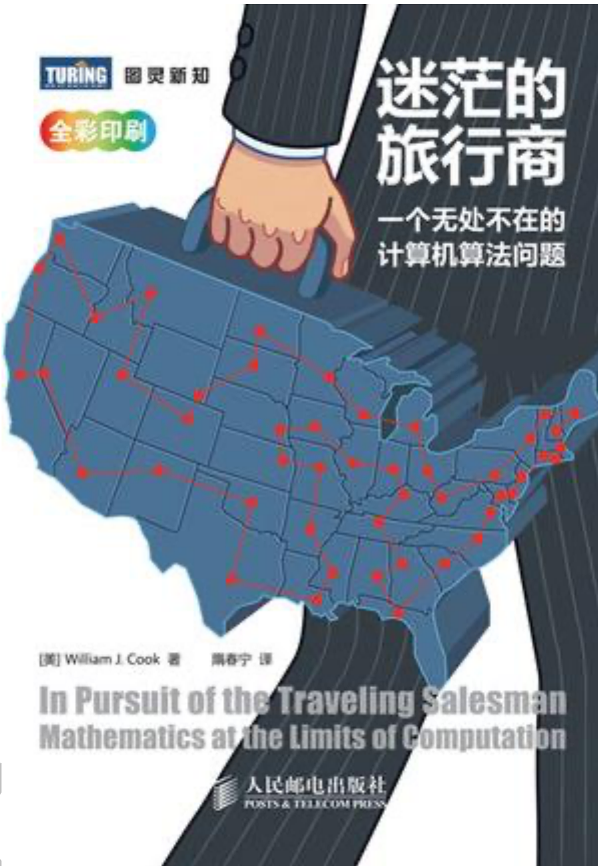

《迷茫的旅行商：一个无处不在的计算机算法问题》提要

北京市陈经纶中学 陈阳 杨熙坤

目录

导读

已经公布的《普通高中信息技术课程标准》（2017 年版 2020 年修订）指出："课程内容紧扣数据、算法、信息系统和信息社会等学科大概念"，"学习算法，可以从系统的角度描述和解决问题，有助于学生未来专业的发展"，"学生应该理解利用算法进行问题求解的基本思想、方法和过程，掌握算法设计的一般方法;能描述算法，分析算法的有效性和效率，利用程序设计语言编写程序实现算法；在解决问题的过程中能自觉运用常见的几种算法"，"掌握贪心、分治、动态规划、回溯等常见算法"，"体验算法效率的差别"，"掌握算法分析的一般方法和过程"，"能有意识地把算法及算法思想迁移应用于实际生活和学习中"，......，等等。

认真阅读《迷茫的旅行商:一个无处不在的计算机算法问题》一书，无论是对于提高教师的施教能力还是对于提高学生的学科素养来说，都是至关重要的。

《迷茫的旅行商：一个无处不在的计算机算法问题》概述了旅行商问题的起源和历史，并阐述了其许多重要的应用范围，如基因组测序、计算机处理器设计、音乐整理和行星寻找，等等。此外还探讨了人类如何在不借助计算机的情况下解决这个令人着迷的数学问题。

这篇《〈迷茫的旅行商：一个无处不在的计算机算法问题〉提要》是两位作者在认真研读《迷茫的旅行商：一个无处不在的计算机算法问题》后合作写成的。文章浓缩了原著的精华，对其中的案例、观点、叙述、解释和分析进行了提炼。

关键词

算法　旅行商问题　TSP　线性规划　割平面法　分支　大计算　复杂性

序

作者说到，旅行商问题的与众不同之处在于，尽管世界各地的一流应用数学家已经研究了几十年，但如今人们依然不知道，对于一般性的旅行商问题如何才能得出远远优于简单暴力枚举检验的通用解法。对于想努力求出通用解法的勇士，克雷数学研究所准备了一份大奖——任何人只要能够提出高效解法或者证明不存在高效解法，便会得到 100 万美元的奖金。

这个问题带来了许多美妙而深刻的结论和猜想。在精确计算领域，一道包含 85900 座城市的难题于 2006 年宣告破解，该题的最优路线经过计算脱颖而出。若是对数目异常庞大的所有路线逐一检验，需要用顶尖计算机工作站连续运转 136 年。在实际应用领域，人们使用该问题的解法，每天可以对大量应用题目计算出最优路线或近优路线。

旅行商问题之所以具有永恒的力量，原因之一在于它是应用数学领域以及运筹学与数学规划方向的驱动力，对新发现起到了有目共睹的带动作用。而且，更多的发现可能就在前方不远处。

第 1 章 难题大挑战

作者举了《游遍 33 座城市》的例子。1962 年春天，宝洁公司发起的一场广告宣传活动在应用数学家中引起了不小的反响。活动的重头戏是奖金高达 1 万美元的一项竞赛，在当时足以买下一座房子。

参赛规则是：假设 Toody 和 Muldoon 打算开车环游美国，地图上用点标出的 33 个地点都要游览，并且要走满足条件的路线中最短的一条。请你为他们规划一条旅行路线，以伊利诺伊州的芝加哥市为旅途的起点和终点，依次用线连接各地点并使得总里程最短。Toody 和 Muldoon 是当时美国一部热门电视剧中的人物。这项游遍 33 座城市的任务是旅行商问题 (Traveling Salesman Problem, TSP) 的一个具体例子。

作者谈到，TSP 的一般形式为：给定一组城市及它们两两之间的距离，求经过每座城市并返回出发地的最短路线。在计算复杂度的本质与人类认识的可能限度这一高深论题中，TSP 是讨论的焦点。虽然公认 TSP 棘手，但从某一方面来看却相当容易:经过给定一组城市的全部可能路线的总数是有限的，

因此 1962 年的某位数学家只需检验每条可能的路线，然后将最短的一条记录下来寄给宝洁公司，便可坐等一万美元支票寄到家中。这个解题策略堪称简单而完美，但有一点潜在的困难。由于路线总数极其庞大，根本不可能逐一检验。

1930 年，奥地利数学家、经济学家 Karl Menger 注意到 TSP 存在的这种困难。正是因为他的工作，数学界才开始关注 TSP 这道难题。他写到:“该问题当然可以在有限多次试验内解决，但是尚未发现能够给出比给定点的全排列数更低的试验次数的解法。”一条路线可以通过到达各城市的顺序来唯一确定。我们依次用字母 A~Z 及数字 1~7 表示 Toody 和 Muldoon 的 33 个目的地，即 A 代表芝加哥市，B 代表维奇托市，以此类推。如此，一条可能的路线便可以记作 ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567 或以上符号排成其他任一序列的形式。每一个这样的序列都是这 33 个符号的一个排列。

由于旅行的起点和终点相同，每个排列对应的路线实际上都是一条环形路线，在这条环形路线上选择另外一座城市作为起点就能得到另一个不同的序列，因此同一条环形路线对应 33 个不同的线性排列。为避免重复计数，不妨固定城市 A 为每条路线的起点，则第二座城市有 32 种可能选择，第三座城市有 31 种可能选择，以此类推。最后，可以得到环形路线的总数为 32×31×30×...×3×2×1。这个数字记作 32!，读作 32 的阶乘，表示 32 个不同物体总共有多少种排列情况，也称为全排列数。

从芝加哥到维奇托的距离等于从维奇托到芝加哥的距离，并且对于任意两座城市来说都是如此，所以我们可以减少一部分工作量。根据对称性，对于同一条路线，总路程与旅行方向无关。因此，字符串 ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567 和它的逆序字符串 7654321ZYXWVUTSRQPONMLKJIHGFEDCBA 虽然写法不同，但对应同一条环形路线。

这样一来，问题的可能路线数便可以减半为 32!/2 种排列等待我们检验。这个数字意味着我们要考察 131565418466846765083609006080000000 条不同的路线。

作者提到，好的算法能保证在至多正比于 n^k 的时间内完成运算。其中，指数 k 可以是任意值，比如取 2、3 或者更大的数。但 k 必须是固定值，不能随着 n 的增加而增加。

TSP 的好算法至今不为人知。迄今为止得到的最好结果发现于 1962 年，算法的运行时间正比于 $n^2 2^n$。尽管这不是个好的算法，但是与经过 n 个点的路线总数$(n-1)!/2$ 相比，运算时间增长得已经相当慢了。

P 类和 NP 类有可能只是同一类问题的两个不同名字。1971 年，Stephen Cook 的理论指出：NP 类中存在一道题，只要我们能找到这道 NP 题的好的算法，就能证明每一道 NP 题都存在好的算法。根据 Cook、Karp 和其他学者的工作，我们现在知道这样的题其实不止一道，它们称为 NP 完全问题 (NP-complete problem)。TSP 就是最著名的 NP 完全问题。

若能对一个 NP 完全问题找到一个好算法，就足以证明 P 类和 NP 类是等价的问题类，即 P=NP。

人们一般认为，P 类和 NP 类并不等价，但是并没有充分的理论依据，只是感觉不应该奢求 P=NP 而已。P=NP 的成立意味着只要能把一道题写成可以验证的形式，便立即得知它存在高效解法。而人们正是假设某些 NP 问题难以求解，并以此为基础建立了现行的密码体系。所以，假如这些 NP 问题都有快速算法，那么网络贸易将陷入停滞。

作者写到，TSP 是算法工程的代表问题。这门学科的研究重视实用性，以不达目的誓不罢休为理念。TSP 的规模一旦大到一定程度，求解某些实例的计算量就可能高得离谱。但这并不代表只要看到规模很大的具体问题，就只能退而求其次地选择粗略估计作为结果。研究人员正是在这种绝不妥协态度的推动下，不断改进计算机代码和技术方法，如今已能解决复杂度惊人的大问题。

第 2 章 历史渊源

作者表示，说起旅行商问题的由来，须知旅行商确实是最早开始规划路线的人。

作者举了《47 天经过 350 站的差旅》的例子。1925 年，旅行商 H.M.Cleveland 在美国培基种子公司上班，工作内容是收集玉米等农作物的订单。他在缅因州巡回的差旅从 7 月 9 日开始，一直持续到 8 月 24 日才结束，全程共经过 350 站。Cleveland 去了很多地方，缅因州只是目的地之一。他的旅途还经过了康涅狄格州、马萨诸塞州、纽约州和佛蒙特州，总共到了一千多个目的地。

15 世纪的《牛津英语词典》里收录了“circuit”一词，意为巡回旅行或巡回路线。这个词的由来与英国的司法管辖区制度有关。当时英国各地的法庭是每年定期开庭审理案件的，因此法官和律师都要在自己管辖的各城镇之间巡回。后来，美国也采用了这种制度。时至今日，虽然法官不再需要巡回奔波了，但美国人仍然把地方法庭称为巡回法庭。

作者又举了《哥尼斯堡七桥问题》的例子。欧拉是历史上最多产的数学家。他的研究范围很广泛，其中有一道难题曾经长期困扰哥尼斯堡的居民。哥尼斯堡当时属于东普鲁士，如今已改名为加里宁格勒。从卫星图像中可以看到，普莱格尔河(今普列戈利亚河)流经哥尼斯堡，形成了错综复杂的水路。中间的长方形岛屿是克奈芳福岛，河道在岛的两侧分流而过;东侧是较大的朗兹岛，两岛隔河相望;河流北岸的部分为老城区，南侧则是郊区。在欧拉那个年代，普莱格尔河上一共架有七座桥梁:克奈芳福岛和老城区之间有绿桥和下桥两座桥梁，克奈芳福岛和郊区之间也有商人桥和铁匠桥两座桥梁，克奈芳福岛和朗兹岛之间是蜜桥，朗兹岛和老城区之间是高桥，朗兹岛和郊区之间则由木桥连接。哥尼斯堡的居民喜欢在城里散步，通过绿桥、下桥、商人桥、铁匠桥、蜜桥、高桥和木桥这七座桥往返于普莱格尔河两岸。当地流传着一道多年未解的难题:如何在一次散步过程中走过全城各地区且每一座桥都恰好走过一次？这就是哥尼斯堡七桥问题。

1735 年 8 月 26 日，欧拉向圣彼得堡科学院提交了一篇论文，发表了对该问题的看法。为了抓住问题的本质，他按照经典的数学思想只把必要的信息提取出来，从而解决了问题并开辟出了一个新的数学分支——图论。

欧拉发现了最关键的一点:若散步的起点和终点不同，则路线中分别以起点或终点为端点的边都是奇数条，以途中任何一点为端点的边都是偶数条;此外，如果散步路线是闭合的，也就是说起点和终点是同一个点，那么以任何一个顶点为端点的边都有偶数条。综上可见，要么所有顶点都是“偶点”，要么恰好有 2 个顶点是“奇点”。

对哥尼斯堡的居民们来说，七桥问题的图里的 4 个顶点都是“奇点”，所以不可能在一次散步过程中经过每座桥恰好一次。欧拉非常简短的论证结束了哥尼斯堡问题的所有争论。

作者再举了《骑士周游问题》的例子。这个题目来自国际象棋，问题是:能否按照一定顺序移动骑士，让骑士从棋盘上的某一格出发，走遍整张棋盘，落在每一格上恰好一次，最后回到出发的格子?

作者指出，骑士周游路线是指一条闭合路线，它到达每个顶点刚好一次。而在哥尼斯堡七桥问题里，所求的散步路线经过每条边刚好一次。

在一个图中，哈密顿回路就是经过所有顶点恰好一次的闭合路线，而欧拉回路则是经过所有边恰好一次的闭合路线。这两种回路都是图论中的基本概念。虽然表面很相似，但实际意义却有天差地别。

判断一张图中是否存在哈密顿回路是一个 NP 完全问题，复杂度与一般形式的 TSP 相当。然而对于欧拉回路的存在性却有一条简单的判定规则：图里可以有不连接任何边的孤立顶点，但除此之外的其他部分必须是连通的。也就是说必须是连成一整片的，每个顶点都应该 与偶数条边关联。

旅行路线和地图上色是有联系的。把地图上每个区域的边界看作图的边，边界的交点看作图的顶点。生成的边界图里若存在一条哈密顿回路，则可以根据它来给地图染色。哈密顿回路不会自交，因此有内部和外部之分。位于回路内部的每条边都横穿内部区域，将其一分为二，从而可以只用两种颜色交替给内部区域染色，每经过一条边就换成另一种颜色即可。同理，回路外部的区域也可以只用另外两种颜色完成染色，最终就得到一幅四色地图。

作者提出，我们用“邮递员问题”一词表示这样一个问题：给定有限数目的点，它们两两之间的距离已知，试求连接所有点的最短路线。

这道题就是要找到一条周游所有点的路线，但不需要最终回到起点。所以只需要在终点和起点之间添加一个“虚城市”就可以把通路两端连接起来形成一条回路，从而转化为 TSP 路线。可以设虚城市和任意一座真实的城市之间的路费都为零。虽然这样增加了一座城市，但并不会干扰我们选择路线的起点和终点。

作者说到，“traveling salesman problem”的英文拼写有时存在细微不同，比如写成“travelling”或者“salesman's”之类的。本书的译者谈到，这道题的中文译名也并不唯一。除了本书使用的“旅行商问题”外，常见译名还包括“旅行推销员问题”和颇有趣味的“货郎担问题”。

作者还举了《孟加拉 600 万处黄麻农田》的例子。统计学家 Prasanta Chandra Mahalanobis 被誉为印度的统计学之父。他的一个主要研究方向是抽样调查的技术方法，并在此领域与 TSP 结下了不解之缘。20 世纪 30 年代，黄麻作物是印度政府财政收入的重要来源，大约占到出口总额的四分之一。如

何收集数据以准确预测黄麻产量是实际生产中的一大问题。当时，黄麻种植地主要分布在孟加拉地区，共有大约 600 万处小农田生产黄麻。由于黄麻种植地分散，因此想要完全调查每一处黄麻田是不现实的。Mahalanobis 提出了一个替代方案:随机抽样调查。把孟加拉地区分成多个区域，各个区域内的田地性质都差不多，在每个区域随机选一些点观察黄麻的种植情况即可。抽样调查需要花费时间和金钱，影响成本的主要因素就是在不同样本区域之间运送人和设备的时间。从这个方面来看，这个问题就变成了一道 TSP，要寻找连接所有选定农田地点的高效路线。

也许因为 Mahalanobis 是一位统计学家，他的研究兴趣在于统计学领域，所以他并没有讨论实际操作任务。换言之，他没有对于具体数据真正求出路线，而是关注最优路线长度的统计估计结果。这种视角可谓独树一帜。

在计算孟加拉地区抽样调查的预计成本时，Mahalanobis 提出的估计值派上了用场。1937 年，人们进行了一次小型试验，并在次年实施了大规模的调查。在做出这些决定的过程中，预计成本都是非常重要的考虑因素。

第 3 章 旅行商的用武之地

作者提到，旅行商问题的实用本质在其名称上就体现得淋漓尽致。

车载 GPS 使用的地图软件中常常包含一个能够求解十几个城市小规模 TSP 的小程序。十几个的数目已经足够满足单日出行的需要了。GPS 设备里存有详细的地图，从而能够准确地估计从某地去往另一地所需的时间。

普林斯顿大学的 Alain Kornhauser 是地图制图技术应用方面的专家。他介绍了 GPS 设备的一种有趣的反向用途。使用者在 GPS 上按照经度和纬度设定目的地之后，有时候会发现这个点不可能落在已知的道路区域内，也就是说根本没有路可以到达那里。但是如果当地的货车司机必须把货物送到，他往往能找到一条路成功抵达目的地，比如抄一条地图上没有标注的小路。这时 GPS 系统会把信息报告给中心服务器，服务器会在相应的地图坐标格里沿着货车经过的路径加上一条新的连接线。于是当下一次有人要去同一个地点送货的时候，地图软件就可以用到这条新添加的路了。

人们常常用小规模的 TSP 模型来规划客车接送乘客或者货车配送货物的路线。

TSP 还可以用于规划另一类路线，就是检查、视察相距较远的地区。作者举了《绘制基因组图谱》的例子。TSP 在遗传学领域的应用最为引人注目。染色体上的标记是基因组图谱的地标，所以过去十年间，精确定位这些标记一直是研究的热点。在基因组图谱中，每一条染色体上都按顺序排列了一些标记，相邻两个标记之间的距离也都有估测的数值。这些图谱中的标记实际上是 DNA 片段，恰好只在整个基因组里出现一次，而且可以通过实验室研究工作准确检测到。研究人员利用了它们的唯一性

和可检测性，对各个实验室绘制的物理图谱进行核实、对照和合并。研究的重中之重是要精确了解这些标记在染色体上出现的顺序，TSP 正是在这一步大显身手。

可以通过不同方法得到这些标记相对位置的实验室数据，其中有一种非常重要的技术称为辐射杂种细胞作图 (radiation hybrid mapping)，简称 RH 作图技术。这种方法分为两步。首先将染色体暴露在高剂量的 X 射线里，使之受到辐射后断裂成若干片段。然后把这些片段和啮齿目动物的遗传物质组合在一起，形成杂交细胞系，再分析确定标记位置。

RH 作图技术的核心思想是，通过分析标记两两之间能否并存于细胞系中，收集总结标记的位置信息。具体说来，如果标记 A 和标记 B 在染色体上距离很近，那么辐射就不太可能使染色体在它们之间断开，所以假如在一个细胞系里有标记 A，那么很可能也有标记 B。反之，如果标记 A 和标记 B 在染色体上距离很远，那么可以预料 A 和 B 将分别出现在不同的细胞系中，两个标记在某个细胞系中并存的可能性极小。研究人员可以巧妙利用这种推理方式，精心得出两个标记间距的实验值。有了两两之间距离的实验值，则测定染色体上标记顺序的问题就转化成为 TSP 模型。事实上，可以把这一顺序看作周游所有标记的哈密顿通路。这条通路很容易转化为回路，正如我们之前的做法那样，只需要增加一个额外的“虚城市”点即可。

在美国国立卫生研究院，Richa Agarwala 和 Alejandro Schäffer 带领一个小组研究这类基因组问题。针对这些具有实际背景的 TSP，他们已经提出了解决方法，开发了软件，还考虑到了如何处理错误数据。他们的软件使用了 Concorde 代码，保证能找出最优路线。这个软件为许多重要的研究作出了贡献，比如用来构建很多物种的基因组图谱，包括人、恒河猴、马、狗、猫、小鼠、大鼠、奶牛、绵羊和水牛。

作者写到，虽然我们讨论 TSP 的应用时，一般考虑的情形都需要跨越遥远的空间距离，实际到达远方的地点，但是有些 TSP 题目产生的背景却不同，无需亲身上路，只是从远处观测目的地而已。通过某种望远镜观测行星、恒星和星系，就是一个很自然的例子。

为了观测，需要把望远镜设备旋转到合适的位置，这一操作称为快动 (slew)。大型望远镜的快动由计算机驱动电机调节操控，整个过程复杂耗时。对于一组观测，有一条 TSP 路线可以让每次快动耗费时间之和达到最小，因此可以用来安排总计划。这道 TSP 里的城市就是需要拍摄照片的观测对象，各地之间的旅行费用就是望远镜在拍摄各个物体之间的快动过程所需的时间。

作者又举了《黄铜模具雕刻的加工时间减半》的例子。人们使用黄铜冲压模具制作浮雕图案，比如巧克力包装盒上凸起的装饰图。黄铜模具以前都是手工制作，但现在一般由重型数控铣床铣削雕刻而成。数控铣床刻完一个字母或者其他设计元素之后，主轴升高，铣头移动，准备刻下一个字母或设计元素。由于每个设计元素都不只是单个小点，因此铣头移动到整个元素上方的任意一点都可以，这就引入了灵活变通的可能。2008 年，数控雕刻师 Bartosz Wucke 写到，如果模具图案有大量字母或者是大量点纹组成的抽象图案，则应用 TSP 模型可以使加工时间减半。

作者再举了《558 位艺术家 15336 首曲目的 MusicRainbow 系统》的例子。在计算机音乐领域，人们同样借助 TSP 来理解大量由计算机编码的音乐。Elias Pampalk 和 Masataka Goto 在日本产业技术综合研究所工作。用户可以借助他们发明的 MusicRainbow 系统找到适合自己音乐品味的其他艺术家。

Pampalk 和 Goto 收集了 558 位艺术家演绎的 15336 首曲目。他们首先通过计算比较这些曲目的音频属性，建立了一种测定艺术家相似度的方法。然后，他们使用 TSP 模型用一条环形路线串起所有艺术家，使彼此相似的艺术家在路线上的次序也比较接近。模型中的城市是艺术家，而两地之间的旅行费用是两人之间的相似度。有了这条环形路线，用户就可以通过旋转一个旋钮在音乐库中进行选择 MusicRainbow 设备配有计算机屏幕，用来显示艺术家的信息。屏幕上有一系列彩色的同心圆，分别对应了音乐的不同高级分类，从而表明了艺术家的种种标识属性。MusicRainbow 系统的一大亮点在于，所有标识信息都是自动搜索网页获得的，因此任何音乐库都可以很方便地使用该系统。

作者表示，在应用数学文献中，隔三差五就会涌现出 TSP 模型的新奇应用。

第 4 章 探寻路线

作者指出，就算听到 TSP 不可解的断言，巡回旅行商也不会大惊小怪。他照旧会发动汽车，上路拜访顾客，履行自己的使命。基于这种脚踏实地的心态，出现了另一种思路：暂时抛开必须找到绝对最优解的念头，转而关注如何尽快求出近优解。在新观点的影响下，人们提出了各种富有想象力的想法。许多技巧和方法在这一方向的 TSP 研究中得到了发展和应用，如今亦成了计算科学领域的重要工具，如模拟退火算法、遗传算法和局部搜索算法等。

作者举了《周游 48 州问题》的例子。这道难题流行于 20 世纪 40 年代，题目是给一名旅行商设计路线，让他以华盛顿特区为起点和终点，周游美国本土的 48 个州。Julia Robinson 建议把各州首府指定为旅行商的目的地，从而缩小路线的范围。

由于无法获得城市间路程的标准数据集，Dantzig、Fulkerson 和 Johnson 自己动手给出了一组数据。他们选择的城市与前人的限定大不相同，在分别选自各个州的 48 座城市中，只有 20 座是所在州的首府。他们之所以选择这组城市，是因为很容易在地图集里查出其中大部分道路的里程。在他们使用的兰德麦克纳利标准地图集中，从华盛顿到波士顿的最短路线会经过美国东北部的另外 7 座城市，这些城市也在旅行商的目的地之列。于是，他们决定不考虑这 7 座城市。

看到地图很容易猜到，最优路线大概会用到直接连接华盛顿和波士顿的这条路。实际确实如此。因此，兰德公司的 Dantzig 等人确实有理由只考虑剩下的目的地。

他们处理了初始数据，把每段距离值先减去 10，再除以 17，然后四舍五入取最接近的整数。他们的论文列出了处理后的距离数据表，从而明确提供了这道题目的条件。

Dantzig 等人的数据并没有使用两点之间的真实距离，因此稍微改变了问题原有的几何学本质。不过，如果数据取为两点之间的直线距离（即欧几里得距离，欧氏距离），可以有效提高比较路线长度的效率。他们当时的解法就完全建立在这种近似的基础之上。

Dantzig 在讲座中透露，研究小组当时使用了一个自制的木质实物模型。模型上面标有题中的 49 个地点，每处都有一枚钉子。代表出发地的钉子上系了一根细绳。他们把这根绳子依次绕过其他钉子，从而勾画出一条路线。

这个模型是他们手工解题的得力助手。绷紧绳子测量绳长就可以获知路线的长度，也能很快直观看出下一段子路线的可能走向。这个模型完全没有提供解题的算法，却帮助 Dantzig 等人确定了一条周游所有城市的路线。后来他们证明，这条路线就是所求的最优路线。按照处理后的数据表来计算，周游美国的最优路线长度为 699 个单位长度，换算成地图集上的实际距离则是 12345 英里（即 19867 千米）。

作者认为，无论是在白纸还是在木制模型上勾画路线，思路都有一点别致之处：先选取一个出发点，然后逐步扩展出一条路线，扩展的过程可以理解为依次加入其他地点，也可以理解为依次加入其他路段。

在从头开始构建路线的时候，最容易想到的方法就是每次都在没有到过的城市中选择最近的一个。这种算法称为最近邻算法。它看起来合情合理，然而得到的路线通常不是最短的。

对于 n 座城市的 TSP 题目，若其旅行成本对称且满足三角不等式$AB+BC\geq AC$，则可以证明最近邻路线成本不会高于最优路线成本的$1+\frac{\log_2 n}{2}$倍。

最近邻算法只会生成一条路线。这条路线四处曲折延伸，最终遍历所有城市，其中每一步都选择最短的一段路程，因此这种算法非常贪心，但是“贪心”一词却被用来命名另一种算法。贪心算法同时生成许多段子路线，每一步都把最短的可用路线片段加入解集中。

用图论语言可以很方便地描述贪心算法这样的 TSP 解法，其中图的顶点就是城市，图的边则是城市之间的路线片段。最终路线是图中一条哈密顿回路，选择的边与旅行商实际出行的路段相互对应。

如果测试范例很大，贪心算法几乎总是明显优于最近邻算法。若城市在正方形内随机分布，以直线距离作为每段路程，则贪心算法得到的路线长度一般不会超过最优解的 1.15 倍，而最近邻路线长度则通常不超过最优解的 1.25 倍。遗憾的是，这个数据只是实验观察得到的。目前已知在最差情况下，对于满足三角不等式的题目，贪心算法的路线长度只能保证不高于最优解的$\frac{1}{2}+\frac{\log_2 n}{2}$倍，因此只是比最近邻算法好一点点而已。

1955 年夏天，John Robacker 投身 TSP 研究。他由随机路线出发，用 Dantzig 等人的方法做了一系列试验，成功解决了好几道包含 9 座城市的题目。这些题目规模太小，说服力不强，但他描述了通用的路线求法。如果题目的数据集很大，这种方法可以实现自动化。

根据这些试验研究，A.W.Boldyreff 提出了求近优路线的一系列步骤。他的方法具有两大优点:原理简单，使用快捷。

这种算法的思路是从一条周游几个城市的子路线出发，逐个增加新的城市，让路线像橡皮筋一样扩展，从而包围更多的城市。

由 Boldyreff 和 Robacker 提出的技巧可以得出一类方法，统称为插入算法。按照选取城市的具体规则，插入算法可分为四种:最小插入法、最近插入法、最远插入法和随机插入法。无论是哪一种插入算法，在插入新城市时都会选择路线上最合适的位点，使插入后的路线长度最短。

最小插入法就是 Robacker 介绍并试验的方法，每次选择城市的原则都是使插入后的路线长度最短。最近插入法每次比较所有剩余城市到当前子路线上每一座城市的距离，选择相距最近的一对城市。其中，不在当前子路线上的那座城市就是要插入的。最远插入法则相反，每次选择的城市都是距离当前子路线上任意城市中最远的一座。而随机插入法每次都是随机选择，把剩余城市中任意一座插入到子路线中。

在这些算法中，作者最喜欢最远插入法，因为它一开始就能把握路线的整体形状，在插入城市的过程中也会不断完善细节。

研究表明，在满足三角不等式的条件下，最小插入法和最近插入法都相当不错，得到的路线长度不会超过最优路线长度的 2 倍。虽然在实际应用中最远插入法往往效果最好，但奇怪的是在理论上其路线长度只能保证不超过最优路线的$\log_2 n$倍。

最近邻算法和贪心算法往往虎头蛇尾，最初的选择似乎很合适，最终的路线却总是令人失望。对旅行商来说，贪心没有好结果。然而出人意料的是，对于一道与 TSP 相关的题目，确实有一种贪心算法能保证得到最优解。这道相关题目是：给定一组城市，要求选出一系列道路，能够连接所有城市并且总成本（代价）最小。

作者的曾曾曾曾曾祖师爷 Arthur Cayley 研究过这样的图。这种图的结构是连通的，而且不包含回路。Cayley 给此类图起了一个合适的名字：树。同时，他还把图的顶点称为节点。

利用树的结构可以给出 TSP 的一种解法，这种解法得到的路线长度不超过最优路线的 2 倍。

道路连接问题的最优解确实就是一棵树。关键在于建设道路网的时候绝对不能形成回路，否则回路的最后一条边就是冗余的，因为它连接的两座城市之前已经通过其他道路连接起来了。这种情况下，使用贪心算法时应该以最短边优先为原则。如果通过已经选择的其他边可以从一条边的一个端点到达另一个端点，那么这条边就不能加入解集。按照贪心算法求解的过程中，各点会逐渐连接起来形成分散的几大部分。各部分规模越来越大，最终生成一棵包含所有城市的树。根据这种简单的方法得到的树一定是最小代价生成树，也就是说总成本一定是最小的。

树并不是环形路线，但是可以给出周游各城市的方式。在一棵树中，到达一座新城市后，如果有一条边从这座城市出发，尚未经过搜索，便沿这条边继续前进，到达另一座新城市；否则，如果从这座

城市出发的所有边都已搜索过，则逐个回溯之前访问的城市，直到发现某座城市连接的某条边未经搜索为止。重复上述过程，最终会到达所有城市并回到出发点。这样的路线称为按照深度优先搜索算法遍历一棵树。

在遍历结束时，每条边都刚好走过两次，这就意味着旅行路线的成本是树的代价的两倍，而最优树的代价不可能超过最优路线的成本，因此这个结果很不错。要想根据遍历路线得到一条实际路线，只需跳过回溯步骤，抄条近道即可。

为了给旅行商指路，“种”一棵树是个不错的主意。若想充分发挥树的作用，需要退一步，从莱昂哈德·欧拉的角度思考问题。把一棵树的每条边复制一遍可以生成一张图，按照深度优先搜索算法遍历树的路线实际上相当于周游对应的图的欧拉回路。复制的目的是保证图的每个顶点都连接偶数条边，而不会像哥尼斯堡七桥问题那样出现奇点。

我们也可以不复制整棵树，而是对树添加一组边，使每个奇点连接的边数都增加一条。这样使得生成的图中没有奇点，因此必然存在欧拉回路，可以截短取直，得到一条周游路线。下面以环游美国问题为例说明这种思路。如图-1 所示，这道题目的树总共包含 42 个顶点。左图用红色标出了 24 个奇点，右图中共有 12 条边染成红色，恰好连接这些奇点各一次。这样的一组边称为一个完美匹配。Jack Edmonds 提出了一种解法，可以在多项式时间内计算出最小代价完美匹配。在最优化领域，这一结论具有里程碑地位。因为最优匹配的代价至多只有最优路线的一半（原因稍后阐明），所以最小代价完美匹配就是最合适、最完美的匹配。把树和其最优匹配相结合，在新的图中截出一条欧拉回路，便得到了一条 TSP 路线，其长度不超过最优路线的 1.5 倍。在理论上这样的性能保证已经很不错，而实际使用时往往可以得到更好的结果。

为了从整体上估算最优路线的长度，首先要注意，在图上沿 TSP 路线前进时，会从一个奇点到达另一个奇点，间或穿插几个偶点。如果抄近道只走奇点，则得到一条周游所有奇点的回路。这条回路可以分拆成两个完美匹配，从某条边开始的第 1、3、5...条边组成了一个完美匹配，第 2、4、6...条边也组成一个完美匹配。这两个完美匹配中，必有一条的长度不超过奇点回路长度的一半。而 Edmonds 的最优匹配只可能更短，不可能更长。

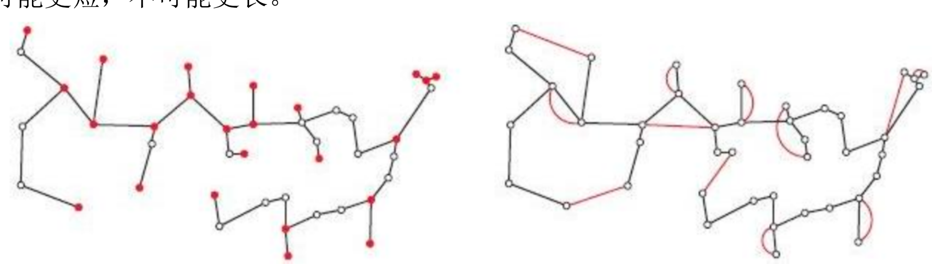

图-1 对于奇点的最小代价完美匹配

1976年，Nicos Christofides首次完整提出这一算法，将欧拉和Edmonds的结论结合在一起。Christofides算法在TSP的神殿里占有重要地位，因为已知的其他多项式时间算法都无法保证给出更短的路线上限。

如果你有意求解这道题，那么目标显然是改进Christofides算法的性能保证。Christofides算法保证路线长度不超过最优解的1.5倍。除了已经讨论的几种著名算法以外，TSP爱好者和研究者也提出了大量其他方法，包括聚类技术、划分算法和空间填充曲线等。

作者说到，贝尔实验室的Shen Lin倡导使用路线改进算法。这类算法名副其实，读取一条路线后便会查漏补缺对其调整修正。

Lin在计算题目的过程中，全面探寻了可以改进路线的方法，包括2-交换、3-交换和更多边的交换操作。2-交换的步骤就是删除路线中的两条边，用另外两条更短的边重新连接路线。

20世纪60年代中期，Lin发表的成果证明3-交换可以成功地改进路线。如果*k*远远大于2和3，搜寻改进路线的*k*-交换就需要巨大的计算量。尽管如此，Lin和计算机科学先驱Brian Kernighan还是成功实现了这种方法。他们巧妙地构建了一种新算法，取得了TSP研究中最伟大的成就之一。

Lin和Kernighan不仅计算出了出色的结果，还深入浅出地阐述了用于实现与改进该算法的诸多思想。过去40年间，在他们两人的原始论文的指导下，Lin-Kernighan算法得到了精确的设计和落实。

新近开发的此类软件能力更强，能够解决超过千万座城市的大规模TSP题目，求出非常优秀的路线。

他们设计这一基本方法原本只为解决几十或几百座城市的TSP题目，却为过去几十年中大多数优秀TSP启发式解法奠定了基础。在此期间，规模大得多的题目也获得了解决。

k-交换方法虽然在应用中表现出了优秀的性能，但在理论上却无法保证优良的最差情况性能。这是Lin-Kernighan算法的缺点所在。即便如此，使用该算法时也不必过分担心，因为它是算法中的王者，一般情况下确实都能够得到非常好的解。

Lin-Kernighan算法在实际计算领域长期占据统治地位，这得力于研究界对它持续不断的改进和增强。1998年，计算机科学家Keld Helsgaun带来了一枚重磅炸弹。

Helsgaun的主要贡献是重写了算法的核心搜索程序。Lin-Kernighan标准算法可以视为搜寻2-交换操作序列的方法，Helsgaun的新方法却以5-交换操作序列为搜寻对象。

他的方法每次都一下子考虑所有此类5-交换，因此能够发现原有标准算法完全找不到的改进操作。

新算法名为Lin-Kernighan-Helsgaun算法，简称为LKH算法。它以5-交换操作为核心，还借用了大量各类技巧，树立了探寻路线方法的新标杆。

在Lin和Kernighan发表原始论文之后、LKH算法公布之前的25年时间里，实现标准LK算法的有效程序代码屈指可数。LK搜索算法虽然有清晰明确的解释表述，但难以转化为能处理大规模数据集的软件。

LK 算法可以用难来形容，但 LKH 算法看起来简直是不可能实现的。LK 算法具有一大特色：在搜寻的每一步，回到根据地城市的路线都仅有一条。换句话说，如果从一条完整路线中删除两条边形成两条子路线，那么把它们连接起来形成新路线的方式是唯一确定的。然而 LKH 算法却不同，简单计算可知，每一步 5-交换操作都来自连续五步 2-交换，涉及 5 条子路线，连接形成新路线的可能方式有 148 种之多。所以 LK 算法和 LKH 算法之间的差别，就是 1 和 148 之间的差别，也就是难和特别特别难之间的差别。

Helsgaun 向研究界公开了完整的程序代码，以此揭示了他取得成功的秘诀。Dave Applegate 和作者通读了他的文件，结果发现其中根本没有真正的独门秘诀可言:他的代码里完整列出了 148 种情形，分别讨论了所有的可能性。他为了写出正确可行的代码、实现极为复杂的算法，付出了堪比愚公移山的努力。

Helsgaun 后来对 LKH 程序进行了有力的升级，新代码允许用户具体指定每次交换操作包含连续几步。2003 年，他以周游瑞典 24978 座城市的问题为例，计算了 10-交换操作，所得 TSP 路线的最优性于次年得到证明。

作者谈到，事实证明把 TSP 视为一类搜索问题的特例，从全局着眼看待寻找路线的过程，这种出发点无论对于找到好的路线还是想出多用途的技术都大有裨益。思路意图在于产生元启发式算法，即用来设计启发式算法的启发式算法。这类研究工作本质具有一般性。

爬山算法代指一类路线改进方法。这类算法之所以得名，是由于其过程可视为沿一系列相邻路线前进，始终向高处移动。算法的每一步都进行一次局部搜索，以寻找附近的高地。若搜索全面彻底，则最终算法将在山峰上终止，或者至少在高原上终止，因为此时所有局部移动必然走下坡路或平路。完整运行时，算法会从初始路线出发，然后迅猛攀升到局部高峰处。

模拟退火算法是一种启发式算法。它按照一定概率接受比当前解更差的邻域，因此条件限制比爬山算法更弱。这种情况下的接受概率一开始比较高，然后随着算法运行而逐渐降低，使算法有机会先跳到更高的山坡上，然后才转而稳步向上攀登。

模拟退火算法由 Scott Kirkpatrick、Daniel Gelatt 和 Mario Vecchi 发明。他们的原始论文研究了 TSP。他们写到，该方法的动力源于统计力学与 TSP 的关联。在统计力学中，退火表示先加热再缓慢冷却材料以改善其结构的工艺过程。

模拟退火算法给当前的寻找路线方法带来的最大影响很可能不在于这种方法本身，而在于它向 TSP 研究领域引入了物理学的思想。计算结果之所以能够首次突破反复 LK 算法所能达到的界限，应当归功于物理学家的跨界贡献。

20 世纪 80 年代末，Olivier Martin、Steve Otto 和 Edward Felten 三人提出了另一种方法。他们的想法是：像 LK 算法这样强大的局部搜索算法得到的路线往往会延伸到所处区域内的高地，因此可以对这

一点加以利用。Martin 等人提议，算法第二次运行时不再从随机城市出发，而是首先观察当前所处的山峰周围，判断能否越过附近的几处壁垒，到达通向更好地点的另外一座山坡上。

具体方案是通过变换原有的 LK 路线获得一条新的初始路线。算法全过程多次重复这一变换步骤，一旦发现更好的路线就用它替换原有路线。要想使算法有效运作，路线在变换后必须离开原有邻域，

因此这种改动难以通过 LK 算法还原。

这样得到的算法性能出色，称为链式 Lin-Kernighan 算法，即链式 LK 算法。

有人以地理思路看待旅行商路线，也有人从生物角度出发把路线当成能够逐渐变异和进化的生命有机体。遗传算法应用了后一种思维方式，其灵感来源是 John Holland 于 1975 年出版的里程碑式著作 Adaptation in Natural and Artificial Systems（《自然系统与人工系统中的适应性》）。Holland 并未讨论如何解决 TSP，但他的遗传算法思想不久后便出现在寻找路线的文献中。

在 TSP 领域，遗传算法的思路大致如下。首先形成路线的初始“种群”，可以通过反复从随机城市出发使用最近邻算法获取路线。然后是一般的循环步骤，在种群中选取若干对路线，让它们交叉配对得到子代路线，再从父子两代路线中选出一个新的种群。反复多次进行这一步骤，最优秀的路线最终将脱颖而出成为唯一的生存赢家。

遗传算法的实质是模拟自然界的进化过程。维持路线的种群这一想法具有明显的优势。

一只蚂蚁单独行动时并无章法可言，但是整个蚁群通过信息素轨迹相互交流却能成功找到省时省力的路线。受到蚂蚁集体行为的启发，一类 TSP 启发式算法——蚁群优化算法诞生。

蚁群算法研究的领军人物是比利时人 Marco Dorigo。此类算法的工作原理是：令一小群蚂蚁沿着图的各边移动，每只蚂蚁探寻一条路线，每次到达新顶点时都选择一条通向未访问过的顶点的边。选择规则是算法的核心，选择的依据是每条边对应的信息素值，信息素值高则选中概率也高。等到所有蚂蚁都走完路线后，各边的信息素值会根据一定的规则变化。增量与路线长度成正比，因此好路线各边的信息素值增加程度大于坏路线的边。

TSP 的各种元启发式思想得到了大量运用，除了上述性能最优秀的几种外，还有神经网络算法、禁忌搜索算法和人工蜂群算法等。

作者提到，TSPLIB 数据库收集了来自学术界和工业界的逾百道 TSP 难题，为各地区各学科的研究人员提供了公共的试验台。

20 世纪 90 年代，罗格斯大学的具体数学与理论计算机科学中心举办了一系列有关算法实现的挑战赛，其中最著名的就是 TSP 挑战赛。

作者写到，启发式算法需要在运行时间和路线质量之间权衡取舍。在该领域，丹麦人 KeldHelsgaun 和日本人永田裕一双双荣登世界冠军的宝座。

第 5 章 线性规划

作者表示，TSP 的全部挑战是:既要选出周游给定点集的最优路线，同时还要清楚它确实是最优的。人们需要的算法应当既能保证路线质量优秀，又无需逐一检验每条可能的路线。在这样的背景下，效率惊人的线性规划便是不二之选。线性规划方法结合所有路线都满足的一系列简单规则，最后提炼出一条规则，形如“所有遍历给定点集的路线长度都不短于 X”。X 的数值提供了直截了当的路线评估标准:如果我们能构造出一条长度恰为 X 的路线，那么它必然是最优路线。

目前所有 TSP 精确解法确实都采用了线性规划方法。而且，线性规划可以应用于 TSP 以外的问题，因此它已成为现代数学中最成功的传奇之一。

George Dantzig 发明了线性规划理论。

在 Dantzig 的线性规划模型中有三大核心要素。第一，他的约束条件不限于相抵的等式，也可以包括不等式，用来限定一个量至少应该和另一个量一样大。这种新特色常常用来声明某物质的数量至少也应该是零。

线性规划模型的第二大核心要素就是对约束条件加以限制，要求它们都是线性的。Dantzig 假定活动消耗的资源正比于活动的强度。

线性规划模型的第三大核心要素则是明确的目标，这为评估不同解的优劣提供了依据。Dantzig 要求目标明确，从而迫使管理人员准确描述预期目的，他本人认为这一点是自己最大的实用成就之一。在指明目标时，我们假定活动的价值正比于其强度，因此要求目标函数是模型中各变量的线性表示，并在活动强度取尽所有可能值时求出目标函数的最大值或最小值。目标函数取得所求最值时，相应各变量的赋值就是线性模型的最优解。

在建立问题模型之后如何给出其最优解？Dantzig 交上的答卷是单纯形算法。作者指出，2000 年评出的“世纪十大算法”中，单纯形算法的名字赫然在列。

2009 年，Richard Karp 举了一堆优秀算法的例子，单纯形算法排在第一个。不过 Karp 当时也不得不指出，单纯形算法在复杂度方面并不优秀，它并不能保证在多项式时间内完成运行。

时至今日，情况依然如此。单纯形算法是如今数学中应用最广的工具之一，但是我们并不清楚今后从实际模型中涌现出的问题是否一直能够迎刃而解。

在单纯形算法的发展史上，有一个阶段格外重要。1984 年，Karmarkar 公布了内点法，这是一种具有竞争力的线性规划新算法。性能强劲的台式工作站和个人电脑恰恰在此时走入寻常百姓家。强强联手，线性规划世界自此跨进超光速发展时代。从 1987 年到 2002 年间，基于单纯形算法的求解程序速度加快了百万倍。机器运行速度提高了三个量级，算法运行速度又提高了三个量级，合起来就让解题能力提高了六个量级。

作者说到，1949 年秋天 Julia Robinson 公布了第一个基于线性规划的 TSP 解法。

作者指出，任何 TSP 都可以准确描述为线性规划问题。

我们需要好好学习如何使用不等式，也就是如何提出对于求解的 TSP 题目有用处的不等式。这是线性规划求解 TSP 的核心所在。

作者说到，对变量附加整数限制之后，线性规划模型就不符合 Dantzig 的理论了，因此无法由单纯形算法或其他线性规划方法直接求解。尽管如此，成千上万的人无法抗拒整数变量带来的灵活性，所以照样迎难而上，每天都在线性规划模型中使用这样的限制。这种拓展的模型称为整数规划。

染色问题是整数规划的典范应用实例。

几乎所有已知的整数规划问题的通用解法，最初都是在求解 TSP 的过程中提出的。

作者谈到，线性规划和整数规划的研究人员遍布数学领域、计算领域、商业界、科学界及工程界。然而，与它们关系最为密切的学科却有个独特的名字：运筹学。

运筹学是关于应用高等分析方法帮助人们做出更好决策的学科。在运筹学的研究中，线性规划、整数规划等种种最优化工具与借鉴自概率论、博弈论以及其他领域的建模方法相辅相成，联合发挥威力。

第 6 章 割平面法

作者提到，单纯形算法无法解决有几十亿约束条件的大型题目。面对这种复杂性，Dantzig、Fulkerson 和 Johnson 提出了一种巧妙的处理方案——割平面法。割平面法的目标不是一举解决整个线性规划问题，而是以“随用随取”为原则，只在必要的时候才产生相应的 TSP 不等式，从而逐步计算线性规划的边界。

作者写到，分离是割平面法的核心问题。

几何形式的 TSP 题目多数都有完全或近乎不自交的线性规划解。图论里有些工具虽然不能用在一般情形下，却适用于此类情况。

作者说到，1979 年，Khachiyan 的线性规划算法登上了《纽约时报》的封面，而它恰恰具有一个有趣的性质：无需列出线性规划问题的显式约束条件，只要有可用的分离程序，就可以执行这个算法。因此，只需服从几个技术条件，Khachiyan 算法就可以用来证明：根据好的分离算法可以得到好的最优化算法；反之亦然，根据好的最优化算法也可以得到好的分离算法。这个数学结论非常深刻。

作者指出，TSP 研究是所有成功的整数规划方法的基础，割平面法也不例外。它绝对是新型整数规划解题程序里最重要的工具。

Ralph Gomory 是第一个认真研究整数规划割平面的人。

Gomory 割平面法中，构造割平面的机制同样出自 Gomory 之手，而如今的整数规划解题商业软件用到的正是这种机制的变体。

第 7 章 分支

作者说到，从 1954 年至 1973 年，研究人员也在倾力研究各种各样的其他解法。其中，最重要的解法是分支定界法，该方法以分而治之为理念。分支定界法和割平面法都是在旅行商问题的背景下提出的通用方法。如今，结合了分支定界算法和割平面法的最先进的 TSP 计算软件能解决包含成千上万座城市的旅行商问题。

在线性规划的松弛问题里寻找最优路线，就像是在干草堆里寻找最精细的针一样。如果有充足的不等式和耐心，割平面法就能解决问题。

虽然这听起来不错，可惜有时候事情并不尽如人意，因为有可能每次新加入的割平面都只能移走一点点干草。此时，我们便可以考虑把大草堆分成两垛小草堆。如果拆分方式合理，那么解决新得到的两个子问题就会比原先搜寻整垛大草堆更容易，也就能在干草堆和所有针之间理清头绪。这个拆分步骤称为分支。Dantzig 等人描述了一般情况下的分支思想，后来 Willard Eastman 根据该理念提出了完整的 TSP 算法。

简而言之，我们首先把干草堆一分为二，接着向模型中添加若干个割平面，然后在两个小草堆中舍弃一个。

作者提到，Eastman 的研究工作带来了一种搜索策略，TSP 研究者 Little 等人将其命名为分支定界法。这种方法的思路非常直截了当。从原问题出发。对于某个子问题，只要由它得到的线性规划下界大于等于一条已知路线的长度，就舍弃该子问题且以后均不予考虑。每一步我们都选择一个剩下的子问题进行分支操作，得出子代的两个子问题。以上过程重复进行，直到每个未分支的子问题均被修剪舍弃为止。

尽管割平面法和分支定界法起初曾经彼此较劲，这两种方法本质上却是天生一对。20 世纪 80 年代，当时的领军人物 Manfred Padberg 和 Giovanni Rinaldi 将两者结合起来，创造了分支切割法。他们给出了谨慎细致的程序实现，让旅行商的计算过程突飞猛进。随着一道 2392 座城市的测试实例宣告破解，他们突破了之前的所有计算纪录。

Padberg 和 Rinaldi 在计算中并没有对割平面法施加人为限制，而是恰恰相反，尽情使用所有能用的割平面，想方设法改进线性规划模型。他们由此发现了一种有用的处理方法：建立一个库，把每次切割都保存在其中，以供所有子问题共用。换言之，如果一个割平面能够改进一个子问题的线性规划下界，那么研究者就把它存起来，等到处理其他子问题时也许能派上用场。建立这个库出于两方面的考虑：第一，搜索该库的速度会比执行复杂分离算法的速度快得多;第二，TSP 计算用到的绝大多数分离算法都是试探性的启发式算法，有时它们试探不中某道子问题，却能试探中另一道子问题。因此，我们可以通过把不等式集结成库，整理出所有正中目标的试探，看看它们能对搜索树的其他部分起到什么作用。

Padberg 和 Rinaldi 的做法可以比作利用婚姻介绍所甄选潜在的伴侣。

Padberg 和 Rinaldi 采用统计学方案，构造出一组分数化程度最高的边，即取值距离 0 和 1 都很远的边。在这一组边中，他们选出旅行成本最高的一条边。因为与短边相比，在长边上进行分支往往对线性规划的界影响更大。

作者说到，分支定界算法植根于 TSP 研究，诞生后不久就成功打入了一般的整数规划领域。整数规划分支定界法的研究先驱是 Ailsa Land 和 Alison Doig。

Land-Doig 的基本方法主导了整数规划计算的实际应用，在一定意义上这是基于割平面的 Gomory 算法未曾取得的成就。直到 20 世纪 90 年代中期以后，分支切割法才成为整数规划商业软件的主力，标志着两大竞争对手终于开始携手合作。

第 8 章 大计算

作者谈到，数学方法不断进步，算法工程精益求精，计算平台日益强大，三者的结合引领 TSP 走向前所未有的显赫高位。

作者举了《游览 64 座随机城市》的例子。Held 和 Karp 攻克了一道 64 座随机城市(采用直线距离)的题目。获胜法宝是基于复杂定界机制的算法和程序。定界机制源自生成树，得到的界则整合到分支定界搜索过程中。该方法能产生非常小的搜索树。

在所有创纪录的 TSP 计算中，Held-Karp 研究是独一无二的，因为它并没有直接利用割平面法。不过，他们用到的定界机制确实与线性规划及 Dantzig 等人的成果有联系。

作者又举了《游览 67 座随机城市》的例子。几个研究小组对 Held-Karp 的分支定界法作了局部细节修正。1975 年，意大利的一个研究团队成功借此解出了一道采用直线距离的题目，规模提高到了 67 座城市。

作者再举了《游览 80 座随机城市》的例子。Ailsa Land 的学生 Panagiotis Miliotis 结合割平面法和一般的整数规划，解决了一系列随机选取地点的题目。测试题目采用直线距离，最多可达 80 座城市。这种综合解题方法最早是由 Glenn Martin 在 20 世纪 60 年代中期提出的。求解过程大致如下：从度约束线性规划的松弛出发，限定所有变量只能取值为 0 或 1，应用 Gomory 的整数规划割平面算法。如果得到的解是条完整回路，则它必定是最优解；否则，找出已知解违背的若干个子回路消去约束条件，加入模型中，重新调用 Gomory 算法，重复该过程。

作者还举了《游览 120 座随机城市》的例子。Martin Grötschel 的方法是纯粹的割平面法:他求解模型使用了线性规划解题软件，但寻找不等式只用了手工计算。他的成果是一条周游德国 120 座城市的最优路线。

这项非凡的成就展示出了梳子不等式用来求解大型 TSP 题目的强大威力。

作者举了《给电路板钻 318 个孔的最佳方案》的例子。Grötschel 计算完 120 座城市的题目之后没过多久，Manfred Padberg 立即启动了一项联合研究，他的合作者是 Saman Hong。他们的项目是将割平面算法自动化，不仅能解决多达 75 座城市的题目，而且在其他题目上可以计算出很好的下界。

一道 318 座城市的钻孔题目，此前由 Shen Lin 和 Brain Kernighan 考查过。

Padberg 不满足于取得好的近似结果，而是在几年后继续探索上述 Lin-Kernighan 题目的解。这一次，他的合作者是 IBM 公司的 Harlan Crowder。

这道 318 座城市的题目以及许多规模较小的题目之解，为 Crowder-Padberg 研究画上了圆满的句号。

作者又举了《游览世界各地 666 座城市》的例子。1987 年，TSP 计算研究再次掀起高潮。那一年公布了两项重大研究，第一项是 Martin Grötschel 和 Olaf Holland 攻克了大量测试题目，最难的一道包括了从世界各地选取的 666 座城市。看到数字 666，《圣经》研究者会说它是“兽的数目”。

Grötschel 选择这些城市时，确实有意创造出一道有如洪水猛兽般的 TSP 计算难题。Grötschel 和 Holland 结合使用了割平面法与一般整数规划，解决了这道难题。他们之所以能获得计算上的成功，主要归功于新发现了许多可以用于梳子不等式的启发式准确分离算法，从而使程序得到非常强的线性规划松弛解。

作者再举了《给电路板钻 2392 个孔的最佳方案》的例子。1987 年的第二项重要研究与众不同。Manfred Padberg 和 Giovanni Rinaldi 的研究清晰地表明，完全把求解过程限制在 TSP 领域内也有诸多好处。他们使用的工具是分支定界法。他们的计算机程序解决了许多测试实例，包括周游美国 532 座城市的题目，前面提到的 666 座城市的 Grötschel-Holland 难题，还有分别包含 1002 座城市和 2392 座城市的钻孔问题。最后这道 2392 座城市的 TSP 题目宣告解决，意味着他们取得了一项惊天动地的成就，而且其计算显然也是截至当时最复杂的最优化问题解答过程。

改进 TSP 结果的动力其实是算法方面的工作，包括用于梳子和团树的新型启发式分离算法以及分支切割法程序实现的众多新方法。

作者还举了《给电路板钻 7397 个孔的最佳方案》的例子。Dave Applegate、Bob Bixby、Vašek Chvátal 和作者（以下简称“作者四人组”）在 1988 年开始合作研究。到了 1989 年，他们已经埋头于分支切割法的具体工作中。他们在 1992 年迎来了第一项成果，解决了一道 3038 座城市的电路板钻孔问题，题目出自 GerdReinelt 的 TSPLIB 题库。然后，他们在程序和算法中加入各种细微改进，在 1993 年得到一条周游旧时东德 4461 座城市的路线，又在 1994 年攻克一道 7397 座城市的计算机电路题目，先后刷新自己的结果。

“作者四人组”从零开始编写全新的称之为 Concorde 的程序。他们把针对规模大得多的 TSP 题目的算法和技术都整合进来。新增内容中，有一个“局部切割”分离例程非常关键，它依靠城市簇收缩来获得非常小的图。如此一来，某种基于线性规划的割平面搜索方法虽然耗时很长，也可以派上用场。

作者举了《环游瑞典 24979 座城市》的例子。“作者四人组”的计算研究主要意在解决一道周游美国 13509 座城市的最优路线题目，并于 1998 年获得成功。“作者四人组”还于 2001 年计算出了周游德国 15112 座城市的最优路线，并于 2004 年计算出了环游瑞典 24979 座城市的最优路线。

《环游瑞典 24979 座城市》的 TSP 计算，将 Concorde 的能力推向了极致。

作者又举了《85900 座城市之旅》的例子。TSPLIB 里还有两道更大的题目，分别包含 33810 座城市和 85900 座城市，它们似乎超出了 Concorde 程序的能力范围。好在正当此时，Daniel Espinoza 和 Marcos Goycoolea 对 Letchford 分离算法的程序实现出现在了网上，为“作者四人组”带来了足够的马力。

2005 年 2 月，“作者四人组”开始对 85900 座城市的 TSP 题目展开最后一轮程序运行。2006 年 4 月，最优解的诞生宣告计算结束。计算出来的下界比当时已知的最好路线的长度只差不到 0.001%了，那条路线是由 Keld Helsgaun 于 2004 年发现的。0.001%的差距已经足够接近了，使用分支切割法只需稍作搜索便能很快完成剩下的这一点工作，证明 Helsgaun 的路线就是真正的最优路线。

作者提到，TSP 的计算之美在于一个简单的事实:题目规模没有最大，只有更大。

Bob Bosch 设计了六道规模各异的 TSP 难题，最小的是 100000 座城市的《蒙娜丽莎》TSP，最大的则是 200000 座城市的维米尔名画《带珍珠耳环的少女》“摹本”。数据集公开在网络上，任何有意寻找更好路线或者确定路线长度下界的人都可以自由获取。

作者再举了《〈蒙娜丽莎〉TSP》的例子。Bosch 测试问题中，《蒙娜丽莎》TSP 迄今吸引的注意力最多，毕竟它只比 85900 座城市的求解纪录高出一丁点儿，难免让人心痒手痒。永田裕一在 2009 年 3 月 17 日发现了《蒙娜丽莎》TSP 的当前最好路线。2010 年 1 月 18 日，5757044 确定为《蒙娜丽莎》TSP 路线的下界。新的下界是由 Concorde 分支切割搜索得到的。至此，下界与永田裕一路线之间仅仅相差 147 个单位长度。看上去近在咫尺，只有 0.0026%而已，可是差距依然存在。

作者还举了《1904711 座城市的世界旅行商问题》的例子。该题于 2001 年问世，覆盖了当时全球各地人类居住的每一处。这些位置由经度和纬度指定，旅行成本则是把地球视为球体，用两点间的大圆劣弧长度近似给定。如此定义的成本函数是由 TSPLIB 里的 GEO 标准稍加变化而得到的。GEO 标准以千米为单位，而这里则换算为米作单位。

2001 年秋季，研究者根据一条长度为 7539742312 米的初始路线，以及数值为 7504218236 的路线下界，确定两者之间有 0.47%的最优性差距。从那时起的 10 年间，新的计算结果不断地蚕食上述差距。路线改进方面主要归功于 Keld Helsgaun，而下界加强方面则基本归功于 Concorde 程序。当前的最好路线是 Helsguan 提出的，长度为 7515790345，最强的线性规划下界则是 Concorde 程序给出的 7512218268，由此可得差距为 0.0476%。

过去 10 年间，最优性差距不断压缩，如今只有原始值的 1/10 强，其中近乎 75%的改进都是由路线优化获得的。Helsgaun 的 LKH 程序表明，寻找路线的方法还有改进空间。

作者举了《526280881 颗恒星的 TSP》的例子。2003 年，作者和 Dave Applegate 设计的这道 TSP 题目囊括了美国海军天文台 USNO-A2.0 星表里记录的全部天体。在可以预见的未来，这或许就是 TSP 计算的终点站。最初作者想要收入每对天体之间的距离估计值，但太粗糙的数据导致所有恒星都落在为数不多的几个同心球面上。因此，作者改为模拟望远镜的移动方式。这样一来，数据就把 TSP 城市指定为天空中的位置，任意一对城市之间的旅行成本便可由两点确定的夹角度数给定。

整体处理恒星旅行商问题非常困难。当 n 取值高达 5 亿时，即使是 n^2 复杂度的运行时间也太漫长了。故而，目前的目标就是发展拆分数据集的方法。确定各部分的界和路线，然后再把它们拼到一起。按照这种方法，作者、Dave Applegate、Keld Helsgaun 和 Andre Rohe 做了初步的试验，并于 2007 年获得了 0.41%的最优性差距。这意味着，恒星旅行商问题的状态落后于世界旅行商问题，相差大约 10 年的光景。

第 9 章 复杂性

作者写到，随着数字不断增长，追踪旅行商的研究带来了数学、计算和工程领域的突破，也推动了无数实际应用的进展。对 TSP 研究者而言，这就是骄傲和快乐的源泉。

TSP 隶属于 P 与 NP 问题，而 P 与 NP 问题则跻身于“七道千禧年难题”(7 Millennium Problems) 之列。克雷数学研究所为每一道难题悬赏百万美金。

NP 问题的典型范例是哈密顿通路问题。

20 世纪伊始，David Hilbert 提出可判定性问题，引起了人们对此的关注。可判定性问题大体相当于询问，是否存在一个算法，对任意给定命题都能判定它是否可以根据一组公理证明。解决此类问题的理论不断发展，成为 20 世纪数学的一项美妙成就，开路先锋是大师级人物 Kurt Gödel（克尔特·哥德尔）、Alonzo Church（阿隆佐·邱奇）和 Alan Turing（阿兰·图灵）。

算法的直观概念就是一系列合起来能够求解某问题的简单步骤。约在距今 2300 年前，欧几里得提供了计算最大公因数的算法。但直到 Hilbert 的年代，人们仍不清楚如何才能给出算法的一般定义。在 1936 年那篇著名的论文中，图灵给出了一个答案并引入了一类数学模型，称为图灵机。

我们可以把算法与图灵机等同看待，这种有效的假定称为邱奇-图灵论题。该论题广受认可，它给出了算法的正式模型，使 P 与 NP 问题及其他复杂性问题得以准确表述。

图灵机是用来描述算法意义的强大模型，但是一种图灵机只是为了一个任务而设计的，比如对两个数求和。图灵指出了至关重要的一点:可以设计通用图灵机，它能够模拟每一种图灵机。

图灵的想法，再加上 Konrad Zuse 和约翰·冯·诺依曼等人的辛勤工作，共同开启了计算时代。

作者表示，自从数字计算机问世以后，效率问题便很快成为最基本、最重要的问题。知道一个问题能由图灵机求解是一码事，但是知道能在有生之年看到图灵机给出问题之解就完全是另一码事了。

有关算法效率的早期讨论围绕 TSP 等整数规划模型展开。所谓问题的解法，指的是能把更多情形的计算工作量降低到可控范围的求解过程。

1961 年，Jack Edmonds 讨论了寻找图中最优匹配的问题。他成功提出了该问题的一个好算法，算法步骤数的增长速率至多正比于 n^4，其中 n 是图中顶点的个数。这个深刻的结果成为了 Edmonds 的奋战重点，他的数学运算非常美妙，动摇了他人的观点。

今天，没有人质疑他的理论。Edmonds 成为了算法和计算复杂性领域的英雄领袖。

作者指出，1967 年，Edmonds 提出了惊世骇俗的猜想，认为 TSP 也许根本没有好算法。

数学家喜欢让一切都井井有条。对于复杂性理论而言，这就意味着他们重视判定问题，即答案为“是”或“否”的问题。

Richard Karp 发明了缩写记号 P，用来表示有好算法的判定问题。P 类的正式定义是指能在多项式时间内由一台单带图灵机解决的问题类。换言之，如果输入带上的符号数目为 n，那么必定存在指数 k 和常数 C，保证图灵机经过至多 Cn^k 步以后必然停机。这种定义相当好，单带图灵机替换为多带图灵机甚至换成功能强大的现代数字计算机也不会影响问题分类。用图灵机模拟先进计算机会拖慢计算速度，但是速度放慢的系数关于 n 仍然只是多项式关系。所以，如果我们有先进计算机上的多项式时间算法，就同样拥有单带图灵机上的多项式时间算法。

对判定问题而言，属于 P 类就意味着完美。不过，Stephen Cook 研究了另一类自然出现的问题，其范围或许比 P 类更大。他概述了 Edmonds 的看法，考察了能够在多项式时间内验证肯定答案的问题。要想验证答案是否正确，我们同时给出问题叙述和肯定证据，让图灵机检验答案是否确实为肯定的。

Karp 提出，可以把 Cook 研究的问题类简称为 NP。

乍一看，似乎 NP 类远没有 P 类那么严格，把问题划分为 NP 类比划分为 P 类容易得多。TSP 就是一例，检验解答容易，找出解答或许很难。类似的例子可以构造出很多。不过，它们只能暗示 NP 类可能比 P 类范围更大。对于只属于 NP 类而不属于 P 类的问题，作者尚未发现。

Stephen Cook 的论文开了 NP 问题正式研究之先河。

Cook 理论的核心思想是归约，即把问题简而化之。

问题归约的正式定义为，在多项式时间内，图灵机接受问题 A 的任意实例并据此构造出问题 B 的实例，使得 A 和 B 的答案完全相同，要么都为“是”，要么都为“否”。问题归约大抵如此，你只需牢记一点，问题 B 的规模不应当过分超出 A 的规模。

显而易见，在对诸多 NP 问题进行分类时，归约用途不小。要说明某问题很简单，可以试试把它归约为另一个简单的问题;要说明某问题很难，可以试试把另一道已知的难题归约为它。Cook 证明，每个 NP 问题都可以归约为可满足性问题。

我们可以理解 Cook 的推理过程。从问题 A 能归约到问题 B，蕴涵结论“若 B 属于 P 类则 A 亦属于 P 类”。因此，如果可满足性问题属于 P 类，那么所有 NP 问题都存在多项式时间算法。Cook 认为 P=NP 的可能性不大。

如果某个 NP 问题可由所有 NP 问题归约得到，那么它就称为 NP 完全问题。Cook 在证明后写到，可满足性是 NP 完全的。

复杂性理论领域研究的领军人物是 Richard Karp，他的论文漂亮地给出了 P 类、NP 类、图灵机和问题归约的专业阐述，还提出了如今赫赫有名的 21 个 NP 完全问题列表，并列出了 Cook 可满足性定理给出的归约。列表中，TSP 以两种不同面目出现，分别是无向图的哈密顿回路问题和有向图的哈密顿回路问题。

1979 年，Michael Garey 和 David Johnson 出版了具有里程碑意义的著作《计算机和难解问题：NP 完全导论》。此书在算法研究圈子里几乎人手一本。每次遇到新问题，研究者总是会先查一遍 Garey-Johnson 的 NP 完全问题名录，看看问题归约是否有可用的参照。

迄今为止，并没有充分靠谱的证据支持 P 和 NP 不是一回事。

作者说到，Gerhard Woeginger 是克雷大奖的民间档案管理员，保管着无数大胆的申领文件。他的“P 与 NP”网页的亮点在于，上面按照时间顺序列出了该问题的里程碑式成果，并在每条成果的开始位置相应注明“相等”或“不相等”。

Woeginger 提供了每篇论文的链接，有些时候还给出了反驳意见的链接。

目前尚不知道，使用欧氏距离的 TSP 是否确实属于 NP 类。

似乎早在 1962 年，研究者在 TSP 领域就遇到了无法翻越的高峰——Michael Held 和 Richard Karp 的研究成果。他们的动态规划算法能在正比于 $n^2 2^n$ 的时间内解决任何一道 n 座城市的 TSP 题目。时至 50 年后的今日，境况依然如此。

割平面法是当之无愧的实际计算冠军，所以当然可以考虑把它用在 TSP 的复杂性分析中。

可惜，割平面法在最差情形下的性能如何，似乎不容易一窥究竟。当前，人们正在搜寻性能更佳的分支切割法实现，而搜寻新的割平面恰是不错的理论目标，两者完全可以相辅相成、同步进行。

第 4 章描述的 Nicos Christofides 提出的基于树的启发式算法保证能给出成本不超过最优路线之 1.5 倍的路线。

Christofides 本人的做法也就是通用标准做法：假定旅行成本是对称的，而且满足三角不等式。满足这种自然条件的题目则称为 TSP 的度量题目。

Christos Papadimitriou 和 Santosh Vempala 业已证明，除非 P=NP，否则没有多项式时间的 α 倍近似算法能解决 TSP 的度量题目且满足 α 小于 1.00452。因此，他们的研究扼杀了在 P≠NP 时获得极好近似算法的幻想。

Sanjeev Arora 证明了一个重要定理，既揭示了近似方法的愿景，也揭露了它的陷阱。他证明，无论选取怎样的 α 值，只要超过 1.0，就存在适用于欧氏距离 TSP 的多项式时间 α 倍近似算法。注意欧氏距离的情形与度量情形不同，后者除非存在计算最优路线的多项式时间算法，否则根本不能指望这么好的结果。这是 Arora 定理的乐观一面，它表明欧氏距离的 TSP 可能比一般的度量题目更容易搞定。

虽然 Arora 定理是出色的理论结论，但当 α 值接近 1.0 时，近似算法的运行时间却会显著增长，实验结果更是很不理想。其实，近似方法一贯如此。因为要获得高质量的解就需要非常精细地划分空间，同时还要以延长搜索时间为代价。

作者指出，图灵风格的计算绝对不是解决旅行商问题的唯一手段。

1994 年，美国南加州大学教授 Leonard Adleman 提出了一种生物思路。Adleman 是一位曾获大奖的计算机科学家，RSA 密码系统中的“A”指的就是他。他的 TSP 解题装置在分子层面运作，力图利用极少量 DNA 存放无数信息。

Adleman 解决的题目是 TSP 的变形，属于哈密顿路径问题。输入为一个图，目标是找到一条路径，起始和终止顶点均已指定，要求该路径周游所有顶点，但不考虑旅行成本问题。

经过七天的缜密工作，Adleman 的团队最终发现了一条得到完整哈密顿路径的双链。

Adleman 的实验需要有位科学狂人在实验室里辛辛苦苦工作一个星期，而处理 DNA 原本就是活的生物的拿手好戏，所以改用其他生物替人干活的主意值得一试。由一群本科生组成的研究小组就实现了这个点子，在细菌体内构建合适的 DNA 序列，成功解决了哈密顿路径问题的一道小例题。

一个来自日本的研究团队创立了使用变形虫解决 TSP 题目的通用做法——变形虫计算可解决一般的 TSP 题目，而不仅限于哈密顿路径问题。

这项研究的最重要之处在于，研究者发明了一种由生物构成的新型可工作计算机。

还有一种基于光学的方案，用于求解哈密顿路径问题。2007 年，这种解法在互联网上一鸣惊人。

令 D 等于每个顶点恰经过一次时整条路线的延迟总时长。

表面上看，这种光学解题机器解决 n 座城市的问题时，需要的步骤数目正比于图的顶点数目。它之所以一举成名，正是因为人们看中了这一点。然而，罗马尼亚计算机科学家 Mihai Oltean 经过仔细分析得出结论，选择延迟时间长度的时候，若想给出可分辨的特征信号，D 必须至少为 2^n 个单位时间。无论是基于 DNA、细菌、变形虫还是光学原理，上述 TSP 解题程序虽然都有完全一步到位的优点，但是也都离不开随城市数目增加而呈指数增长的资源。一种更有可能成功的思路要利用量子物理学特性。第一个提出把量子物理用在计算设备中的人是 Richard Feynman。

量子计算设备最为基本的组成部分是量子位 (qubit)。传统计算机使用 0/1 二进制位表示信息，量子位与传统二进制位类似，但很特别。它能够存放值 0，也能存放值 1，还能同时存放 0 和 1 两个值。

根据量子力学的神奇特性，如果有 100 个量子位，那么它们合起来就能同时编码 2100 种可能性，因此确实有可能实现真正的“一步到位”，一次检验所有 TSP 路线。

如果认为数目众多的量子位就能保证轻松求解旅行商问题，那就错了。诚然，一百万个量子位足以编码周游 1000 座城市的每条路线，问题出在物理学上。虽然所有路线都能同时表示，但是真正考察所有量子位状态的时候，却只剩下一条路线，其他路线全都消失得无影无踪。

作者喜欢的一种猎奇的计算模型是时间旅行解题程序。

回到脚踏实地的现实世界，别忘了 Dantzig、Fulkerson 和 Johnson 使用的那台实体装置，还有 20 世纪初期货真价实的旅行商背后的助手团队。他们使用的都是钉绳法，即用图钉在地图上标出目的城市，再用一条绳子勾勒出可能的路线。绷紧绳子便可以测量路线长度，所以这种装置既比纯手工计算更快，又比穿越时空的量子计算机更容易搭建，直至今日仍是最实用的 TSP 实体辅助解题工具。

第 10 章 谋事在人

作者说到，无论是旅行商、律师、牧师、作家还是游客，多年来一直都在计划路线。所有的网球运动员在漫长的训练课程结束后，还要把一地的球都捡起来。

在 2007 年的一次数学研讨会上，SylviaBoyd 提出一道 50 座城市的 TSP，掀起一场挑战赛，规则是要求所有计算一律人工完成。挑战赛持续了一天，作者和 Dave Applegate 得出了获胜路线，以微弱的优势战胜另一位 TSP 研究者 Gérard Cornuéjols。可惜的是，这条路线不是最优路线。

作者谈到，最优路线需要避免自交，所以路线经过边界各点的顺序必须和凸包边界经过边界各点的顺序相同。

仔细分析 10 座和 20 座城市 TSP 题目的人工解题实验结果之后，MacGregor 和 Ormerod 得出结论:寻找 TSP 题目近似解的难度是由内部点的个数决定的。

作者提到，对于规模非常小或者大部分城市都落在凸包边界上的 TSP 题目，实验参与人总是可以正常解决，彼此之间几无差别。然而，城市数目达到 20 座以后，参与人表现的个体差异立刻就变得非常明显。

在类似 TSP 的问题上，人类实验者的表现差异一直是神经心理学临床研究的依据。Halstead-Reitan 成套测验（或称 HRB 神经心理成套测验）中的连线测验就是一个精彩的实例。

连线测验的第一个环节要求被试者画一条路线顺次连接 25 座分别带有标号 1~25 的城市，完成速度应当尽可能快，主试者会随时指出错误。正确路线通常不是周游所有城市的最优路线，不过也算是没有自交的较短路线。测验的第二个环节与之大同小异，只是把城市标号由 1~25 改为 1，A，2，B，...，12，L，13 的混合标记。这项分为两部分的测验由美国军方心理学家于 20 世纪 40 年代发明，现在通用的计分系统由 Reitan 设计，评分标准完全基于被试者完成任务花费的时间。

无数实验结果表明，连线测验可以鉴别出遭受大脑损伤的患者，而且灵敏度很高。1990 年的一篇综述称，它是国际神经心理学会各成员机构中最广为使用的测验。

第 11 章 错综之美

作者写到，数学一直是复杂图案——圈外人眼里的神秘图案的源泉。

数学家如果形容某项研究极为美妙，完全不表示这种美可以实体化。TSP 也不例外。也许有条周游一组点的路线形状很优美，但是数学家并不会因此着迷。TSP 的几何结构和复杂度结合之美，才真正令数学家流连忘返。

尽管如此，一些迷人的美术作品中也出现了 TSP 的影子。有些画作甚至成功抓住了旅行商引人入胜的数学本质。

Julian Lethbridge 是一位知名的当代艺术家。

在谈起旅行商系列作品时，他向作者解释说，他在一本期刊里看到了 TSP 的描述，立刻被好路线的简洁思想和简约空间深深吸引。

作者表示，球面上的路线没有内外之分。球面会分为两部分，像相邻的拼图一样镶嵌在一起。

Robert Bosch 的名字在第 1 章已经出现过，那道含有 100000 座城市的《蒙娜丽莎》TSP 题目正是出自他之手。

Bosch 很喜欢最优化方法，包括求解 TSP 的精确算法和启发式算法。他借助最优化方法创造了无数 TSP 艺术品，除了《蒙娜丽莎》还包括《维纳斯的诞生》等。Craig Kaplan 是加拿大滑铁卢大学的计算机科学家。他们两人合作开发了可以巧妙设定城市位置的精细复杂的方法，从而能用一条好的 TSP 路线有趣地重现一幅画作。

Bosch 和 Lethbridge 都巧妙地使用了 TSP 和相应的若尔当曲线。Bosch 作品的巧妙之处在于设计中的对称性，而 Lethbridge 在画作纹理布局上独具匠心。相比之下，Philip Galanter 则采取了鲜明直接的做法，绘制了一系列大型 TSP 壁画。

作者说到，艺术家 J.Eric Morales 常住美国波特兰市，圈内人都称他为 Mo。他的手绘若尔当曲线用法自成一体，这种艺术形式被他命名为“迷宫投影”。Mo 只用一条连续曲线就能在画中创造丰富的表现。他的一笔画技巧广受瞩目，相关作品出现在耐克鞋和苹果 iPod 保护套上，画像还得到了篮球明星迈克尔·乔丹的授权。

据 Mo 回忆，他小时候常玩磁性画板 (Etch-A-Sketch)。用一根不自交的曲线在画板上绕来绕去，画满整个屏幕，一画就是好几个钟头。后来，他学习艺术，知道了线条疏密和色调明暗之间的原则与关系。这时，他又想起了童年的一笔画，便意识到自己可以用连续曲线绘制出类似照片画质的画作，只需要操控各部分线条的间距——间距近就代表色调暗，间距远则代表色调亮。由此得到的曲线明显类似于短的 TSP 路线。

对于如何将 TSP 风格的画作拓展到其他介质上，Mo 有不少点子。在他最新的作品中，阳光穿过仿照迷宫雕塑而成的物体，投影出清晰可辨的人脸。“迷宫人”是他设计的另一项应用，这是一个虚拟的皮肤透明的无头人体模型。他在人体上附以一层颜色渐变的迷宫图案，从视觉上模拟其环境，并在表层传递文字信息，使虚拟人体栩栩如生。

TSP 艺术项目旨在用连续曲线一笔画出原版画作的摹本。作画时需要精心选取一组城市位置，使用启发式算法生成周游点集的短路线。

该项目诞生于欧柏林学院，Bosch 及其学生 Adrianne Herman 在该校发明了一种算法，按照数码图像各部分的灰度，成比例地设定城市分布。他们将一幅图片转化为网格，根据每个格子的平均灰度，放置 0 到 k 座城市。格子几乎为白色时对应 0 座城市，黑色则对应 k 座。城市在格子内的位置是随机放置的。k 值和格子的大小决定了 TSP 题目中城市的数目。

由一条周游 Bosch-Herman 点集的短路线可得到一幅图，并能辨认出原始图像。但是密集的点往往导致路线呈锯齿状，无法充分重现原始图像的连续色调。为此，Bosch 和 Kaplan 采用一种多遍算法，通过计算重心重新分布城市位置，大大改进了图像的色调。计算时，参照原始图像的色调确定城市间几何距离的权重，使得各点向色调较暗的区域集中。整体上，这一过程避免城市密度突然改变，从而可以由暗色调细腻地过渡到明亮色调。

作者说到，Jaroslav Nešetřil 不仅是离散数学界的重要研究者，也是成功的艺术家。

从很久以前开始，Nešetřil 便成为德国波恩大学离散数学研究所的常客，那里的负责人 BernhardKorte 同样是数学界和艺术界的双栖人才。波恩离散数学研究所不仅是离散最优化的顶尖研究中心，还设有计算博物馆，藏品主题为计算、美术和音乐。

在应用领域，波恩离散数学研究所的专长是集成电路——构成现代电子设备核心的计算机芯片的最优化设计。它的研究人员最擅长应用离散数学来提高芯片速度，改善组织结构。

随着人们不断深入了解 TSP 内在的复杂性，TSP 与艺术的密切联系也将越来越多地显露在世人面前。

第 12 章 超越极限

作者说到，从今往后，TSP 之美仍将继续吸引数学家和计算机科学家，这一点毫无疑问。

本书涉及的许多研究题目依然悬而未决，比如《蒙娜丽莎》难题、世界旅行商问题、4/3 猜想、突破 Held-Karp 运行时间下界以及改进 Christofide 近似算法界限。只有非常渴望深入探究 TSP 计算之谜的人，才有可能对它大彻大悟。

如果能证明 P=NP，那么人类将大步跨进新时代，拥有能够建模并理解世界的高效计算工具。另一方面，如果事实符合多数专家的猜测，有 P≠NP，那么无数重大问题将永远得不到解答。因为只要解法的运行时间呈指数增长，则用于计算的机器速度再怎么提高都无法望其项背。

若 P≠NP，则无论是科学界还是非科学界，通用解法都将受到限制。

参考文献

[1][美] William J.Cook,著.隋春宁,译.迷茫的旅行商：一个无处不在的计算机算法问题 [M].中国北京:人民邮电出版社,2013 年 10 月.