

## 算法研究的方向和方法的比较研究

贺思敏

清华大学 计算机科学与技术系 计博一

1997年5月

计算机是二十世纪人类最伟大的发明，它极大地推动了科学技术的进步，强烈地影响着人类社会生活的各个方面，并且对人类精神与智力的至尊地位提出了巨大的挑战。最近，计算机“深蓝”战胜了国际象棋天才霸主卡斯帕罗夫，三、四十年前人工智能前辈们的预言终于成为现实，从任何意义上将这都是极具震撼力的事件。这既是人类的胜利，也是计算机的胜利。

计算机之所以有力量，硬件的高速发展是一个重要原因，但还不是最主要的原因。计算机的力量主要来自其软件，软件的核心是算法。算法是计算机的灵魂，算法的改进所产生的作用要比单纯提高硬件速度有效得多。算法的历史比计算机长得多，例如求两个整数的最大公约数的欧几里德算法距今已有两千多年的历史了。算法的理论也极为深奥，它不仅已是数学的一部分，而且有些内容可以说是元数学的一部分，例如交互式证明理论。因此，算法的研究一直是计算机研究与应用中最具挑战性的工作。

本文作者受算法研究的挑战性吸引而进入这个领域，几年来最大的感受是算法研究的各个方面都存在着两种既互相斗争、又共同发展的矛盾运动。一方面，矛盾双方都有深刻的理论，都有优美的结果，可以说从事任一方面的研究都有吸引力；另一方面，无论从个人特长，还是从国家和社会的需要来看，又必须有所选择，有所侧重。这就要对矛盾双方的特点有清晰的了解。由于算法研究的博大精深，本文不可能对所有重要方面都予以评述，只能就作者研究组合优化问题时涉及到的有关算法研究的方向与方法的问题谈一谈其中的矛盾运动特征。

### 1. 计算机不能做什么 → 计算机能做什么

计算机算法的研究有两条截然不同的路线：一条路线研究计算机不能做什么，另一条路线研究计算机能做什么。

研究计算机不能做什么早在计算机诞生之前就已经开始并有了明确的结论。1936年，图林(Turing)提出了图林机模型，用这个抽象的计算机模型定义了算法、计算、可计算等概念，证明了存在计算机不可计算的问题，从而给出了计算机能力的界限。

70年代初期，Cook、Karp定义了NP完全的复杂性类，证明了相当多的现实问题是NP完全问题。NP完全问题用现有的算法求解均需要指数长的时间，因此，在现有计算资源下实际上不大可能求解。这是人类对计算机能力的局限性又一次深刻的揭示。由此诞生的计算复杂性理论成为理论计算机科学的中心内容。

计算复杂性既是对计算机能力的一大限制，又是对人类认识能力的一大限制。今天，计算机已成为人类认识自然、改造自然必不可少的最有力的武器，因此，计算机不能做什么就意味着许多事人类同样也不能办到。研究连续问题计算复杂性的著名学者Traub认为，或许有可能从形式上证明某些科学问题是无法解答的，因为宇宙中不存在能够解答这类问题的计算资源(时间，存储器，能量等)。

计算复杂性理论也使我们对什么是数学中的美有了新的认识。例如，图论中一个著名定理是平面图判定的Kuratowski定理，这个定理说一个图可平面化当且仅当图中不含两种类型的子图。从数学上讲，这是个优美的定理。但是，直接按这个定理设计计算机程序即算法，复杂度极高！真正有效的计算机算法选择的完全是另外一条路，其中主要是深度优先搜索。这从数学上看似乎不太优美，但是此算法的复杂度是线性的，因而是极其有效的。因此我们说，计算复杂性使我们对美与效率有了新的认识，没有效率的美至少是不完美的。

但是，以NP完全性理论为代表的计算复杂性理论也逐渐暴露出它的局限性。

NP完全性理论指出，NP完全问题的求解可能存在本质困难。但是，NP完全问题在科学研究和生产实践中广泛存在，而且迫切需要解决。无论多么困难，这些问题是不会消失的。因此，仅仅指出问题的难解性是不够的，更重要的是给出求解方法。而NP完全性理论最大的不足就是它不提供正面的解决方法。NP完全性理论的所有结论基本上是否定性的，非常容易使人在面对真实问题时持一种悲观、消极的态度。

NP完全性理论的第二个缺陷是它仅指出用一个算法求解一个问题的所有实例时在最坏情况下可能是指数复杂度的，而我们在真实世界遇到的问题并不一定正好是最坏实例。即使对每一个算法均存在最坏实例，也并

不意味着某一个实例对所有算法均是最坏的。换句话说，NP 完全性理论只是指出以不变的算法对付万变的问题是存在困难的，而以万变的算法对付万变的问题则就不受 NP 完全性理论的限制了！

可见，NP 完全性理论给我们带来对计算机能力局限性的深刻认识的同时，它的非正面的、非构造性的研究方法和研究结论也有很大的局限性。在当前大量现实问题迫切需要解决的形势下，自然应当更加重视正面的、构造性的算法研究方向，这个方向的主要内容就是算法的设计与分析。

算法设计与分析的目的是探讨计算机能做什么。1995 年，一批优秀的理论计算机科学家代表理论计算机科学界总结了理论计算机科学在通讯网络、并行计算机体系结构、软件系统、超大规模集成电路设计、学习理论、生物学、数学、制造、天文学等领域做出的贡献，所有的贡献都是算法的形式。可以说计算机对人类社会所做的贡献必须通过算法才能被接受，而计算机不能做什么的研究对大多数人来说是不必关心的问题。那么，从学术角度，算法复杂性理论与算法设计与分析理论是什么样的关系呢？

算法复杂性理论是算法设计与分析理论的元理论。元理论分析这个理论本身，例如对这个理论的基本概念给予精确的定义，并研究这个理论的局限性。一般而言，理论确立有用的正面结果，而元理论则多半由反面结果组成，它限定这种理论的研究范围。

因此，算法复杂性理论常给出消极结果，算法设计与分析理论常给出积极结果，这分别是它们的本质特点，我们不必过多指责算法复杂性理论。算法复杂性理论的反面结果对算法设计与分析的方向具有很大影响。最初，人们追求的是问题的最优解。NP 完全性理论诞生后，人们意识到追求最优解是不太现实的，转而寻找问题的近似解，但希望能有一定的性能保证，比如比最优解差不超过 5%。NP 理论的进一步研究表明，对相当多的一些问题，要使近似解具有任意的性能保证，也不大可能在多项式时间内完成。这样，算法设计的目标开始转向启发式算法，即直观合理，但在数学上不再具有性能保证的算法。在算法设计目标逐渐现实化的过程中，NP 完全性理论提供了重要的理论依据。

但是，算法设计与分析理论从 NP 完全性理论中得到的启发毕竟太少、太抽象、太不具体了，特别是在技巧上得不到任何帮助，因此，算法设计与分析虽然在计算复杂性理论限定的范围内发展，但却基本上走着一条相对独立的道路。吴文俊先生在总结我国古代数学发展的特点时说：“我国的古代数学基本上遵循了一条从生产实践中提炼出数学问题，经过分析综合，形成概念与方法，并上升到理论阶段，精练成极少数一般性原理，进一步应用于多种多样的问题。从问题而不是从公理出发，以解决问题而不

是以推理论证为主旨，这与西方之以欧几里德几何为代表的所谓演绎体系旨趣迥异，途径亦殊。”这一段话实际上也是对算法设计与分析理论的发展道路的最好描述。

总之，算法复杂性理论和算法设计与分析理论分别从不同角度研究算法，既相互补充对方的不足，又具有很大的独立性。我国目前的科研和生产中有大量的实际问题需要解决，更关心计算机能做什么，因此应当优先研究算法的设计与分析。我们在下两节着重讨论算法设计与分析中存在的矛盾运动特点。

## 2. 算法设计的技巧性 → 算法设计的统一性

组合问题的算法设计是学术界公认的难题，其中一个重要的原因是组合问题个性太强，几乎是一个问题一个算法，没有什么规律可寻。这样，算法在设计时需要高度的技巧性，在应用时仍需要较高技巧把实际问题化成为数不多的几个现有模型后才能求解，在学习时除了死记外，很难学到算法设计的方法。

但是，在组合问题及其算法极强的特殊性背后，实际上蕴涵着相当强的统一性，其中非常重要的一种统一性就是我们在论文中研究的主要内容——局部搜索。

局部搜索通俗地讲，就是瞎子爬山，拐杖能探到的点就是“局部”的范围，在这个局部范围内调整到一个更好的点就是局部搜索。如此简单的一种思路却代表着相当多典型算法的共同特征。

线性规划的单纯形算法是典型的局部搜索算法。单纯形算法沿着约束域的凸多面体的棱逐渐向更好的点改进，当不能再改进时就达到了最优解。几个经典的组合优化问题，如最小支撑树、最大流、最大匹配等，现有的好算法均是局部搜索算法。我们常用的排序问题的算法，最自然的想法也是把相邻的逆序的两个数对换位置，也是局部搜索的思路。尽管所有这些算法是完全独立地设计出来的，根据具体问题的不同细节上有所区别，但其中的共性是相当大的，当然不挖掘是无法发现的。

局部搜索不仅是上面谈到的经典问题的算法特征，而且是求解 NP 完全问题的常用方法。六、七十年代在求解旅行商问题和图划分问题时发展形成了局部搜索的各种技术。近些年来流行的计算智能的一些方法，如模拟退火算法、遗传算法等，均是在局部搜索的基础上所做的变形。而且，对

于连续优化问题，由于本质上不能枚举，因此可以说所有算法均是局部搜索性质的算法。这样，多项式可解问题与 NP 完全问题、离散优化问题与连续优化问题虽然有本质区别，但是在算法上却有着深刻的相似性。

我们之所以探求算法的统一性，第一个目的是整理现有的算法。现有的算法建立在高度的技巧性上，象散落的一粒粒珍珠。我们发掘出算法的统一性之后，就可以以这种统一性为线索将算法有机地联系在一起，在问题之间、算法之间相互关系的基础上把握它们各自的特殊性，这要比过去单纯记忆每个算法，在认识上要深刻得多。

探求算法统一性的第二个目的是设计新算法。许多问题的算法，最初是凭着高度智慧才设计出来的，这是非常宝贵的创造，不充分利用这一创造是非常大的浪费。探索算法统一性可以把一个问题上算法设计的经验有效地转移到另一个问题上，使这个新问题的算法设计有一个高的起点。换言之，统一性的探求实际上就是一种方法的探求，有了方法，就相当于有了规律可循，算法设计就不困难了。

我们在论文中提出的利用可读性变换进行算法设计的思想就是利用统一性的一个例子。一个真实问题往往有多种建模方式，不同的模型下有不同的求解算法。这些不同的模型及算法既然是求解同一个问题，它们之间必然有一定联系。进一步讲，不同问题之间也存在联系。例如，NP 完全问题之间存在着多项式变换，因而一个问题的多项式算法必然导致另一个问题的多项式算法。这样，我们既可以利用同一个问题一种形式下的算法设计另一种形式下的算法，也可以利用一个问题的算法设计另一个问题的算法。论文中我们用可读性变换的思想把可满足性问题化成多项式方程组后用吴文俊消元法求解，不仅使吴方法求解可满足性问题具有了可读性，而且使可满足性问题有了一种全新的算法。

总结起来，我们首先需要高度技巧性来设计第一个问题的第一个算法，设计多种问题的多种算法，没有这个条件，探求统一性就失去了基础，成为无米之炊。但是停留在毫无关联的算法上并不能说我们已经很好地掌握了这些技巧。探求统一性可以使我们超越技巧，形成方法，进一步推广到其他问题上。统一后的方法往往具有更大的简单性，更符合美的标准。

### 3. 算法的理论分析 → 算法的实验分析

算法的分析历来被认为是非常数学化或理论化的。其实在算法早期发展过程中并不是这样。Hopcroft 在他的图灵奖演说中谈到，60 年代的算法

研究非常令他不满意：杂志上发表的算法仅给出在一些样本实例上的执行时间，几年后又有人提出一个改进算法，声称在同样的实例上速度更快，但搞不清这是由于计算机更快了，还是程序实现更好了，还是算法本身有改进。因此，Hopcroft 力主在最坏实例情况下的渐近复杂性基础上建立算法的分析理论，使之与计算机和程序实现无关。

随后，以渐近复杂性为基础的算法的理论分析成为算法分析的重要工具，它以最坏情况下的渐近复杂性给出算法复杂性的上界，并由此给出了判定一个问题是否属于多项式时间可解类的判据。

但是，随着问题、算法及其分析方法的发展，发现以最坏情况下的渐近复杂性作为算法复杂性的度量具有很大局限性。最典型的例子是线性规划问题的单纯形算法，它在最坏情况下是指数时间复杂度，但是在日常应用中的表现非常好。因此人们开始怀疑最坏情况下的渐近复杂性是否是合适的算法复杂性度量。

后来，人们引入了平均情况的复杂性，发现单纯形算法的平均复杂性不高，这就解释了理论分析与现实表现的矛盾。但平均情况的复杂性又引入了新的问题，因为平均情况分析要假设实例服从某一概率分布，相当于每个实例有一个权重，如何保证这种分布假设具有现实意义下的代表性呢？也许难例的权重过小，导致了过分乐观的结论呢？

进一步，人们发现渐近复杂性也有不合理之处。渐近是考虑问题规模无限大时的情况，而现实世界的问题规模总有一个范围，因而渐近复杂性好的算法在求解现实规模的问题时并不一定优于一个渐近复杂性差的算法。有时渐近分析的结论完全不能用。最典型的例子是对模拟退火算法的分析。模拟退火算法是通过模拟固体由高温到低温缓慢降温到达能量最小状态的过程来求解组合优化问题的一种算法，理论上渐近收敛到最优解的性能。但使用马尔可夫链等复杂工具进行分析的结果是：仅当允许无限多次变换时，模拟退火算法才是一种最优化算法；对最优解任意近似的逼近，对多数组合优化问题都导致比解空间规模还大的变换数，从而导致算法的指数执行时间。这使我们搞不清楚模拟退火算法到底有没有效果。直至 1989 年 Johnson 等人通过详尽的实验才证实了模拟退火算法确有优点。Johnson 等人是算法理论分析的权威，他们带头转向算法的实验分析是极其耐人寻味的。90 年代，算法的实验分析逐渐兴起，开始显示自己的威力。

实验分析的最大优点是技巧性少，现实性强。通过选择一些具体问题的实例，通过算法的真实运行，我们可以知道现有的计算资源下算法可以求解多大规模的问题，达到什么样的质量。实验分析可以在完全真实的条

件下进行，不需要做各种假设。因此，实验分析比理论分析更值得作为一般方法推广。但是，正是由于实验的限制条件较少，比较容易进行，因而更需要有所选择，否则会陷入数据的汪洋大海之中。因此，实验分析常常以理论分析为指导。

例如，实验分析也研究理论分析常研究的算法的渐近性能，方法是通过测试从小到大各种规模的实例上算法的性能，然后用某种函数形式进行最小二乘方拟合，便可估计出算法的渐近性能大概是个什么趋势。这种趋势分析是建立在精确的数量化结果的基础上，因而比纯粹的理论分析给人一种更为现实的感觉。

又如，实验分析也研究算法的平均和最坏性能，方法是选择各种结构的实例，有随机实例，有真实应用的实例，也有故意设计的刁钻古怪的实例，在这些实例上测试算法的性能，则可以估计出算法平均和最坏的性能，以及算法对实例结构的敏感性。

实验分析不仅仅按理论分析的一些模式来分析算法，它有自己的生命。实验分析可以发现理论分析不便于发现的结果和现象。我们在论文中通过实验分析提出了一个重要猜想，即模拟升温比模拟退火具有更好的性能。这与模拟退火从物理背景到理论分析都不相同，但却有实验证据的支持。尽管仍要进行进一步的验证，但这已经表明实验分析可以发现非平凡的现象。

总结起来，算法的理论分析与算法的实验分析是算法分析必不可少的两个方面，互相补充对方的不足。如果一个算法的理论性能已经分析得很清楚了，那么实验分析往往主要起一个验证作用，但应当强调的是，即使这种情况下，实验分析仍是不可或缺的。如果一个算法的理论分析非常困难，难以给出有价值的信息，特别是求解 NP 完全这类本质困难的问题时，实验分析的重要性就显现出来了。

上面我们对算法复杂性理论、算法的设计与分析方法进行了比较研究，从中可以看到，没有一种理论或方法是完美的，可以用于解决所有我们感兴趣的问题。因此，保持算法研究的多样性是必要的。从当前的研究现状看，算法研究对复杂性理论、算法设计的技巧性、算法分析的理论性过于偏重，因此，有必要加强对算法的构造性、统一性和实验分析方法的研究，以保持一定的平衡性。算法研究是一个富于挑战性和创造性的研究领域，把握正确的方向，掌握正确的方法，必然会为人类社会带来更大的精神和物质财富！