第 10 章 旅行商问题

旅行商问题（Travelling Salesman Problem, TSP）作为图论与组合优化领域的经典难题，长期以来吸引着数学、运筹学、人工智能等多领域研究者的目光。其核心目标——寻找访问所有节点且路径最短的闭合回路，看似简单，却蕴含着深刻的理论挑战与广泛的现实意义。从 18 世纪欧拉研究的骑士周游问题起源，到如今在物流配送、交通规划、生产调度等领域的深度应用，旅行商问题不仅展现了组合优化的复杂性，更成为测试新算法性能的标准问题。本章将系统梳理旅行商问题的问题起源、数学建模、精确、启发式与元启发式求解算法、实际应用场景及衍生变种，揭示其在理论探索与实践创新中的双重价值。

10.1 旅行商问题介绍

10.1.1 问题的产生和起源

欧拉七桥问题是 18 世纪著名的古典数学问题之一。在哥尼斯堡的一个公园里，有七座桥将普雷格尔河中的两个岛与河岸连接起来。问是否可能从这四块陆地中的任意一块出发，恰好通过每座桥一次，再回到起点。欧拉于 1736 年研究并解决了此问题，他把此问题归结为“一笔画”问题，证明上述走法是不可能的。

欧拉通过对七桥问题的研究，不仅圆满地回答了哥尼斯堡居民提出的问题，而且得到了三条更为广泛的有关一笔画的结论，称之为欧拉定理。对于一个连通图，通常把从某节点出发恰好经过每条边一次的路线叫做欧拉路。人们又通常把从某节点出发恰好经过每条边一次，最终回到出发点的欧拉路叫做欧拉回路。具有欧拉回路的图叫做欧拉图。欧拉七桥问题作为图论中路径遍历问题的开创性探索，其将实际场景抽象为图模型并研究遍历逻辑的思路，为旅行商问题的形成提供了重要的早期启发。

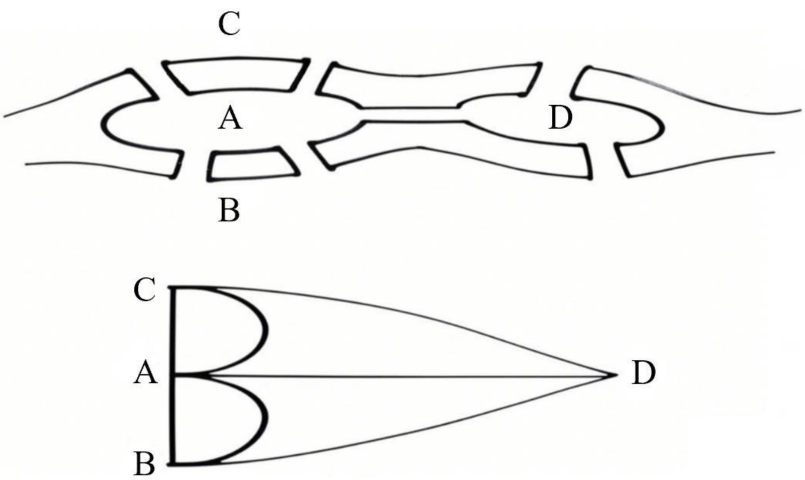

图 10.1 欧拉七桥问题示意图

旅行商问题的历史很久，最早的雏形可追溯至 1759 年欧拉研究的骑士周游问题，即对于国际象棋棋盘中的 64 个方格，走访全部 64 个方格且仅走访一次，并且最终返回到起始点。尽管骑士周游问题因“移动规则受限”与旅行商问题存在差异，但二者的核心逻辑一致，为旅行商问题的早期探索提供了思路。美国 RAND 公司 1948 年正式引入旅行商问题，该公司的声誉以及线性规划这一新方法的出现，使得旅行商问题成为一个知名且流行的问题。

旅行商问题吸引了数学、运筹学、物理、生物、人工智能等领域的学者展开研究。旅行商问题展示了组合优化的多个核心方面，已经成为并将继续成为测试新算法的标准问题，如模拟退火、禁忌搜索、神经网络以及进化算法等都使用旅行商问题来测试。

10.1.2 问题描述

有一个旅行商需要前往n个城市开展业务，要求从某一城市出发，遍历所有n个城市，每个城市仅访问一次，最终返回出发城市，目标是找到总路程最短的闭合路径。

旅行商问题也可称为货担郎问题、推销员问题等。1962 年中国数学家管梅谷先生提出的中国邮递员问题（Chinese Postman Problem, CPP），是著名的图论问题之一。一个邮递员从邮局出发，到所辖街道投递邮件，最后返回邮局。如果他必须走遍每条街道至少一次，应如何选择投递路线，使所走的路程最短。旅行

商问题与中国邮递员问题均为图论中经典的路径优化问题，且均要求从起点出发并最终返回起点，目标是最小化闭合路径的总距离，二者的核心联系在于对“闭合最短路径”的追求及实际场景中的路线规划应用。

二者的本质区别在于遍历对象与规则：中国邮递员问题以图中的边为核心，要求遍历每条边至少一次（允许重复经过节点），核心是通过构造欧拉回路解决“边覆盖”问题（如邮递员需走遍所有街道）；而旅行商问题以图中的节点为核心，要求访问每个节点恰好一次（起点与终点重合除外，不允许重复访问其他节点），目标是寻找哈密顿回路以优化“节点遍历”顺序（如旅行商需访问所有城市）。这种“边遍历”与“节点遍历”的根本差异，导致中国邮递员问题存在多项式时间精确解法，而旅行商问题属于 NP 难问题，大规模场景下求解依赖近似算法。简言之，中国邮递员问题解决“如何不遗漏地走完所有路”，旅行商问题解决“如何不重复地走完所有点”，二者分属不同的图论问题范畴，并非同一问题。

10.2 旅行商问题数学模型

给定无向完全图$G=(V,E,d)$，其中V是顶点集合，$|V|=n$，E则是边集合，$\boldsymbol{d}$是顶点之间的距离矩阵。旅行商问题建立在无向图G上，要求除起点外的每个顶点必须被访问一次，寻找一条环路，使得总路程长度最短。

用数学语言描述旅行商问题不难，但用数学公式描述很难。最早的数学模型是 Dantzig 在 1959 年提出的。其中，d_{ij}是从城市i到城市j的距离（或成本），可以是任意值。决策变量x_{ij}为 0-1 变量，表示解中是否包含边$[i,j]$。

$$x_{ij}=\begin{cases}1, & \text{如果解中包含边}[i,j], \\ 0, & \text{否则},\end{cases} \qquad \forall [i,j]\in E.$$

旅行商问题的目标是最小化总旅行路程，即

$$\min \sum_{i\in V}\sum_{j\in V} d_{ij}\, x_{ij}$$

需要保证每个点仅有一个其他点能够到达该点，即

$$\sum_{i\in V} x_{ij} = 1, \qquad \forall j \in V.$$

同理，每个点仅能到达其他点中的一个，即

$$\sum_{j\in V} x_{ij} = 1, \qquad \forall i \in V.$$

但是仅考虑上述两个条件会产生子回路，所以需要破除子回路，避免解中出现不包含所有顶点的闭合回路，即

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1, \qquad \forall S \subsetneq V\colon |S| \geq 2.$$

破除子回路也可以使用下面的公式，即

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - 1, \qquad \forall S \subset V\colon |S| \geq 2.$$

综上，旅行商问题的完整数学模型如(10.1)或(10.2)：

$$\begin{aligned}
& \min \sum_{i\in V}\sum_{j\in V} d_{ij}\, x_{ij} \\
s.t.& \\
& \sum_{i\in V} x_{ij} = 1, && \forall j \in V, \\
& \sum_{j\in V} x_{ij} = 1, && \forall i \in V, \\
& \sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1, && \forall S \subsetneq V\colon |S| \geq 2, \\
& x_{ij} \in \{0,1\}, && \forall [i,j] \in E.
\end{aligned} \tag{10.1}$$

或，

$$
\begin{aligned}
&\min \sum_{i\in V}\sum_{j\in V} d_{ij}\, x_{ij} \\
s.t.& \\
&\sum_{i\in V} x_{ij} = 1, && \forall j \in V, \\
&\sum_{j\in V} x_{ij} = 1, && \forall i \in V, \\
&\sum_{i\in S}\sum_{j\in S} x_{ij} \le |S| - 1, && \forall S \subset V: |S| \ge 2, \\
&x_{ij} \in \{0,1\}, && \forall [i,j] \in E.
\end{aligned}
\tag{10.2}
$$

公式(10.1)和公式(10.2)是 Dantzig-Fulkerson-Johnson 模型，但该模型使用的破解子回路的方法并不好，约束太多。另外一种方法是 Single Commodity Formulation，是通过引入流量变量来消除子回路的经典方法，约束更少。其中，y_{ij}表示边$[i,j]$上的流量。相应的完整数学模型如(10.3)：

$$
\begin{aligned}
&\min \sum_{i\in V}\sum_{j\in V} d_{ij}\, x_{ij} \\
s.t.& \\
&\sum_{i\in V} x_{ij} = 1, && \forall j \in V, \\
&\sum_{j\in V} x_{ij} = 1, && \forall i \in V, \\
&y_{ij} \le (n-1) x_{ij}, && \forall [i,j] \in E \\
&\sum_{j=2}^{n} y_{1j} = n - 1, && \\
&\sum_{i\in V} y_{ij} - \sum_{k\in V} y_{jk} = 1, && \forall j \in V \setminus \{1\}, \\
&y_{ij} \ge 0, && \forall [i,j] \in E, \\
&x_{ij} \in \{0,1\}, && \forall [i,j] \in E.
\end{aligned}
\tag{10.3}
$$

例题 10.1 假设一位快递员需要在城市中 6 个不同的社区间进行配送，所有可能的路线连接如图 10.2 所示。图中每条连线上的数字代表从一个社区到另一个社区的配送时间。面对繁杂的路线选择，快递员如何才能规划出一条既逐个访问每个社区、又能确保总配送时间最短的最优路径呢？

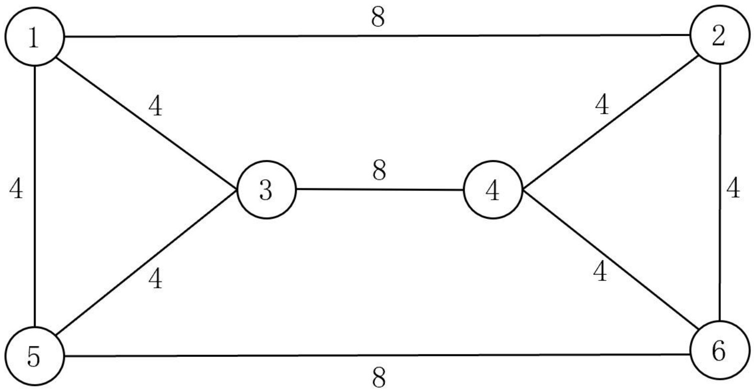

图 10.2 例题 10.1 中的城市路线连接图

（1）这个问题可以被认为是一个对称的旅行商问题吗，即每条路线的配送时间在两个方向上是相同的？并说明原因。

（2）基于该实际情况，建立一个合适的数学模型，用以求解这个最短配送路径的问题。

解 可以。因为快递员需要一条能够访问每个点的环路，因此该问题为旅行商问题。而且该问题中从点i到点j和从点j到点i的配送时间是相同的，所以它是对称的旅行商问题。

$$
\begin{aligned}
&\min 8x_{12}+4x_{13}+4x_{15}+4x_{24}+4x_{26}+8x_{34}+4x_{35}+4x_{46}+8x_{56}\\
&s.t.\\
&x_{12}+x_{13}+x_{15}=2(\text{节点 }1),\\
&x_{12}+x_{24}+x_{26}=2(\text{节点 }2),\\
&x_{13}+x_{34}+x_{35}=2(\text{节点 }3),\\
&x_{24}+x_{34}+x_{46}=2(\text{节点 }4),\\
&x_{15}+x_{35}+x_{56}=2(\text{节点 }5),\\
&x_{26}+x_{46}+x_{56}=2(\text{节点 }6),\\
&\sum_{i\in S}\sum_{j\notin S,j>i}x_{ij}+\sum_{i\notin S}\sum_{j\in S,j>i}x_{ij}\geq 2,\qquad \forall S\subset\{1,\dots,6\},2\leq|S|\leq 4(\text{子回路消除约束})\\
&x_{ij}\in\{0,1\},\qquad \forall i,j\in V,j>i.
\end{aligned}
$$

例题 10.2 在考虑上述城市配送问题的基础上，引入一个新的变化，从点i到点j与从j到i的配送时间不同。例如，快递员若由编号较大的节点向编号较小的节点配送，所需时间会多 50%。比如，若点 1 到点 2 的配送时间为 8，那么从点 2 到点 1 的配送时间为$8\times 1.5=12$。请在此基础上，重新分析和建模。

（1）请给出在这种非对称路径条件下的目标函数。

（2）给出相应的约束条件，确保每个节点的到达和离开都恰好一次。

（3）设计一个约束，以避免在节点集合$S=\{1,3,5\}$中形成任何子回路，从而确保整个路径的正确性和唯一性。

解 （1）根据题目可知，该问题为非对称旅行商问题，其目标函数如下：

$$\min \begin{matrix} 8x_{12}+4x_{13}+4x_{15}+12x_{21}+4x_{24}+4x_{26}+6x_{31}+8x_{34}+4x_{35}+6x_{42} \\ +12x_{43}+4x_{46}+6x_{51}+6x_{53}+8x_{56}+6x_{62}+6x_{64}+12x_{65} \end{matrix}$$

（2）使得路径中每个节点的到达和离开都恰好一次的约束条件如下：

$$x_{21}+x_{31}+x_{51}=1(\text{进入 }1),$$

$$x_{12}+x_{42}+x_{62}=1(\text{进入 }2),$$

$$x_{13}+x_{43}+x_{53}=1(\text{进入 }3),$$

$$x_{24}+x_{34}+x_{64}=1(\text{进入 }4),$$

$$x_{15}+x_{35}+x_{65}=1(\text{进入 }5),$$

$$x_{26}+x_{46}+x_{56}=1(\text{进入 }6),$$

$$x_{12}+x_{13}+x_{15}=1(\text{离开 }1),$$

$$x_{21}+x_{24}+x_{26}=1(\text{离开 }2),$$

$$x_{31}+x_{34}+x_{35}=1(\text{离开 }3),$$

$$x_{42}+x_{43}+x_{46}=1(\text{离开 }4),$$

$$x_{51}+x_{53}+x_{56}=1(\text{离开 }5),$$

$$x_{62}+x_{64}+x_{65}=1(\text{离开 }6).$$

（3）为了避免在集合$S=\{1,3,5\}$中形成任何子回路的约束条件如下：

$$x_{12}+x_{34}+x_{56}\geq 1,$$

$$x_{13}+x_{15}+x_{31}+x_{35}+x_{51}+x_{53}\leq 2.$$

10.3 旅行商问题求解

旅行商问题是一个NP完全问题，随着城市数量的增加，求解最优路线的计算复杂度呈指数级增长。早期的研究者使用精确算法求解该问题，常用的方法包括动态规划法、分支定界法等，但随着问题规模的增大，精确算法无法求解。因此，面对大规模旅行商问题，国内外学者主要使用启发式算法或元启发式算法这类近似算法，包括最近邻点法、最近插入法、局部搜索算法、C-W节约算法、遗传算法、模拟退火算法、蚁群算法、禁忌搜索算法等。

10.3.1 精确算法求解旅行商问题

旅行商问题最简单的求解方法是枚举法，但规模变大以后，枚举几乎是不可能的，所以考虑其他几种精确算法，如动态规划法、分支定界法、回溯法等。接下来以动态规划法为例进行介绍。代码见电子资源。

动态规划法通过将问题分解为更小的子问题来高效地处理问题的复杂性。设已知存在n个城市，阶段的划分基于中间经过城市的数量，用k来表示阶段，$k=0,1,2,\dots,n-1$，城市i到城市j的距离定义为d_{ij}。设S表示从城市1出发到达城市i中间所经过城市的集合。状态(i,S)表示经过S集合中所有城市最后到达城市i。定义最优值函数$f_k(i,S)$为从城市1出发经由S集合中的k个中间城市到达城市i的最短路径长度。动态规划的状态转移方程为：

$$\begin{cases} f_k(i,S)=min_{j\in S}\{f_{k-1}(j,S\backslash\{j\})+d_{ji}\}, k=1,2,\dots,n-1, \\ f_0(i,\emptyset)=d_{1i}, \end{cases} \quad i=2,3,\dots,n.$$

例题 10.3 运用动态规划法解决四城非对称旅行推销员问题。表10.1详细列出了这四个城市之间的距离信息。假设推销员从城市1出发，途经每个城市仅一次，最终返回城市 1，寻找一条最短的路线。请确定这种最优路线的路径安排，从而实现总行程距离的最小化。

表 10.1 例题 10.3 中的各城市间距离表

城市	1	2	3	4
1	0	10	5	6
2	6	0	8	4

3	7	9	0	5
4	9	6	8	0

解 由边界条件可知

$$f_0(2,\emptyset)=d_{12}=10,\qquad f_0(3,\emptyset)=d_{13}=5,\qquad f_0(4,\emptyset)=d_{14}=6.$$

当 k=1 时，从城市 1 开始，中间经过一个城市到达城市i的最短距离为

$$f_1(2,\{3\})=f_0(3,\emptyset)+d_{32}=5+9=14,$$

$$f_1(2,\{4\})=f_0(4,\emptyset)+d_{42}=6+6=12,$$

$$f_1(3,\{2\})=10+8=18,\qquad f_1(3,\{4\})=6+8=14,$$

$$f_1(4,\{2\})=10+4=14,\qquad f_1(4,\{3\})=5+5=10.$$

当 k=2 时，从城市 1 开始，中间经过两个城市到达城市i的最短距离为

$$\begin{aligned}f_2(2,\{3,4\})&=\min[f_1(3,\{4\})+d_{32},f_1(4,\{3\})+d_{42}]\\&=\min[14+9,10+6]=16.\end{aligned}$$

$$f_2(3,\{2,4\})=\min[12+8,14+8]=20.$$

$$f_2(4,\{2,3\})=\min[14+4,18+5]=18.$$

当 k=3 时，从城市 1 开始，中间经过三个城市回到城市 1 的最短距离为

$$\begin{aligned}f_3(1,\{2,3,4\})&=\min[f_2(2,\{3,4\})+d_{21},f_2(3,\{2,4\})+d_{31},f_2(4,\{2,3\})+d_{41}]\\&=\min[16+6,20+7,18+9]=22.\end{aligned}$$

所以，旅行推销员的最短旅行路线是 1→3→4→2→1，最短距离为 22。

10.3.2 启发式算法求解旅行商问题

求解旅行商问题还有很多启发式方法，如最近邻点法、最近插入法。这两种启发式算法都属于贪婪算法，也属于近似求解方法当中的“巡回路径构造算法”。这类算法是把城市逐个地加入到路径中，全部加入以后就得到了一条巡回路径。

（1）最近邻点法

首先从初始城市开始，作为整个回路的起点，然后找到离刚刚加入到回路的上一节点最近的一个（未访问）节点，并将其加入到回路中。重复上一步，直到

所有的节点都加入到回路中。最后，将最后一个加入的节点和起点连接起来，构成了一个旅行商问题的解。代码见电子资源。

例题 10.4 某快递公司在市区内有 4 个重要配送站点，分别为 A、B、C 和 D。公司利用地图软件精确测算了各站点之间的交通距离（公里），并整理成表 10.2。为了优化配送路线，降低运输成本，公司决定采用“最近邻点法”——从起点站 A 出发，不断选择距离当前位置最近的未访问站点，逐步构建一条覆盖所有站点且最终返回起点的最短巡回路线。

表 10.2 例题 10.4 中的各城市快递站点间交通距离表

站点	A	B	C	D
A	0	5	3	6
B	5	0	4	2
C	3	4	0	7
D	6	2	7	0

解 起始节点为 A。从 A 出发，距离 A 最近的节点是 C（距离为 3），将 C 加入回路，此时回路为 A-C。

从 C 出发，在剩下的 B 和 D 中，距离 C 最近的节点是 B（距离为 4），将 B 加入回路，此时回路为 A-C-B。

从 B 出发，剩下的节点只有 D，将 D 加入回路，此时回路为 A-C-B-D。

最后将 D 和起始节点 A 连接起来，得到完整回路 A-C-B-D-A。所以，总路程$= 3 + 4 + 2 + 6 = 15$公里。

（2）最近插入法

首先从任意一个节点出发，选择与其最近的节点形成初始子回路。在剩下的节点中，寻找一个离子回路中某一节点最近的节点，再在子回路中找到一个弧，使弧的两端节点到刚寻找到的最近节点的距离之和减去弧长的值最小，实际上就是把新找到的节点加入子回路以后使得增加的路程最短，就把这个节点增加到子回路中，并去掉子回路中相应的弧，同时将新节点连接到该弧的两端节点，形成

新的子回路。重复以上过程，直到所有的节点都加入到子回路中。代码见电子资源。

例题 10.5 某连锁便利店在城市内设有四个仓库，分别为 P、Q、R 和 S。为确保库存信息的及时性和准确性，仓库的盘点工作需要尽可能快。为了提高效率，最大程度地减少工作人员在仓库间的来回奔波，公司决定采用“最近插入法”来规划最佳的检查路线。此次巡检将从仓库 P 出发，逐步插入距离最近的未访问仓库，最终形成一条高效连贯的巡检路径。各仓库之间的距离经过实地测量后，整理在表 10.3 中，作为规划路线的基础数据。

表 10.3 例题 10.5 中的各仓库间距离表

仓库	P	Q	R	S
P	0	15	8	12
Q	15	0	7	10
R	8	7	0	14
S	12	10	14	0

解 从 P 出发，找到最近的节点 R（距离为 8），形成子回路 P-R-P。

在剩下的 Q 和 S 中，距离子回路中节点最近的是 Q（距离 R 为 7）。计算将 Q 插入不同弧的情况：

若插入 P-R 之间，增加的距离为：$d(P,Q)+d(Q,R)-d(P,R)=15+7-8=14$。

若插入 R-P 之间，增加的距离为：$d(R,Q)+d(Q,P)-d(R,P)=7+15-8=14$。

此时，子回路变为 P-Q-R-P。剩下节点 S，距离子回路中节点最近的是 Q（距离为 10）。计算插入情况：

插入 P-Q 之间，增加的距离为：$d(P,S)+d(S,Q)-d(P,Q)=12+10-15=7$。

插入 Q-R 之间，增加的距离为：$d(Q,S)+d(S,R)-d(Q,R)=10+14-7=$

17。

插入 R-P 之间，增加的距离为：$d(R,S)+d(S,P)-d(R,P)=14+12-8=18$。

因为插入 P-Q 之间增加的距离最小，所以子回路变为 P-S-Q-R-P。总路程$=12+10+7+8=37$。

（3）局部搜索算法

局部搜索算法，即 k-OPT 算法，属于近似求解方法当中的“巡回路径优化算法”，搜索一个解空间邻域里的最优值。基本思想是先产生一条初始巡回路径，再改变其中某些城市的顺序，使路径优化，逐渐接近最优解。代码见电子资源。

k-OPT 算法的思路是，从巡回路径中找出k条边，把它们换成另外k条边（换过以后仍然是一条巡回路径，没被断开），通过搜索邻域内的可能组合寻找优化方案，使路径得到优化。如果无论选出哪k条边来替换都不能进一步优化，则称这条巡回路径是k-最优。显然，k越大，优化后的效果越好，可是随着k的增大，计算量也会随之增大。

例题 10.6 某地质勘探队在一片广大区域内分布着五个重要的勘探点，分别是 M、N、O、P 和 Q。为了获得详尽的地质样本和信息，队员们必须前往每个点进行采集作业。各勘探点之间的行程距离（公里）已通过详细测量，整理在表 10.4 中。考虑到车辆燃油有限，行驶距离过长会增加成本和时间，勘探队最初规划了一条连接所有点的巡回路径：M-N-O-P-Q-M。为了进一步缩短行驶距离，提高效率，现采用 2-OPT 算法对这条路径进行优化调整，旨在找到更短、更高效的路线，从而节省燃料、减少时间。

表 10.4 例题 10.6 中的各勘探点间距离表

勘探点	M	N	O	P	Q
M	0	7	6	9	8
N	7	0	5	8	7
O	6	5	0	6	5

P	9	8	6	0	4
Q	8	7	5	4	0

解 初始路径 M-N-O-P-Q-M，总路程$d_1 = 7 + 5 + 6 + 4 + 8 = 30$公里。根据 2-OPT 算法原理，需通过删除两条不相邻的边并重新连接，生成新的合法巡回路径以尝试优化。

例如，删除边(N,O)和(Q,M)，重新连接(N,Q)和(O,M)。新路径为 M-N-Q-P-O-M。新路程$d_2 = 7 + 7 + 4 + 6 + 6 = 30$公里，该操作未优化路径。

再尝试删除边(O,P)和(Q,M)，重新连接(O,M)和(P,Q)。新路径为 M-N-O-M-Q-P-N。（这不是合法的巡回路径，舍去）

继续尝试删除边(N,O)和(P,Q)，重新连接(N,P)和(O,Q)。新路径为 M-N-P-Q-O-M。新路程$d_3 = 7 + 8 + 4 + 5 + 6 = 30$公里，该操作未优化路径。

继续尝试删除边(M,N)和(O,P)，重新连接(M,O)和(N,P)。新路径为 M-O-N-P-Q-M。新路程$d_4 = 6 + 5 + 8 + 4 + 8 = 31$公里，该操作未优化路径。

最后尝试删除边(M,N)和(P,Q)，重新连接(M,P)和(N,Q)。新路径为 M-P-Q-N-O-M。新路程$d_5 = 9 + 4 + 7 + 5 + 6 = 31$公里，该操作未优化路径。

在 2-OPT 算法中，合法边对需满足“无公共节点”。对于含 5 个独立节点的初始路径，总边数 5 条，有效不相邻边对共 5 种。上述尝试已覆盖全部 5 种组合，均未优化路径，故当前最优路径为 M-N-O-P-Q-M 为，总路程为 30 公里。

（4）C-W 节约法

假定有n个访问地，把每个访问地看成一个点，并取其中的一个点作为基点。首先将每个点与基点相连接，构成线路$1 - j - 1 (j = 2,3, \dots, n)$这样就得到一个具有$n - 1$条线路的图。旅行者按照此路线访问的$n$个点所走的总路程为$z = 2\sum c_{1j}$，其中$c_{1j}$为点 1 到点$j (j = 2,3, \dots, n)$的路段长度，这里假定$c_{1j} = c_{j1}$（对所有点$j$）。若连接点$i$和$j$，即旅行者走弧$(i, j)$，路程节约值$s(i, j)$可计算如下：$s(i, j) = 2c_{1i} + 2c_{1j} - (c_{1i} + c_{1j} + c_{ij})$。对不同的点对$s(i, j)$越大，所节约的路程越多，因此应优

先将这段弧插入到旅行线路中。代码见电子资源。算法的步骤如下：

I.选取基点，将基点与其他各点连接，得到$n-1$条线路$1-j-1(j=2,3,\dots,n)$；

II.对不违背条件的所有可连接点对(i,j)，计算节约值$s(i,j)=c_{1i}+c_{1j}-c_{ij}$；

III.将所有的$s(i,j)$按其值由大到小排列；

IV.按$s(i,j)$值的上述顺序，逐个考察其端点i和j，若满足以下条件，就将弧(i,j)插入到线路中。其条件是：点i和点j不在一条线路上；点i和点j均与基点相邻；

V.返回步骤 IV，直至考察完所有的弧为止。

通过上面的步骤，使问题的解逐步得到改善，最后得到满意解。

例题 10.7 本题采用 C-W 节约算法，对下列旅行商问题进行求解。已知各访问点的具体位置详见表 10.5。通过算法的计算步骤，寻求一条最优路径，以实现最短的整体行程距离。

表 10. 5 例题 10. 7 中的访问点位置表

访问点	A	B	C	D	E	F	G
坐标	(10,23)	(0,13)	(1,0)	(21,2)	(13,4)	(11,6)	(10,10)

解 各点对之间的距离如下（保留两位小数），$c_{ij}=c_{ij}$，

序号	弧	节约值	序号	弧	节约值
1	(D,E)	34.70	9	(E,G)	25.53
2	(E,F)	33.44	10	(C,G)	24.25
3	(C,E)	31.29	11	(D,G)	23.11
4	(C,F)	30.07	12	(B,F)	18.13
5	(D,F)	29.97	13	(B,E)	17.57
6	(C,D)	28.31	14	(B,G)	16.70
7	(F,G)	25.91	15	(B,D)	14.14

8	(B,C)	25.80			

按各段弧节约值由大到小的顺序进行排列，

序号	弧	节约值	序号	弧	节约值
1	(D,E)	34.70	9	(E,G)	25.53
2	(E,F)	33.44	10	(C,G)	24.25
3	(C,E)	31.29	11	(D,G)	23.11
4	(C,F)	30.07	12	(B,F)	18.13
5	(D,F)	29.97	13	(B,E)	17.57
6	(C,D)	28.31	14	(B,G)	16.70
7	(F,G)	25.91	15	(B,D)	14.14
8	(B,C)	25.80			

得到，

序号	弧	线路及说明	插入该弧的节约值
0		A-B-A, A-C-A, A-D-A, A-E-A, A-F-A, A-G-A	
1	(D,E)	A-B-A, A-C-A, A-D-E-A, A-F-A, A-G-A	34.70
2	(E,F)	A-B-A, A-C-A, A-D-E-F-A, A-G-A	33.44
3	(C,E)	E 点与基点 A 不相邻，不插入	0
4	(C,F)	A-B-A, A-D-E-F-C-A, A-G-A	30.07
5，6	(D,F) (C,D)	这些点已在同一条线路上	0
7	(F,G)	F 点与基点 A 不相邻，不插入	0
8	(B,C)	A-D-E-F-C-B-A, A-G-A	25.8
9，10	(E,G) (C,G)	E 点、C 点与基点 A 不相邻，不插入	0
11	(D,G)	A-G-D-E-F-C-B-A	23.11

最后得到的线路为 A-G-D-E-F-C-B-A，线路总长度为 76.52。

10.3.3 遗传算法求解旅行商问题

旅行商问题已成为遗传算法学界的一个主要研究目标。对旅行商问题的研究为遗传算法求解组合优化问题提供了丰富的经验和牢固的基础。主要的努力在三个方面：采用适当的表达方法进行编码；设计可用的遗传算子，以保持父辈特性并避免不可行性；防止过早收敛。代码见电子资源。

通常的二进制编码，不能较好地适用于旅行商问题。人们为旅行商问题提出了几种染色体表达方法，如随机键表达、换位表达。这两种表达方式不仅适用于旅行商问题，也适用于其他组合优化问题。

（1）随机键表达

首先是由 Bean 提出的，这种表达方法将解表达为一串(0,1)间的随机数。这些随机数用作解码的分类键。编码中位置i代表城市i，位置i的随机数表示城市i在旅行商问题巡回中的顺序。这些随机键按升序排列，即可得到访问城市的顺序。

随机键消除了解的表达中的不可行性，即无论如何进行交叉变异，所得到的子代染色体都是可行的。适用于各种顺序优化问题，包括机器调度、资源分配、车辆线路和二次指派问题等。但是这种表达的染色体的基因含义不明确。随机键的基因值通过相对排序间接决定城市顺序，因此基因的绝对数值不直接对应特定城市。同样的基因值，在不同的染色体中代表的顺序可能完全不一样。即便如此，随机键表达，也不失为一种很好的表达方式。

（2）换位表达

也称为路径表达或顺序表达，是旅行商问题巡回的最自然的表达。染色体中基因的值表示城市，基因的顺序表示访问城市的顺序。这种表达的搜索空间是城市顺序“换位”的集合。

这种表达采用传统的单点交叉时，可能导致非法的巡回。为此，人们研究了多种交叉算子，如部分映射交叉、顺序交叉、循环交叉、基于位置的交叉、基于顺序的交叉、子巡回交换交叉、启发式交叉等。

这些交叉算子可以分为两类：

I.规范法：可看作二进制串的两点或多点交叉在换位表达中的扩展。后代中一些城市会丢失或重复，从而得到非法染色体。为此，需要嵌入修复程序来解决后代的非法性，然而这种修复程序是盲目的，它不能保证交叉后的后代比双亲更好。

II.启发式法：启发式法的目的是为了产生改进的后代。

（a）部分映射交叉（Partial-Mapped Crossover, PMX）

由 Goldberg 和 Lingle 提出。可看作二进制串的两点交叉在换位表达中的扩展。它用特别的修复程序来解决简单两点交叉引起的非法性。所以部分映射交叉基本上是简单的两点交叉+修复程序。步骤如下：

I.随机选取两个位置，由两点定义的子串称为映射段；

II.交换双亲的子串，形成原始后代；

III.确定两个映射段之间的映射关系；

IV.根据映射关系，将后代合法化。

（b）顺序交叉（Order Crossover, OX）

由 Davis 提出。可看作是一种带有不同修复程序的部分映射交叉的变型。步骤如下：

I.从第一双亲随机选择一个子串；

II.通过拷贝子串，产生一个原始后代；

III.删去第二双亲中子串已有的城市，得到原始后代需要的其他城市的顺序；

IV.从左到右将这些城市定位到后代的空缺位置上。

（c）基于位置的交叉（Position-Based Crossover, PBX）

由 Syswerda 提出。基本上是一种换位表达的均匀交叉+一个修复程序。也可看作是顺序交叉的一种变型，其中城市的子串是不连续的。步骤如下：

I.从第一双亲上随机选取一组基因；

II.复制这组基因到相应位置，产生一个原始后代；

III.删去第二双亲中子串已有的城市，得到原始后代需要的其他城市的顺序；

IV.从左到右将这些城市定位到后代的空缺位置上。

（d）基于顺序的交叉（Order-Based Crossover, OBX）

也是 Syswerda 提出的。双亲 2 中“被选择基因”的顺序，被双亲 1 中相应基因的顺序所替代。实际上是基于位置的交叉的变型，步骤如下：

I.从双亲 1 随机选取一组基因；

II.从双亲 2 拷贝非选择的基因形成原始后代；

III.从双亲 1 复制选择的基因形成后代。

（e）循环交叉（Cycle Crossover, CX）

由 Oliver, Smith 和 Holland 提出。和基于位置的交叉一样，该方法从一个双亲中取一些城市，而其他城市则取自另一个双亲。不同之处是来自第一个双亲的城市不是随机产生的，而是根据两个双亲相应位置城市构成的循环确定的。步骤如下：

I.根据双亲相应的城市位置，找出一个循环；

II.从双亲 1 复制循环中的基因到原始后代；

III.删去双亲 2 中已在循环上的城市；

IV.用剩余的城市填满后代剩余的位置。

（f）子巡回交换交叉（Subtour Exchange Crossover, SXX）

由 Yarnamura, Ono 和 Kobayashi 提出。从双亲中选择子巡回，子巡回中含有共同的城市。后代由交换子巡回的方法产生。步骤如下：

I.在双亲中随机选择子巡回；

II.交换子巡回。

（g）启发式交叉（Heuristic Crossover, HX）

由 Grefenstette, Gopal, Rosrnaita 和 Gucht 首先提出。启发式交叉，就是把一些启发式算法嵌入到交叉算子当中来，这样更有可能产生优于父代的后代，从而提高算法的效率，明显优于其他方法。例如，采用最近邻点法的启发式交叉步骤如下：

I.从一对双亲中随机地选取一个城市作为开始城市；

II.由当前城市出发，选择一条不构成循环的最短边（由双亲表达的）。若两条边都构成循环，则随机选取一个能使巡回继续的城市；

III.如巡回完成，停止；否则转第二步。

对于换位表达的染色体编码方式，变异运算相对容易。人们已经提出了几种用于换位表达的变异运算，如反转变异、插入变异、移位变异、互换变异以及启发式变异等方法。

（a）反转变异（Inversion Mutation）：在染色体上随机地选择两点，将两点间的子串反转。

（b）插入变异（Insertion Mutation）：随机地选择一个城市，并将它插入到一个随机的位置中。

（c）移位变异（Displacement Mutation）：随机地选择一个子巡回，并将其插入到一个随机的位置中。插入变异可以看作是子巡回只含有一个城市的移位变异。

（d）互换变异（Swap Mutation）：随机地选择两个位置，并将这两个位置上的城市相互交换。

（e）启发式变异（Heuristic Mutation）：由 Cheng 和 Gen 提出。采用邻域技术，以获得后代的改进。启发式变异过程如下：

I.随机地选出λ个基因；

II.按所有选出基因的可能的换位产生邻域；

III.评估所有邻域点，选出最好的作为变异产生的后代。

10.4 旅行商问题应用

旅行商问题作为经典优化问题，核心是寻找访问所有节点的最短闭合路径。其应用覆盖物流配送、交通规划、生产调度等领域，通过按顺序遍历所有节点并最小化总路径成本的逻辑，指导路径优化。跨领域特性推动其理论与实践结合，显著提升效率并降低成本。例如：

（1）旅行路线的确定

例如：现有一个旅行商要乘火车在北京、上海、广州、大连、西安、昆明、重庆、南京、宁波、青岛、成都、桂林、杭州、开封、长春这 15 个城市之间做一次环游，每个城市都到达一次且仅到达一次，并最终回到出发的城市。试求得一条总交通费用最小的环游路线。

（2）配送路线问题

在物流领域，一个物流配送公司的单个快递员（或单辆车），欲将n个客户的订货从仓库出发沿最短路线全部送到后返回，如何确定最短路线。例如：单个快递员从仓库出发为多个客户配送包裹的路线确定问题、单辆押运车完成多个网点的押运调度问题。客户地点相当于城市，这类问题属于单旅行商问题。

（3）最短连线问题

平面上有n个点，用最短的线将全部的点连起来。类似的问题如：机械加工当中，要在一个零件（或者一个电路板）上加工很多孔，如何安排刀具的走刀路径，能使走刀距离最短。孔相当于城市、孔之间的距离相当于城市间的距离。

（4）机器人焊接路径规划问题

例如汽车制造的车身焊接，焊点多达 4-5 千个，一条合理的焊接路径可以减少机器人工位的生产时间，缩短整体工时，提高生产效率。这里焊点相当于城市、焊点间的距离相当于城市间的距离。再如一个印刷线路板需要安装大量的元器件，不同的安装点可以看作是不同的城市，问题是：如何规划机器人的路径，以缩短运动时间，提高工作效率。

（5）多品种装配顺序确定问题

装配企业中，对许多品种的产品进行装配时，一个产品装配完以后要卸下夹具并换上装配另一个产品的夹具，更换两个不同的夹具所需的时间是不一样的。问题是如何安排多品种的装配顺序，能使更换夹具的时间最短。这里的品种相当于城市、更换时间相当于距离。

（6）车间配送路径优化问题

工厂当中要将一批零件分配到很多个车间，如何选择一条路径使得每个车间走一遍后回到起点，且所走的路径最短，这也属于旅行商问题。

（7）投币电话的硬币收集问题

在一个城市当中，如何收集投币电话的硬币，能使行走的路程最短。投币电话就相当于城市，由于有些道路是单向的，所以这是一个非对称旅行商问题。

（8）管道铺设问题

铺设一些半径大小不等的圆形管道，要求所有管道与底边相切，管道之间可以相切，尽可能使铺好后的总宽度最窄。此类问题通常也称为圆排列问题。把管道看作城市、管道圆心之间的水平距离看作是城市间的距离，这个问题可以归结为旅行商问题，增加一个虚拟城市 0，城市 0 到所有管道的距离等于对应管道的半径。

（9）中药自动配药系统中的取药路径规划问题

不同的中药放置在不同位置，这些位置相当于城市。针对一张药方涉及的所有中药，需规划取药车的路径。按顺序访问所有中药的放置位置，将中药放入取药车，使取药车的总移动距离最短。

（10）垃圾运输问题

垃圾车每天从垃圾处理厂出发，到各垃圾集中点运回垃圾，构成旅行商问题。需要规划最短运输路线，提高运输效率，降低成本。垃圾处理厂相当于起始城市，各垃圾集中点则是需要访问的其他城市，通过优化路线，能够减少运输车辆的空驶里程，提高整体运输的经济效益和时间效益。

（11）超市商品的上架问题

超市工作人员需将商品从仓库搬运到各货架。这类似于旅行商问题，要确定最佳上架顺序，使行走的总路程最短，提高工作效率。仓库可看作起始城市，每个货架是需要访问的城市。合理规划货架访问顺序，能避免工作人员重复经过某些通道，加快商品上架速度，提升超市运营效率。

（12）图书馆的图书上架问题

图书馆管理员需将归还图书放回正确书架位置，这相当于旅行商问题，要确定最优上架路线，减少行走时间。图书管理员从还书台出发，依次将图书放回书架，通过优化路线，能提高图书上架的及时性，方便读者更快地找到所需图书，提升图书馆的服务质量。

（13）钢铁企业热轧工序的生产调度问题

钢铁企业在实际编制热轧生产调度时,一般都是从预选池的N个订单当中依次编制出M个轧制单元计划。编制轧制计划，必须满足轧制规范（有很多限制条件），例如两个连续的订单（假设一个订单一个板坯）之间，宽度、厚度和硬度的变化越小越好，基于此原则，为不同的连续订单设置惩罚费用，目标是使总的惩罚费用最小。通过设置一个虚拟订单，它与其他订单之间的惩罚费用都是 0，如果把订单看作城市，惩罚看作距离，那么热轧调度问题就变为一个，从虚拟订单开始的多旅行商问题，也就是从虚拟订单开始，找到M个轧制计划，使得总的惩罚费用最小。

（14）轧钢厂的轧制批量计划问题

轧钢厂需合理安排不同钢材轧制顺序，把不同轧制任务看作城市，轧制设备初始状态当作起始城市，寻找最优顺序以减少设备调整成本，提高生产效率。合理安排轧制顺序可以降低设备在转换不同轧制参数时的能耗和时间成本，提高轧钢厂的生产效益和市场竞争力。

（15）堆垛机拣选作业调度问题

堆垛机在仓储物流中心用于拣选货物，从初始位置出发拣选所有货物后返回。确定最优拣选路线，减少行驶距离，提升拣选作业效率。通过优化拣选路线，可以加快货物的出库和入库速度，提高仓储物流中心的货物周转率，满足现代物流

高效运作的需求。

10.5 旅行商问题的变种问题

旅行商问题作为经典组合优化模型，在应对现实场景的复杂约束时衍生出诸多扩展问题。随着实际应用中对路径规划的要求从单一的“最短距离”向多元化目标演变，各类针对特定场景约束的旅行商问题变种应运而生。例如：

（1）对称旅行商问题和非对称旅行商问题

在对称旅行商问题（Symmetric Traveling Salesman Problem, STSP）中，城市i到城市j的距离与城市j到城市i的距离相同，即$c_{ij} = c_{ji}$。例如，城市之间的距离遵循公路网络，彼此之间的行驶成本是对称的。而在非对称旅行商问题（Asymmetric Traveling Salesman Problem, ATSP）中，城市i到城市j的距离与城市j到城市i的距离不同，即$c_{ij} \neq c_{ji}$。

（2）多旅行商问题

由于限制条件的增加，旅行商问题可以衍生出多旅行商问题（Multiple Travelling Salesman Problem, MTSP），即从一个出发点出发，由多个旅行商协作遍历所有客户的问题，每个客户仅被一个旅行商访问，且所有旅行商最终返回出发点。该问题中，客户无需求，车辆无装载限制，目标是使总行驶里程最短。在实际生活中，多旅行商问题广泛应用于物流配送、快递派送、旅游路线规划等领域。例如，一家快递公司需要从一个中心仓库出发，安排多名快递员分别前往不同的客户地点送货，最后返回仓库。如何规划路线使得总行驶距离最短，就可以用多旅行商问题来建模。

多回路运输问题是多旅行商问题的扩展形式。当客户不仅需要被访问，还存在一定容积和重量的商品装卸需求，且涉及不同种类、型号车辆的调度策略时，多旅行商问题可扩展为多回路运输问题。

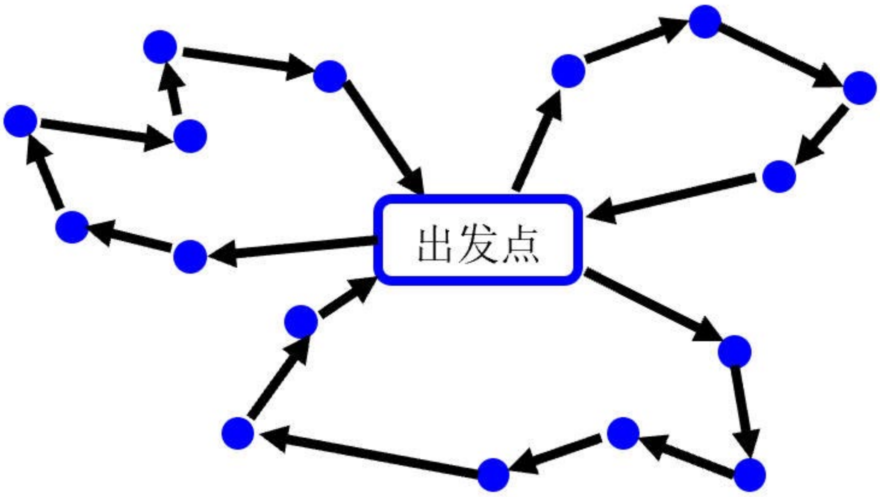

图 10.3 多旅行商问题示意图

（3）多回路运输问题

多回路运输问题是一类涉及多车辆、多路径的运输规划问题，其中车辆路径问题（Vehicle Routing Problem, VRP）是其典型代表。车辆路径问题的核心是对一系列客户需求点设计适当的路线，使多辆车辆有序遍历所有点，在满足货物需求量、车辆载重量、时间窗、行驶里程等约束条件下，实现总里程最短、总费用最少、车队规模最小或车辆利用率最高等优化目标。

多回路运输问题与旅行商问题的核心区别在于：旅行商问题是单车单回路问题，仅需规划单个主体遍历所有点的最短路径，无容量等约束；多回路运输问题（以车辆路径问题为例）是多车多回路问题，需同时确定车辆数量、客户分配方案及各车辆的行驶顺序，且必须满足车辆容量、客户需求等约束。因其约束更复杂、需协同优化多维度目标，求解难度更高，但更贴合物流配送、多车调度等实际场景。

（4）带约束的旅行商问题

例如：在带时间窗的旅行商问题（Travelling Salesman Problem with Time Windows, TSPTW）中，每个城市必须在指定的时间窗口内访问。这种问题常见于物流配送场景，客户要求货物在特定时间段送达。

在带容量约束的旅行商问题（Travelling Salesman Problem with Capacity

Constraints, TSPCC）中，物流配送中车辆的载货量是有限的，每个城市需要配送的货物量不同，旅行商（配送车辆）的载货量不能超过其容量限制。数学模型中需要增加变量来表示旅行商的载货量，并添加相应的约束条件，确保在访问每个城市后，旅行商的载货量不超过其容量。

在带优先级约束的旅行商问题（Travelling Salesman Problem with Precedence Constraints, TSPPC）中，某些城市访问之间存在先后顺序关系。例如在项目管理中，某些任务必须在其他任务之前完成。类比旅行商问题，这就意味着某些城市必须在其他城市之前被访问，在数学模型中需要通过添加相应的逻辑约束来体现这种先后顺序关系。

（5）瓶颈旅行商问题

瓶颈旅行商问题（Bottleneck Travelling Salesman Problem, BTSP）要求巡回路线中经过的最长距离最短。在一些实际问题中，不仅关注总的旅行距离，还需要关注旅行路线中最长的一段距离。例如，在通信网络的维护中，维修人员需要访问多个通信基站进行维护，为了避免某一段过长的路程导致维修时间过长，影响整个网络的正常运行，就需要考虑瓶颈旅行商问题，尽量使最长的一段路程最短。

（6）多目标旅行商问题

在实际的物流配送、旅游规划等场景中，往往需要同时考虑多个目标，所以产生了多目标旅行商问题（Multi-Objective Traveling Salesman Problem, MOTSP）。例如：不仅考虑距离最短，同时考虑费用最低等。在旅游路线规划中，游客不仅希望总旅行距离最短，还希望旅游费用最低，同时还能尽可能多地游览到著名景点。在物流配送中，除了考虑运输距离和成本，还可能需要考虑配送时间、服务质量等因素。

（7）奖金收集旅行商问题

奖金收集旅行商问题（Prize-Collecting Traveling Salesman Problem, PCTSP）涉及到在访问特定城市时收集奖励并尽量最小化总成本。与经典的旅行商问题不同，奖金收集旅行商问题允许决策者在某些城市中选择是否访问，而不是强制访

问所有城市。这意味着在选择访问哪些城市时，旅行商既要考虑收集的奖励，也要考虑旅行的成本。

10.6 本章小结

旅行商问题是图论与组合优化领域的经典 NP 难问题，其路径优化思想可追溯至欧拉研究的骑士周游问题等早期图论探索，核心是寻找访问所有节点且仅访问一次、最终返回起点的最短闭合回路。它与中国邮递员问题存在本质区别，前者以节点遍历为核心，后者聚焦边的覆盖，二者分属不同图论问题范畴，适配不同实际场景需求。

问题建模基于无向完全图展开，Dantzig-Fulkerson-Johnson 模型是经典的数学模型，通过 0-1 决策变量定义边的选择，搭配节点进出约束与子回路消除约束，精准刻画问题核心诉求；此外还有通过引入流量变量减少约束的 Single Commodity Formulation 等建模方式，为问题求解提供灵活支撑。

求解方法分为精确算法、启发式算法与元启发式算法三类。精确算法如动态规划法、分支定界法能获取最优解，但计算复杂度随节点规模指数增长，仅适用于小规模问题；启发式算法如最近邻点法、最近插入法、C-W 节约法等，通过贪心策略快速构造近似最优解；元启发式算法如遗传算法、禁忌搜索算法等，通过模拟自然进化或记忆机制跳出局部最优，更适配大规模复杂场景。

旅行商问题应用场景广泛，涵盖物流配送、生产调度、机器人路径规划、仓储拣选等多个领域，为各类路径优化需求提供解决方案。同时，针对现实场景中的复杂约束，衍生出对称与非对称旅行商问题、多旅行商问题、带时间窗/容量约束的旅行商问题等多种变种，进一步拓展了其适用范围，更好地满足多元化实际需求。

研学互通

旅行商问题作为运筹学与图论交叉领域的核心难题，其研究价值始终聚焦于算法创新与复杂系统优化的深度联动，通过理论突破与技术赋能的双重驱动，持续革新资源配置效率与路径规划的科学范式。动态适应性建模是当前研究的前沿

焦点，针对物流网络中的交通管制、需求突变等动态干扰，鲁棒优化框架将实时数据嵌入路径决策，结合强化学习与边缘计算技术，实现毫秒级路径重规划，使传统静态旅行商问题模型向“感知-决策-执行”的智能闭环演进。例如，在无人机应急配送场景中，基于时空网络的动态旅行商问题模型可根据天气、空域管制等实时信息动态调整航线，将配送准时率大幅提高，为灾害救援等紧急场景提供关键技术支撑。

跨学科技术融合推动旅行商问题研究向智能化、绿色化方向跃迁。物联网与数字孪生技术的应用，构建了物理世界与算法模型的实时映射系统，通过车载传感器网络采集路况数据，结合深度学习预测交通流量，可使城市配送路径的全局优化效率明显提升。同时，可持续性约束的引入（如碳排放核算、新能源车辆续航限制）催生了“绿色旅行商问题”研究分支，通过多目标优化算法平衡路径成本与环境负荷，在电商物流中实现单车碳排放量大幅降低。此外，人机协作决策模式的创新，如交互式遗传算法允许人类专家动态调整路径优先级，在医疗急救路线规划中兼顾时间最短与路况安全，体现了算法理性与人文关怀的协同。

旅行商问题的学术演进直接映射着产业升级的技术需求，其理论成果已深度渗透至智慧物流、智能制造、智慧城市等战略领域。从早期动态规划对小规模问题的精确求解，到现代无人机集群路径优化中的分布式计算，研究范式的革新持续推动着效率革命与社会价值创造。若想系统把握该领域的学术脉络与创新方向，可重点研读以下文献：

(1) Bellman, R. (1962). Dynamic programming treatment of the travelling salesman problem. Journal of the ACM, 9(1), 61 – 63.

这篇论文中，Bellman 首次将动态规划思想引入旅行商问题，提出利用子结构最优性分解路径问题的方法，奠定了后续精确算法的理论基础。文章为状态转移方程和递推求解旅行商问题提供了数学支撑，是理解精确求解方法必读的经典文献。

(2) Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. Operations Research, 21(2), 498 – 516.

这篇论文提出 Lin-Kernighan 启发式，创新性地采用了多点交换的局部搜索策略，有效优化大规模旅行商问题实例。该算法因其实用性和高效性成为旅行商问题解决方案的基石，被广泛采纳于工业界与学术界。

(3) Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (1985). The traveling salesman problem: A guided tour of combinatorial optimization.

这本经典专著系统梳理了旅行商问题的理论、模型、精确算法与启发式算法，是旅行商问题及组合优化领域的权威入门与进阶教材。

(4) Poikonen, S., Golden, B., & Wasil, E. A. (2019). A branch-and-bound approach to the traveling salesman problem with a drone. INFORMS Journal on Computing, 31(2), 335–346.

这篇论文研究的带无人机的旅行商问题是一种卡车和无人机混合型送货模式，其中无人机搭载在卡车上，从卡车上起飞运送包裹。通过动态求解程序，可以给出每个节点的近似下限。对分支定界思想与旅行商问题有兴趣的读者，推荐阅读。

(5) Baniasadi, P., Foumani, M., Smith-Miles, K., & Ejov, V. (2020). A transformation technique for the clustered generalized traveling salesman problem with applications to logistics. European Journal of Operational Research, 285(2), 444–457.

聚类广义旅行商问题是对经典旅行商问题的扩展，其中节点集合被划分为多个节点簇，而这些簇又进一步被划分成节点子簇。其目标是找到一条最小路径，该路径恰好访问每个子簇中的一个节点，并且每个簇的所有子簇都要依次被访问。聚类广义旅行商问题可应用于现代物流问题，推荐感兴趣的读者阅读。

(6) Kloster, K., Moeini, M., Vigo, D., & Wendt, O. (2023). The multiple traveling salesman problem in presence of drone- and robot-supported packet stations. European Journal of Operational Research, 305(2), 630–643.

这篇论文研究具有无人机站点的多重旅行商问题，这是对经典多重旅行商问题的扩展。研究发现使用无人机站点相较于传统问题的解决方案，能显著节省配送时间，在能耗方面也更优。推荐对多重旅行商问题感兴趣的研究者阅读。

(7) Pop, P. C., Cosma, O., Sabo, C., & Sitar, C. P. (2024). A comprehensive survey on the generalized traveling salesman problem. European Journal of Operational Research, 314(3), 819 – 835.

这篇论文填补了广义旅行商问题综述的空白，提供关于广义旅行商问题的数学建模、解决方法和最新进展的全面综述。此外，文章还探讨了一些开放性问题以及潜在的研究方向，适合对广义旅行商问题感兴趣的研究者阅读。

(8) Bock, S., Bomsdorf, S., Boysen, N., & Schneider, M. (2025). A survey on the traveling salesman problem and its variants in a warehousing context. European Journal of Operational Research, 322(1), 1 – 14.

这篇论文系统地回顾了已知旅行商问题的变体，并考虑在仓库场景中是否存在相应有意义的应用。如果对旅行商问题在实际仓储情景中的应用感兴趣的话，推荐阅读这篇论文。

思行经世：旅行商问题的技术赋能与乡村振兴实践

在乡村物流领域，顺丰速运的无人机配送系统为旅行商问题的社会价值转化提供了典型范式。针对赣南脐橙主产区地形复杂、村落分散的特点，顺丰在南康区构建了“无人机+地面物流”的混合路径优化模型。这种系统通过动态调整飞行路径，在确保最短运输距离的同时，确保将 80 个行政村纳入服务网络，即使部分航线需绕行更多的里程，仍实现了“村村通快递”的民生目标。

这种技术理性与人文关怀的平衡，本质上是“共同富裕”理念在物流领域的具象化。无人机配送不仅使脐橙损耗率显著下降，更通过电商平台提升果农的年均收入。这种实践既通过技术创新提升了物流效率，更通过服务覆盖于经济赋能促进了社会公平，印证了“技术创新必须服务于社会公平”的价值取向。

这一实践的深层意义在于，它突破了传统算法单纯追求效率的局限。系统在路径规划中嵌入“行政村覆盖率”、“农产品时效性”等约束条件，使技术决策与乡村振兴战略形成协同。正如浙江省《低空经济高质量发展三年行动计划》提出的“无人机下乡”战略，其通过构建 200 个乡村起降枢纽与 30 条专用航线，将旅行商问题转化为“数字鸿沟弥合工具”，体现了“科技平权”的治理智慧。

这种将算法优化与公共服务供给深度融合的模式，正是中国特色社会主义制度下“技术向善”的生动注脚。通过这种创新方法，顺丰速运不仅解决了物流效率问题，还解决了乡村振兴战略中服务覆盖不均的问题。这种方法的成功印证了中国政府推动科技创新与社会发展深度融合的战略方向，也为其他地区提供了一个可复制的乡村物流发展模式。

此外，顺丰速运在推动这一项目过程中，也积极与当地企业、农民协同合作。例如，在南康区隆木乡，顺丰投入无人机开展助农配送。这种紧密的协作，不仅提升了物流效率，还促进了农村经济发展，帮助更多农民通过电商平台销售农产品。这种实践不仅增加了农民的收入，还提高了农村生活质量，真正实现了“共同富裕”的目标。

在技术层面，顺丰速运的无人机配送系统依赖于先进的技术手段，如高精度定位系统、智能航控系统等。这些技术手段使得无人机能够精确地规划和执行飞行路线，确保货物安全地到达目的地。同时，这些技术也为数据分析提供了丰富的信息，从而进一步优化物流路径，提高运输效率。

此外，顺丰速运还积极推动了无人机产业链的发展。例如，公司与明德新材等企业合作，采购碳纤维外壳等先进材料。截至目前，赣州低空经济产业园已入驻投产企业 7 家、通航驻场单位 3 家、已签约待入驻企业 4 家，签约金额超 50 亿元。这一系列合作，不仅促进了无人机技术的发展，还带动了当地经济的发展，真正实现了“共享发展”的目标。

总之，顺丰速运在乡村物流领域的无人机配送系统，不仅解决了传统算法单纯追求效率的问题，还通过嵌入社会价值转化的约束条件，真正实现了“科技平权”的治理智慧。这种方法的成功，体现了中国特色社会主义制度下“技术向善”的强烈信心，也为全球其他地区提供了一个可复制的乡村物流发展模式。

习题

习题 10.1 某垃圾回收车负责五个地点的收集任务，起点设在垃圾处理厂（点 X），各地点的具体坐标详见表 10.6。请设计一条最优化的行车路线，确保回收任务覆盖所有点的同时，总行驶成本（如时间或距离）达到最低。要求使用最近邻

点法作为求解方法，逐步选择距离当前位置最近的下一个收集点，以实现高效而实用的路线规划。

表 10.6 习题 10.1 中的收集点位置表

位置	X	Y	Z	W	V	U
坐标	(3,15)	(8,12)	(12,5)	(15,2)	(5,8)	(10,10)

习题 10.2 在某地区，六个城市的自来水供应中心分布点如表 10.7 所示。为了高效连接这些中心，现需设计一条管道铺设路径，确保材料用量最少，以降低基础建设的成本。请采用最近插入法，逐步选择离已连接中心最近的未连通点，将其加入到管道网络中，直至全部覆盖，实现一条既经济又实用的最优铺设路线。

表 10.7 习题 10.2 中的自来水供应中心位置表

位置	P	Q	R	S	T	U
坐标	(4,20)	(1,12)	(9,8)	(14,5)	(6,15)	(11,3)

习题 10.3 在下面的旅行商问题中，城市的具体位置如表 10.8 所示。设想一位旅行商从 A 城出发，依次访问每个城市且仅访问一次，最后返回 A 城，形成一条完整的巡游路线。为了实现行程的最优化，利用遗传算法寻找最佳路径方案，确保总行驶距离达到最短。

表 10.8 习题 10.3 中的城市位置表

城市	A	B	C	D	E	F	G
坐标	(2,18)	(5,13)	(10,8)	(16,3)	(14,7)	(9,5)	(6,10)

习题 10.4 在解决旅行商问题时，精确算法虽然能找到最优解，但计算成本高昂；而近似算法则以快速近似的方式提供可行方案，但未必绝对最优。假设负责为一家拥有 50 个配送点的物流企业规划配送路线，时间紧迫，计算资源有限。在这样的情况下，应该优先选择哪种算法？请简要说明选择理由，以及考虑的重点在哪些方面，以确保物流配送既高效又合理。

习题 10.5 在求解旅行商问题的启发式算法中，常用的有最近邻点法和最近插入法，这两者都属于贪婪策略。请分析这两种算法在实际应用场景中的优缺点。

结合具体场景，举例说明在哪些情况下，某一种算法会比另一种更为适合使用，从而帮助在不同的实际需求中做出最合适的选择。