组合数学丛书

组合优化

Combinatorial Optimization

□ William J. Cook　William H. Cunningham
William R. Pulleyblank　Alexander Schrijver　著
□ 李学良　史永堂　译

高等教育出版社
HIGHER EDUCATION PRESS

□ 组合数学丛书

ZUHE YOUHUA

组合优化

Combinatorial Optimization

□ William J. Cook，William H. Cunningham
William R. Pulleyblank，Alexander Schrijver 著

□ 李学良 史永堂 译

高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

图字：01－2008－4895 号

图书在版编目(CIP)数据

组合优化/(美)库克(Cook, W. J.)等著；李学良，史永堂译. —北京：高等教育出版社，2011.3
书名原文：Combinatorial Optimization
ISBN 978－7－04－031959－0

Ⅰ.①组… Ⅱ.①库…②李…③史… Ⅲ.①组合规划 Ⅳ.①O221.7

中国版本图书馆 CIP 数据核字(2011)第 025559 号

策划编辑 赵天夫 **责任编辑** 赵天夫 **封面设计** 张 楠
责任绘图 尹 莉 **责任印制** 刘思涵

出版发行	高等教育出版社	购书热线	010－58581118
社 址	北京市西城区德外大街 4 号	咨询电话	400－810－0598
邮政编码	100120	网 址	http://www.hep.edu.cn
			http://www.hep.com.cn
经 销	蓝色畅想图书发行有限公司	网上订购	http://www.landraco.com
印 刷	北京中科印刷有限公司		http://www.landraco.com.cn
		畅想教育	http://www.widedu.com
开 本	787×1092 1/16	版 次	2011 年 3 月第 1 版
印 张	21	印 次	2011 年 3 月第 1 次印刷
字 数	400 000	定 价	59.00 元

物 料 号 31959－00

著者简介

William J. Cook 现任美国佐治亚理工学院教授, 1983 年获得加拿大滑铁卢大学博士学位, 1998 年被邀请在国际数学家大会上作 45 分钟报告, 2003 年、2004 年、2009 年分别担任 Beale-Orchard-Hays 奖、George Pólya 奖、Fulkerson 奖的评审主席. 主要研究领域为整数规划与组合优化, 所出版的专著《The Traveling Salesman Problem: A Computational Study》于 2007 年获 Lanchester 奖.

William H. Cunningham 现任加拿大滑铁卢大学数学系教授, 1971 年获得博士学位, 主要研究领域为组合优化、多面体组合学、拟阵等.

William R. Pulleyblank 现任 IBM 业务咨询服务事业部商业优化中心副总裁, 1973 年获得加拿大滑铁卢大学博士学位, 曾任加拿大滑铁卢大学教授, 曾在 IBM 研究中心身兼数职 (包括 IBM 研究中心数学科学院总监), 他推动了 IBM 研究中心在超大规模计算领域的多项研究, 主要研究领域为运筹学、组合优化以及优化应用等.

Alexander Schrijver 现任荷兰国家数学和计算机科学研究院 (CWI) 教授. 因在组合优化领域基础的开创性工作, Alexander Schrijver 与 Martin Grötschel, László Lovász 一起于 2006 年获得 John von Neumann Theory 奖; 于 2003 年获得 Dantzig 奖, 分别于 1982 年、2003 年两次获 Fulkerson 奖, 于 2005 年获 Spinoza 奖, 所出版的专著《Combinatorial Optimization: Polyhedra and Efficiency》、《Theory of Linear and Integer Programming》分别于 2004 年、2005 年获 Lanchester 奖.

序言

组合优化 (也称为组合最优化), 是应用数学的一个活跃领域, 它把组合数学、线性规划以及算法理论的技术结合起来解决离散结构上的最优化问题. 这个领域里已经有许多经典的教材, 但我们觉得还存在空间对这门学科进行新的论述, 以覆盖过去十年里所取得的一些进展. 我们本来打算将这些素材形成一本适合一个学期课程的基本教材, 然而事实证明包含前沿课题的迫切要求是不可抵挡的, 因而手稿内容逐渐增加, 最终超过了人们对一学期课程所能覆盖内容的合理期望. 本书可以允许教师从所论述的众多课题中适当挑选授课内容, 我们希望这是一个优点. 这样, 本书可以作为数学系、运筹学系和计算机科学系本科生和研究生的教程. 高等理论课程可能需要用一到两次课讲授第二章、3.1 和 3.2 节, 然后集中讲授 3.3、3.4、4.1 节以及第五、六章的大部分和第八、九章的一些部分. 入门课程可能要覆盖第二章、3.1 到 3.3 节、4.1 节、4.2 或者 4.3 节两节之一, 以及 5.1 到 5.3 节. 更多关于整数线性规划和多面体方法的课程可以主要基于第六、七章, 并将包含 3.6 节.

最具挑战性的习题已经用黑体标出, 这些应该只可能用在高等课程中.

阅读我们的教材仅需的先决条件是有一定的数学基础. 我们频繁地使用了线性规划对偶, 因此不熟悉这个主题的读者应该在学习课本主要内容之前预先学习附录中的线性规划.

许多阅读过本书早期手稿的同行们给出了中肯贴切的评论和建议, 我们从中获益很大. 特别地, 我们要感谢 Hernan Abeledo, Dave Applegate, Bob Bixby, André Bouchet, Eddie Cheng, Joseph Cheriyan, Collette Coullard, Satoru Fu-

jishige, Grigor Gasparian, Jim Geelen, Luis Goddyn, Michel Goemans, Mark Hartmann, Mike Jünger, Jon Lee, Tom McCormick, Kazuo Murota, Myriam Preissmann, Irwin Pressman, Maurice Queyranne, André Rohe, András Sebő, Éva Tardos 以及 Don Wagner. 这本书是在 Bellcore、Bonn 大学、Carleton 大学、CWI Amsterdam、IBM Watson Research、Rice 大学以及 Waterloo 大学完成的.

有关本书的信息, 包括勘误表, 能够在网站

http://www.math.uwaterloo.ca/~whcunnin/bookpage.html

上找到.

译者序

本书的四位作者 William J. Cook, William H. Cunningham, William R. Pulley-blank, Alexander Schrijver 均为组合优化方面的著名专家, 他们不仅研究成果卓著, 而且出版了一些很有影响力的著作和教材. 这本《组合优化》就是其中的一部经典教材.

本书共分九章, 另外还添加了一个附录 A. 第一章从旅行售货商问题和匹配问题这两个重要例子出发, 简单介绍了什么是组合优化问题, 什么是算法, 以及如何度量算法的运行时间; 第二章介绍了最小生成树和最短路算法; 第三章从对最大流问题的刻画开始, 详细阐述了最大流与最小割定理及其在多个不同实际问题中的应用, 深入探讨了最大流算法, 另外还简单介绍了多商品流问题; 第四章介绍了最小费用流问题及相关算法, 其中包括原始对偶算法和对偶尺度放大算法; 第五章介绍了最优匹配问题及相关算法, 包括最大匹配的花算法、最小权完美匹配的花算法、一般匹配问题的算法等, 还讨论了匹配算法在邮递员问题及双层嵌入等问题中的应用; 第六章介绍了整多面体和有界多面体的相关概念和基本知识, 针对匹配问题介绍了割平面算法和整数线性规划算法; 第七章介绍了旅行售货商问题的多种启发式算法, 以及割平面算法和分支定界法; 第八章介绍了拟阵的相关概念和基本知识, 阐述了拟阵交算法及其相关应用; 第九章简单介绍了 $\mathcal{NP}$ 类问题和 $\mathcal{NP}$-完全性, 并证明了几个著名问题的 $\mathcal{NP}$-完全性, 形式地定义了问题、算法及运行时间, 并简单介绍了图灵机; 最后, 附录 A 罗列了线性规划的基本知识, 这在前面的章节中将要用到.

受高等教育出版社的邀请翻译这本著名教材, 我们感到非常荣幸, 但也感到

诚惶诚恐, 而且越翻译就越感到压力大. 我们深深地感到, 读懂一部原著是一回事, 而翻译好一部原著又是一回事, 区别关键在于精准上. 我们本着尽量遵循作者原意的原则进行翻译, 而不是像自己重写一部这方面的教材那样, 以便使得读者能够品尝到本书的原味. 但由于我们的翻译水平有限, 不妥之处在所难免, 错误之处还请读者不吝赐教, 我们当深表感谢.

本书的翻译初稿是在 2009 学年讲授组合优化这门课程时形成的, 我们要感谢参加的博士研究生, 他们是李莎莎、李玮、李静、火博丰、马红平、李磊、孙跃方、李一阳、计省进、陈莉莉、刘蒙蒙. 初稿之后, 我们又核对了多遍, 并对有些单词反复查询词典, 体会作者的用意, 在此过程中, 前三位博士生付出了很多的时间和精力. 本书由高等教育出版社引进出版, 编辑赵天夫先生一直鼓励我们翻译此书, 并耐心回答我们的各种问题, 我们衷心地感谢他为本书的出版所做的努力, 可以说没有他的鼓励和督促就没有这本翻译教材. 最后感谢南开大学组合数学中心对译者们的大力支持.

译者
2011 年 1 月
南开大学组合数学中心

目录

第一章　问题和算法

§1.1　两个问题

旅行售货商问题 (The Traveling Salesman Problem)

一家石油公司在尼日利亚的海上油田有 47 座钻井平台, 每座平台有一套操纵装置, 它能控制从平台关联的油井到岸上储油罐的原油出油量. 为了调节出油率, 人们需要定期检查钻井平台. 这项工作由直升飞机完成, 它从岸上的直升飞机基地出发, 飞到要求的平台, 然后返回基地.

直升飞机飞行的费用昂贵, 所以石油公司希望找到一种方法, 给直升飞机规定飞行路线, 既能检查所要求的钻井平台, 还能使全部的飞行时间最少. 如果我们假设飞行时间与飞行距离成正比, 那么这个问题就是一个*欧氏旅行售货商问题* (Euclidean traveling salesman problem) 的例子. 在欧氏平面上给定点集 V, 每个点有一对坐标 (x, y), 而坐标为 (x_1, y_1) 和 (x_2, y_2) 的两点间的距离是 $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$. 我们希望找到一个经过 V 中所有点的简单圈 (或*环游* (tour)), 使它的长度最小, 我们称这样的环游是*最优环游* (optimal tour). 这个例子中点集 V 由所有需要检查的钻井平台和岸上基地组成.

欧氏旅行售货商问题
输入: 欧氏平面上的一个点集 V.
目标: 找到经过这些点的一个简单圈, 使得边的长度之和最小.

试图解决这一问题有很多方法. 大多数简单的方法有一个共同特点, 要么

从解的质量上, 要么从运行时间上看, 它们表现得都不够好. 比如, 假定我们希望直接测试所有解, 然后从中选择一个最好的, 这种方法肯定能够找到最短圈. 但是如果 $|V| = n$, 那么存在 $(n-1)!/2$ 种不同的可能解 (习题 1.1). 假定我们有一台随意支配的计算机, 它能在一纳秒 ($=10^{-9}$ 秒) 内测定单个的可能事件. 如果我们只需要检查 23 个钻井平台, 那么遍历所有可能的环游就要用将近 178 个世纪!

另一方面, 假定我们要求一种较快的方法, 但不必保证产生最优解. “*最近邻点算法* (Nearest Neighbor Algorithm)” 是这样进行的: 任取一个起始点, 然后去最近的一个还未被访问过的点, 从那里再继续去下一个最近的未被访问过的点. 重复这一过程直到所有的点都已被访问过, 然后回到起始点. 图 1.1 给出了将这一算法应用到一个样本问题的结果 (取自 Gerd Reinelt 的 TSPLIB). 注意到尽管每一次移动局部来看都是最好的选择, 但最终的结果却可能很坏. 首先, 这种算法容易遗漏这样的点, 稍后它一定会以很高的代价被访问. 其次, 有时也许会 “身陷困境”, 就是算法使你访问到了角落中的某个点, 在那里为了继续这次环游, 你被迫做一次长距离移动以到达下一个最近的点.

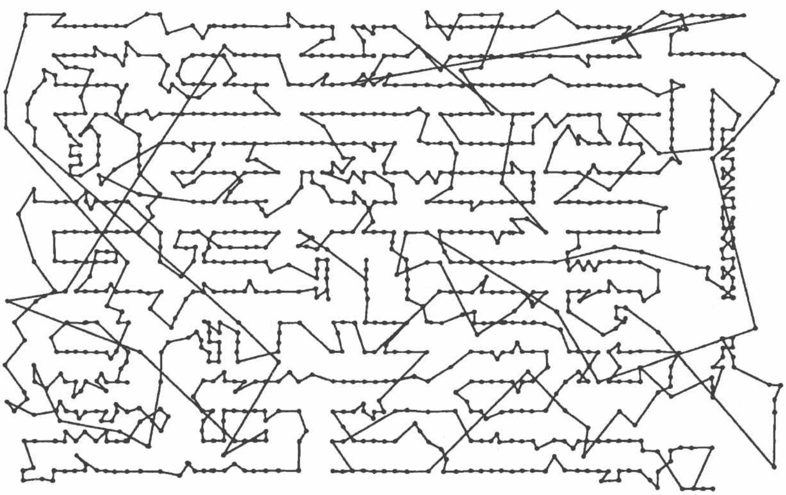

图 1.1　最近邻点解

匹配问题 (Matching Problem)

一名逻辑电路设计师要用绘图仪绘制一个计划好的电路以便于直观检测. 绘图仪通过来回移动一支笔, 同时在笔的下面前后卷动一张纸来操作. 每绘制一种新颜色的线之前要换上这种颜色的笔, 所以每种颜色的线都是独立绘制的. 问

题是要把绘制图形所需要的时间减到最短. 这个时间由两部分构成: “落笔” 时间, 也就是实际绘制线条发生的时间, 和 “抬笔” 时间, 这时笔不接触纸面, 只是简单地从要绘制的一条线的终点移向另一条的起点. 令人惊讶地是, 常常一半以上时间被用在抬笔移动上, 在落笔时间上我们几乎无法控制, 但是我们可以大大减少抬笔时间.

比如, 假定我们希望绘制一张如图 1.2 所示的电路图. 首先注意到我们将要绘制的图形是连通的, 这就使事情简单了, 原因我们稍后陈述. 你能否确信有些抬笔移动是必要的呢? 我们定义图形的*点* (node) 就是两条或多条线相遇或交叉的点, 或是一条线的端点. 换句话说, 它就是图形上这样的点, 使得有正数 (2 除外) 条线从它发出. 我们称一个点为*奇* (odd) 的, 如果有奇数条线从它发出, 否则称它为*偶* (even) 的, 参见图 1.3.

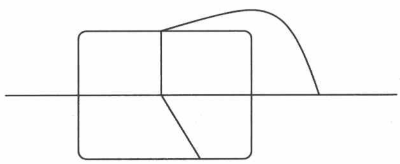

图 1.2 电路图

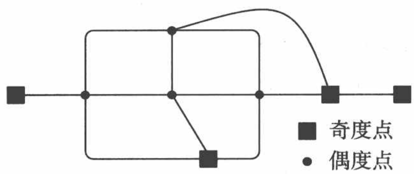

图 1.3 奇度点与偶度点

图论中最古老的定理之一蕴含着图中总存在偶数个奇度点这样一个结论, 另一个源自欧拉的古老定理声称: 可以在没有抬笔移动的情况下描绘一个图并回到起点, 当且仅当它是连通的并且没有奇度点.

找到一组新的线, 把它们添加到图形上去, 使每个奇度点变为偶度点, 并使新的线上全部的行进时间尽可能的少, 通过这种方法我们能使抬笔时间最短.

设 $t(p,q)$ 是从点 p 到 q 画一条线所需要的时间 (抬笔或落笔), 如果我们假设 $t(p,q)$ 与 p 和 q 间的欧氏距离成正比, 那么 t 满足*三角不等式* (triangle inequality): 对任意点 p, q, r, 有 $t(p,r) \leqslant t(p,q)+t(q,r)$. 另外, 如果 $t(p,q)$ 与无论哪个方向 —— 垂直或水平 —— 上较大的移动成正比时, 三角不等式同样成立.

每当 t 满足三角不等式时, 新线的最优集合使奇度点成对. 在欧氏距离的情形下, 找到这些线的问题就是一个*欧氏匹配问题* (Euclidean Matching Problem) 的例子.

欧氏匹配问题
输入: 欧氏平面上的一个点集 V.
目标: 找到一个线集合, 使得每个点恰好是一条线的端点, 并且这样的线的长度之和最小.

如果最初的图形不是连通的, 那么我们可以添加一些额外的线, 得到一个无奇度点的图形, 但它本身不连通. 在这种情况下, 一定数量的额外抬笔移动是必要的. 此外, 该移动的最小化问题包含欧氏旅行售货商问题作为特殊情形. 因为假定有一个要解决的欧氏旅行售货商问题的实例, 我们在每个点的位置上画一个小圈, 这样构成一个图形. 如果取这些圈之一作为笔的起始位置, 假设笔的行进时间与欧氏距离成正比, 那么最小化抬笔时间就是旅行售货商问题.

相似与差异

欧氏旅行售货商问题与欧氏匹配问题是组合优化中两个著名的模型, 这两个问题有几个相似之处. 第一, 每个都涉及在平面上选择连接一些点的线的集合. 第二, 在两种情形下, 可行解的数目都太大, 以至于不可能在令人满意的时间内考虑所有的可行解. 第三, 对这两个问题来说, 大多数简单的启发式算法表现得都不是很好.

然而, 两个问题之间还存在着隐藏在表面下的主要差异. 一方面, 存在有 Edmonds 的有效算法, 它能对欧氏匹配问题的任何实例找到最优解. 另一方面, 对欧氏旅行售货商问题, 不仅没有这样的已知算法, 而且大多数的研究者认为, 简直就不存在这样的算法!

对欧氏旅行售货商问题所持的这样一种悲观看法是由于将在第九章讨论的计算复杂性理论. 非正式地说, 这一论断就是如果对欧氏旅行售货商问题存在一个有效算法, 那么对每个问题, 只要满足能够有效地检验至少一个最优解的可行性的条件, 也将存在一个那样的有效算法. 这里最后一个条件不是很强, 几乎每个组合优化问题都满足它.

在本书中, 自始至终我们都将在那些已知有有效算法的问题和那些已知与欧氏旅行售货商问题同样困难的问题之间谨慎拿捏分寸, 大部分的精力将用于描述位于分界线 “好的一边” 的模型, 包括第五章和第六章对匹配问题的深入讨论. 除了它们自己很重要外, 这些好的模型为进攻那些位于 “坏的一边” 的问题也构建了基础. 在汇集了一系列组合优化的方法之后, 第七章中我们将通过对旅行售货商问题的讨论来说明这一点.

§1.2 度量运行时间

尽管我们上面用过的 “有效” 这个词是凭直觉的而且对于某些目的也足够了, 但用一些方法去量化这个概念还是很重要的. 我们遵循成规, 通过给出算法在解决给定规模的问题时所需步骤数的上界来估计该算法的有效性. 在使之精确前, 一些警示的话是适宜的. 界定步骤数仅仅提供了算法有效性的估计, 它不

应被看作是一种硬性的规定, 以为有更好的界就意味着在实际中有更好的运行性能, 因为这个界是从给定问题的所有可能实例中得出的, 而实际中你只需要算法尽可能快捷地解决手头的实例 (某些病态的实例可能让你的算法没完没了地运行, 这样的事也许并不会真的困扰你). 这种现象的一个熟知的例子是线性规划的单纯形法: 对广泛类型的问题它表现得相当好, 但一般地说, 它的行为仍然没有好的界. 尽管如此, 公平地说, 算法复杂性的思想的确常常指出一种方法相对于另一种方法的优势. 此外, 这种概念的广泛使用使得很多算法被发现, 这些算法被证实不仅在理论意义上更优, 而且在实际中也更快捷. 记住这些提醒以后, 让我们更准确地定义 "给定步骤数的上界" 是什么意思.

算法的概念可以用图灵机或别的计算模型来表示 (参见第九章), 但就目前而言, 把算法看作解决问题的一列指令, 这一直观概念已经足够了. 我们关心的是: 某个算法解决一个给定问题需要用多长时间? 计算机结构的快速变化使得按照某个特定机器测定所有运行时间的做法几乎失去了意义. 因此我们在一种抽象的计算机模型上估计运行时间, 在这种模型上我们对算法执行中的 "基本" 运算计数. 粗略地说, 一个基本运算就是其工作量被一个常数界定的运算, 即它不依赖于问题实例的规模. 但是对加、乘、除和比较等算术运算, 我们有时对这一规则做例外处理, 并以单位时间代价计数这样的运算, 也就是说运算涉及的数的长度不影响运算的时间代价. 这样做经常是恰当的, 因为出现在很多算法中的数不随着算法的进行而趋于增大. 另一种更精确的模型是按 "字节运算" 来计数: 数字用二进制来表示, 并且算术运算被逐字节地执行. 当涉及的数字的长度明显影响了问题的复杂性时, 这样的计数更恰当 (比如对一个数是否为素数的检测).

组合优化问题通常由诸如某个网络或一族集合这样的离散结构与一个可以代表例如费用或容量这样的数的集合一起构成. 我们通过对结构的某种编码长度 (比方说用二进制符号) 加上数字集合的规模来估计问题的规模 (当我们计数算术运算时, 每个数字按单个单位计数; 当我们计数字节运算时, 每个数字按用二进制表示时所用到的位数来计数). 当然, 这种估计依赖于被选择的特定编码, 但是如果使用的编码类型是一致的, 那么我们就获得了一个稳健估计. 另外, 在大多数情况下, 编码的各种选择在规模上仅会有一个常数因子的差别. 因此给定一个问题的实例, 我们用一个整数 n 来估计它的规模, 这里 n 代表编码的字节数加上数的集合的规模. 我们因此可以作出例如 "步骤数被 $5n^2 + 3n$ 界定" 这样的论断.

在分析一个算法时, 我们的主要兴趣在于它在大规模实例上的性能表现. 这源于一个显然的理由, 那就是差不多任何方法都应该能解决一个小规模问题. 当问题的规模使得劣等的方法不能在合理的时间内解决这个实例时, 优等的算法

就将闪现它的光泽. 因此, 如果一个算法以 $5n^2+3n$ 作为运行时间界, 那么我们常常略去 $3n$ 这一项, 因为对大的 n 值, 它是可以忽略不计的. 此外, 尽管界 $5n^2$ 显然比 $17n^2$ 更好, 但对一个问题实例, 在能够解决它和不能够解决它之间, 这两种界多半没什么差别. 因此我们通常只注意界的量级, 把 $5n^2+3n$ 描述为 "n^2 阶". 对此有一个形式的记号: 假设 $f(n)$ 和 $g(n)$ 是非负整数集合上正的实值函数, 如果存在常数 $c>0$, 使得对所有足够大的 n, $f(n)\leqslant c\cdot g(n)$, 那么我们称 $f(n)$ 是 $O(g(n))$ (记号 $O(g(n))$ 读作 "大 O $g(n)$"). 因此 $5n^2+3n$ 是 $O(n^2)$ 而 $35\cdot 2^n+n^3$ 是 $O(2^n)$.

作为这些想法的例子, 再次考虑旅行售货商问题的最近邻点算法. 我们描述过这种快 (但有时粗劣) 的方法, 用它来作为枚举所有 $(n-1)!/2$ 个可能的环游的替代. 我们可以很容易地用 O 记号来量化它.

让我们首先考虑算术运算模型. 欧氏旅行售货商问题的一个实例可以通过给定将要访问的 n 个点的 (x,y) 坐标来确定, 所以一个实例的规模就简单地是 $2n$.

执行最近邻点搜索算法的一种容易 (尽管有时是低效率) 的方法就是建立一个 n 元数组, 这里每个对象在数组中有三个域 $\boxed{x \mid y \mid \text{标记}}$. 我们通过把每个点 v_i 的坐标 (x,y) 放置在第 i 个对象上并取所有的标记域为 0 来初始化这个数组. 算法的一般过程是取一个点 v_j (比如在第一次通过时取 $j=1$), 扫描全部 n 个对象, 对所有标记域等于 0 的点 v_i, 计算从 v_i 到 v_j 的距离, 同时记住具有这个最小距离的点 v_{i^*}. 然后我们输出 v_{i^*} 作为环游上的下一个点, 取 v_{i^*} 的标记域为 1 并且从 v_{i^*} 开始继续这一搜索. 当我们遍访所有的 n 个点后算法终结.

初始化过程取 $3n$ 个基本运算 (除一个循环和控制运算的近似值外). 通常的过程取 n 步来检查标记域, 再加上最多 $n-1$ 步距离计算, 其中的每一步取 3 次加, 2 次乘和 1 次比较 (为保持最小距离) (注意到在距离计算时不必计算平方根, 因为我们只需要比较 $(x_j-x_i)^2+(y_j-y_i)^2$ 的值来找到对点 v_j 来说的最小距离点 v_{i^*}). 因为执行一般步骤 $n-1$ 次, 所以我们得到运算的一个上界 $3n+(n-1)(n+6(n-1))$, 即最近邻点算法需要 $O(n^2)$ 次算术运算.

为了以字节运算模型分析算法, 我们需要用在 (x,y) 坐标中考虑字节数的方法度量输入规模. 一种标准的估计是 $2nM$, 这里 M 是在 (x,y) 坐标下 $1+\lceil\log(|x|+1)\rceil$ 和 $1+\lceil\log(|y|+1)\rceil$ 的最大值 (log 为以 2 为底的对数. 如果 t 是有理数, 那么 $\lceil t\rceil$ 是大于或等于 t 的最小整数, 而 $\lfloor t\rfloor$ 是小于或等于 t 的最大整数). 现在我们必须读 M-字节长的数 (这样初始化需要 $2nM+n$ 步), 并且以字节来计算和比较值 $(x_j-x_i)^2+(y_j-y_i)^2$ (需要 $O(M^2)$ 次运算), 算法中基本运算的数目只在上述事实下改变, 所以对字节运算数目的一个快速估计是 $O(n^2M^2)$.