第 7 章 指派问题

在现代社会的各类组织与生产活动中，如何将有限的资源高效、合理地分配到多样化的任务，是管理决策中最为普遍且核心的问题之一。无论是企业中的员工调度、制造系统中的设备分配，还是交通运输、医疗服务、教育资源配置等领域，资源与任务的最佳匹配直接影响着整体系统的运行效率与经济效益。指派问题，作为运筹学中最具代表性的优化模型之一，正是应对这一挑战的有力工具。本章将系统介绍指派问题的基本概念、数学模型及其多样化的扩展形式，深入剖析经典的匈牙利算法、分支定界法等方法，并通过丰富的案例展示其在实际中的广泛应用。同时，随着大数据与人工智能技术的发展，指派问题的研究与应用也不断突破传统边界，呈现出更强的复杂性与动态性。通过本章的学习，读者不仅能够掌握指派问题的理论基础与求解技巧，更能体会其在推动社会资源优化、提升组织管理效能中的重要作用，为后续深入学习运筹学的其他优化模型奠定坚实基础。

7.1 指派问题介绍

7.1.1 背景介绍

指派问题是一类特殊的线性规划问题，旨在将一组资源（如工人、机器）分配给一组任务（如工作岗位、订单），以实现总成本最小化或总收益最大化。该问题由数学家 Harold W.Kuhn 于 1955 年在匈牙利数学家 König 的零元素覆盖定理基础上提出，并开发了高效的“匈牙利算法”。指派问题在人员调度、任务分配、生产线优化和物流配送等领域具有广泛应用。

在现实世界中，资源分配问题无处不在。无论是在生产制造、项目管理、交通运输，还是在服务行业和公共政策制定中，指派问题都扮演着重要的角色。通过合理的分配，可以优化资源利用，减少浪费，提升整体效率。因此，掌握指派问题的解决方法，对管理者和决策者来说至关重要。

7.1.2 问题描述和分类

指派问题是指如何在满足独占性约束的前提下，优化资源与任务的分配方案。

独占性约束要求每个资源只能分配给一个任务，同时每个任务只能由一个资源完成。优化目标通常是最小化总成本或最大化总收益。例如，在一个工厂中，需要将 n 名工人分配到 n 个工作岗位，每个工人完成不同岗位的成本各不相同。如何分配才能使总成本最低？在实际问题中，资源、任务和成本之间的关系可能较为复杂，直接通过直觉或手工计算难以找到最优解。因此，需要将问题抽象为数学模型，以便系统化地分析和求解。数学模型通过量化资源与任务的分配关系、成本和约束条件，为优化问题提供了清晰的分析框架。

指派问题可以根据不同的约束和目标进行分类。标准指派问题是最基本的类型，每个资源只能被分配给一个任务，每个任务也只能被一个资源完成，目标是最小化总成本或最大化总收益。在平衡指派问题中，资源和任务的数量相等，且每个资源必须被分配给一个任务。而在不平衡指派问题中，资源和任务的数量不相等，此时需要引入虚拟资源或虚拟任务，将问题转化为平衡指派问题进行求解。二次指派问题（Quadratic Assignment Problem）则涉及到两个成本矩阵，分别表示资源和任务之间的基本分配成本以及资源和资源之间的交互成本。目标是同时最小化这两部分的成本。

7.2 指派问题的数学模型

7.2.1 标准指派问题模型

指派问题的数学模型主要用于表示如何将一组资源（如工人、机器等）最优地分配给一组任务（如工作岗位、订单等）。在该模型中，通常以成本矩阵的形式给出资源与任务之间的分配成本，并通过数学优化方法找到使总成本最小的分配方案。

假设有n个资源和n个任务。对于每个资源i和每个任务j，有一个分配成本c_{ij}，表示将资源i分配给任务j所需要的成本。需要找到一种分配方案，使得总成本最小。

为了表示资源和任务之间的分配情况，引入 0-1 变量x_{ij}，其定义如下：

$$x_{ij}=\begin{cases}1\ ,\text{如果资源}i\text{被分配给任务}j\\0\ ,\text{否则}\end{cases}$$

$$\forall i\in\{1,2,\dots,n\}，\forall j\in\{1,2,\dots,n\}$$

这个变量x_{ij}表示资源i是否被分配给任务j。由于每个资源只能分配给一个任务，每个任务也只能被一个资源分配，因此x_{ij}只有两种可能的取值：0 或 1。

指派问题的目标是最小化所有资源分配到任务上的总成本。基于上述的决策变量x_{ij}，总成本可以表示为：

$$\min\ Z=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij}$$

$$\forall i\in\{1,2,\dots,n\}，\forall j\in\{1,2,\dots,n\}$$

目标函数 Z 即为所有分配方案的总成本之和。为了确保每个资源和每个任务的分配都是唯一且完整的，需要引入以下约束条件：

对于每一个资源i，它只能被分配给一个任务j。数学上，这一约束可以表示为：

$$\textstyle\sum_{j=1}^{n}x_{ij}=1,\forall i\in\{1,2,\dots,n\}$$

该约束确保了每个资源i被分配到且仅分配到一个任务。

对于每一个任务j，只能有一个资源i被分配给它。数学上，这一约束可以表示为：

$$\textstyle\sum_{i=1}^{n}x_{ij}=1,\forall j\in\{1,2,\dots,n\}$$

最后，由于决策变量x_{ij}为 0-1 变量，因此需要以下约束：

$$x_{ij}\in\{0,1\},\forall i\in\{1,2,\dots,n\},\forall j\in\{1,2,\dots,n\}$$

综上所述，指派问题的数学模型可以用以下形式来表示：

$$\min\ Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{7.1}$$

s.t.

$$\textstyle\sum_{j=1}^{n} x_{ij}=1\ \ ,\forall i \in \{1,2,\dots,n\},$$

$$\textstyle\sum_{i=1}^{n} x_{ij}=1\ \ ,\forall j \in \{1,2,\dots,n\},$$

$$x_{ij} \in \{1,0\}\,, \forall i\{1,2,\dots,n\}, \forall j \in \{1,2,\dots,n\} \tag{7.2}$$

这一模型概括了指派问题的本质，即在给定的资源和任务集合中，通过唯一分配，实现总成本最小化或总收益最大化。上述模型是标准的指派问题模型，但在实际应用中，指派问题可以根据不同的场景进行扩展。来看一个具体的例子。

例题 7.1 假设某公司有 3 名员工（A、B、C），需要完成 3 个项目（项目 1、项目 2、项目 3）。不同员工完成不同项目的成本不同，公司希望找到一个最优的分配方案，使得总成本最低。已知各员工完成各项目的成本如表 7.1 所示：

表 7.1 例题 7.1 中的员工完成各项目的成本

	项目 1	项目 2	项目 3
A	13	24	18
B	22	19	23
C	17	16	20

解 定义用 0-1 变量x_{ij}来表示员工i是否被分配给项目j，其中$i \in \{\mathrm{A},\mathrm{B},\mathrm{C}\}$，$j \in \{1,2,3\}$。

$$x_{ij} = \begin{cases} 1,\text{如果资源}i\text{被分配给任务}j \\ 0,\text{否则} \end{cases}$$

$$\forall i \in \{1,2,3\}, \forall j \in \{1,2,3\}.$$

目标是最小化总成本。根据给定的数据，成本矩阵c_{ij}表示为：

$$c_{ij}=\begin{pmatrix}13 & 24 & 18\\ 22 & 19 & 23\\ 17 & 16 & 20\end{pmatrix}$$

因此，总成本可以表示为

$$\min\ Z=13x_{11}+24x_{12}+18x_{13}+22x_{21}+19x_{22}+23x_{23}+17x_{31}+16x_{32}+20x_{33}$$

需要确保每名员工只能被分配给一个项目，同时每个项目也只能分配给一名员工。加入了这些约束条件后该模型可表示为：

$$\min\ Z=13x_{11}+24x_{12}+18x_{13}+22x_{21}+19x_{22}+23x_{23}+17x_{31}+16x_{32}+20x_{33}$$

s.t.

$$\sum_{j=1}^{3}x_{ij}=1,\forall i\in\{1,2,3\},$$

$$\sum_{i=1}^{3}x_{ij}=1,\forall j\in\{1,2,3\},$$

$$\mathrm{x_{ij}}\in\{1,0\},\forall i\in\{1,2,3\},\forall j\in\{1,2,3\}.$$

7.2.2 指派问题的对偶理论

指派问题作为一类特殊的线性规划问题，其本质是通过优化资源与任务的匹配关系实现目标函数的最小化。从标准指派问题的原始模型出发，构建其对偶问题，并分析二者的关联性。

根据线性规划对偶理论，原始问题的每个等式约束对应一个自由变量（无符号限制的对偶变量），而每个原始变量$x_{ij}\geq 0$ 对应一个不等式约束。引入对偶变量u_i（对应资源约束）和v_j（对应任务约束），得到对偶模型：

$$\mathrm{Max}\ W=\sum_{i=1}^{n}u_i+\sum_{j=1}^{n}v_j \tag{7.3}$$

$$\mathrm{s.t.}\,u_i+v_j\leq c_{ij},$$

$$\forall i \in \{1,2,\dots,n\}, \forall j \in \{1,2,\dots,n\}. \tag{7.4}$$

其中u_i,v_j为自由变量，通常被称为影子价格或对偶价格。

在对偶模型中，对偶变量u_i和v_j分别代表资源的机会成本。u_i是指第i个人员（行）的“时间价值”。例如，在任务分配中，若某技术工人未执行某项任务，其剩余时间可以用于其他高优先级工作，u_i即衡量该人员未参与该任务的潜在损失。v_j是指完成第j个任务（列）的“最低隐性成本”。该成本可理解为任务本身的价值底线，即人员能力与任务需求不匹配时付出的额外隐形成本。

约束条件u_i+v_j≤c_{ij}可解释为总隐性成本不超过实际成本，若对某个任务-人员组合(i,j)，实际成本c_{ij}较高，而隐性成本u_i+v_j较低，说明该分配在经济上不划算，此时应避免选择该组合。例如，若雇用一名高薪员工完成一个简单任务（c_{ij}高但u_i+v_j低），相当于浪费了高级人才的时间资源。

最优解的经济平衡是指在最优指派方案中，对选中的任务-人员组合(i,j)，其实际成本c_{ij}必须等于隐性成本u_i+v_j，即约束紧致性。这代表资源的价格完全与其使用价值匹配，不存在经济浪费。

未选中组合的冗余成本是指，未选中的组合(i,j)，必须满足u_i+v_j<c_{ij}，表明若强行分配该任务，实际成本将高于隐性成本，造成效率损失。

连接原始问题最优解与对偶问题最优解的桥梁是至关重要的互补松弛条件（Complementary Slackness）。这一条件揭示了原始变量与对偶变量之间的密切联系，是线性规划最优性的重要判据。在指派问题中，原始决策变量x_{ij}与对偶约束c_{ij}−u_i−v_j⩾0 满足互补松弛条件。

$$\left(c_{ij} - u_i - v_j\right)\mathrm{x}_{\mathrm{ij}} = 0 \tag{7.5}$$

$$\forall i \in \{1,2,\dots,n\}, \forall j \in \{1,2,\dots,n\}.$$

这意味着若x_{ij}=1（即选择该分配），则$u_i + v_j = c_{ij}$，即对应的对偶约束必须紧致。若$u_i + v_j < c_{ij}$，则必有x_{ij}=0，即不选择该分配。

只有当资源的隐性成本与实际成本完全匹配时，才允许分配；任何未充分利用资源或任务价值的机会成本必须为零。通过对偶变量的平衡，避免高估人员能力（虚高u_i）或低估任务价值（虚低v_j），确保价格体系自洽。

综上所述，对偶理论通过提供对偶模型和互补松弛条件，为深入理解指派问题最优解的经济内涵和数学结构提供了一个强有力的框架。它不仅证明了最优指派方案的效率特性，也启示了基于价格调整来逐步逼近最优解的算法思路，这正是后续匈牙利算法的核心思想所在。

例题 7.2 某公司有 3 名技术工人（A、B、C）和 3 个任务（T1、T2、T3）需要完成。通过合理分配工人与任务，使得总成本最小化。假设每个工人只能分配到一个任务，每个任务只能由一个工人完成。构建该指派问题的原始线性规划模型。根据对偶理论，构建其对偶模型，并解释对偶变量的经济意义。

每个工人完成每个任务的成本（以小时为单位）如表 7.2 所示：

表 7.2 例题 7.2 中的每个工人完成每个任务的成本

工人/任务	T1	T2	T3
A	4	6	8
B	5	3	7
C	6	5	4

解 决策变量$x_{ij} = \begin{cases} 1,如果工人i被分配到任务j \\ 0,否则 \end{cases}$

$$\forall i \in \{A, B, C\}, j \in \{1,2,3\}.$$

$$\min\ Z = 4x_{A1} + 6x_{A2} + 8x_{A3} + 5x_{B1} + 3x_{B2} + 7x_{B3} + 6x_{C1} + 5x_{C2} + 4x_{C3}$$

s.t.

$$x_{A1} + x_{A2} + x_{A3} = 1,$$

$$x_{B1}+x_{B2}+x_{B3}=1,$$

$$x_{C1}+x_{C2}+x_{C3}=1,$$

$$x_{A1}+x_{B1}+x_{C1}=1,$$

$$x_{A2}+x_{B2}+x_{C2}=1,$$

$$x_{A3}+x_{B3}+x_{C3}=1,$$

$$x_{ij}\in\{0,1\},\forall i\in\{A,B,C\},j\in\{1,2,3\}.$$

对偶模型构建：

$$\max W=u_A+u_B+u_C+v_1+v_2+v_3$$

s.t.

$$\begin{aligned} u_A+v_1\le 4,\quad u_A+v_2\le 6,\quad u_A+v_3\le 8,\\ u_B+v_1\le 5,\quad u_B+v_2\le 3,\quad u_B+v_3\le 7,\\ u_C+v_1\le 6,\quad u_C+v_2\le 5,\quad u_C+v_3\le 4. \end{aligned}$$

其中u_A是工人 A 的时间价值（机会成本），表示若 A 未被分配某任务，其时间可用于其他工作的潜在收益或损失。

v_1是任务 T1 的最低隐性成本，反映完成该任务所需的最低资源价值。

$u_A+v_1\le 4$代表的意思是工人 A 完成任务 T1 的隐性成本（时间价值+任务价值）不超过实际成本 4 小时。若超过则分配不经济。

7.2.3 二次指派问题

在许多实际问题中，资源之间的相互作用同样不可忽视。例如，在设施布局问题中，一个设施的位置不仅影响其自身的工作效率，还会因为与其他设施的相对位置而影响整体系统的效率。这类问题被称为二次指派问题（Quadratic Assignment Problem, QAP）。

二次指派问题在许多实际应用中具有广泛的应用，如生产设施布局设计、计算机芯片布局设计、通信网络设计等。QAP 是运筹学中最具挑战性的组合优化问题之一，属于 NP 难问题。它不仅涉及线性成本，还需要考虑资源之间的相互

作用成本，因此求解难度较大，但也更加贴近现实问题。

（1）问题描述

二次指派问题旨在将n个设施分配到n个不同的地点，使总成本最小化。总成本由两部分组成：线性成本和二次成本。线性成本表示设施与地点之间的直接分配成本。而二次成本则是由两个设施之间的交互作用决定。例如，在工厂布局中，设备间的距离可能影响物料运输效率；在任务分配中，工作人员的协作关系可能影响整体进度。问题的核心在于找到一个指派方案，综合优化这两类成本，同时满足每个设施分配到一个且仅一个地点的约束。由于二次成本引入了非线性项，QAP 的求解通常需要采用高级优化方法或启发式算法。

（2）数学模型

令c_{ij}为资源i指派给工作j时的线性成本。a_{ijkl}为资源i被指派给工作j且资源k被指派给工作l时的二次相互作用成本。

决策变量x_{ij},表示资源i是否被指派给工作j。如果i被指派给j，则x_{ij}=1；否则x_{ij}=0。

二次指派问题的目标是最小化总成本，其中总成本包含线性成本和二次相互作用成本。约束条件仍是每个资源必须指派到一个且仅一个工作，每个工作必须由一个且仅一个资源完成。以及决策变量的约束。

$$\min\ Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij+}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} a_{ijkl}x_{ij}x_{kl} \tag{7.6}$$

s.t.

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i \in \{1,2,\dots,n\} \tag{7.7}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j \in \{1,2,\dots,n\} \tag{7.8}$$

$$x_{ij} \in \{0,1\}, \forall i \in \{1,2,\dots,n\}, \forall j \in \{1,2,\dots,n\}.$$

二次指派问题（QAP）在实际中有着广泛的应用，涉及多个领域的复杂分配和布局问题。在工厂或车间中，为设备安排维修工人时，除了要考虑每个工人的技能和每台设备的维修需求外，还需考虑工人与工人之间的协作效率；在项目管理中，为不同的任务安排员工时，除了员工与任务的匹配度，还需要考虑员工之间的合作效应。

在军事行动中，如武器目标分配（炮兵火力任务分配）或空战目标分配问题，需要综合考虑武器与目标的匹配情况以及不同武器系统之间的协同作战能力；在仓库管理中，合理分配货物位置以最小化取货和运输的时间；将飞机分配到不同的停机位时，不仅要考虑飞机与停机位的匹配，还要考虑其他飞机的出入和滑行路径。

例题 7.3 某公司指派 3 个员工（编号 1, 2, 3）到 3 个城市（编号 1, 2, 3）工作（每个城市单独一人），希望最小化总电话费用。员工两两之间的通话时间（单位：小时）在矩阵的上三角部分给出，城市两两之间的通话费率（单位：元/小时）在矩阵的下三角部分给出。目标是求最小总电话费用，其费用矩阵如下：

$$\begin{bmatrix} 0 & 10 & 5 \\ 8 & 0 & 3 \\ 5 & 2 & 0 \end{bmatrix}$$

上三角（不包括对角线）：员工通话时间；

位置(1,2)=10：员工 1 与员工 2 的通话时间；

位置(1,3)=5：员工 1 与员工 3 的通话时间；

位置(2,3)=3：员工 2 与员工 3 的通话时间。

下三角（不包括对角线）：城市通话费率；

位置(2,1)=8：城市 1 与城市 2 的通话费率；

位置(3,1)=5：城市 1 与城市 3 的通话费率；

位置(3,2)=2：城市 2 与城市 3 的通话费率。

解 本问题属于二次指派问题，目标函数包含二元变量的乘积项。总电话费

用由员工对（i,k）的通话时间t_{ik}和其所在城市对(j,l)的费率d_{jl}共同决定。

$$\min Z = \sum_{i=1}^{3} \sum_{k=i+1}^{3} t_{ik} \left(\sum_{j=1}^{3} \sum_{l=1}^{3} d_{jl} x_{ij} x_{kl} \right)$$

s.t.

$$x_{ij} = \begin{cases} 1 & \text{员工}i\text{分配到城市}j \\ 0 & \text{否则} \end{cases}$$

$$\sum_{j=1}^{3} x_{ij} = 1, \forall i \in \{1,2,3\},$$

$$\sum_{i=1}^{3} x_{ij} = 1, \forall j \in \{1,2,3\}.$$

7.2.4 多目标指派问题

在标准的指派问题中，通常只有一个目标，例如最小化总成本或最大化总收益。然而，现实中的许多资源分配问题通常涉及多个目标，这些目标可能是相互冲突的。因此，多目标指派问题（Multi-objective Assignment Problem）是指在指派问题中同时考虑多个优化目标，要求在满足这些目标的条件下找到一个最优的资源分配方案。

（1）多目标指派问题的背景

多目标指派问题在许多实际场景中都有广泛应用。在制造业中，决策者可能需要在降低生产成本的同时提高生产效率。在物流管理中，需平衡运输成本、交货时间和客户满意度之间的关系。在项目管理中，通常希望任务在最短时间内完成，同时保证高质量。这些目标往往相互制约，例如降低成本可能导致交货延迟，因此需要在多目标之间找到合理的平衡。

（2）多目标指派问题的问题描述

多目标指派问题可以被描述为在一个资源分配场景中，存在多个需要优化的目标函数。给定一组资源（如工人、机器）和一组任务，每种资源分配到某个任务会产生与多个目标相关的不同结果（例如成本、时间、质量等）。问题的核心

在于找到一个指派方案，使得所有目标在某种平衡下达到最优或接近最优的状态。由于目标之间可能存在冲突（例如最小化成本可能导致时间增加），决策者需要通过权衡来确定一个满足实际需求的解。解决该问题通常需要定义目标的优先级或权重，或者通过生成一组非支配解（Pareto 最优解）来供决策者选择。

（3）多目标指派问题的数学模型

在多目标指派问题中，需要同时考虑多个目标函数，每个目标函数可能有不同的权重或重要性。为了使模型更加通用和可解，可以采用一种加权和法（Weighted Sum Method）来将多个目标函数转化为一个单一的优化目标。

假设有n个资源和n个任务。每个资源和任务之间的分配具有多个目标，如成本、时间、效益等。可以为每个目标函数赋予一个权重w_k，表示该目标的相对重要性。

假设有m个目标函数，每个目标函数对应资源i和任务j的分配决策x_{ij}有一个特定的贡献。对于每个目标函数k，有以下数学模型：

$$\min Z=\sum_{k=1}^{m} w_k \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{(k)} x_{ij} \tag{7.9}$$

s. t.

$$\sum_{i=1}^{n} x_{ij}=1, \forall j \in \{1,2,\dots,n\} \tag{7.10}$$

$$\sum_{j=1}^{n} x_{ij}=1, \forall i \in \{1,2,\dots,n\} \tag{7.11}$$

$$x_{ij} \in \{0,1\}, \forall i \in \{1,2,\dots,n\}, \forall j \in \{1,2,\dots,n\}.$$

其中，$c_{ij}^{(k)}$是资源i分配给任务j的第k个目标的成本。将这些多个目标通过加权和的方式合并成一个总目标函数Z。约束条件仍然与标准指派问题相同，即每

个资源只能分配给一个任务，每个任务只能分配给一个资源，且决策变量x_{ij}为 0-1 变量。

多目标指派问题是一类在多个目标之间权衡优化的复杂问题，广泛应用于实际的资源分配、任务调度和生产管理等场景中。通过加权和法或 Pareto 最优解等方法，可以在多个目标之间找到合适的平衡。由于目标函数之间存在相互冲突，求解多目标指派问题通常比单目标问题更为复杂，往往需要借助进化算法、模拟退火或其他启发式算法来有效解决。

例题 7.4 在医院管理中，手术室的高效调度是一个关键问题。假设某医院有 5 名外科医生（D1,D2,D3,D4,D5）和 5 台待安排的手术（S1,S2,S3,S4,S5），每位医生完成不同手术的成本（以时间衡量，单位：分钟）不同，同时需考虑患者的等待时间（反映满意度）。目标是最小化总时间成本和最小化患者总等待时间。

时间成本关系如表 7.3 所示：

表 7.3 例题 7.4 中的不同手术的时间成本

c_{ij}	S1	S2	S3	S4	S5
D1	60	90	50	70	80
D2	70	80	60	65	75
D3	55	85	45	60	70
D4	65	75	55	50	85
D5	80	70	65	60	55

手术开始前的等待时间如表 7.4 所示：

表 7.4 例题 7.4 中的手术开始前的等待时间

	S1	S2	S3	S4	S5
D1	20	30	15	25	35
D2	25	25	20	20	30
D3	15	35	10	20	25
D4	20	30	15	15	40

	S1	S2	S3	S4	S5
D5	30	20	25	20	15

解 这是一个多目标指派问题。定义两个目标函数：

目标 1 是最小化总时间成本

$$Z_1 = \sum_{i=1}^{5} \sum_{j=1}^{5} c_{ij} x_{ij}$$

目标 2 是最小化患者总等待时间

$$Z_2 = \sum_{i=1}^{5} \sum_{j=1}^{5} w_{ij} x_{ij}$$

采用加权和法将两个目标合并：

$$\min Z = \alpha Z_1 + (1-\alpha) Z_2 = \sum_{i=1}^{5} \sum_{j=1}^{5} (\alpha c_{ij} + (1-\alpha) w_{ij}) x_{ij}$$

其中，c_{ij}为医生D_i执行手术S_j的时间成本。w_{ij}为医生D_i执行手术S_j时患者的等待时间。α∈[0,1]为权重（例如，α=0.6 更重视时间成本）。约束条件同理标准指派问题。

完整数学模型如下：

$$\min Z = \sum_{i=1}^{5} \sum_{j=1}^{5} [\alpha c_{ij} + (1-\alpha) w_{ij}]\, x_{ij}$$

$$\text{s.t.} \quad \sum_{i=1}^{5} x_{ij} = 1, \forall j = 1, \dots, 5,$$

$$\sum_{j=1}^{5} x_{ij} = 1, \forall i = 1, \dots, 5,$$

$$x_{ij} \in \{0,1\}, \forall i = 1, \dots, 5, \forall j = 1, \dots, 5,$$

其中$\alpha \in [0,1]$.

7.3 指派问题的求解算法

7.3.1 匈牙利算法

匈牙利算法适用于求解高度约束的组合优化问题，虽然可以使用传统方法计算，但是这些方法在计算效率上并不理想。相比枚举法或通用整数规划求解器，匈牙利算法更高效。接下来介绍匈牙利算法求解步骤。

首先，将指派问题的成本矩阵设为C=[c_{ij}]，

其中c_{ij}表示指派员工i完成任务j的成本。对矩阵的每一行，减去该行的最小值，使得每一行中至少有一个元素为 0。此步骤的目的是将矩阵的每一行都包含至少一个 0 元素，以便后续步骤进行匹配。

在行减法之后，对矩阵的每一列，减去该列的最小值，使得每一列中至少有一个元素为 0。经过行减法和列减法后，矩阵中将包含多个零元素，但这些零元素不一定能够直接构成一个独立的匹配。

从矩阵中的零元素开始寻找独立的零元素匹配，即不同行不同列的零元素匹配。具体步骤如下：

(1)从只有一个零元素的行开始，给该零元素加圈（记作⓪），表示该行已经被匹配。

(2)将⓪所在列中的其他零元素划去（记作Φ），表示这些零元素不可用。

(3)重复上面的过程，直到无法继续为止。

如果找到的独立零元素数量等于矩阵的阶数（即行列的数量），则最优解已找到；如果独立零元素的数量小于矩阵的阶数，则需要进一步寻找最优解。

如果找到的独立零元素数量等于矩阵的阶数（即行列的数量），则最优解已找到；如果独立零元素的数量小于矩阵的阶数，则需要进一步寻找最优解。

当找到的独立零元素数量不足以覆盖矩阵的所有行和列时，需要通过添加最小覆盖线来调整矩阵，使得零元素的数量增加。这一步骤的关键在于如何有效地

找到覆盖所有零元素的最少直线，并在矩阵中引入新的零元素以继续寻找最优解。其具体步骤如下：

(1)首先，检查哪些行中没有标记为独立零元素的“⓪”。这些行中所有的元素都是可以考虑进一步调整的。对所有没有“⓪”的行打上“√”标记。

(2)检查刚才打了“√”标记的行中，哪些列包含被标记为“Φ”的零元素。对这些列打上“√”标记。

(3)检查刚才打了“√”标记的列中，哪些行包含独立零元素“⓪”。对这些行打上“√”标记。

(4)重复上述的标记过程，即不断在打了“√”标记的行和列之间交替标记，直到无法继续标记新的行或列为止。此时，已标记了所有可以通过独立零元素（⓪）关联的行和列。

(5)根据上面的标记情况，将进行划线。对没有打“√”标记的行划水平线，对打了“√”标记的列划垂直线。这些划线将覆盖矩阵中的所有零元素。由于每一条线可以覆盖一整行或一整列，因此，这些线数目是最少的覆盖所有零元素的直线数l。

(6)如果划线数l与矩阵的阶数n相同，说明已经找到了最大可能数量的独立零元素，可以找到最优解；如果划线数$l<n$，说明当前独立零元素数量不足以覆盖所有行和列，需要调整矩阵以增加新的零元素。

(7)找到未被覆盖的元素中的最小值δ。对未打‘√’的行（未覆盖的行）所有元素减去δ，对打‘√’的列（覆盖的列）所有元素加上δ。此操作保持覆盖线上的零元素不变，并可能引入新的零元素。

(8)经过上述调整后，矩阵中可能出现新的零元素。可以回到之前的步骤（寻找独立零元素），直到找到与矩阵阶数相同的独立零元素，得到最优解。

代码见电子资源

在实际应用中，某些场景允许一个参与者同时完成多个任务。为此，可以通过复制该参与者的行来扩展成本矩阵，表示其可多次分配。对于新添加的行，保持与原行相同的成本值，以确保分配结果等价。此方法适用于需要放宽一对一分配约束的情况，例如在任务量较大时允许高效参与者承担额外任务。在另一些场景中，某些任务不能由特定参与者完成。为避免此类分配，可将该参与者与该任务的指派成本设为一个极大的值 M。通过设置高成本，匈牙利算法将自动排除这些不可行分配，确保任务分配给合适的参与者。

例题 7.5 有四位学生（甲、乙、丙、丁）需要分别完成四道不同的数学题目（题目 1、题目 2、题目 3、题目 4）。每位学生解决每道题目的预计耗时（以分钟为单位）不同，耗时关系如表 7.5 所示。使用匈牙利算法，求解一个最优分配方案，使得四位学生各自完成一道题目，且总耗时最小。

表 7.5 例题 7.5 中的学生的耗时关系

	题目 1	题目 2	题目 3	题目 4
甲	12	15	11	14
乙	13	10	12	17
丙	14	13	16	18
丁	11	14	12	16

解 首先进行行减操作和列减操作，得到图 7.1:

$$\begin{pmatrix} 1 & 4 & 0 & 0 \\ 3 & 0 & 2 & 4 \\ 1 & 0 & 3 & 2 \\ 0 & 3 & 1 & 2 \end{pmatrix}$$

图 7.1 例题 7.5 中的矩阵图 1

寻找独立零元素并进行标记

$$\begin{pmatrix} 1 & 4 & 0 & 0 \\ 3 & 0 & 2 & 4 \\ 1 & 0 & 3 & 2 \\ 0 & 3 & 1 & 2 \end{pmatrix}$$

图 7.2 例题 7.5 中的矩阵图 2

标记行和列，划线覆盖所有零元素，得到图 7.3：

$$\begin{pmatrix} 1 & 4 & 0 & 0 \\ 3 & 0 & 2 & 4 \\ 1 & 0 & 3 & 2 \\ 0 & 3 & 1 & 2 \end{pmatrix}$$

图 7.3 例题 7.5 中的矩阵图 3

判断划线数量小于 4 故需要调整矩阵。如图 7.4 所示：

$$\begin{pmatrix} 1 & 5 & 0 & 0 \\ 2 & 0 & 1 & 3 \\ 0 & 0 & 2 & 1 \\ 0 & 4 & 1 & 2 \end{pmatrix}$$

图 7.4 例题 7.5 中的矩阵图 4

重复以上操作直至划线数量等于 4，算法结束。

匈牙利算法的优缺点：

匈牙利算法的时间复杂度为$O(n^3)$，对于中小规模的指派问题（任务和资源数量在几十到几百之间），能够在合理时间内找到全局最优解。因此，在成本最小化或收益最大化的指派问题中，使用匈牙利算法可以确保找到最佳的解决方案。

匈牙利算法不仅适用于标准的指派问题，还可以通过一些扩展处理不平衡指

派问题或部分可行的指派问题。通过增加虚拟任务或虚拟资源，可以将不平衡问题转化为标准的方阵，从而使用匈牙利算法进行求解。

虽然$O(n^3)$的时间复杂度对于中小规模问题是可以接受的，但在任务和资源数量非常大的情况下（如数千甚至数万个任务或资源），算法的运行时间可能会变得不可接受。此时，匈牙利算法的计算效率可能不足以处理实际需求，需要考虑启发式算法或近似算法来提高求解速度。

7.3.2 二次指派问题求解

二次指派问题（QAP）是一类典型的复杂组合优化问题，因资源分配和相互作用成本的复杂关系而难以求解。随着问题规模的扩大，传统的精确算法在计算效率和可行性方面常常面临挑战。为在合理时间内获得高质量解，**元启发式算法**成为解决此类问题的重要工具。

启发式算法通常基于问题特性或经验规则，能够在较短时间内找到可行解，但往往依赖于具体问题，适用范围有限。而元启发式算法则是在启发式算法的基础上发展起来的，具有更高的通用性和适应性。元启发式算法不依赖于具体问题的领域知识，而是通过模拟自然界或社会系统中的一些通用机制（如进化、群体协作等）来进行搜索。遗传算法（Genetic Algorithm）就是一种典型的元启发式算法。它通过模拟自然选择和遗传机制，在解空间中进行全局搜索，能够有效避免陷入局部最优，并适用于多种复杂的组合优化问题。

在解决二次指派问题时，主要介绍遗传算法的应用。Andrew D.Tate 和 Alice E.Smith 在他们的研究中设计了一种遗传算法来求解二次指派问题，并进行了详细的实验以验证其性能。遗传算法通过编码、选择、交叉和变异等操作，不断进化种群中的解，从而逐步逼近最优解。

为了评估该遗传算法的效果，Tate 和 Smith 将实验结果与现有的一些知名启发式方法以及已知的最佳解进行了比较。他们的比较主要包括以下几个方面：

他们选择了一些在二次指派问题求解中表现优异的启发式算法（如禁忌搜索、模拟退火等）作为参考基准。通过实验数据的对比，发现他们的遗传算法在许多情况下能够达到与这些方法相当甚至更优的解。

为了进一步验证遗传算法的有效性，他们将遗传算法的解与文献中已知的二次指派问题最优解进行了对比。结果显示，遗传算法在多个测试实例中能够逼近或达到这些最优解。

Tate 和 Smith 还测试了遗传算法在不同参数配置（如种群大小、变异率、交叉率等）下的表现。结果表明，遗传算法在这些不同配置下依然能够保持良好的性能，这说明该算法对参数设置的依赖性较低，具有较强的鲁棒性。

代码见电子资源

例题 7.6 某物流公司需要在 9 个仓库位置（编号为 1 到 9）分配 9 辆不同的货车（编号为 1 到 9），以完成当天的货物配送任务。每辆货车分配到每个仓库位置会产生不同的配送成本（例如时间、油耗等）。目标是通过优化货车与仓库位置的分配方案，使总配送成本最小化。

表 7.6 例题 7.6 中的配送成本

货车/仓库	1	2	3	4	5	6	7	8	9
货车 1	10	15	20	25	30	35	40	45	50
货车 2	12	14	18	22	28	32	38	42	48
货车 3	15	13	17	21	27	33	37	43	47
货车 4	20	18	16	24	26	34	36	44	46
货车 5	25	22	24	20	25	30	35	40	45
货车 6	30	28	26	22	24	28	34	38	42
货车 7	35	32	30	28	22	26	32	36	40
货车 8	40	38	36	34	30	24	28	34	38
货车 9	45	42	40	38	36	30	26	32	36

解 为了表示货车与仓库的分配关系，采用顺序编码法。每个个体是一个长度为 9 的排列，排列中的数字代表货车编号，位置则对应仓库编号。例如，个体[7,3,1,2,9,5,6,8,4]表明仓库 1 分配给货车 7，仓库 2 分配给货车 3，仓库 3 分配给货车 1，依此类推。这种编码方式能够确保每辆货车只服务一个仓库，同时每个仓库只由一辆货车服务，满足分配问题的约束条件。

算法从随机生成的初始种群开始。种群规模通常设为问题规模的 1.5 至 2 倍，本例建议使用 15-20 个个体。为简化说明，这里展示两个示例个体：父代 1 为[7,3,1,2,9,5,6,8,4]，父代 2 为[4,3,6,7,9,8,1,2,5]。实际应用中，种群规模过小可能导致早熟收敛，过大则会增加计算开销。

适值函数定义为分配方案的总配送成本。以父代 1 为例：仓库 1→货车 7（成本 35），仓库 2→货车 3（成本 13），仓库 3→货车 1（成本 20），仓库 4→货车 2（成本 22），仓库 5→货车 9（成本 36），仓库 6→货车 5（成本 30），仓库 7→货车 6（成本 34），仓库 8→货车 8（成本 34），仓库 9→货车 4（成本 46），总成本为 270。成本越低的个体适应度越高，在进化过程中保留概率越大。本问题的优化目标就是最小化该适应度值。

采用部分匹配交叉（PMX）生成新个体。首先随机选择两个交叉点（例如位置 3 和 6），将两个父代个体的中间片段交换。父代 1 的片段为[1,2,9,5]，父代 2 的片段为[6,7,9,8]，交换后得到初始子代[7,3,6,7,9,8,6,8,4]。此时存在货车重复（位置 4 和 7 的货车 7、6、8 重复），需通过映射关系解决冲突：根据交换片段建立 1↔6、2↔7、5↔8 的映射，将位置 4 的货车 7 映射为 2，位置 7 的货车 6 映射为 1，位置 8 的货车 8 映射为 5，最终得到有效子代[7,3,6,2,9,8,1,5,4]。该子代的总成本为 35（仓库 1）+13（仓库 2）+26（仓库 3）+22（仓库 4）+36（仓库 5）+24（仓库 6）+40（仓库 7）+40（仓库 8）+46（仓库 9）=282。

采用逆序变异增加种群多样性。随机选择基因序列的一段（例如位置 3 至 7），将该子序列反转。以子代[7,3,6,2,9,8,1,5,4]为例，子序列[6,2,9,8,1]反转为[1,8,9,2,6]，得到变异个体[7,3,1,8,9,2,6,5,4]。该个体的总成本为 35（仓库 1）+13（仓库 2）+20（仓库 3）+34（仓库 4）+36（仓库 5）+32（仓库 6）+34（仓库 7）+40（仓库 8）+46（仓库 9）=290。

根据适应度（总成本）选择优秀个体进入下一代。示例中父代 1 成本 270，子代成本 282，变异个体成本 290，优先保留成本最低的父代 1。通过重复执行交叉、变异和选择操作，种群在迭代中逐步进化，最终收敛到总成本最低的分配方案。实际应用中常采用锦标赛选择策略，即随机选取若干个体，保留其中适应度最高的进入下一代。

接下来介绍遗传算法求解二次指派问题优缺点。遗传算法是一种全局优化算法，特别适合解决像二次指派问题这样的复杂问题。它不依赖于问题的特定结构，因此可以处理多种约束条件和复杂的目标函数。遗传算法是一种基于群体的搜索算法，不需要问题的梯度信息或连续性，这使得它适合求解非线性、不可微的二次指派问题。遗传算法的收敛通常较慢，尤其是在问题规模较大或者解空间复杂的情况下，找到高质量解所需的迭代次数可能非常多。遗传算法通常没有明确的停止准则，通常是通过预设迭代次数或观察收敛速度来决定何时停止，这可能导致资源的浪费或未能充分搜索解空间。

7.3.3 算法比较与选择指南

表 7.7 求解指派问题的算法比较

算法名称	适用条件	核心优势	关键局限
匈牙利算法	线性网络模型	高精度、理论最优解保证	只能处理小规模模型，对大规模问题求解困难
遗传算法	非线性优化问题	全局搜索、鲁棒性强、适用范围广	收敛速度较慢、对参数敏感
模拟退火算法	单目标优化问题	避免陷入局部最优解，易实现	参数选择困难，收敛速度较慢
深度强化学习	动态环境，适合实时动态调整	适应性强、控制能力灵活	算法复杂度高，对数据量要求大

在运筹学中，优化算法的选择本质上是一种问题特性与求解效率的权衡。经典算法（如匈牙利算法）依托严格的数学基础，能够提供理论最优解，但适用性通常受限于线性、静态和小规模问题。启发式算法（如模拟退火）则突破了传统框架，以近似最优解为代价，能够处理复杂的非线性、多峰及动态约束问题，并且更易与工程实践融合。数据驱动型算法（如深度强化学习）通过动态学习与自适应策略，能够实时响应高维问题，但对数据质量和算力资源的要求较高。未来，混合算法的创新（如遗传算法嫁接局部搜索、强化学习嵌入数学规划）将推动运筹学从“单一算法最优”迈向“多策略协同求解”，尤其在应对突发事件、绿色计算等新型挑战中更具潜力。理解算法的核心思想而非机械应用其步骤，方能在面对

工业调度、物流优化等实际问题时，实现效率与精度的精准平衡。

7.4 本章小结

在本章中，深入探讨了指派问题这一运筹学中的经典问题。通过标准指派问题和二次指派问题等问题的数学建模，掌握了如何将资源（如工人、机器）最优地分配给任务（如工作岗位、订单），以实现总成本的最小化或总收益的最大化。此外，通过介绍匈牙利算法和遗传算法等求解方法，学会了应对不同复杂度的指派问题，特别是在实际应用中的高效求解策略。

指派问题不仅在学术研究中具有重要地位，更在实际操作中广泛应用于诸如人员调度、生产线优化、物流配送等领域。无论是制造业的生产管理，还是交通运输的路径优化，本章内容都为读者提供了有力的工具和方法。

在结束本章的学习后，大家可以思考以下几个问题，以加深对指派问题的理解：

在现实世界中，你能想到哪些具体场景可以应用指派问题的求解方法？这些场景下的成本矩阵如何构建？

设想如果指派问题的约束条件发生变化（如某些资源无法执行某些任务），应如何调整求解方法？

二次指派问题的复杂性给求解带来了挑战，除了遗传算法，你还能想到其他可能有效的启发式算法吗？

通过这些思考，相信大家能够更好地掌握指派问题的核心理念，并在实际中灵活运用这些知识。

研学互通

随着机器学习和深度学习技术的迅速发展，越来越多的研究者开始将这些技术应用到传统的运筹学问题中，尤其是在复杂和大规模的指派问题领域。深度学习不仅提供了一种自动化、数据驱动的方式来构建优化模型，还能在复杂环境中实时做出决策。

传统的指派问题求解方法，如匈牙利算法、分支定界法，通常适用于规模较小、结构清晰的问题。然而，当面对动态、多维度、实时变化的指派问题时，传统算法往往无法满足计算效率和求解质量的需求。在这种背景下，深度学习和强化学习技术为指派问题的求解提供了新的视角和方法。

在许多实际场景中，指派问题并不是静态的。例如，在共享交通系统中，车辆与乘客的匹配需要根据实时变化的交通状况、乘客需求、车辆位置等因素进行动态调整。这类问题难以通过传统的确定性算法求解，而深度强化学习（Deep Reinforcement Learning,DRL）提供了一个有效的框架。如滴滴、Uber 等共享交通平台，通过深度强化学习，系统可以在实时交通状况下自动调度车辆，以最小化乘客的等待时间并优化车辆利用率。

为进一步探索组合优化、运筹学及相关算法的理论与应用，以下推荐的经典文献和书籍涵盖了指派问题的核心方法、历史发展和实际案例。这些资源适合希望深入理解优化算法的读者，既提供严谨的数学框架，又通过应用场景激发灵感，助力在学术研究与实际问题解决间架起桥梁：

(1) Kuhn, H. W. (1955). The Hungarian method for the assignment problem. Naval Research Logistics Quarterly, 2(1 – 2), 83 – 97.

这篇开创性论文首次系统提出了匈牙利算法，为线性指派问题提供了高效的多项式时间解法。Kuhn 在文中不仅详细描述了算法的步骤和数学原理，还通过实例演示了算法的实际操作过程。匈牙利算法的提出极大推动了运筹学和组合优化领域的发展，并在生产调度、交通运输、人员安排、物流配送等实际问题中得到广泛应用。对于希望了解经典算法起源、优化方法设计思路以及算法与实际结合的读者，这篇论文是不可多得的学习资源。

(2) Pardalos, P. M., & Rendl, F. (1993). Quadratic assignment problems. European Journal of Operational Research, 65(3), 410 – 425.

本文系统综述了二次指派问题，这是组合优化领域最具挑战性的问题之一，广泛应用于工厂设施布局、电子电路设计、数据关联等场景。作者梳理了 QAP 的数学建模方法，详细分析了其 NP 难度、复杂性来源及典型求解算法，并结合

实际案例展示了 QAP 在现实中的重要意义。文中还评述了启发式算法和元启发式算法在大规模 QAP 中的应用效果。适合具备一定优化基础、希望深入探索高难度优化问题理论和算法的进阶学习者和研究人员。

(3) Burkard, R. E., Dell'Amico, M., & Martello, S. (2009). Assignment problems. SIAM.

这本权威专著被誉为“指派问题领域的百科全书”，系统全面地介绍了线性指派问题（LAP）、二次指派问题（QAP）及其变种问题的理论基础、数学模型、精确算法（如分支定界法、匈牙利算法）和启发式算法（如模拟退火等）。书中不仅深入解析了每类算法的原理、优缺点，还通过丰富的实际应用案例（如运输、制造、网络分配等）阐释了理论与实践的结合。全书结构清晰、内容详实，既适合本科生和研究生系统学习，也适合工程师和科研人员查阅与参考。

(4) Lawler, E. L. (1963). The quadratic assignment problem. Management Science, 9(4), 586–599.

Lawler 的这篇经典论文是 QAP 领域的奠基之作，首次深入剖析了二次指派问题的数学结构、建模框架及其计算复杂性，提出了早期的精确解法和理论界限。作者通过严密的数学推导，揭示了 QAP 与线性指派问题的本质区别，并为后续研究提供了理论基础和研究方向。该论文对理解高阶优化问题的内在难点、算法设计的理论依据具有重要意义，适合对优化理论、运筹学基础感兴趣的读者深入研读。

(5) Koopmans, T. C., & Beckmann, M. (1957). Assignment problems and the location of economic activities. Econometrica, 25(1), 53–76.

这篇论文是运筹学与经济学交叉研究的典范，首次将指派问题模型应用于经济活动中的资源分配和设施选址优化。作者通过构建数学模型，分析了如何在考虑运输成本和地理分布的前提下，实现经济活动的最优布局。文中理论严谨，兼具现实指导意义，对理解运筹学在经济、地理、城市规划等领域的实际应用具有重要启发。适合对跨学科应用、资源配置优化和经济建模感兴趣的读者。

(6) Burkard, R. E., Cela, E., Pardalos, P. M., & Pitsoulis, L. (1998). The quadratic

assignment problem: An overview of exact and heuristic solution methods. Handbook of Combinatorial Optimization, 3, 1713 – 1800.

作为 QAP 领域的权威综述，这篇文章系统梳理了二次指派问题的各类求解方法，包括精确算法（如分支定界、割平面法、拉格朗日松弛等）和启发式算法（如模拟退火、禁忌搜索等），并评述了其理论进展和实际应用效果。文中还总结了 QAP 的最新研究热点和未来发展方向。内容详实、结构清晰，是学术研究者和工程实践者深入了解 QAP 求解技术、掌握前沿动态的重要参考资料。

思行经世：指派问题助力科技强国

指派问题模型作为运筹学中的核心方法，通过科学分配资源实现效率最大化或成本最小化，其原理在于将任务精准匹配给执行者，确保每个任务仅由一个最适合的资源完成，成为解决复杂分配问题的利器。在我国科技强国建设中，指派问题模型不仅提升了各领域的效率，还深刻体现了中国特色社会主义制度下科技创新与管理优化的精神内核。

以三峡工程为例，这一世界级水利工程的成功离不开指派问题模型的支持。面对数万名工人和海量设备的调度，建设团队通过模型将专业工程师分配至关键工序，如发电机组安装，量化每个任务的工时和能力需求，优化资源配置，确保工程按时优质完成。这一过程不仅保障了国家能源安全，更体现了“敢为人先、追求卓越”的三峡精神，彰显了指派问题在重大工程中的关键作用。

在国防科技领域，北斗导航系统的全球覆盖得益于指派问题模型的优化。科研团队通过模型将有限的地面站资源分配至最优的卫星跟踪任务，构建匹配矩阵解决资源冲突，确保信号覆盖的精准性。这一技术支撑了北斗系统的成功运行，体现了“波形图里的中国脉动”，为我国从航天大国迈向航天强国注入了动力。此外，在教育资源配置中，指派问题模型推动了公平发展。教育部门通过模型将优质教师资源分配至偏远地区学校，将擅长数学的教师匹配至需求较高的学校，量化教师能力和学校需求，实现最优匹配，提升了农村教育质量。这一实践体现了“立德树人、科教兴国”的教育精神，为国家人才战略奠定了基础。

指派问题模型的每一次应用，都是资源优化与国家精神的生动交融。三峡工

程的精准调度诠释了“敢为人先”，北斗系统的技术优化体现了“波形图里的中国脉动”，教育资源的均衡配置体现了“立德树人”。这些实践不仅提升了效率，还激励读者在科技强国建设中继承优良传统、勇于创新。指派问题模型作为运筹学的瑰宝，深刻融入我国的重大工程、国防科技和教育事业，为实现中华民族伟大复兴贡献了智慧与力量。

习题

习题 7.1 某工厂需要将 4 项生产任务（T1,T2,T3,T4）分配给 4 台机器（M1,M2,M3,M4），该工厂专注于高效生产和成本控制，面对日益激烈的市场竞争，合理的任务分配不仅能最大限度地降低生产成本，还能提升设备利用率和生产效率。每台机器的性能和技术参数存在差异，导致不同机器执行同一任务的成本不同。每台机器执行不同任务的生产成本如表 7.8 所示,请根据以上信息，合理安排任务与机器的匹配关系，以实现生产成本的最小化。

表 7.8 习题 7.1 中的机器的生产成本

	M1	M2	M3	M4
T1	13	8	7	15
T2	12	9	6	11
T3	14	6	9	12
T4	11	5	10	7

习题 7.2 某公司有 4 个项目需要分配给 3 个可用的团队。然而，并不是所有团队都能够处理所有项目。例如，团队 A 只能处理项目 1 和 3，团队 B 只能处理项目 2 和 4，而团队 C 只能处理项目 1 和 2。如何安排使得项目分配的总成本最低？

表 7.9 习题 7.2 中的团队的项目成本

	项目 1	项目 2	项目 3	项目 4
团队 A	9	-	4	-
团队 B	-	7	-	3

	项目 1	项目 2	项目 3	项目 4
团队 C	6	8	-	-

习题 7.3 某制造公司有 5 个工厂（A,B,C,D,E）和 5 个仓库（1,2,3,4,5）。每个工厂生产的产品需要分配到某个仓库。每个工厂分配到仓库时，会产生一定的运输成本，且两个工厂之间存在交互影响，即一个工厂的位置会影响其他工厂的位置成本。编写代码求解问题。具体的成本关系如表：

表 7. 10 习题 7. 3 中的工厂的运输成本

	仓库 1	仓库 2	仓库 3	仓库 4	仓库 5
工厂 A	12	7	9	15	10
工厂 B	8	10	11	5	7
工厂 C	10	13	6	9	12
工厂 D	14	9	12	11	13
工厂 E	11	8	10	14	9

表 7. 11 习题 7. 3 中的工厂的位置成本

	工厂 A	工厂 B	工厂 C	工厂 D	工厂 E
工厂 A	0	3	4	2	5
工厂 B	3	0	2	6	1
工厂 C	4	2	0	3	7
工厂 D	2	6	3	0	4
工厂 E	5	1	7	4	0

习题 7.4 某跨国公司需要从 5 个不同国家的分部各选派一名员工担任 5 个不同职位的项目负责人。每个员工只能担任一个职位，每个职位只能由一名员工担任。表 7.12 给出了各国员工担任各职位的能力评分（满分为 100 分）：

表 7. 12 习题 7. 4 中的各国员工的能力评分表

	技术总监	市场经理	财务主管	人力资源	运营总监
美国分部	85	92	78	70	88
中国分部	90	75	82	85	79
德国分部	78	80	95	75	83
英国分部	92	78	80	88	90
印度分部	75	85	84	92	76

此外，由于各国文化背景不同，员工之间的协作难度（按 1-10 分制，分值越高表示协作难度越大）如表 7.13 所示：

表 7.13 习题 7.4 中的员工之间的协作难度表

	美国分部	中国分部	德国分部	英国分部	印度分部
美国分部	0	6	4	7	5
中国分部	6	0	5	4	3
德国分部	4	5	0	6	7
英国分部	7	4	6	0	5
印度分部	5	3	7	5	0

1.如何分配各国员工到各个职位，使得总能力评分最大？请给出最优分配方案和最大总评分。

2.若公司规定技术总监和财务主管必须能够良好协作（协作难度不超过 5），最优分配方案和最大总评分会如何变化？

3.若公司决定优化团队整体协作性，希望在保证总能力评分不低于原最优方案 90%的前提下，使得团队成员间的平均协作难度最小，请给出满足此约束的最优分配方案、总能力评分和平均协作难度。

习题 7.5 某大学计算机系有 8 名优秀学生需要分配到 8 个不同公司的暑期实习岗位。每个学生只能分配到一个公司，每个公司只接收一名实习生。系主任根据学生的专业技能和公司岗位要求进行了匹配度评估（满分为 100 分），评分

如表 7.14 所示：

表 7.14 习题 7.5 中的学生的评分表

	谷歌	微软	亚马逊	苹果	腾讯	阿里	百度	字节跳动
张同学	92	88	85	79	90	82	78	86
李同学	85	90	82	88	75	93	80	79
王同学	78	82	95	85	88	80	92	85
赵同学	90	85	80	92	86	78	88	91
陈同学	86	79	88	90	95	84	80	82
刘同学	80	94	85	78	82	90	85	88
杨同学	93	82	79	85	78	88	94	80
周同学	85	87	90	83	84	85	82	96

此外，每个公司提供的实习薪资（单位：千元/月）如表 7.15 所示：

表 7.15 习题 7.5 中的公司提供的实习薪资表

公司	谷歌	微软	亚马逊	苹果	腾讯	阿里	百度	字节跳动
薪资	15	14	13	15	12	13	11	14

同时，考虑到学生的通勤成本，各学生到各公司的单程通勤时间（单位：分钟）如表 7.16 所示：

表 7.16 习题 7.5 中的各学生的通勤时间表

	谷歌	微软	亚马逊	苹果	腾讯	阿里	百度	字节跳动
张同学	45	30	25	40	35	50	20	45
李同学	35	45	30	25	40	20	35	30
王同学	20	35	45	30	25	40	50	35
赵同学	30	20	35	45	30	25	40	20

	谷歌	微软	亚马逊	苹果	腾讯	阿里	百度	字节跳动
陈同学	40	30	20	35	45	30	25	50
刘同学	25	40	30	20	35	45	30	40
杨同学	50	25	40	30	20	35	45	25
周同学	35	50	25	40	30	20	35	30

1.如何分配学生到各公司，使得总匹配度评分最高？请给出最优分配方案和最高总评分。

2.如果系主任希望在保证总匹配度不低于最优方案 95%的前提下，使学生获得的总薪资最高，请给出满足此约束的最优分配方案、总匹配度评分和总薪资。