敌方目标，来最大化对于敌方目标的销毁期望，随着时代的发展，战场环境也变得愈发复杂，实际的作战场景中往往需要雷达在内的侦察资源来指导武器对目标进行打击，因此在对问题进行数学建模时，仅仅考虑武器与目标两个作战要素会导致模型与现实相脱节。因此本文考虑的情景中，首先对于一个更复杂的作战场景进行了数学建模，将武器目标分配问题拓展为武器-雷达-目标分配问题。

在过去的研究中，武器目标分配问题已经被证明为 NP-完全问题，并且一般采用非线性方式对问题进行建模，已有的求解思路包括包括启发式算法，近似算法和精确算法。在精确算法中，目前两种最有效的方式是对问题进行线性化与利用问题的凸性对问题采用外逼近方法来进行求解，在本文中，将对这两种求解思路进行整合与拓展。针对非线性形式的武器目标分配问题，通过将所有作战场景列出来的方式，实现对问题的线性化，将问题转化为一个具有指数多个列的线性整数规划问题，再对这个问题采用基于列生成的分支定界算法，并且将列生成子问题转化为一个凸整数规划题，并使用外逼近方法进行求解，并提出了一系列有效不等式，最终得到整体的求解方式。

在实际的作战场景中，作战过程往往并非是一个静态的过程，无论是敌方的来袭目标还是我方的作战策略通常都会随着时间发生变化。为了能够对这个过程进行数学建模，我们采用了非均匀时间离散的方式，将整个作战过程拆分为多个决策节点，以兼顾对于近期决策点的精确抉择与远期决策点的估计。同时在这个数学模型的基础上，同样采用列生成求解框架，对于列生成子问题进行了进一步的转化，将问题转化为了一个更紧的形式，以实现问题的快速求解。

关键词：武器目标分配，列生成，外逼近，动态武器目标分配

Abstract

Since its introduction by Manne in 1958, the weapon-target assignment (WTA) problem has become a classic issue in the field of operations research and optimization. Its fundamental form involves allocating m weapons to n enemy targets to maximize the expected destruction of the enemy targets. As time has evolved, the battlefield environment has become increasingly complex, and actual combat scenarios often require reconnaissance resources, including radar, to guide weapons in striking targets. Therefore, merely considering the two combat elements of weapons and targets in mathematical modeling can lead to a disconnect between the model and reality. In this context, this paper first mathematically models a more complex combat scenario, extending the WTA problem to a weapon-radar-target assignment problem.

In past research, the WTA problem has been proven to be NP-complete, and it is generally modeled in a nonlinear manner. Existing solution approaches include heuristic algorithms, approximation algorithms, and exact algorithms. Among exact algorithms, the two most effective methods currently are linearizing the problem and using the convexity of the problem to solve it with an outer approximation method. This paper integrates and expands upon these two solution approaches. For the nonlinear form of the WTA problem, the problem is linearized by listing all combat scenarios, transforming it into a linear integer programming problem with exponentially many columns. This problem is then solved using a branch-and-bound algorithm based on column generation, and the column generation subproblem is transformed into a convex integer programming problem and solved using the outer approximation method. A series of effective inequalities are proposed, resulting in an overall solution approach.

In actual combat scenarios, the combat process is often not static, as both the enemy's incoming targets and our combat strategy typically change over time. To mathematically model this process, we adopt a non-uniform time discretization approach, dividing the entire combat process into multiple decision nodes to balance precise choices for near-term decision points with estimates for long-term decision points. Based on this

mathematical model, the column generation solution framework is also used, and the column generation subproblem is further transformed, converting the problem into a tighter form to achieve rapid solution of the problem.

Keywords: Weapon-Target-Assignment, Column Generation, Outer Approximation, Dynamic-Weapon-Target-Assignment

目 录

图形列表

表格列表

符号与缩写列表

符号

符号	描述
I	武器集合，i 代表武器的下标
J	雷达集合，j 代表雷达的下标
K	目标集合，k 代表目标的下标
n	目标的数量。
m	武器类型的数量。如果每种类型只限定一件，这个变量代表的是武器的数量。
l_i	武器容量上限。
L_i	武器容量下限（若存在）。
r_j	雷达容量上限。
R_j	雷达容量下限（若存在）。
V_k	目标 k 的价值。
p_{ik}	武器 i 命中目标 k 的概率。
p_{ijk}	使用武器 i 和雷达 j 命中目标 k 的概率。(x_{ijk}, c_{ijk} 类似)
p_{ijt}^k	使用武器 i 和雷达 j 在时间点 t 命中目标 k 的概率。(x_{ijt}^k, c_{ijt}^k 类似)
x_{ik}	分配给目标 k 的类型为 i 的武器的数量，可能退化为 0-1 变量
c_{ik}	将武器 i 分配给目标 k 的成本。

缩写

WTA	Weapon Target Assignment, 武器目标分配
SWTA	Static Weapon Target Assignment, 静态武器目标分配
DWTA	Dynamic Weapon Target Assignment, 动态武器目标分配
CG	Column Generation, 列生成
BAP	Branch-and-Price, 分支定价

OA	Outer Approximation, 外逼近
MP	Master Problem, （列生成）主问题
SP	Subproblem, （列生成）子问题
RMP	Restricted Master Problem, （列生成）受限主问题
RC	Reduced Cost, 简约成本

第 1 章　引言

1.1　武器目标分配问题

1.1.1　静态武器目标分配问题

武器目标分配问题是一个军事领域中经典的数学优化问题，最初由 Manne(1958)[1] 引入，最开始的目标为为来袭导弹分配可用的拦截弹，以最小化导弹摧毁受保护物资的概率。随着对问题研究的深入与算力的进步，武器目标分配逐渐发展出很多不同的细分领域，在众多的划分标准中，一个重要的划分标准为考虑一阶段问题还是多阶段问题，即将问题分为了静态武器目标分配问题（静态武器目标分配问题）与动态武器目标分配问题（动态武器目标分配问题）。静态武器目标分配问题中不考虑时间，仅考虑针对一次导弹来袭所做的拦截。而动态武器目标分配问题则主要考虑拦截 - 观察 - 拦截的模式，即每次打击过后观察打击的结果并安排下一次打击。Lloyd 和 Witsenhausen(1986)[2] 证明了即使是较为简单的静态武器目标分配仍然是 NP 完全的，因此在之前的研究中，大多数研究者的研究课题是寻找合适的智能方法，通过启发式算法在一个较短的时间内寻找接近最优解的方法。但启发式算法并不能在理论上保证最终结果的最优性，在之前的研究中，使用整数规划方法求解问题精确解的研究内容较少，求解规模较小，直到近几年才有相对有效的算法出现。

在比较早的时期，由于计算机解决大型非线性问题的能力有限 (Day, 1966)[3]，当时的研究者将研究种地放在了形式较为简单的静态武器目标分配问题 (den-Broeder et al., 1959)[4]，即：

$$\min \quad \sum_{j=1}^{n} V_j \left(\prod_{i=1}^{m} (1 - p_{ij})^{x_{ij}} \right) \tag{P1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le l_i, \qquad \forall\, i \in I, \tag{P1.2}$$

$$x_{ij} \in \mathbb{N}, \qquad \forall\, j \in J,\ i \in I. \tag{P1.3}$$

其中约束表示每个武器的使用上限不能超过其容量。他们结合当时的计算能力开发了二相对应的比较简单的求解方法 (Lemus and David，1963)[5]，(Day，1966)[3]。

Eckler and Burr (1972)[6] 提出并讨论了采用动态规划的方式求解静态武器目标分配的可能性，但未能较好地得到解决此类问题的算法。随着计算能力的提高，计算机解决更加复杂的问题的能力也在提高。Burr et al. (1985)[7] 建模并解决了最早的动态武器目标分配问题之一，与此同时 Chang (1987)[8] , Soland (1987)[9] 和 Hosein(1989)[10] 等人也对求解动态武器目标分配问题提供了一些方法。同时，用新方法解决了具有较少假设的静态武器目标分配模型（包括 Kwon et al., (1999)[11]；Metler et al., (1990)[12]；Wacholder, (1989)[13]），较少的假设使得问题与现实的情况更加贴合，但是同样增加了问题的难度。这种对于早期算法进行修补的过程一直持续到 2000 年。在 20 世纪 00 年代，模型和求解算法得到了进一步发展：一个发展方向是在模型中引入更多参数，使得模型能够更好地反应现实情况（即 Shang et al. (2007)[14] 和 Karasakal (2008)[15]）；另一个发展方向则是保证模型能够更加快速地得到最优解或者最优解的近似解（即 Malcolm, (2004)[16]，Ahuja et al. (2007)[17]，以及 Ahner and Parson (2015)[18]）。随着新的模型的建立，后续研究者基于这些模型开发了更新的算法（即 Bertsekas et al., (2000)[19]；Kline et al., (2017)[20]；Wu et al., (2008)[21] ）或将原有的方法进行了效果显著的改进（即 Ahuja et al., (2007)[17]；Lee et al., (2002)[22]；Su et al., (2008)[23]；Xin et al., (2010)[24]）。

1.1.2 动态武器目标分配问题与多作战元素的武器目标分配问题

随着计算机算力的不断增长，武器目标分配现有的研究方向除了在已经存在的现有模型上改进求解方法之外，还提出了关于目标飞行路径的时间依赖性的动态模型（Khosla, (2001)[25]; Leboucher et al., (2013)[26]），但与现有模型相比受到的关注较少，且这些模型中解决方案技术的改进尚未出现。

第一个模型是 shoot-look-shoot 模型，它对于每次打击结果都可以进行观察，因此在打击的过程中目标是否中存在都是确定的。在这种模型中，假设问题共有两阶段，武器在第一阶段中将武器分配给目标，之后随着打击结束对问题进行一次观察。针对第一次打击带来的结果，将剩余的武器分配给幸存的目标。

第二个模型被称为两阶段模型，或者更一般地说多阶段模型，它与 shoot-look-shoot 的不同之处在于它不允许在打击后观察，且每个目标仅被考虑一次，若在某一阶段没被摧毁，后续将不再有摧毁该目标的机会。也就是说，在 shoot-look-shoot 问题中，作战目的是对给定数量的目标进行反复攻击，直到所有目标

都被摧毁或达到迭代次数的限制。而在 2 阶段问题中，第一次会出现的目标是给定的，但是在第二阶段，有可能会重新出现新的目标，且目标的数量和类型仅仅能够给出一个概率分布而不是确定性的。在已知多阶段的目标的概率分布的前提下，希望能够给出对目标毁伤的数学期望最高的方案。可以看出，动态武器目标分配问题的复杂程度明显高于静态武器目标分配，但是却能够更好地结合实际情况。在更加复杂的模型中，也存在将两种动态模型结合在一起的情况，但是目前针对此问题尚没有比较好的精确求解方法。

在动态武器目标分配问题发展的过程中，关于问题的困难程度也曾经被多人指出。Khosla (2001)[25] 指出，“尽管采用了两步法，但每个优化问题仍然即使对于数量不多的威胁、武器系统和时间点，也有巨大的搜索空间。”改进两步法的方法尚未出现。同样，Leboucher at el. (2013) [26] 评论了问题的指数增长并提出了一种两步解决技术，并补充说另一个问题是“如何评估一个解的质量”。武器目标分配的未来将需要解决前面提到的以调度为中心的动态武器目标分配的困难与能够利用问题的特殊结构的技术。此外，存在许多问题中本应该被引入的参数，由于它们会带来增加的计算复杂性而被删除，随着计算能力的增加，这些参数可以使用新的理论模型重新引入，以保证模型更加贴合实际情况。

除了从时间的维度将问题拓展为动态武器目标分配问题，具体的作战场景刻画也有。在现代化的作战场景中，除了武器与目标两个关键要素外，多种战场资源的综合调配将起到关键性的作用。举例而言，在防空反导作战当中，一次基本的打击包含雷达对目标的锁定，武器对目标的打击和指挥所对整体作战场景的控制，因此将这些作战资源进行抽象，针对每个来袭目标，我们可以为其设计一条或多条“目标 - 雷达 - 指控 - 发射车 - 弹药”的杀伤链，其中，指控部分可以解耦，而弹药与发射车则是绑定在一起的，相比于传统的武器目标分配问题多了雷达这个作战要素。针对这个更复杂的场景，对应的数学模型也需要进行对应的调整，其数学形式为：

$$
\begin{aligned}
\max \quad & \sum_{k=1}^{n} v_k \left[1 - \prod_{i=1}^{m} \prod_{j=1}^{t} \left(1 - p_{ijk}\right)^{x_{ijk}} \right] \\
\text{s.t.} \quad & \sum_{j=1}^{t} \sum_{k=1}^{n} x_{ijk} \leq l_i, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m} \sum_{k=1}^{n} x_{ijk} \leq r_j, \quad \forall j = 1, \cdots t \\
& x_{ijk} \in \mathbb{N}
\end{aligned}
$$

其中两个约束分别代表武器和雷达不能超过其容量限制。

因此，在武器目标分配这个研究课题上，无论是进一步优化对于静态武器目标分配问题的求解方法以增加求解规模，还是将求解方法拓展到动态武器目标分配问题与更复杂作战场景下，为这些问题设计高效、使用的求解算法都具有研究意义。

1.2 预备知识

1.2.1 整数规划问题与分支定界求解框架

整数规划问题是数学规划的一个重要分支，它在工程设计、生产计划、资源分配、物流管理等领域有着广泛的应用。整数规划问题的特点是决策变量必须取整数值，这使得问题的求解相比于连续规划更为复杂。

根据决策变量的不同，整数规划问题可以分为纯整数规划问题、混合整数规划问题和 0-1 整数规划问题。纯整数规划问题要求所有决策变量均为整数，而混合整数规划问题则允许部分变量为连续变量。0-1 整数规划问题是纯整数规划的一个特例，其中所有决策变量只能取 0 或 1，常用于表示选择与否的决策情形。从约束和目标函数的形式来看，整数规划问题又可以分为整数线性规划问题（ILP）和非线性整数规划问题（INLP）。整数线性规划问题的目标函数和约束条件都是线性的，这是最常见的整数规划类型。非线性整数规划问题则涉及非线性的目标函数或约束条件，凸整数规划问题作为一种特殊的非线性整数规划问题，它的目标函数是凸函数且约束条件构成的可行域为凸集。

由于整数解的空间是离散的，整数规划问题的求解通常不能直接应用连续优化的方法。常用的求解方法包括分支定界法 [1]、割平面法（[2], [3]）和启发

式算法等。在现代的求解软件中，通常是在分支定界的框架下结合多种割平面与启发式方法来对问题进行求解。

定义 1.1. 与整数规划相关的基本定义

- **分数变量 (Fractional Variable)**: 在线性松弛问题的解中，如果一个整数变量的值不是整数，则称该变量为分数变量。
- **上界 (Upper Bound, UB)**: 对于最小化问题，上界是目标函数的一个估计值，该值大于或等于最优解的目标函数值，通常由一个可行解计算得到。
- **下界 (Lower Bound, LB)**: 对于最小化问题，下界是目标函数的一个估计值，该值小于或等于最优解的目标函数值，通常由一个松弛解得到。
- **相对间隙 (Relative Gap)**: 相对间隙用于衡量上界和下界之间的差距相对于上界的比例，定义为

$$\text{Relative Gap} = \frac{\text{UB} - \text{LB}}{|\text{UB}|} \times 100\%$$

其中，UB 是上界，LB 是下界。

- **刻面 (Facet)**: 对于一个多面体（如线性规划的可行域），一个刻面是多面体的极大面，即不存在其他的多面体的一部分。由于刻面能够提供多面体的最紧凑表示，它对于描述多面体的结构非常重要。
- **有效不等式 (Valid Inequality)**: 对于一个给定的整数规划问题，一个有效不等式是指一个不等式，它对于该问题的所有可行整数解都成立。有效不等式用于增强线性松弛的表达，从而更紧密地逼近整数可行域，提高求解效率。

分支定界方法被证明是一个精确算法，它是一种对可行解进行穷举的算法，但是和普通穷举法所不同的是，分支定界算法在对某一分支进行检索之前会预先计算出该分支的上界或下界，如果界限无法比目前的最优可行解更好，那么该分支就会被舍弃，通过降低搜索空间的规模节约了大量的搜索时间。而割平面则可以通过割掉不可行的部分，来让解空间逼近整数可行集的凸包，因此二者能够相互结合。在本文中会用到分支定界的部分知识，但是仅仅在外逼近算法中有与经典分支定界方法有所不同的敌方，因此在此处并不详述分支定界方法的算法流程。

1.2.2　列生成方法

列生成算法概述

列生成方法是一种在大规模优化问题中广泛应用的技术，它的核心思想是在一个较小的、更易于管理的问题（即主问题）上求解，而不是在问题的所有可能性上一次性求解。这种方法主要应用在大规模的线性规划和整数规划问题中，尤其适用于那些具有大量可能决策变量，但在任何最优解中只有少数决策变量非零的情况。

列生成算法的算法流程

列生成方法求解的是一个标准的线性规划问题：

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

其中 x 代表的决策变量个数非常多，通过以下步骤迭代地求解这个线性规划问题：

步骤一：初始化主问题。

这一步通常通过启发式方法或随机选择的方式，选取一个较小的、可解的问题作为主问题的初始版本，主问题的数学模型可以描述为：

$$\begin{aligned} \min \quad & c_B^T x_B \\ \text{s.t.} \quad & A_B x_B = b, \\ & x_B \geq 0, \end{aligned}$$

使用 x_B 表示仅选择了原问题的部分列。

步骤二：求解主问题。

接下来我们使用线性规划方法（如单纯形法或者内点法）来求解这个初步限制的主问题，获取当前的最优解和对偶变量值。

步骤三：求解子问题。

对于子问题，我们实际上是在尝试寻找那些未被包含在当前主问题中的列（即尚未选取的变量），若加入主问题后有可能改进当前解。这一过程被称为定价过程。在定价过程中，我们会计算每个未选取变量的约化费用（Reduced cost）：

$$\min \quad c_i - \pi^T A_i,$$

若该问题的最优值小于 0，对应的 x 将被加入到主问题的备选解中。

步骤四：判断当前主问题是否达到最优解。

如果在步骤三的定价过程中，我们找到了一个约化费用小于零的决策变量，那么我们就将这个决策变量（也即新的列）加入到主问题中，并回到步骤二，对扩充了的主问题进行求解。否则，如果所有未选取的决策变量的约化费用都大于或等于零，那么我们就停止算法，因为这表明当前的主问题解就是全局最优解。

算法 1 列生成算法

procedure 列生成 (A, b, c) ▷ 输入：线性规划参数 A, b, c
　初始化主问题的变量集合 $S \subset \{1, 2, \ldots, n\}$
　repeat
　　求解主问题 $\min\{c^T x : Ax = b, x \geq 0, x_i = 0$ 对于所有$i \notin S\}$
　　得到当前解 x^* 和对偶变量值 π^*
　　求解子问题 $\min\{c_j - \pi^* A_j : j \notin S\}$
　　if $\exists j \notin S$ 使得 $c_j - \pi^* A_j < 0$ **then**
　　$S = S \uplus \{j\}$
　　end if
　until 不存在 $j \notin S$ 使得 $c_j - \pi^* A_j < 0$
　return 主问题的最优解 x^* ▷ 输出：线性规划的最优解
end procedure

列生成方法的最优性证明及它在武器目标分配问题中的具体表现形式将在后续章节中进行陈述。

1.2.3　外逼近方法

外逼近算法概述外逼近方法是一种求解凸优化问题的技术，如果目标函数或约束条件满足凸性条件的同时形式又比较复杂，则希望通过逐步构造可行域的一个线性外逼近来近似原问题的可行域，从而将问题转化为迭代地求解一系列线性规划。当这个方法应用在凸整数规划问题（属于 MINLP 的一种特殊情况）上时，根据已有的结论，仅仅考虑那些整数点的外逼近割就足够完成对于原问题的等价转化，将这个方法与分支定界框架进行有效的集合，则可以在有限的步骤内得到问题的精确解。

原问题与等价转化

考虑一个凸优化问题：

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & x \in X, \end{aligned}$$

其中 $f(x)$ 是凸函数，$g_i(x)$ 是凸约束，X 是凸集。为了将原问题转化为一个具有线性目标函数的优化问题，我们引入一个新变量 η 并考虑以下问题：

$$\begin{aligned} \min \quad & \eta \\ \text{s.t.} \quad & f(x) \leq \eta, \\ & g_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & x \in X. \end{aligned}$$

这个新问题在满足原问题的约束条件下最小化 η，其中 η 是 $f(x)$ 的一个上界。因此，这个新问题与原问题是等价的。

外逼近算法的算法流程

通过以下步骤迭代地求解转化后的凸优化问题：

步骤一：初始化。

选择一个初始点 $x^0 \in X$，构造初始外逼近集合 $S^0 = \{x \in X : g_i(x^0) + \nabla g_i(x^0)^T(x - x^0) \leq 0, \forall i\}$。

步骤二：求解近似问题。

在第 k 次迭代中，求解近似问题：

$$\begin{aligned} \min \quad & \eta \\ \text{s.t.} \quad & f(x^k) + \nabla f(x^k)^T(x - x^k) \leq \eta, \\ & x \in S^k, \end{aligned}$$

得到解 (x^{k+1}, η^{k+1})。

步骤三：更新外逼近集合。

更新外逼近集合 $S^{k+1} = S^k \cap \{x \in X : g_i(x^{k+1}) + \nabla g_i(x^{k+1})^T(x - x^{k+1}) \leq 0, \forall i\}$。

步骤四：终止条件。

如果 (x^{k+1}, η^{k+1}) 满足终止条件（如 $\eta^{k+1} - \eta^k < \epsilon$），则停止迭代，输出 (x^{k+1}, η^{k+1}) 作为最优解；否则，令 $k = k + 1$，返回步骤二。

算法 2 外逼近算法

procedure 外逼近 (f, g, X) ▷ 输入：目标函数 f，约束函数 g，可行域 X
　初始化 $x^0 \in X$，$S^0 = \{x \in X : g_i(x^0) + \nabla g_i(x^0)^T(x - x^0) \leq 0, \forall i\}$
　$k \leftarrow 0$
　repeat
　　求解近似问题 $\min\{\eta : f(x) \leq \eta, x \in S^k\}$，得到解 (x^{k+1}, η^{k+1})
　　更新 $S^{k+1} = S^k \cap \{x \in X : g_i(x^{k+1}) + \nabla g_i(x^{k+1})^T(x - x^{k+1}) \leq 0, \forall i\}$
　　$k \leftarrow k + 1$
　until 满足终止条件
　return 解 (x^k, η^k) ▷ 输出：凸优化问题的最优解
end procedure

当问题转化为凸整数规划时，外逼近方法将与分支定界方法相结合，并结合这个问题的具体形式，具体的组合方式将在第 2 章与第 3 章中进行详细地介绍。

1.3　本文主要工作

本文主要工作集中在武器目标分配问题的研究和求解方法上。首先，本文针对经典的武器目标分配问题进行了深入分析，提出了一种线性化建模方法，将非线性问题转化为线性整数规划问题，为后续的求解算法设计奠定了基础。在此基础上，本文设计了一个基于列生成框架的求解算法，并采用外逼近方法求解列生成子问题，有效地解决了指数多列的挑战。

进一步，本文将研究范围拓展到包含复杂资源约束和动态变化的武器目标分配问题，提出了相应的数学模型和评价方式。特别地，针对动态武器目标分配问题，本文采用了非均匀时间离散的方法，建立了一个能够精确求解的模型，并将列生成求解框架进行了相应的推广，以适应动态变化的作战环境。

在算法设计方面，本文针对列生成子问题的特殊性，提出了两种不同的建模方式，并采用外逼近求解框架进行求解。为了提高求解效率，本文还设计了基于问题特性和实际含义的有效不等式，并对经典的列生成方法进行了改进，允许在求解过程中选择非最优子问题和多个改进列，从而加速了算法的收敛速度。

最后，本文通过数值实验验证了所提出的模型和算法的有效性和效率。实验结果表明，本文的研究不仅在理论上对武器目标分配问题的求解方法做出了贡献，而且在实际应用中具有重要的指导意义。

第 2 章　武器目标分配问题及基本求解方法

2.1　引言

武器目标分配问题是军事领域的一个经典问题，它描述的作战场景是当敌方有多个来袭目标时，我方如何合理地分配作战资源，来尽可能摧毁敌方的来袭目标。在 Manne 在 1958 年提出的经典模型中，假设敌方有多个目标来袭，我方有不同的武器来对目标进行拦截，且对整体有如下的假设：

- 确定性假设：每个目标的价值是固定的，不随着时间发生变化。同时在给定目标和武器的条件下，毁伤概率是定值且不随时间发生变化。
- 毁伤概率假设：武器的打击之间是相互独立的。如果一个目标被多个武器同时打击，那么该目标的存活概率就是每个杀伤链打击后的存活概率的乘积。我们计算该目标中的毁伤概率就是用 1 减去存活概率的大小。
- 资源限制假设：由于武器特性，第 i 个武器最多使用 l_i 次

在这些前提假设下，可以写出问题的数学模型：

$$\min \quad \sum_{j=1}^{n} V_j \left(\prod_{i=1}^{m} (1 - p_{ij})^{x_{ij}} \right) \tag{P1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le l_i, \qquad \forall\, i \in \boldsymbol{I}, \tag{P1.2}$$

$$x_{ij} \in \mathbb{N}, \qquad \forall\, j \in \boldsymbol{J},\ i \in \boldsymbol{I}. \tag{P1.3}$$

同样的实际场景从不同角度建模可能会得到形式不同的数学模型，而不同的建模方式对应了不同的求解方法，也显著地影响了问题的求解速度，因此如何对问题进行数学建模具有重要的研究意义。在 2.1 节中，我们将展开讨论对于的建模方法，首先给出武器目标分配问题在历史上的不同建模方式并指出它们的一些优劣，之后会给出对于问题进行线性化的方法，目前全部的精确求解方法都是基于不同角度的线性化，因此在这一小节中将展示两种比较重要的线性化思路，最后将给出数学模型在一个更复杂的作战场景下的推广。

同时根据问题的形式可以看出，这个问题的难点主要在于目标函数中决策变量在指数项，具有明显的非线性，同时在后续的拓展模型中，对于概率的计算方式都与这个经典模式相似，因此如何有效地处理这个非线性项将是求解问题

的关键。在 2.2 节中，将首先证明这个问题是一个凸优化问题，之后会给出应用外逼近方法来对该问题进行求解的一个思路。在凸整数规划问题中，外逼近方法的应用与连续问题有一定的区别，在这一节中也将详述如何有效地将分支定界方法与外逼近方法进行相互结合。

在本章的内容中，如果不加以特殊说明，武器目标分配问题指的都是静态武器目标分配问题，即不考虑时间维度，而是仅仅在一个瞬时考虑如何进行武器的分配。

2.2 武器目标分配问题的数学建模

2.2.1 经典模型的凸性证明

首先我们给出经典模型（P1）的凸性证明。

命题 2.1. *函数* $f_j(x)=\prod_{i=1}^{m}(1-p_{ij})^{x_{ij}}$ *是一个凸函数*

证明. 考虑问题的海森矩阵 $\boldsymbol{H}$ 中第 a 行 b 列的元素，即原函数对两个变量求偏导，可以得到项为：

$$\frac{\partial f_j(x)}{\partial x_{aj}\partial x_{bj}}=\ln(1-p_{aj})\ln(1-p_bj)f_j(x)$$

从而可以得到问题的海森矩阵为

$$\boldsymbol{H}=f_j(x)\begin{bmatrix}\ln(1-p_{1j})\ln(1-p_{1j}) & \ln(1-p_{1j})\ln(1-p_{2j}) & \cdots & \ln(1-p_{1j})\ln(1-p_{mj})\\ \ln(1-p_{2j})\ln(1-p_{1j}) & \ln(1-p_{2j})\ln(1-p_{2j}) & \cdots & \ln(1-p_{2j})\ln(1-p_{mj})\\ \vdots & \vdots & \ddots & \vdots\\ \ln(1-p_{mj})\ln(1-p_{1j}) & \ln(1-p_{mj})\ln(1-p_{2j}) & \cdots & \ln(1-p_{mj})\ln(1-p_{mj})\end{bmatrix}$$

如果我们记：

$$l=\left[\ln(1-p_{1j}),\ln(1-p_{2j}),\cdots,\ln(1-p_{mj})\right] \tag{2.1}$$

则

$$\boldsymbol{H}=f_j(x)l\cdot l^T$$

由于 $f_j(x)\geq 0$，因此它是正定矩阵，从而原函数是一个凸函数。 □

由于多个凸函数的线性组合一定也是凸函数，因此我们可以得到如下命题

命题 2.2. *经典武器目标分配问题的目标函数是一个凸函数，优化问题是一个凸优化问题。*

2.2.2　武器目标分配问题的数学模型构建

在这一小节中，我们首先讨论对于经典模型的不同形式并对比它们之间的优劣，再给出动态武器目标分配问题的一些经典模型。

模型 2 如果限制每个武器最多使用一次，则变量 x_{ij} 就会退化为一个 0-1 变量，此时由于

$$(1-p_{ij})^{x_{ij}} = 1 - x_{ij}p_{ij}, \quad x = 0, 1$$

因此原问题可以转化为：

$$\min \quad \sum_{j=1}^{n} V_j \left(\prod_{i=1}^{m} (1 - x_{ij}p_{ij}) \right) \tag{P1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le l_i, \qquad \forall\, i \in \boldsymbol{I}, \tag{P1.2}$$

$$x_{ij} \in \mathbb{N}, \qquad \forall\, j \in \boldsymbol{J},\ i \in \boldsymbol{I}. \tag{P1.3}$$

此时与原问题相比，目标函数从指数函数变成了多项式函数，但是此时函数将不再具有凸性，因此这个模型更适合一些不利用问题凸性的算法（比如一些基于枚举算法），对于一些凸优化算法这个模型并不合适。

同时对于原模型，可以采用取对数的方式来给问题进行等价的转化，假设 $w_k \triangleq \sum_{i=1}^{m} \ln(1-p_{ij})x_ij$，并引入变量 η 来指示这个指数变量，则原模型可以被转化为：

$$\min \quad \sum_{k=1}^{n} V_k \eta_k \tag{P1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le l_i, \qquad \forall\, i \in \boldsymbol{I}, \tag{P1.2}$$

$$e^{w_k} \le \eta_k \tag{2.2}$$

$$\sum_{i=1}^{m} \ln(1-p_{ij})x_{ij} = w_k \tag{2.3}$$

$$x_{ij} \in \mathbb{N}, \quad w, \eta \in \mathbb{R} \qquad \forall\, j \in \boldsymbol{J},\ i \in \boldsymbol{I}. \tag{P1.3}$$

这个模型将问题的非线性项转化为了一个比较形式上比较简单的函数，$e^{w_k} \le \eta_k$，这个思想最早是由 Ahuja (2007)[17] 提出的，Ahuja 在文章中利用这个等价的转化将问题转变为网络流问题来进行求解，不过他在文章中采用以 2 为底的对数

变换。同时根据 Kline et al.(2017)[20] 得到得数值实验结果，采用 BARON 等商业全局优化求解器时，这个模型也比经典模型求解效果更好，在限定的时间内能够增加问题的求解规模。这个模型的方式相当于对空间进行了一些映射，因此在计算割平面时，不同的形式也会有不同的线性割平面，因此在后面的讨论中，也会针对这两种不同的形式提出不同的割平面。

2.2.3 静态武器目标分配问题的线性化方法

为了解决目标函数的非线性，Lu 提出了一种对问题直接进行线性建模的方式，这种方式通过引入指数多的列个数来消除问题的非线性，虽然在文章中 Lu 假设了每种武器的容量都是 1，但是可以将其推广到任何整数容量。由于新的建模方式可以有效利用多种整数线性规划和线性规划的技巧，这种线性化的建模方式有比较大的启发性，因此在这里将介绍推广后的线性化模型。

假设一共有 m 个武器，则对于任意一个目标 j，每个武器可以选择打击 j 或不打击 j，并且之多打击 c_w 次，因此共有 $(c_w+1)^m$ 个不同的打击方案，我们称一个打击方案为一个场景，用 s_j(scene) 表示。由于每个目标备选的方案都是一样的，因此当不产生歧义的情况下，也使用 s 来表示一个场景。整体的决策过程就是给每个目标分配一个打击场景（包括不分配武器也是一种打击场景），而且各个目标之间的打击场景之间不能产生冲突，必须要满足与原来约束等价的各种约束。

对于一个打击场景 s，我们使用符号 n_{si} 表示在场景 s 中，第 i 个武器的使用次数。同时我们定义 $q_{js}=V_j\prod_{i=1}^{m}(1-n_{si}\cdot p_{ij})$ 用来表示使用第方案 s 打击目标 j 的加权概率。最后我们使用决策变量 y_{js} 来表示是否对目标 j 使用场景 s，这是一个 0-1 变量，因为一个场景要么被分配给了一个目标，要么没有被分配给这个目标。

举例而言，假设共有 8 个武器，则使用第 1，3，6 号武器的攻击方案即记为 $s^*=s_{[1,0,1,0,0,1,0,0]}$，$|S|=2^8=256$，此时有：

$$n_{s1}=n_{s3}=n_{s6}=1,\ n_{s0}=n_{s2}=\cdots=n_{s7}=0.$$

假设用这个打击场景打击目标 1，则 $q_{js}=V_1(1-p_{11})(1-p_{31})(1-p_{61})$，可以看出，就是原模型中 $x_{11}=x_{31}=x_{61}=1$ 时目标 j 的赋权存活概率。此时 $y_{1s^*}=1$，$y_{1s}=0,\ \forall s\neq s^*$

使用上面的符号，我们可以写出这个武器目标分配问题的线性模型：

$$\min \quad \sum_{j=1}^{n}\sum_{s=0}^{2^m-1} q_{js}y_{js} \tag{L1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n}\sum_{s=0}^{2^m-1} n_{si}y_{js} \le l_i, \quad \forall\, i \in \boldsymbol{I}, \tag{L1.2}$$

$$\sum_{s=0}^{2^m-1} y_{js} = 1, \qquad \forall\, j \in \boldsymbol{J}, \tag{L1.3}$$

$$y_{js} \in \{0,1\}, \qquad \forall\, j \in \boldsymbol{J},\ s \in \boldsymbol{S}. \tag{L1.4}$$

这是一个整数线性规划（ILP）问题，求解整数线性规划问题的一个常见办法是求解问题的松弛问题（Relaxation），并结合分支定界算法来找到最优解。因此对该问题进行松弛，可以得到如下松弛后的模型：

松弛后的线性模型 (L2)

$$\min \quad \sum_{j=1}^{n}\sum_{s=0}^{2^m-1} q_{js}y_{js} \tag{L2.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n}\sum_{s=0}^{2^m-1} n_{si}y_{js} \le l_i, \quad \forall\, i \in \boldsymbol{I}, \tag{L2.2}$$

$$\sum_{s=0}^{2^m-1} y_{js} = 1, \qquad \forall\, j \in \boldsymbol{J}, \tag{L2.3}$$

$$0 \le y_{js} \le 1, \qquad \forall\, j \in \boldsymbol{J},\ s \in \boldsymbol{S}. \tag{L2.4}$$

在这里，我们将原模型的整数约束放宽为连续约束，从而得到了松弛模型，同时注意到，在满足约束 (L2.3) 的情况下，变量 y_{js} 的取值范围自然就会限制在 0 到 1 之间，因此，将 (L2.4) 修改为 $y_{js} \ge 0$ 的约束对最优解的求解没有任何影响，从而得到下面的模型：

修改后的线性模型 (L2')

$$\min \quad \sum_{j=1}^{n}\sum_{s=0}^{2^m-1} q_{js}y_{js} \tag{L2'.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n}\sum_{s=0}^{2^m-1} n_{si}y_{js} \le l_i, \quad \forall\, i \in I, \tag{L2'.2}$$

$$\sum_{s=0}^{2^m-1} y_{js} = 1, \qquad \forall\, j \in J, \tag{L2'.3}$$

$$y_{js} \ge 0, \qquad \forall\, j \in J,\ s \in S. \tag{L2'.4}$$

可以看出，这个模型虽然是线性模型，但是却具有指数数量级的列个数。线性化建模的方式能够比较容易地推广到后文中提到的多作战要素模型和时间模型。关于这个问题的求解方式，我们将放在第三章及第四章中进行详细阐述。

2.2.4 数学模型在多作战要素场景下的推广

在现代化的作战场景中，仅仅考虑武器与目标将无法有效地刻画作战场景，因此在模型中引入雷达作为另外一种作战要素，即针对每个来袭的目标，我们可以设计一条或多条“目标 - 雷达 - 发射车 - 弹药”的组合，我们称之为杀伤链。此时 x_{ijk} 和 p_{ijk} 代表的是对于目标 k，使用武器 i 和雷达 j 进行打击的个数与概率。此时优化目标与之前类似，依然是最大化消除的来袭目标威胁值：

$$f_1(x) = \sum_{j=1}^{n} v_j \left[1 - \prod_{i=1}^{m}\prod_{k=1}^{t}\left(1 - p_{ikj}\right)^{x_{ikj}}\right]$$

但是约束在原来的基础上需要进行调整，即需要同时包含：每个发射车仅有 r_i 枚导弹，每个雷达最多连通 l_k 个发射车，在有的要求中，还会增加每个目标最多分配 s_j 个杀伤链的限制，由于增加该限制只会降低问题的求解难度，因此我们

在本文的范围内并不考虑这个约束，因此我们可以得到新的数学模型为：

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} v_j \left[1 - \prod_{i=1}^{m} \prod_{k=1}^{t} \left(1 - p_{ikj}\right)^{x_{ikj}} \right] \\
\text{s.t.} \quad & \sum_{k=1}^{t} \sum_{j=1}^{n} x_{ikj} \le r_i, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ikj} \le l_k, \quad \forall k = 1, \cdots t \\
& \sum_{i=1}^{m} \sum_{k=1}^{t} x_{ijk} \le s_j \quad \forall j = 1, \cdots, n \\
& x_{ijk} \in \mathbb{Z}^+
\end{aligned}
$$

此时问题也可以比较自然地推广到线性化模型中：

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} \sum_{s \in S} q_{js} y_{js} & \\
\text{s.t.} \quad & \sum_{j=1}^{n} \sum_{s \in S} n_{si} y_{js} \le l_i & \forall\, i \in I \\
& \sum_{j=1}^{n} \sum_{s \in S} n_{sk} y_{js} \le r_j & \forall\, k \in K \\
& \sum_{s \in S} y_{js} = 1 & \forall\, j \in J \\
& y_{js} \in \{0, 1\} & \forall\, j \in J,\ s \in S
\end{aligned}
$$

在第三章中，我们将详细地介绍针对这个模型如何使用列生成方法进行求解。

2.3　武器目标分配问题的外逼近求解方法

在 2.1 节中，我们已经证明了问题具有凸性，因此一个直观的求解方法就是使用一些凸优化的手段来对问题进行直接求解。（差参考文献），如何对将目标函数中的非线性转化为约束中的非线性会影响问题的求解难度，而对于一个凸整数规划问题，外逼近方法的使用需要与分支定界方法进行结合。因此在本节中的第一部分将首先讨论两种对于问题的凸优化处理方式，并证明其中一种方式相比于另外一种方式更紧，而第二部分将介绍外逼近方法如何与分支定界框架结合，并给出这个问题使用外逼近进行求解的具体形式。

2.3.1 武器目标分配问题的两种凸优化建模方式

对于经典形式的武器目标分配问题，有两种建模方式能够将目标函数中的非线性部分转化到约束中，即：

模型一：

$$\min \quad \sum_{k=1}^{n} V_k \eta_k \tag{P1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le l_i, \qquad \forall\, i \in \boldsymbol{I}, \tag{P1.2}$$

$$\prod_{i=1}^{m} (1 - p_{ij})^{x_i} \le \eta_k \tag{2.4}$$

$$x_{ij} \in \mathbb{N}, \quad \eta_k \in \mathbb{R} \qquad \forall\, j \in \boldsymbol{J},\ i \in \boldsymbol{I}. \tag{P1.3}$$

与模型二：

$$\min \quad \eta \tag{P1.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le l_i, \qquad \forall\, i \in \boldsymbol{I}, \tag{P1.2}$$

$$\sum_{j=1}^{n} V_j \prod_{i=1}^{m} (1 - p_{ij})^{x_{ij}} \le \eta \tag{2.5}$$

$$x_{ij} \in \mathbb{N}, \quad \eta \in \mathbb{R} \qquad \forall\, j \in \boldsymbol{J},\ i \in \boldsymbol{I}. \tag{P1.3}$$

在这两个模型中，当 x_{ij} 的取值一定时，目标函数的取值是一样的，同时如果仅仅考虑 x 是整数时，二者的定义域也是一样的，但是当对问题进行线性松弛后，对应的约束会有所区别，在外逼近方法中计算出的外逼近割也会有所不同，由于更紧的松弛能够带来更好的计算效果，因此我们希望对比这两个模型松弛的紧致程度。

命题 2.3. 模型一比模型二的非线性模型与线性外逼近模型都更紧（模型一松弛的定义域被包含在模型二中）

证明. 对于任意 j，若满足 $\prod_{i=1}^{m} (1 - p_{ij})^{x_i} \le \eta_k$，则对这些不等式的进行线性加和，取系数为 V_j，则可以得到模型二中的不等式，因此模型二的松弛定义域不会比模型一的松弛定义域更紧。

当凸约束被转化为线性外逼近割后，证明类似。 □

因此模型二的定义域比模型一更紧，在对多个凸函数进行线性加和时，使用类似模型二的方式来将约束转化到变量上效果更好。

2.3.2　基于分支定界框架的外逼近方法

求解凸整数规划时，外逼近方法将发生一定变化。最早由 Duran and Grossmann[]（差参考文献）提出整数规划的外逼近方法，它的核心思想是通过有限的外逼近割来将问题等价地转化为一个 MILP 问题。MILP 中的约束时全体整数点处的外逼近割，且如果可行域中的整数点个数是有限的，则只需要在这些整数点设置外逼近割，即可将问题转化为约束个数是有限的（与连续的情形不同）MILP。用数学的语言来表述这一点，即当 $f(x)$ 和 $g(x)$ 都是凸函数时，下面两个模型是等价的：模型一 (MINLP)：

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_j(x) \leq 0, \forall j \ in J \\ & x \in X \subseteq Z^{|I|} \end{aligned}$$

模型二 (MILP - OA)：

$$\begin{aligned} \min \quad & \eta \\ \text{s.t.} \quad & \eta \geq f(x^*) + \nabla f(x^*)^T (x - x^*, \forall x^* \in X) \\ & f(x^*) + \nabla f(x^*)^T (x - x^*) \leq 0, \forall j \in J, x^* \in X \\ & x \in X \end{aligned}$$

命题 2.4. *假设 $X \neq \Phi$, f 和 g 连续，凸，可微，且没有其他的非连续变量时，MINLP 模型的最优值与 MILP-OA 模型的最优值相等，且任何一个 MINLP 的最优解也一定时 MILP-OA 问题的最优解。*

而在武器目标分配问题当中，只有目标函数是非线性的，因此只需要将问题转化为

$$\begin{aligned} \min \quad & \eta_j \\ \text{s.t.} \quad & \eta_j \geq f(x^*) + \nabla f(x^*)(x - x^*), \quad \forall x^* \in X, \\ & x_i \in \{0, 1\} \end{aligned}$$

其中每一个由外逼近带来的约束：

$$\eta_j \geq f(x^*) + \nabla f(x^*)(x - x^*)$$

被称为一个外逼近割

从实际求解的角度讲，虽然这两个问题等价，但是这个形式中的外逼近割个数等于整数点个数，如果将约束全部列出，相当于需要将全部的整数点进行枚举，这从计算量上而言是不现实的，因此基本的求解方式是先生成一个简约外逼近主问题 (OA-MP($\hat{X}$))，它只包含一部分问题的约束，即

$$\begin{aligned} \min \quad & \eta_j \\ \text{s.t.} \quad & \eta_j \geq f(x^*) + \nabla f(x^*)(x - x^*), \quad \forall x^* \in \hat{X}, \\ & x_i \in \{0, 1\} \end{aligned}$$

这个算法的具体迭代方式与第一章中提到的外逼近迭代方式相似，但是终止条件有所不同。

算法 3 外逼近算法

1: **procedure** 外逼近 (f, X) ▷ 输入：f，可行域 X
2: 初始化 $x^0 \in X$，$S^0 = \{(x, \eta) \in X \times \mathbb{R} : f(x^0) + \nabla f(x^0)(x - x^0) \leq \eta, \}$
3: 初始化 $\eta^0 = \arg\min\{\eta \in \mathbb{R} : \eta \geq f(x^0) + \nabla f(x^0)(x - x^0), x \in X\}$
4: $k \leftarrow 0$
5: **while** $f(x^k) + \nabla f(x^k)(x - x^k) > \eta^k$ **do**
6: 更新 $S^{k+1} = S^k \cap \{(x, \eta) \in X \times \mathbb{R} : f(x^k) + \nabla f(x^k)(x - x^k) \leq \eta^k, \forall i\}$
7: 求解近似问题 $\{\min \eta : (x, \eta) \in S^{k+1}\}$，得到解 (x^{k+1}, η^{k+1})
8: $k \leftarrow k + 1$
9: **end while**
10: **return** 解 (x^k, η^k) ▷ 输出：凸优化问题的最优解
11: **end procedure**

这个算法与经典的外逼近方法类似，同时根据定理，能够在有限步中终止。但是这个算法在第 6 步中需要求解一个整数规划问题，而且每次进行一步迭代都需要重新计算这个整数规划问题，这会带来额外的开销，因此需要将这个求解与分支切割方法（branch and cut）相结合，来降低问题的求解复杂度。

同时我们注意到，虽然在全部的整数点增加外逼近割就可以保证两个问题的最优解等价，但是在任何一个分数点增加外逼近割依然是 MILP 的一个有效不等式，在进行分支定界的过程中计算线性松弛不可避免地会计算出多个分数点，此时由于外逼近割的计算比较简单，因此加入分数点处的外逼近割有可能会降低问题的求解复杂度。

算法的基本思想是通过 callback 函数，在分支定界算法的基础上增加了外逼近割的部分，其基本思想与外逼近方法类似，先选择一部分外逼近割来得到一个 OA-MP 问题，在进行分支定界进行求解时，如果得到了一个整数可行解或者分数松弛解，就进行一次验证，如果此时得到的 (x,η) 组合不满足 $f(x)\le\eta$ 的约束，就意味着当前的 OA-MP 的定义域过大，无法割掉这个错误的解，此时就需要增加外逼近割来作为分支定界方法的全局割。其算法流程如下：

算法 4 分支定界算法

procedure 分支定界 (f, X) ▷ 输入：目标函数 f，可行域 X
  初始化根结点 q_0，q_0.LB $\leftarrow -\infty$，$Q \leftarrow \{q_0\}$，GLB $\leftarrow -\infty$，GUB $\leftarrow \infty$;
  while $Q \neq \varnothing$ 且 GLB < GUB **do**
    选取 $q \in Q$，更新 $Q \leftarrow Q \setminus \{q\}$，更新 GLB $\leftarrow \min\{q.\text{LB} \mid q \in Q\}$;
    if q 的线性松弛问题可行 **then**
      记线性松弛问题的解为 $(x^k, \eta$，最优值为 c;
      if $f(x^k) > \eta^k$ **then**
        (可选) 增加外逼近割： $f(x^k) + \nabla f(x^k)(x - x^k) \leq \eta^k$
      end if
      if $c <$ GUB **then**
        if x 是原问题可行解 **then**
          令 $x^* \leftarrow x^k$，GUB $\leftarrow c$;
          if $f(x^k) > \eta^k$ **then**
            增加外逼近割： $f(x^k) + \nabla f(x^k)(x - x^k) \leq \eta^k$
          end if
        else
          将 q 分成 n 个子问题 $q_1, \ldots, q_n$，更新 $Q \leftarrow Q \cup \{q_1, \ldots, q_n\}$;
        end if
      end if
    end if
  end while
  if GUB $< \infty$ **then**
    return 最优解 x^* 和最优值 GUB $= \eta^*$;
  else
    return 问题不可行;
  end if
end procedure

2.4　数值实验

具目前我能找到的公开数据集而言，针对 WTA（武器-目标分配）问题的大规模数据集相对较少。目前公开的主要数据集是 Sonuc 和 Bayir（2017 年）提供的，可在http://web.karabuk.edu.tr/emrullahsonuc/wta上获取。以往的研究通常根据问题规模的不同采用不同的数据处理方式：对于小规模问题，数据通常直接在论文中给出；而对于大规模问题，则通常采用随机生成的数据集，并说明数据生成的方法和参数。

在历史上，研究者们考虑了不同规模的 WTA 问题实例，从最小的 4 个武器和 4 个目标到最大的 200 个武器和 400 个目标不等。例如，早期的研究如 Manne（1958 年）和 Wang 等人（2008 年）分别考虑了最多 4 个武器和 4 个目标以及最多 7 个武器和 10 个目标的情景。中等规模的情景包括 Lee 等人（2002 年）和 Lee 等人（2003 年）分别研究的最多 120 个武器和 80 个目标以及 120 个武器和 120 个目标的情景。大规模的情景则包括 Ahuja 等人（2007 年）研究的最多 200 个武器和 400 个目标的情景，以及 Gelenbe 等人（2010 年）和 Sonuc 和 Bayir（2017 年）研究的最多 200 个武器和 200 个目标的情景。文献中考虑的最大武器-目标比（W-T 比）为 4。

在近年的研究中，Lu(2021) 扩展了这一研究范围，考察了达到 400 个武器和 400 个目标的规模，进一步推动了 WTA 问题在大规模情境下的研究。此外，Alexandre Colaers Andersen 等人（2022 年）也对 WTA 问题进行了深入研究，他们考虑了武器-目标比为 1.5 的情景，其中最大规模达到了 700 个武器和 700 个目标，这一规模远超过之前的研究，为 WTA 问题的大规模数据集研究提供了新的参考。

在本论文中的第二章与第三章中，我们考虑以下数据集：我们首先采用了 Sonuc 和 Bayir（2017 年）公开的数据集，并将其用于与我们的算法进行比较。此外，我们还随机生成了多种不同的场景，包括不同的武器目标比例和武器规模，这些情景覆盖了 Ahuja 等人（2007 年）、Gelenbe 等人（2010 年）和 Sonuc 和 Bayir（2017 年）Lu(2021) 和 Alexandre Colaers Andersen 等人（2022 年）考虑的问题的规模。我们的最大问题规模达到了 1200 个武器和 1200 个目标，最大 W-T 比达到了 4，同时我们假设 $p_H = 0.9$ 为杀伤概率最大值，$p_L = 0.6$ 为杀伤概率最小值。所有链的杀伤概率为在 $p_L = 0.6$ 到 $p_H = 0.9$ 之间均匀分布的随机数。$v_H = 100$

为目标威胁值的最大值，$v_L = 25$ 为目标威胁值的最小值。所有目标的威胁值设定为在 $v_L = 25$ 到 $v_H = 100$ 之间均匀分布的随机整数。

在第二章中，我们主要考虑 Sonuc 和 Bayir（2017 年）公开的数据集，并且将计算结果作为后续测试的基准，并通过计算结果体现 WTA 问题求解的难点。在这个数据集中，武器与目标的数量一致，取自 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200。

我们对比了 Lu、Ahuja 的方法，这是我们得到的数值结果:

第 3 章　基于列生成的复杂静态武器目标分配求解算法

3.1　引言

对于武器目标分配问题而言，无论是经典模型还是改进后的包含雷达的复杂模型，其求解的难点都在于目标函数的非线性。为了解决非线性的

3.2　列生成算法

3.2.1　复杂的武器目标分配问题数学模型及线性化

为了应对具体的作战场景，我们分别考虑问题的约束条件和目标函数：优化目标一：最小化敌方目标的加权存活概率 **优化目标**

$$f_1(x)=\sum_{j=1}^{n} v_j\left[1-\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ikj}}\right] \quad \text{(毁伤概率)}$$

优化目标二：最小化整体的作战时间

$$f_2(x)=\max_{i,k,j}\{C_{ikj}x_{ikj}\} \quad \text{(打击时间)}$$

约束雷达通道约束：杀伤链使用雷达 j 的次数应小于等于其通道数 r_j。

$$\sum_{i\in I}\sum_{k\in J} x_{ikj}\le r_j,\quad \forall j\in K \tag{3.1}$$

发射车通道约束：杀伤链使用发射车 i 的次数应小于等于弹药数 l_i。

$$\sum_{k\in J}\sum_{j\in K} x_{ikj}\le l_i,\quad \forall i\in I \tag{3.2}$$

杀伤链数目约束：目标 k 最多可构建的杀伤链条数为 s_k。

$$\sum_{j\in K}\sum_{i\in I} x_{ikj}\le s_k,\quad \forall k\in J \tag{3.3}$$

我们将两个优化目标进行线性组合，并使用参数 θ 控制二者之间的权重，可

以得到问题的模型为：

$$
\begin{aligned}
\max \quad & f_1(x) - \theta f_2(x) \\
\text{s.t.} \quad & \sum_{k=1}^{t}\sum_{j=1}^{n} x_{ikj} \le r_i, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ikj} \le l_k, \quad \forall k = 1, \cdots t \\
& \sum_{i=1}^{m}\sum_{k=1}^{t} x_{ijk} \le s_j \quad \forall j = 1, \cdots, n \\
& x_{ijk} \in \mathbb{Z}^+
\end{aligned}
$$

此时我们希望通过与第二章中提到的类似的方式来对问题进行线性化。与之前的转化方式类似，假设共有 m 个发射车与 t 个雷达，最多共可以生成 $m \times t$ 条杀伤链。此时对于任意一个目标 j，其最多分配 s_j 个杀伤链，因此最多有 $(1+s_j)^{(m\times t)}$ 种不同的场景。此时使用整数变量 n_{si}，表示在第 s 个场景下，使用武器 i 的个数. 类似地，n_{sk} 表示在第 s 个场景下，使用雷达 k 的个数。由于一个杀伤链一定对应着一个雷达和一个武器，因此在任何场景下，都有

$$
\sum_{i=1}^{m} n_{si} = \sum_{k=1}^{t} n_{sk}
$$

而 q_{js} 依然代表问题加权后的毁伤概率，

$$
q_{js} = V_j \left[1 - \prod_{i=1}^{m}\prod_{k=1}^{t} \left(1 - p_{ikj}\right)^{n_{sik}} \right]
$$

同时用 y_{js} 来表示是否对目标 j 使用场景 s，假设 $\hat{S}$ 表示满足约束的场景，比如在上述模型中要求 $\sum_{j\in K}\sum_{i\in I} x_{ikj} \le s_k, \quad \forall k \in J$，则要求所有 $\hat{S}$ 中的场景中杀伤链的个数都小于等于 s_k，整体就可以得到如下的线性化模型：

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n}\sum_{s\in S} q_{js}y_{js} + \theta \max_{i,k,j}\{C_{ikj}x_{ikj}\} && \\
\text{s.t.} \quad & \sum_{j=1}^{n}\sum_{s\in S} n_{si}y_{js} \le r_i && \forall\, i \in I \\
& \sum_{j=1}^{n}\sum_{s\in S} n_{sk}y_{js} \le l_k && \forall\, k \in K \\
& \sum_{s\in S} y_{js} = 1 && \forall\, j \in J \\
& y_{js} \in \{0,1\} && \forall\, j \in J,\ s \in \hat{S}
\end{aligned}
$$

而此时目标函数中的时间项依然不是线性形式，因此增加变量 η 来控制目标函数中的时间项，来把问题转化为一个线性规划

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n}\sum_{s\in S} q_{js}y_{js} + \theta\eta && \\
\text{s.t.} \quad & \sum_{j=1}^{n}\sum_{s\in S} n_{si}y_{js} \le l_j && \forall\, i \in I \\
& \sum_{j=1}^{n}\sum_{s\in S} n_{sk}y_{js} \le r_i && \forall\, k \in K \\
& t_{js}y_{js} - \eta \le 0 && \forall\, j \in J, s \in \hat{S} \\
& \sum_{s\in S} y_{js} = 1 && \forall\, j \in J \\
& y_{js} \in \{0,1\} && \forall\, j \in J,\ s \in \hat{S} \\
& \eta \ge 0 &&
\end{aligned}
$$

3.2.2　一个例子

我们在这里举一个最简单的例子来指出线性化方法时如何转化模型的。假设一个最简单的情形，共有 2 个目标，2 个武器，2 个雷达，即 $i = j = k = 2$，此时雷达与武器的组合共有 $2 \times 2 = 4$ 种，即有四种可以选的杀伤链。我们假设每个杀伤链给每个目标最多分配一次（即仅仅考虑 0-1 变量），此时可能的打击场景共有 $2^4 = 16$ 种，即每个杀伤链都可以选择或者不选择，我们用一个 4 位的二

进制数表示这几种杀伤链是否被选择。在对内容进行枚举时，先对武器进行内层循环，再对雷达进行外层循环，即四个位置分别是（W0R0, W1R0, W0R1, W1R1)

由于二进制数可以与 10 进制数一一对应，因此这 16 个不同的打击场景可以被 0 到 15 的数字来表示，比如 6 对应的二进制数就是 (0,1,1,0)，它所代表的打击场景就是选择“武器 1 和雷达 0”与“武器 0 和雷达 1”这两条杀伤链对某个目标进行打击。下面如果我们用一个 10 进制数表示一个打击场景，指的就是它转化为二进制数后对应的打击场景。

而 $q_j s$ 表示的就是对目标 j 使用打击场景 s 时的加权概率，比如 $q_{1,6}$ 对应的就是使用 $(0,1,1,0)$ 来打击目标 1 是的毁伤概率，即

$$q_{1,6} = V_1(1 - p_{1,1,0})(1 - p_{1,0,1})$$

因此问题可以被表述为

$$\text{Minimize} \quad (q_{1,0}y_{1,0} + q_{1,1}y_{1,1} + \cdots + q_{1,15}y_{1,15}) + (q_{2,0}y_{2,0} + q_{2,1}y_{2,1} + \cdots + q_{2,15}y_{2,15}) \tag{3.4}$$

$$\text{s. t.} \quad (n_{0,i_0} \cdot y_{1,0} + n_{1,i_0} \cdot y_{1,1} + n_{2,i_0} \cdot y_{1,2} + \cdots + n_{15,i_0} \cdot y_{1,15}) \tag{3.5}$$

$$+ (n_{0,i_0} \cdot y_{2,0} + n_{1,i_0} \cdot y_{2,1} + \cdots + n_{15,i_0} \cdot y_{2,15}) \leq 1 \tag{3.6}$$

$$(n_{0,i_1} \cdot y_{1,0} + n_{1,i_1} \cdot y_{1,1} + n_{2,i_1} \cdot y_{1,2} + \cdots + n_{15,i_1} \cdot y_{1,15}) + \tag{3.7}$$

$$(n_{0,i_1} \cdot y_{2,0} + n_{1,i_1} \cdot y_{2,1} + \cdots + n_{15,i_1} \cdot y_{2,15}) \leq 1 \tag{3.8}$$

$$(n_{0,r_0} \cdot y_{1,0} + n_{1,r_0} \cdot y_{1,1} + n_{2,r_0} \cdot y_{1,2} + \cdots + n_{15,r_0} \cdot y_{1,15}) + \tag{3.9}$$

$$(n_{0,r_0} \cdot y_{2,0} + n_{1,r_0} \cdot y_{2,1} + \cdots + n_{15,r_0} \cdot y_{2,15}) \leq 1 \tag{3.10}$$

$$(n_{0,r_1} \cdot y_{1,0} + n_{1,r_1} \cdot y_{1,1} + n_{2,r_1} \cdot y_{1,2} + \cdots + n_{15,r_1} \cdot y_{1,15}) + \tag{3.11}$$

$$(n_{0,r_1} \cdot y_{2,0} + n_{1,r_1} \cdot y_{2,1} + \cdots + n_{15,r_1} \cdot y_{2,15}) \leq 1 \tag{3.12}$$

$$y_{1,0} + y_{1,1} + \cdots + y_{1,15} = 1 \tag{3.13}$$

$$y_{2,0} + y_{2,1} + \cdots + y_{2,15} = 1 \tag{3.14}$$

$$t_{1,0} \cdot y_{1,0} + t_{1,1} \cdot y_{1,1} + \cdots + t_{1,15} \cdot y_{1,15} = 1 \tag{3.15}$$

$$t_{2,0} \cdot y_{2,0} + t_{2,1} \cdot y_{2,1} + \cdots + t_{2,15} \cdot y_{2,15} = 1 \tag{3.16}$$

$$y_{1,0}, y_{1,1}, \ldots, y_{1,7}, y_{2,0}, y_{2,1}, \ldots, y_{2,7} \in \{0,1\} \tag{3.17}$$

而每一个打击场景都能够计算出它使用了多少武器和多少雷达，比如上面提到的 6 场景，就是使用了武器 1 与雷达 0 和武器 0 与雷达 1 的组合，此时一共

使用了武器 0，武器 1，雷达 0，雷达 1 各一次，即 $n_{6,i_0} = n_{6,i_1} = n_{6,r_0} = n6, r_1 = 1$，因此这个场景对应原问题中的列就是 $(1,1,1,1,1,0,t_{1,6},0)^T$。用这样的方式就能够得到原问题转化后的线性化模型。

可以看出，这个形式中的变量个数非常多，而列生成为解决这个问题提供了一个框架，下一节中将详细介绍如何利用列生成来求解这个问题。

3.2.3　线性化后的武器目标分配问题的列生成形式

首先我们会求解这个问题的线性松弛，因此整数限制（5）可以被转化为 $0 \le y_j s \le 1$。此时由于约束（4），当所有的 $y_j s \le 0$ 成立时，$y_j s \le 1$ 自然成立，因此可以舍去 $y_j s \le 1$ 部分。

再观察约束（3），可以看到这系列的约束数量非常多，有指数多个，但是由于约束（4）的存在，这些约束可以进行聚合，将其整合为一个约束，且该整合方式在不影响整数可行域的同时，还会让问题的松弛变得更紧

$$\sum_{s \in S} t_{js} y_{js} - \eta \le 0 \quad \forall j \in J$$

命题 3.1. *在 $\sum_{s \in S} y_{js} = 1$ 且 $y_{js}, t_{js} \ge 0$ 的条件下新的约束形式比原来的约束形式 $t_{js} y_{js} - \eta \le 0 \quad \forall j \in J, s \in \hat{S}$ 更紧*

证明. 我们假设 j 是固定的，当

$$\sum_{s \in S} t_{js} y_{js} - \eta \le 0$$

成立时，由于 $t_{js} \ge 0, y_{js} \ge 0$，因此

$$t_{js^*} y_{js^*} \le \eta - \sum_{s \in S, s \ne s^*} t_{js} y_{js} \le \eta \tag{3.18}$$

因此新的约束比原来的约束更紧。同时假设原来一组解满足约束 $\sum_{s \in S} y_{js} = 1$，则

$$\sum_{s \in S} t_{js} y_{js} \le \max_{s \in S}(t_{js}) \cdot \sum_{s \in S} y_{js} = \max_{s \in S} t_{js} \le \eta$$

因此新的约束不会割掉原来的整数点。因此新的约束比原来的约束更紧。　□

在对约束进行上述调整后，我们可以得到问题的线性松弛：

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n}\sum_{s\in S} q_{js}y_{js} + \theta\eta \\
\text{s.t.} \quad & \sum_{j=1}^{n}\sum_{s\in S} n_{si}y_{js} \le r_i && \forall\, i \in I \\
& \sum_{j=1}^{n}\sum_{s\in S} n_{sk}y_{js} \le l_j && \forall\, k \in K \\
& \sum_{s\in S} t_{js}y_{js} - \eta \le 0 && \forall\, j \in J \\
& \sum_{s\in S} y_{js} = 1 && \forall\, j \in J \\
& y_{js}, \eta \ge 0 && \forall\, j \in J,\ s \in S
\end{aligned}
$$

可以看出，这个模型的约束个数为 $m+n+2\times k$，但是变量个数却是 $2^{(}m*n)\times k+1$，具有指数多列，因此对于这个模型我们希望采用列生成的框架来对问题进行求解，因此我们把这个问题称为武器目标分配问题列生成的主问题（MP）

根据上面的数学形式，该问题中的约束分为 4 类，我们从上到下依次称其为：武器约束、雷达约束、时间约束、目标约束。记这四个约束的对偶变量分别为武器约束 u_i，雷达约束 μ_k，时间约束 w_j，目标约束 v_j，即可写出对偶问题：

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{m} 4u_i + \sum_{k=1}^{t} 8\mu_k + \sum_{j=1}^{n} v_j \\
\text{s.t.} \quad & \sum_{i=1}^{m} n_{si}u_i + \sum_{k=1}^{t} n_{sk}\mu_k + t_{js}w_j + v_j \le q_{js}, \quad \forall j \in J, s \in S_j^w \quad (P) \\
& \sum_{j=1}^{J} -w_j \le 1 \\
& u_i \le 0, \quad \mu_k \le 0 \\
& w_j \le 0, \quad v_j \text{ free}
\end{aligned}
$$

列生成的基本思路是先选择其中一部分列作为备选列，其他列对应的变量值设置为 0。我们假设第 W 次计算时，每个目标 j 备选的场景即列的集合为 S_j^w，

则可以得到列生成的约化主问题：

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n}\sum_{s\in S} q_{js}y_{js} + \theta\eta \\
\text{s.t.} \quad & \sum_{j=1}^{n}\sum_{s\in S_j^W} n_{si}y_{js} \le r_i && \forall\, i \in I \\
& \sum_{j=1}^{n}\sum_{s\in S_j^W} n_{sk}y_{js} \le l_j && \forall\, k \in K \\
& \sum_{s\in S_j^W} t_{js}y_{js} - \eta \le 0 && \forall\, j \in J \\
& \sum_{s\in S_j^W} y_{js} = 1 && \forall\, j \in J \\
& y_{js}, \eta \ge 0 && \forall\, j \in J,\ s \in S_j^W
\end{aligned}
$$

求解这个问题得到的最优解也会获得一组对偶变量，记它们为武器约束 u_i^W，雷达约束 μ_k^W，时间约束 w_j^W，目标约束 v_j^W，此时它们是 RMP 对偶问题的最优解，但是未必是 MP 对偶问题最优解，在经典的列生成方法中，我们会找到那些被违背最严重的对偶约束，即希望找到

$$\sum_{i=1}^{m} n_{si}u_i + \sum_{k=1}^{t} n_{sk}\mu_k + t_{js}w_j + v_j \le q_{js}$$

这些约束哪一项被最严重地违背，即

$$q_{js} - \left[\sum_{i=1}^{m} n_{si}u_i + \sum_{k=1}^{t} n_{sk}\mu_k + t_{js}w_j + v_j\right] < 0$$

何时成立，并且希望尽可能地小。而针对每个目标，不同的对偶约束对应了原问题中的不同列，即不同的场景 s，因此问题就会被转化为

$$\min q_{js} - \left[\sum_{i=1}^{m} n_{si}u_i + \sum_{k=1}^{t} n_{sk}\mu_k + t_{js}w_j + v_j\right]$$

如果最小值比 0 还小，则意味着依然没有求到最优解，而如果全部最小值不小于 0，则迭代终止。否则把选择出的列加入到 RMP 中，即得到了新的约化主问题，循环迭代知道没有违背的对偶约束，即终止迭代。

为了证明上述改进的列生成算法能够在有限步内终止且达到最优解，我们需要分两步进行证明，第一步是如果上述终止条件成立，则一定可以得到问题的

最优解。第二步则是算法迭代一定可以在有限步终止于这个条件。对于第一步，证明需要用到如下引理：

引理 3.2 (对偶定理). 对于任意的线性规划问题，如果原问题

$$\begin{aligned} min \quad & c^T x \\ s.t. \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

和对偶问题

$$\begin{aligned} max \quad & b^T y \\ s.t. \quad & A^T y \leq c \end{aligned}$$

存在可行解 x 和 y，则有

(1) 原问题的目标函数值大于等于对偶问题的目标函数值 $c^T x \geq b^T y$。

(2) 如果存在一组原问题的解 x^* 和对偶问题的解 y^*，使得原问题的目标函数值和对偶问题的目标函数值相等，即有 $c^T x^* = b^T y^*$，则 x^* 和 y^* 分别是原问题和对偶问题的最优解。

在列生成方法中，简约主问题 (RMP) 的最优解 x_B 是原问题的一组可行解（即其他变量取 0），但简约主问题的对偶问题的解 π 未必是主问题 (MP) 对偶问题的可行解。如果验证了所有的约化费用都大于等于 0，即保证了对偶问题的约束 $A^T y \leq c$ 一定成立，从而 π 是主问题对偶问题的一组可行解。由于 x_B 和 π 分别是简约主问题和简约主问题对偶问题的最优解，因此可以保证 $c^T x_B = b^T \pi$，即根据对偶定理，即得到 x_B 和 π 是原始主问题 MP 的最优性。

对于第二步，证明如下：

假设原问题有 m 列（变量），主问题初始化有 m_0 列（变量），如果在步骤四的判断过程中发现存在使得约化系数小于 0 的变量（列），则将其加入到主问题的备选列当中，每次迭代至少使得备选列增加 1，因此经过至多 $m - m_0$ 次迭代，主问题就会包含原问题全部的列，此时一定会得到原问题的最优解。而当在步骤四的判断过程中发现不存在使得约化系数小于 0 的变量（列），则问题已经达到了最优解。

综上所述，可以得到如下定理：

定理 3.3. 使用上述的列生成算法，问题一定可以在 $m-m_0$ 次迭代内终止，且此时可得到原问题的最优解。

而如何有效地求解列生成子问题是这个算法的关键，因此在下一小节中我们会详尽地讨论问题的求解方式。

3.3　子问题的求解

3.3.1　子问题的两种外逼近建模方法

在求解子问题时，列生成需要求解的子问题形式为：

$$\min q_{js}-\left[\sum_{i=1}^{m} n_{si}u_i+\sum_{k=1}^{t} n_{sk}\mu_k+t_{js}w_j+v_j\right]$$

$$\min \quad q_{js}-\left[\sum_{i=1}^{m} n_{si}u_i+\sum_{k=1}^{t} n_{sk}\mu_k+t_{js}w_j+v_j\right] \text{ s.t.} \qquad s\in S_k^W$$

将 q_{js} 的定义代入，同时 s 就对应的是全部的可行场景，因此可以重新转化为使用 x 来表现的形式，从而得到的形式是：

$$V_j\left[\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ikj}}\right]-\sum_{i=1}^{m}\left(u_i\sum_{k=1}^{t}x_{ik}\right)-\sum_{k=1}^{t}\left(\mu_k\sum_{i=1}^{m}x_{ik}\right)-\max\{t_{ikj}\cdot x_{ikj}\}w_j-v_j$$

这个问题的每一个可行解对应的都是一个 MP 问题的列（变量），求解这个列生成子问题就可以得到下一步往列生成主问题中增加的变量。

而对这个式子进行一下整合，对于时间项的处理与主问题类似，需要引入一个额外的变量来消除式子中的 max 项，为了叙述简单，在这里不考虑时间项，而仅仅考虑其他的项，我们就可以得到子问题的目标函数：

$$V_j\left[\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ikj}}\right]-\sum_{i=1}^{m}\sum_{k=1}^{t}(u_i+\mu_k)x_{ik}$$

而这个问题的约束则需要准确地刻画所有的 S 中的场景，这与原问题的刻画方式类似，因此可以得到约束：

$$\sum_{k=1}^{t}x_{ik}\le r_i,\quad \forall i\in I$$

$$\sum_{i=1}^{m} x_{ik} \le l_k, \quad \forall k \in K$$

$$x_{ik} \le s_j$$

将目标与约束进行整合后，假设不考虑时间，即可得到**子问题形式为：**

$$\begin{aligned}
\min \quad & V_j \left[\prod_{i=1}^{m} \prod_{k=1}^{t} \left(1 - p_{ikj}\right)^{x_{ik}} \right] - \sum_{i=1}^{m} \sum_{k=1}^{t} (u_i + \mu_k) x_{ik} - v_j \\
\text{s.t.} \quad & \sum_{k=1}^{t} x_{ik} \le r_i, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m} x_{ik} \le l_j, \quad \forall k = 1, \cdots t \\
& x_{ik} \le s_j \\
& x_{ik} \in \mathbb{N}
\end{aligned}$$

为了能够更好地讨论问题的数学形式，记 x_{ik} 的线性系数 $u_i + \mu_k$ 为 c_{ik}，$-v_j$ 记为 c_0，表示目标函数的常数项，此时有 $c_{ik} \ge 0$，则问题的形式可以被写为：

$$\begin{aligned}
\min \quad & V_j \left[\prod_{i=1}^{m} \prod_{k=1}^{t} \left(1 - p_{ikj}\right)^{x_{ik}} \right] + \sum_{i=1}^{m} \sum_{k=1}^{t} c_{ik} x_{ik} + c_0 \\
\text{s.t.} \quad & \sum_{k=1}^{t} x_{ik} \le r_i, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m} x_{ik} \le l_j, \quad \forall k = 1, \cdots t \\
& x_{ik} \le s_j \\
& x_{ik} \in \mathbb{N}
\end{aligned}$$

可以看出，此时子问题与原问题具有一定的相似性，但是把原问题中多个目标之间的耦合拆解为每个目标单独计算，但是对于选择的杀伤链增加了一个类似于费用的线性惩罚项。当我们假设问题是 0-1 变量时，原模型的变量个数是 $m \times n \times k$，它的搜索空间为 $2^{m \times n \times k}$，但是子问题则仅仅有 $m \times k$ 个变量，因此它的搜索空间为 $2^{m \times k}$，因此比原问题更容易进行求解。

同样，由于我们已经证明了 $f_j(x)=\prod_{i=1}^{m}(1-p_{ij})^{x_{ij}}$ 是一个凸函数，子问题同样也是凸的，因此我们对这个凸问题同样也有转化为指数的处理方式，它们影响了使用外逼近方法来求解问题的形式，第二种建模方式首先考虑引入变量 η 来将非线性转移到约束中

$$
\begin{aligned}
\min \quad & V_j\eta+\sum_{i=1}^{m}\sum_{k=1}^{t}c_{ik}x_{ik}+c_0 \\
\text{s.t.} \quad & \prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ik}}\le\eta \\
& \sum_{k=1}^{t}x_{ik}\le r_i, \quad \forall i=1,\cdots,m \\
& \sum_{i=1}^{m}x_{ik}\le l_k, \quad \forall k=1,\cdots t \\
& x_{ik}\le s_j \\
& x_{ik}\in\mathbb{N}
\end{aligned}
$$

增加一个中间参数 w，来让非线性部分的结构变得更加简单即考虑对于：

$$
\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ik}}\le e^{w}
$$

两边取对数，即可得到

$$
\sum_{i=1}^{m}\sum_{k=1}^{t}x_{ik}\ln\left(1-p_{ikj}\right)\le w
$$

从而问题的形式即为：

$$
\begin{aligned}
\min \quad & V_j\eta + \sum_{i=1}^{m}\sum_{k=1}^{t} c_{ik}x_{ik} + c_0 \\
\text{s.t.} \quad & e^{w} \le \eta \\
& \sum_{i=1}^{m}\sum_{k=1}^{t} x_{ik}\ln\left(1-p_{ikj}\right) \le w \\
& \sum_{k=1}^{t} x_{ik} \le r_i, \quad \forall i = 1,\cdots,m \\
& \sum_{i=1}^{m} x_{ik} \le l_k, \quad \forall k = 1,\cdots t \\
& x_{ik} \le s_j \\
& x_{ik} \in \mathbb{N}
\end{aligned}
$$

3.3.2 子问题的求解

对于上面两种数学模型，由于它们都是凸的，因此都可以使用外逼近的求解框架，即考虑给定一个凸函数 $f(x)$ 与其中一点 x^*，由于不等式一定成立：

$$
f(x) \ge f(x^*) + \nabla f(x^*)(x - x^*)
$$

在该问题中

$$
f(x) = V_j\left[\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ik}}\right] - \sum_{i=1}^{m}\sum_{k=1}^{t}(u_i+\mu_k)x_{ik} - w_j - v_j
$$

因此有 x_{ik} 前的线性化系数为：

$$
\frac{\partial f(x)}{\partial x_{ik}} = V_j\ln(1-p_{ikj})\prod_{i=1}^{m}(1-p_{ikj})^{x_{ik}} - u_i - \mu_k \triangleq l_{ik}
$$

因此，在 $\overline{x_{ik}}$ 处的外逼近割为：

$$
\eta_j \ge V_j\sum_{i=1}^{m}\sum_{k=1}^{t}\left[\ln(1-p_{ikj})\prod_{i=1}^{m}(1-p_{ikj})^{x_{ik}} - u_i - \mu_k\right]\left(x_{ik} - \overline{x_{ik}}\right) + f(x_{ik})
$$

该割的右端项即为：

$$
V_j\sum_{i=1}^{m}\sum_{k=1}^{t}\left[\ln(1-p_{ikj})\prod_{i=1}^{m}(1-p_{ikj})^{x_{ik}} - u_i - \mu_k\right]\overline{x_ik} - f(\overline{x_ik}) \triangleq r_{\overline{x_{ik}}}
$$

$$
\begin{aligned}
\min \quad & \eta \\
\text{s.t.} \quad & \sum_{i=1}^{m}\sum_{k=1}^{t} l_{ik}x_{ik} - \eta \le r_{\overline{x_{ik}}} \\
& \sum_{k=1}^{t} x_{ik} \le 4, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m} x_{ik} \le 8, \quad \forall k = 1, \cdots t \\
& x_{ik} \in \{0, 1, 2, 3, 4\}
\end{aligned}
$$

而对于另外一个模型，其非线性项为：

$$e^{w} \le \eta$$

此时，假设有一个点 (x^*, w^*, η^*)。则问题增加的外逼近割就是

$$e^{w^*}(w - w^*) + e^{w^*} \le \eta$$

整理成变量在左侧的形式就是

$$e^{w^*}w - \eta \le e^{w^*}(w^* - 1)$$

注意到根据问题的形式，在 x_{ij} 给定时，显然有 $e^{w} = \eta = \prod_{i=1}^{m}\prod_{k=1}^{t}\left(1 - p_{ikj}\right)^{x_{ik}}$ 时，目标函数值最小，因此可以得到问题的外逼近主问题形式就是

$$
\begin{aligned}
\min \quad & V_j\eta + \sum_{i=1}^{m}\sum_{k=1}^{t} c_{ik}x_{ik} + c_0 \\
\text{s.t.} \quad & e^{w^*}w - \eta \le w^* - e^{w^*}, \quad \forall (x, w, \eta) \\
& \sum_{i=1}^{m}\sum_{k=1}^{t} x_{ik}\ln\left(1 - p_{ikj}\right) \le w \\
& \sum_{k=1}^{t} x_{ik} \le 4, \quad \forall i = 1, \cdots, m \\
& \sum_{i=1}^{m} x_{ik} \le 8, \quad \forall k = 1, \cdots t \\
& x_{ik} \in \{0, 1, 2, 3, 4\}
\end{aligned}
$$

应用外逼近求解方法就可以求解这两种不同形式的外逼近

在使用外逼近方法进行求解时，需要一组初始外逼近割作为外逼近方法的启动，选择合适的初始割能够有效地降低外逼近方法的求解速度。考虑到问题的目标函数为

$$\min V_j\left[\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ik}}\right]-\sum_{i=1}^{m}\sum_{k=1}^{t}(u_i+\mu_k)x_{ik}-w_j-v_j$$

由于随着 x_{ik} 的增加，第一项会减少，第二项会增加，因此可以通过选择 $1-p_{ijk}+\alpha(ui+\mu_k)$ 来选择问题的初始割。

设置初始割为 $\overline{x_{ik}}$ 使 $1-p_{ijk}+\alpha(ui+\mu_k)$ 最大的一个 x_{ij}, 并设置其值为 4 来满足约束条件

3.4 加速技巧

3.4.1 有效不等式

在进行组问题求解时，如果能够增加一些不等式，使得约束构成的空间能够尽可能地逼近问题的凸包，能够有效降低分支定界方法的迭代速度。因此我们希望能够对问题的性质进行分析，来得到一些有效不等式。

武器数量不等式

在求解武器目标分配问题的子问题的动机是，希望找到对偶模型中，约束尚未满足, 即

$$V_j\left(\prod_{i=1}^{m}(1-p_{ij})^{x_i}\right)-\sum_{i=1}^{m}u_ix_i-v_j<0$$

的行，我们希望分析这个函数的性质，来给子问题的求解带来一些帮助。

首先我们注意到，上个式子可以被拆分成三个部分：$V_j\left(\prod_{i=1}^{m}(1-p_{ij})^{x_i}\right)$，$-\sum_{i=1}^{m}u_ix_i$，$v_j$。由于对偶模型保证了 $u_i\leq 0$，因此这三个部分中的前两个部分都是大于等于 0 的。因此如果有

$$\sum_{i=1}^{m}-u_ix_i-v_j\geq 0$$

或者

$$V_j\left(\prod_{i=1}^{m}(1-p_{ij})^{x_i}\right)-v_j\geq 0$$

对 Reduced cost 的检验都不可能小于 0。基于此，我们进行了一个简单的处理：

下界

我们考虑这部分

$$\sum_{i=1}^{m} -u_i x_i - v_j \geq 0$$

由于线性相加，加的越多就越大，因此可以对 $-u_i$ 的值进行排序。若其中前 k 小的 $-u_i$ 之和已经满足大于等于 v_j 了，就一定会导致上述不等式成立。

上面这个思想的数学表述是，假设对 u_i 已经根据大小进行了排序，排序后为 $u'_1 < u'_2 < \cdots < u'_m$，则假设

$$U_j \triangleq \min\left\{m, \{k : k = 1, \cdots, m, \sum_{i=1}^{k}(-u'_i) - v_j > 0\}\right\}$$

若 $\sum_{i=1}^{m} x_i > U_j$，则一定有

$$\sum_{i=1}^{m} -u_i x_i - v_j \geq 0$$

即该解无法满足 $rc < 0$ 的条件，一定不会被加入到改进解中。

上界与下界类似，我们考虑

$$V_j\left(\prod_{i=1}^{m}(1-p_{ij})^{x_i}\right) - v_j \geq 0$$

由于这个相乘的形式，乘得越多，第一项的值越小，因此我们同样可以对 $(1-p_{ij})$ 来进行从小到大的排序（即对 (p_{ij}) 进行从大到小的排序）。如果武器分配的比较少，导致加权后的毁伤概率已经大于了 v_j，就会导致上述不等式一定成立。

类似地，假设对 (p_{ij}) 已经根据大小进行了排序，排序后为 $p'_{1j} < p'_{2j} < \cdots < p'_{mj}$，则假设

$$W_j \triangleq \max\left\{0, \{k : k = 1, \cdots, m, V_j\prod_{i=1}^{k}(1-p_{ij}) - v_j > 0\}\right\}$$

若 $\sum_{i=1}^{m} x_i < W_j$，则一定有

$$V_j\left(\prod_{i=1}^{m}(1-p_{ij})^{x_i}\right) - v_j \geq 0$$

即该解无法满足 $rc < 0$ 的条件，一定不会被加入到改进解中。

命题 3.4. 假设对 u_i 已经根据大小进行了排序，排序后为 $u'_1 < u'_2 < \cdots < u'_m$，则假设

$$U_j \triangleq \min\left\{m, \{k : k = 1, \cdots, m, \sum_{i=1}^{k}(-u'_i) - v_j > 0\}\right\}$$

且对 (p_{ij}) 已经根据大小进行了排序，排序后为 $p'_{1j} < p'_{2j} < \cdots < p'_{mj}$，则假设

$$W_j \triangleq \max\left\{0, \{k : k = 1, \cdots, m, V_j \prod_{i=1}^{k}(1 - p_{ij}) - v_j > 0\}\right\}$$

则必须要满足条件

$$W_j \leq \sum_{i=1}^{m} x_i \leq U_j$$

才有可能得到使得目标函数值小于 *0.*

证明已经由上面的内容保证。

命题 3.5. 当 $p_i \in (0, 1),\ x \in \mathbb{N}$ 时，有：

$$\prod_{i=1}^{n}(1 - p_i)^{x_i} \geq 1 - \sum_{i=1}^{n} p_i x_i$$

证明. 使用归纳法进行证明。

当 $x_i = 0$ 的时候 $(1 - p_i)^{x_i} = 1$，从而仅仅考虑那些非 $0x$ 的项，即希望证明

$$\prod_{i=1}^{n}(1 - p_i) \geq 1 - \sum_{i=1}^{n} p_i$$

当 $n = 0$ 的时候，这个不等式是显然成立的。假设不等式在 n 的时候成立，此时由于而考虑由于

$$\prod_{i=1}^{n}(1 - p_i)p_{n+1} \leq p_{n+1}$$

从而二式相减可以得到

$$\prod_{i=1}^{n}(1 - p_i) - \prod_{i=1}^{n}(1 - p_i)p_{n+1} \geq 1 - \sum_{i=1}^{n} p_i - p_{n+1}$$

$$\prod_{i=1}^{n+1}(1 - p_i) \geq 1 - \sum_{i=1}^{n+1} p_i$$

从而归纳地证明了这个问题 □

在子问题将非线性项转移到约束中的模型中，凸约束为：

$$\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ik}} \le \eta$$

因此根据上面的不等式，有：

$$\begin{aligned} -\eta &\le -\prod_{i=1}^{m}\prod_{k=1}^{t}\left(1-p_{ikj}\right)^{x_{ik}} \\ &\le \sum_{i=1}^{m}\sum_{k=1}^{t} p_{ik}x_{ik} - 1 \end{aligned}$$

这就是问题的一个有效不等式

3.4.2　选择非最优的子问题结果

基本思想

在经典的列生成的求解算法框架中，每次的列生成子问题求解的都是使得简约价格最小的对偶问题的行，其终止条件是最小的行也大于 0 时，就得到了对偶问题的可行解，从而依据对偶定理，证明了求解得到的解就是问题的最优解。这个方法能够有效执行的基本思想是，那些在对偶问题中简约价格较低的变量增加到原问题中有可能更有效地接近问题的最优解，这个思想在一个通用的问题中可能会有一定的效果，但是在这个具体的问题结构中，在一些特殊的情境下，比如武器和目标的个数相当且各个武器对目标的打击概率相近时，最优解的方案往往对应着每个目标仅分配了 1 到 2 个武器的打击，这意味着每次选择求解到最优的子问题可能会陷入局部极值但并不能正确地选择最终适合的变量。同时因为每次子问题求解的时间相对比较久，且利用分支定界求解问题的过程中有机会获得多个不同的可行解，且它们都可能对 OA 子问题起到改进的作用，因此可以每次不止选择一个解，而是可以同时选择多个解，这样将会更充分地利用每次求解子问题的信息，以降低主问题的求解次数。

3.5　数值实验

第 4 章　基于列生成的动态武器目标分配问题研究

4.1　动态武器目标分配问题的数学建模

4.1.1　动态武器目标分配问题的场景描述与数学建模

对于动态武器目标分配问题，在目前的研究中主要有两种不同的建模思路

第一类模型被称为 shoot-look-shoot 模型，它对于每次打击结果都可以进行观察，因此在打击的过程中目标是否中存在都是确定的。在这种模型中，假设问题共有两阶段，武器在第一阶段中将武器分配给目标，之后随着打击结束对问题进行一次观察。针对第一次打击带来的结果，将剩余的武器分配给幸存的目标。

第二类模型被称为两阶段模型，或者更一般地说多阶段模型，它与 shoot-look-shoot 的不同之处在于它不允许在打击后观察，且每个目标仅被考虑一次，若在某一阶段没被摧毁，后续将不再有摧毁该目标的机会。也就是说，在 shoot-look-shoot 问题中，作战目的是对给定数量的目标进行反复攻击，直到所有目标都被摧毁或达到迭代次数的限制。而在 2 阶段问题中，第一次会出现的目标是给定的，但是在第二阶段，有可能会重新出现新的目标，且目标的数量和类型仅仅能够给出一个概率分布而不是确定性的。在已知多阶段的目标的概率分布的前提下，希望能够给出对目标毁伤的数学期望最高的方案。

动态武器目标分配问题的复杂程度明显高于静态武器目标分配，但是却能够更好地结合实际情况。在更加复杂的模型中，也存在将两种动态模型结合在一起的情况，但是目前针对此问题尚没有比较好的精确求解方法。由于动态武器目标分配问题的算法复杂程度较高，因此目前针对动态武器目标分配的精确算法往往都对模型进行了非常严格的限制，比如 Eckler 和 Burr (1972)[6] 假设所有武器具有相同的杀伤概率并且所有目标具有相同的价值，这导致该问题的数学模型过于简略，从而对实际应用场景不高。

为了能够解决解决实际的作战问题，我们希望能够对于作战场景进行数学建模，使得模型既能够从全局的角度解决包含时间的问题，又控制求解的复杂度在一定的时间范围内。下面我们就将描述对于问题的建模与数学描述。

首先我们对于作战场景进行简短的直观描述。

假设整个战场可以被离散为多个时间节点，在每个时间节点处都需要完成

一次决策，并且也仅仅允许在这些时间节点分配对应的杀伤链进行打击，优化的目标函数是包括毁伤概率、作战花费、作战时间等在内的多个作战目标，整个模型有如下的假设：

- 确定性假设：每个目标的价值是固定的，不随着时间发生变化。同时在给定打击时点、目标、雷达、发射车、弹药的条件下，雷达-发射车-弹药组成的杀伤链对目标的毁伤概率是定值，不随时间发生变化。

- 毁伤概率假设：我们假设每条杀伤链之间是相互独立的。如果一个目标被多个杀伤链同时打击，那么该目标的存活概率就是每个杀伤链打击后的存活概率的乘积。如果一个目标在不同时间被多个不同的杀伤链打击，则它的存活概率是累积到该时点的所有杀伤链打击下存活概率的乘积。优化目标中的毁伤概率就是用 1 减去存活概率的大小。

- 优化目标假设：我们的目标是整合多个优化目标，包括最大化消除的来袭目标威胁值、最小化作战时间和最小化作战消耗。这几个目标之间的关系为线性相加，其系数可随着战场态势由指挥官负责调整，以体现多个作战目标的优先级。

- 资源限制假设：由于资源特性，我们假设为每个目标分配杀伤链时，需要考虑到各类限制，具体而言包括：

i. 在每个时点杀伤链使用雷达的次数应小于等于其通道数

ii. 在每个时点杀伤链使用发射车的次数应小于等于发射车载弹量。

iii. 每个发射车具有总体的弹药上限，在各个时点使用这个发射车的次数累加也应该有上限。

iv. 针对每个目标，最多可构建的杀伤链条数有上限。

v. 每个雷达与一个目标绑定，每个目标占用雷达的一个通道。当雷达锁定一个目标时，多个武器可以同时使用这个雷达的锁定信息。

vi. 每个武器与一个雷达绑定，在武器发射到击中目标这段时间内，不允许更换武器使用的雷达。

vii. 雷达窗口约束：针对每个来袭目标，不同的雷达仅在部分时间能够扫描到这个目标，雷达对目标的锁定必须在这个窗口期内。

viii. 打击窗口约束：针对每个来袭目标，能够被打击得窗口有限，由此限制了给定杀伤链攻击的时点，不能在超出时点范围外对目标进行打击。

对于时间的离散，如果对整体采用均匀离散，如果离散间隔较小，会导致计算复杂度过大，并且过度详细地考虑了比较久的未来，未来节点可能随态势发生较大变化，因此太过详细的考虑或许并不合理。但如果离散间隔较大，则可能导致近期的决策没有办法有效制定，降低了整体的复杂度。因此采用了这种先疏后密的离散方式，近期节点详细考虑，安排较为精细的作战策略。长期节点进行粗略考虑，未来随战场态势进行调整。

作为一个求解系统，在这个数学模型中系统的输入与输出分别是：

- 输入：

1. 敌方有多个目标来袭，可对目标路径进行初步预测，以得到每个目标的可打击时间、雷达可探测时间。

2. 不同时间使用不同资源对敌方目标的杀伤概率、目标的权重。

3. 其他的目标及资源特性。

- 输出：在每个打击时间点，如何给每个目标安排杀伤链。

4.1.2　动态武器目标分配问题的数学模型

为了对上述假设进行数学建模的构建，我们首先引入以下符号

- t：时间，$t \in T \triangleq \{t_0, \cdots, t_N\}$，为了有效降低考虑的时点，$T$ 并非均匀分布. 举例而言，可以设置为

$$T = \{1, 2, 3, 4, 5, 10, 15, 20, 25, 50, 75, 100, 200, 300, 400, 500, \cdots\}$$

- N：离散化的武器发射时间点数量。
- T_k：时间窗，表示第 k 个目标允许打击的时间点，$T_k \subseteq T$
- p_{ijt}^k：杀伤概率，表示对目标 k，在时间 t，使用武器 i 和雷达 j 组成的杀伤链的杀伤概率。
- w_k：目标 k 最后的打击时间，达到 w_k 则认为目标已经命中我方关键内容。
- R_k：目标的权重
- $\hat{T}_{i,t}^k$：在时间 t，使用武器 i 攻击目标 k 所需要的时间

决策变量

- x_{ijt}^k：决策变量，表示对目标 k，在时间 t，使用武器 i 和雷达 j 组成的杀伤链的使用次数。变量类型为整数变量，如果限制每个杀伤链最多使用 1 次，则

决策变量为 0-1 变量。

- z_{jt}^{k}：决策变量，表示对目标 k，在时间 t，是否使用雷达 j 进行锁定。变量类型为 0-1 变量。

时间窗口

- $[L_k, U_k]$：目标允许打击的时间窗口。只有在这个时间窗口内，允许武器进行打击。

- $[L_{j,k}^{R}, U_{j,k}^{R}]$，雷达 j 能够扫描到目标 k 的时间窗口。只有在这个窗口内，雷达能够锁定目标，因此武器打击的整个过程应该囊括在这个范围内。

资源能力约束

首先在每个具体的时间阶段，武器和雷达限于本身的能力，会有一个允许同时锁定/攻击的目标上限

武器资源约束

每个时间点武器最多用于 l_i 条杀伤链。

$$\sum_{j,k} x_{i,j,t}^{k} \le l_i, \quad \forall t, i$$

雷达的通道约束

同一时间，雷达最多锁定 r_j 个目标。

$$\sum_{k} z_{j,t}^{k} \le r_j, \quad \forall t, j$$

弹药上限约束每种武器都有给定数量的弹药，在各个阶段使用的总弹药数有上限，但雷达却没有，假设其个数为 L_i，则这个约束可以表示为

$$\sum_{j,k,t} x_{ijt}^{k} \le L_i, \quad \forall i \in I$$

锁定约束：要求当武器进行打击直到打到目标的过程中，必须要有**唯一一个雷达进行锁定**

$$x_{i,j,t}^{k} \le z_{j,t+n}^{k}, \quad \forall n = 0, 1, \cdots, \hat{T}_{i,t}^{k}$$

雷达窗口约束：雷达必须在规定的窗口内

$$z_{j,t}^{k} \in \begin{cases} \{0,1\} & \text{if } t \in \left[L_{j,k}^{R}, U_{j,k}^{R}\right] \\ \{0\} & \text{else} \end{cases} \quad \forall j, k$$

打击窗口约束：武器的打击也必须在规定的窗口内

$$x_{i,j,t}^{k} \in \begin{cases} \{0,1\} & \text{if } x_{i,j,t}^{k} + \hat{T}_{j,t}^{k} \in [L_k, U_k] \\ \{0\} & \text{else} \end{cases} \quad \forall i,j,k$$

在途打击约束要求同时在途的打击数量是有上限的（不知道如何刻画）

选择一：在每个时间阶段，如果限制给每个目标能够分配的杀伤链上限为 q_k，则有约束

$$\sum_{i,j} x_{ijt}^{k} \le q_{k,t}, \quad \forall t,k$$

选择二：在整个决策的过程中，每个目标分配的打击数量有上限

$$\sum_{i,j,t} x_{ijt}^{k} \le q_{k,t}, \quad \forall k$$

目标函数

中间参数：

- $\eta_{t,k}$：在时间 t，目标 k 的带时间权重摧毁概率，具体计算方式底下会写
- $\alpha_{t,k}$：在时间 t，目标 k 的时间权重与目标权重的系数
- d_k：辅助参数，用于计算目标的时间紧迫程度。

截止到时刻 t_a，目标的 k 的存活概率为

$$\prod_{t \in T_k, t \le t_a} \prod_{i=1}^{m} \prod_{j=1}^{n} (1 - p_{ijt}^{k})^{x_{ijt}^{k}}$$

为了符号简便，我们可以记在时刻 t，打击目标 k 的击毁概率为 $\hat{p_{t,k}}$

$$\hat{p_{t,k}} = \prod_{i=1}^{m} \prod_{j=1}^{n} (1 - p_{ijt}^{k})^{x_{ijt}^{k}}$$

因此存活概率就是

$$\prod_{t \in T_k, t \le t_a} \hat{p_{t,k}}$$

假设随着时间的推移，这个目标的紧迫程度会上升，因此对于给定目标 k，我们用下面的系数刻画它的紧迫程度

$$\eta_{t_a,k} = \frac{1}{d_k + (w_k - t)}$$

因此同时考虑目标和时间的加权

$$\alpha_{t,k} = R_k \cdot \eta_{t_a,k} = \frac{R_k}{d_k + (w_k^u - t)}$$

其中 d_k 为一个参数，$w_k - t$ 表示剩余的时间间隔数。当 d_k 为 0 的时候，我们认为和打击目标的急迫程度与剩余时间窗数呈反比，加入平衡参数可以让这个曲线变得更平缓，但同样呈现随着时间窗变小紧迫性彼变大。利用这两个，我们可以计算出在时刻 t_a，增加了紧迫程度加权的存活概率为

$$\eta_{t,k} \cdot \prod_{t=w_k^l}^{t_a} \prod_{i=1}^{m} \prod_{j=1}^{n} (1 - p_{i,j,t}^k)^{x_{i,j,t}^k}$$

从而可以得到以下三个目标函数：

毁伤概率的目标函数

$$f_1 = \sum_k [R_k \sum_{t_a \in T_k} \eta_{t,k} \cdot \prod_{\substack{t \in T_k \\ t \le t_a}} \prod_{i=1}^{m} \prod_{j=1}^{n} (1 - p_{i,j,t}^k)^{x_{i,j,t}^k}]$$

花费的目标函数：

$$f_2 = \sum_{i,j} (c_{ij} \sum_k \sum_{t \in T_k} x_{ijt}^k)$$

时间的目标函数：

$$f_3 = T_{\text{MAX}}$$

由此还会增加约束：

$$(\hat{T}_{j,t}^k + t) x_{i,j,t}^k \le T_{\text{MAX}}, \quad \forall i, j, k, t$$

总的优化函数即考虑对上述三个优化目标的线性加权

$$\min \theta_1 f_1 + \theta_2 f_2 + \theta_3 T_{\text{MAX}}$$

由于系数只跟目标函数有关，与约束无关，因此可以通过比例来让 f_1 前的系数为 1，即目标函数为：

$$\min f_1 + \theta_2 f_2 + \theta_3 T_{\text{MAX}}$$

由于花费和 θ 都是线性项，因此可以把它们二者相乘得到整合后的系数，假设整合后的系数继续用 f_2 代替，则

上述把所有目标和约束进行整合，即可得到总体的数学规划模型

$$
\begin{aligned}
\min \quad & f_1 + f_2 + \theta_3 T_{\mathrm{MAX}} \\
\text{s.t.} \quad & \sum_{j,k} x_{i,j,t}^{k} \le l_i, \quad \forall t, i \\
& \sum_{k} z_{j,t}^{k} \le r_j, \quad \forall t, j \\
& \sum_{j,k,t} x_{ijt}^{k} \le L_i, \quad \forall i \in I \\
& x_{i,j,t}^{k} \le z_{j,t+n}^{k}, \quad \forall i, j, k, n = 0, 1, \cdots, \hat{T}_{i,t}^{k} \\
& \sum_{i,j,t} x_{ijt}^{k} \le q_k, \quad \forall k \\
& (\hat{T}_{j,t}^{k} + t) x_{i,j,t}^{k} \le T_{\mathrm{MAX}}, \quad \forall i, j, k, t \\
& z_{j,t}^{k} \in \begin{cases} \{0,1\} & \text{if } t \in \left[L_{j,k}^{R}, U_{j,k}^{R} \right] \\ \{0\} & \text{else} \end{cases} \quad \forall j, k \\
& x_{i,j,t}^{k} \in \begin{cases} \{0,1\} & \text{if } x_{i,j,t}^{k} + \hat{T}_{j,t}^{k} \in \left[L_k, U_k \right] \\ \{0\} & \text{else} \end{cases} \quad \forall i, j, k
\end{aligned}
$$

4.2　基于列生成的求解方法

4.2.1　列生成框架

为了求解这个问题，我们依然希望考虑将问题线性化后使用列生成方法，考虑针对每个目标 k，全部的打击场景，假设为 S_k

即针对每个目标 k，一套完整的打击方案（在时刻 t，是否使用武器 i 和雷达 j，给定后，可以计算出这个打击方案的总花费（因为目标函数中的两项都是可以对 k 进行分离的）。

同时，一套给定的打击方案对应在每个时刻使用的雷达与武器也是固定的，从而可以计算出对应的数值，我们使用 n 作为标记，举例而言，使用 $n_{s_{ti}}$ 表示某个场景 s 在时间 t 使用武器 i 的个数。详细来说，与之前不同的符号约定可以写成：

从而可以得到整体的数学模型

$n_{s_{ti}}$ ： 打击场景 s 在时间 t 使用武器 i 的个数。

$n_{s_{tj}}$ ： 打击场景 s 是时间 t 使用雷达 j 的个数。

n_{s_i} ： 打击场景 s 在使用武器 i 的个数，即所有时间点使用的 i 的个数相加。

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{K}\sum_{s\in S_k} q_{ks}y_{ks} + \theta_3 T_{\max} \\
\text{s.t.} \quad & \sum_{k=1}^{K}\sum_{s\in S_k} n_{s_{t,i}}y_{ks} \le l_i && \forall\, i\in I, t\in T \\
& \sum_{k=1}^{K}\sum_{s\in S_k} n_{t,j}y_{ks} \le r_j && \forall\, j\in J, t\in T \\
& \sum_{k=1}^{K}\sum_{s\in S_k} n_{s_i}y_{ks} \le L_i && \forall\, i\in I \\
& \sum_{s\in S_k} y_{ks} = 1 && \forall\, k\in K \\
& \sum_{s\in S_k} t_{ks}y_{ks} \le T_{\max} && \forall\, k\in K \\
& ny_{ks} \ge 0 && \forall\, k,\ s\in S_k
\end{aligned}
$$

其中 q_{js} 的为：

$$
\sum_{t_a\in T_k} \alpha_{t_a} \prod_{t\in T_k, t<t_a} \prod_{i=1}^{m}\prod_{j=1}^{n} (1-p_{ijt})^{x_{ijt}} + \sum_{t_a\in T_k} c_{ij} \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijt}
$$

如果考虑这个问题的列生成子问题，假设上面五类约束的对偶变量记为 u_{ti}, v_{tj}, w_i， z_k， TM_k，则问题的对偶模型是：

$$
\begin{aligned}
\max \quad & l_i \sum_{t\in T_k}\sum_{i=1}^{m} u_{ti} + r_j \sum_{t\in T_k}\sum_{j=1}^{n} v_{tj} + L_i \sum_{i=1}^{m} w_i + \sum_{k\in K} z_k \\
\text{s.t.} \quad & \sum_{t\in T_k}\sum_{i=1}^{m} n_{s_{ti}}u_{ti} + \sum_{t\in T_k}\sum_{j=1}^{n} n_{s_{tj}}\mu_{tj} + \sum_{i=1}^{m} w_i + z_k + t_{ks}TM_k \le q_{js}, \quad \forall j\in J, s\in S_j^w \quad (P) \\
& \sum_{k\in K} -TM_k \le \theta_3 \\
& u_{ti},\ v_{tj},\ w_i,\ TM_k \le 0 \\
& z_k \text{ free}
\end{aligned}
$$

就需要从对偶问题的多个不同的行中选择出一列 Reduced cost < 0 的一列，由于 TM_k 是定值，因此这个约束一定满足，因此对偶问题可能导致 RC < 0 的约束可以被写为：

$$\sum_{t\in T_k}\sum_{i=1}^{m} n_{s_{ti}}u_{ti} + \sum_{t\in T_k}\sum_{j=1}^{n} n_{s_{tj}}\mu_{tj} + \sum_{i=1}^{m} w_i + z_k + t_{js}TM_k \le q_{js}$$

因此列生成子问题的判断依据就是对于一个给定的目标 k，计算是否存在

$$q_{js} - \left(\sum_{t\in T_k}\sum_{i=1}^{m} n_{s_{ti}}u_{ti} + \sum_{t\in T_k}\sum_{j=1}^{n} n_{s_{tj}}\mu_{tj} + \sum_{i=1}^{m} w_i + z_k + t_{js}TM_k\right) < 0$$

从而子问题就是

$$\min \quad \sum_{t_a\in T_k}\alpha_{t_a}\prod_{t\in T_k, t<t_a}\prod_{i=1}^{m}\prod_{j=1}^{n}(1-p_{ijt})^{x_{ijt}} + \sum_{t_a\in T_k}\sum_{i=1}^{m}\sum_{j=1}^{n}(c_{ij}-u_{it}-v_{tj}-w_i)x_{ijt} - z_k - t_{js}TM_k$$

$$\text{s.t.} \quad \sum_j x_{ijt} \le l_i, \quad \forall t, i$$

$$z_{jt} \le r_j, \quad \forall t, j$$

$$\sum_{j,t} x_{ijt} \le L_i, \quad \forall i \in I$$

$$x_{i,j,t} \le z_{j,t+n}, \quad \forall i, j; n = 0, 1, \cdots, \hat{T}^k_{i,t}$$

$$\sum_{i,j,t} x_{ijt} \le q, \quad \forall k$$

$$z_{j,t} \in \begin{cases} \{0,1\} & \text{if } t \in \left[L^R_{j,k}, U^R_{j,k}\right] \\ \{0\} & \text{else} \end{cases} \quad \forall j$$

$$x_{i,j,t} \in \begin{cases} \{0,1\} & \text{if } x^k_{i,j,t} + \hat{T}^k_{j,t} \in \left[L_k, U_k\right] \\ \{0\} & \text{else} \end{cases} \quad \forall i, j$$

$$x_{ijt} \in \mathbb{N}$$

其中 α_t 是当前的权重，表示当前目标随着 t 的变化的权重，随着时间越大，距离最终允许的打击时间越近，这个数就会越大

4.2.2　子问题求解技巧

应用之前的记号，并假设线性项系数被整合为

$$\hat{c_{ijt}} = c_{ij} - u_{it} - v_{th} - w_i$$

同时记问题的截距为：

$$c_0 \triangleq -z_k - TM_k$$

同时因为

$$t_{ks} = \max x_{i,j,t} \cdot t_{i,j,t}$$

因此可以增加一个变量 $T_{\max}$，从而约束多了

$$x_{i,j,t} * t_{i,j,t} \le T_{\max}$$

因此新的子问题形式为：

$$\begin{aligned}
\min \quad & \sum_{t_a \in T_k} \alpha_{t_a} \prod_{t \in T_k, t \le t_a} \hat{p}_t + \hat{c_{ijt}} x_{ijt} + c_0 + c_1 t_{\max} \\
\text{s.t.} \quad & \sum_j x_{ijt} \le l_i, \quad \forall t, i \\
& z_j t \le r_j, \quad \forall t, j \\
& \sum_{j,t} x_{ijt} \le L_i, \quad \forall i \in I \\
& x_{i,j,t} \le z_{j,t+n}, \quad \forall i, j, t; n = 0, 1, \cdots, \hat{T}^k_{i,t} \\
& (t + \hat{T}^k_{i,j,t}) \cdot x_{i,j,t} \le t_{\max} \\
& \sum_{i,j,t} x_{ijt} \le q, \quad \forall k \\
& z_{j,t} \in \begin{cases} \{0,1\} & \text{if } t \in \left[L^R_{j,k}, U^R_{j,k}\right] \\ \{0\} & \text{else} \end{cases} \quad \forall j \\
& x_{i,j,t} \in \begin{cases} \{0,1\} & \text{if } x^k_{i,j,t} + \hat{T}^k_{j,t} \in \left[L_k, U_k\right] \\ \{0\} & \text{else} \end{cases} \quad \forall i, j \\
& t_{\max} \in \mathbb{R}
\end{aligned}$$

考虑上面目标函数中的非线性项

$$F(x) = \sum_{t_a \in T_k} \alpha_{t_a} \prod_{t \in T_k, t \le t_a} \prod_{i=1}^{m} \prod_{j=1}^{n} (1 - p_{ijt})^{x_{ijt}}$$

针对每项引入一个 η_{t_a}，则需要增加如下约束：

$$\prod_{t \in T_k, t \le t_a} \prod_{i=1}^{m} \prod_{j=1}^{n} (1 - p_{ijt})^{x_{ijt}} \le \eta_{t_a}, \quad t_a \in T_k$$

与之前类似，记这个不等式左边的项为 $f_{t_a}(x)$，对于任何一个整数点 x^*，由这个凸约束带来的外逼近割是

$$f_{t_a}(x^*)+\nabla f_{t_a}(x^*)(x-x^*)\le \eta_{t_a}$$

即：

$$f_{t_a}(x^*)+\sum_{t\in T_k,t\le t_a}\sum_{i=1}^{m}\sum_{j=1}^{n}\{\ln(1-p_{ijt})\cdot(x_{ijt}-x_{ijt^*})\}\le \eta_{t_a}$$

如果采用这个形式的凸函数，则将上述步骤直接带入外逼近的框架进行求解即可。

但是此时如果使用上面提到的第二种建模方式进行建模，由于多个非线性函数都和时间相关，因此问题的数学形式会有一定程度的改善。

我们考虑额外引入 w，可以得到每一个非线性项的形式是

$$e^{w_{t_a}}\le \eta_{t_a},\quad t_a\in T_k$$

$$\prod_{t\in T_k,t\le t_a}\prod_{i=1}^{m}\prod_{j=1}^{n}(1-p_{ijt})^{x_{ijt}}\le e^{w_{t_a}},\quad t_a\in T_k$$

从而我们可以得到

$$\sum_{t\in T_k,t\le t_a}\sum_{i=1}^{m}\sum_{j=1}^{n}(\ln(1-p_{ijt})x_{ijt})\le w_{t_a},\quad t_a\in T_k$$

对于这些约束，我们可以将其替换为：

$$\sum_{t_a\le t<t_{a+1}}\sum_{i=1}^{m}\sum_{j=1}^{n}(\ln(1-p_{ijt})x_{ijt})\le w_{t_a+1}-w_{t_a},\quad t_a\in T_k$$

可以看出，当采用新的约束形式时，原来的约束一定成立，因此新的约束比原来的约束更紧。但是由于在最有解处，有

$$\sum_{t\in T_k,t\le t_a}\sum_{i=1}^{m}\sum_{j=1}^{n}(\ln(1-p_{ijt})x_{ijt})=w_{t_a},\quad t_a\in T_k$$

此时在新的约束下也会成立，因此更紧的边界不会割掉最优解，因此对于这种建模场景下的武器目标分配问题，使用新的外逼近形式能够降低问题的搜索空间。

4.3　数值实验

第 5 章　总结与展望

本文主要工作内容围绕武器目标分配问题展开，首先针对经典的武器目标分配问题进行了线性化，并对线性化后的模型设计了一个基于列生成框架并用外逼近方法求解列生成子问题的算法，再将问题和算法进行拓展到复杂资源约束和动态的武器目标分配问题，并给出了动态武器目标分配问题的一种评价方式。

本文在第二节中，介绍了武器目标分配问题的经典模型及一些已经存在的精确算法，同时利用了经典模型的凸性，给出了两种不同的凸优化建模方式，并应用外逼近求解框架对问题进行求解。除此以外，还给出了一种问题的线性化建模方式，将问题等价地转化为一个线性化的模型，后续章节中的基于列生成的求解算法都是在这个线性化模型的基础上展开的。最后结合实际的作战需求，给出了问题在多作战要素下的推广，将武器目标分配问题拓展为武器-雷达-目标分配问题，并给出了问题的数学模型构建与线性化方法。

在第三章中，详细地介绍了武器-雷达-目标分配问题的数学模型与线性化方式，并给出了一个例子来详细地展示线性化模型的含义。在线性化模型的基础上，针对问题具有指数多列的特点，设计了基于列生成框架的求解算法。在这个具体的问题中，列生成的子问题并非一个简单的线性规划，而是可以被转化为一个凸优化问题，它的形式与武器目标分配问题相似，但是却实现了目标之间的分离，在缩减了问题规模的同时可以对问题进行并行化。在针对子问题的算法设计上，我们提出了两种不同的建模方式，并使用外逼近的求解框架来对问题进行求解。为了能够增加问题的求解速度，我们针对子问题设计了基于数学特性和实际含义的有效不等式，并且对经典的列生成方法进行了改进，允许子问题选择非最优的子问题并且可以同时选择多个不同的改进列。最后应用数值实验证明了这个框架和求解技巧的效果。

本文在第四章中，介绍了如何将静态武器目标分配拓展到动态武器目标分配问题，给出了问题的场景描述并提出了对应的数学模型。相比于经典的动态武器目标分配问题，这个问题采用了非均匀的时间离散，希望在不过多牺牲问题内容信息的前提下，建立一个能够精确求解的模型。基于建立的模型，我们将之前

提到的列生成求解框架进行推广，此时子问题依然是一个凸优化问题。针对子问题，我们证明了采用对数方式进行建模的模型会更紧，并基于此完成了算法的设计。

在武器目标分配问题上，本文还有很多可能改进的地方。首先是这个问题与分支定界框架更有效的结合，在使用传统的分支定界方法来进行求解过程中，在分支过程中会增加对于某一个变量的限制，但是在这个问题中它会导致子问题的求解失去凸性，从而影响问题的求解。其次在子问题的求解上，还有一些可以选择的加速技巧没有使用，同时同一目标的多个子问题之间非线性项相同，不同之处仅仅在于线性项系数，如何有效利用这一点来实现热启动以实现子问题的求解加速也是一个有趣的研究课题。最后在列生成过程中使用的非最优子问题选择方式也依然有改进空间，如何自适应地选择需要的列依然有改进的空间。

附录 A　中国科学院大学学位论文撰写要求

学位论文是研究生科研工作成果的集中体现，是评判学位申请者学术水平、授予其学位的主要依据，是科研领域重要的文献资料。根据《科学技术报告、学位论文和学术论文的编写格式》（GB/T 7713-1987）、《学位论文编写规则》（GB/T 7713.1-2006）和《文后参考文献著录规则》（GB7714—87）等国家有关标准，结合中国科学院大学（以下简称“国科大”）的实际情况，特制订本规定。

参考文献

[1] Land A H, Doig A G. An automatic method for solving discrete programming problems [J]. Econometrica, 1960, 28(3): 497.

[2] Gomory R E. Outline of an algorithm for integer solutions to linear programs [J]. Bulletin of the American Mathematical Society, 1958, 64(5): 275-278.

[3] Crowder H, Johnson E L, Padberg M. Solving large-scale zero-one linear programming problems [J]. Operations Research, 1983, 31(5): 803-834.

致 谢

感激 casthesis 作者吴凌云学长，gbt7714-bibtex-style 开发者 zepinglee，和 ctex 众多开发者们。若没有他们的辛勤付出和非凡工作，LaTeX 菜鸟的我是无法完成此国科大学位论文 LaTeX 模板 ucasthesis 的。在 LaTeX 中的一点一滴的成长源于开源社区的众多优秀资料和教程，在此对所有 LaTeX 社区的贡献者表示感谢！

ucasthesis 国科大学位论文 LaTeX 模板的最终成型离不开以霍明虹老师和丁云云老师为代表的国科大学位办公室老师们制定的官方指导文件和众多 ucasthesis 用户的热心测试和耐心反馈，在此对他们的认真付出表示感谢。特别对国科大的赵永明同学的众多有效反馈意见和建议表示感谢，对国科大本科部的陆晴老师和本科部学位办的丁云云老师的细致审核和建议表示感谢。谢谢大家的共同努力和支持，让 ucasthesis 为国科大学子使用 LaTeX 撰写学位论文提供便利和高效这一目标成为可能。

作者简历：

casthesis 作者

李冠达，男，北京人，1998 年出生。2016.9 - 2021.6 就读于清华大学数学科学系，获得学士学位 2021.9 - 2024.6 就读于中国科学院数学与系统科学研究院，攻读硕士学位。

已发表（或正式接受）的学术论文：

1. 暂无

参加的研究项目及获奖情况：

可以随意添加新的条目或是结构。