计算方法丛书

非数值并行算法

第一册

——模拟退火算法

康立山 谢 云 尤矢勇 罗祖华 著

科学出版社

计算方法丛书

非数值并行算法(第一册)

模拟退火算法

康立山 谢云
尤矢勇 罗祖华 著

科学出版社

1998

内 容 简 介

本书系统地介绍模拟退火算法以及这一方法的并行实现和在优化、搜索、机器学习、统计物理中的应用. 主要内容包括: 模拟退火算法、并行模拟退火算法、渐近收敛性、冷却进度表、模拟退火算法的应用、改进和变异、Boltzmann 机及其在组合优化中的应用.

本书可供计算机科学、计算数学、生命科学与医学等学科的高校师生、研究人员、工程技术人员阅读.

计算方法丛书

非数值并行算法(第一册)

模拟退火算法

康立山 谢 云 尤矢勇 罗祖华 著

责任编辑 林 鹏 李成香

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1994 年 4 月第 一 版 开本：850×1168 1/32

1998 年 7 月第三次印刷 印张：7 7/8

印数：3001—5000 字数：205 000

ISBN 7-03-003736-7/O · 659

定价: 14.00元

(如有缺页倒装,本社负责掉换。〈新欣〉)

《计算方法丛书》编委会

前　　言

非数值算法是基础科学、工程技术和管理科学等领域中常用的一类计算方法，如许多解组合优化问题的算法就是典型的非数值算法. 由于这些问题尤其是其中的 NP 完全问题本身所固有的计算复杂性，求其精确解的计算量往往随问题规模呈指数型增长，以致使用任何高速计算机都需耗费大量的时间，甚至根本无法实现. 因此，研究非数值计算的近似算法及其并行实现的途径具有十分重要的实际意义.

模拟退火算法是近年提出的一种适合解大规模组合优化问题，特别是解 NP 完全问题的通用有效近似算法. 它与以往的近似算法相比，具有描述简单、使用灵活、运用广泛、运行效率高和较少受初始条件限制等优点，而且特别适合并行计算. 因此不仅具有很高的实用价值，而且对推动并行计算的研究也有着重要的理论意义.

本书就是系统讨论模拟退火算法的一本专著. 书中详细介绍了模拟退火算法的数学和物理背景、理论基础及实现形式，并着重研究其用于解组合优化问题的具体方法和并行策略，最后介绍了可视为模拟退火算法大规模并行实现的 Boltzmann 神经计算机模型，将模拟退火算法的研究引入到一个新的境地. 书中汇集了作者在模拟退火算法的理论和应用上的最新研究成果和国内外一些专家学者的研究成果，所列实例和程序均经计算机实际运行通过，力图为读者提供一些模仿的范例.

本书内容自成体系，无需太多预备知识. 当然，懂得一些组合数学和计算机编程的基础知识对阅读是有帮助的. 本书可供高等学校计算数学和计算机科学等专业的学生及研究生学习，也可供理工科其它专业和管理专业的师生参考，还可供利用计算机从事

优化和管理工作的科技人员阅读.

本书是在武汉大学并行计算研究小组集体研究的基础上完成的. 在撰写过程中,得到了该小组全体同仁的大力支持和指导,并引用了他们的许多研究成果. 所以本书实际上是该小组集体智慧的结晶. 在此谨向这些同志致以衷心的感谢.

本书由康立山指导并最后定稿;谢云提出撰写提纲,撰写第五至八章,并统一全书稿;尤矢勇撰写第一、二、四章;罗祖华撰写第三章.由于作者水平有限,加之本书是国内第一部系统讨论模拟退火算法的中文书籍,难免有错误和不足之处,恳请广大读者指正.

本项目的研究得到了国家863高技术计划、国家自然科学基金和国防科工委"八五"预研计划的资助,国家自然科学基金委员会和中国科学院科学出版基金专家委员会将本书特列为国家自然科学基金委员会优秀成果专著出版基金和中国科学院科学出版基金共同资助项目,科学出版社林鹏同志对本书的出版给予了大力支持与帮助. 在此一并致谢!

作　　者

1993 年 9 月

目　录

第一章 引 论

在管理科学、计算机科学、分子物理学和生物学以及超大规模集成电路（VLSI）设计、代码设计、图象处理和电子工程等科技领域中，存在着大量组合优化问题．其中许多问题如货郎担问题、图着色问题、设备布局问题以及布线问题等，至今没有找到有效的多项式时间算法．这些问题已被证明是 NP 完全问题[1]．

用最优算法如线性规划求 NP 完全问题的最优解，需要问题规模的指数阶时间，在问题规模增大时，往往由于计算时间的限制而丧失可行性．用近似算法如贪心法求解 NP 完全问题，在多项式界的时间里，只能给出近似最优解．

本章介绍组合优化问题和计算复杂性理论的基本概念，并结合几个组合优化的 NP 完全问题实例，介绍其近似算法．最后，在引入邻域结构概念的基础上，介绍一种通用的近似算法——局部搜索算法．

§1 组合优化问题

组合优化问题的目标是从组合问题的可行解集中求出最优解．本书研究那些可以用数学语言精确描述的组合优化问题，并假定其可行解集是有限的或可数无限的，同时解的质量可以量化，因而可以比较不同解间的质量差异．

1.1 组合优化问题的基本概念

优化问题有三个基本要素：变量、约束和目标函数．在求解过程中选定的基本参数称为变量，对变量取值的种种限制称为约束，表示可行方案衡量标准的函数称为目标函数．

货郎担问题（TSP）是组合优化中最为著名的问题，它易于陈述而难于求解．自 1932 年 K. Menger 提出以来，已引起许多数学家的兴趣，但至今尚未找到有效的求解方法．由于货郎担问题综合了一大类组合优化问题的典型特征，下面以它为例说明组合优化问题的基本概念．

例 1.1　货郎担问题（TSP）

给定 n 个城市和每两个城市间的距离．一个货郎自某一城市出发巡回售货，问这个货郎应该如何选择路线，使每个城市经过一次且仅一次，并且路径长度最短．

设 $D=[d_{ij}]$ 是距离矩阵，其元素 d_{ij} 表示城市 i,j 间的距离．则对变量 D 的约束是：

（1）每个元素是非负整数，即

$$d_{ij}\geqslant 0,\quad 1\leqslant i,j\leqslant n;$$

（2）对角线上的元素为 0，即

$$d_{ii}=0,\quad 1\leqslant i\leqslant n;$$

（3）是对称矩阵，即

$$d_{ij}=d_{ji},\quad 1\leqslant i,j\leqslant n;$$

（4）任意三个元素满足三角不等式，即

$$d_{ij}+d_{jk}\geqslant d_{ik},\quad 1\leqslant i,j,k\leqslant n.$$

TSP 的一个解可表述为一个循环排列

$$\pi=(\pi_1,\pi_2,\cdots,\pi_n),$$

它也可表示为

$$\pi_1\rightarrow\pi_2\rightarrow\cdots\rightarrow\pi_n\rightarrow\pi_1$$

的一条路径，$\pi_i(1\leqslant i\leqslant n)$ 是该路径中第 i 个经过的城市．显然，满足

$$\pi_i\neq\pi_j,\quad 若\ i\neq j$$

的解才是可行解．所有可行解的集合构成解空间 S，即

$$S=\{n\ 个城市的所有循环排列\}.$$

解空间的规模为 $|S|=\dfrac{(n-1)!}{2}$．路径长度

$$f(\pi)=\sum_{i=1}^{n} d_{\pi_i,\pi_{i+1}}, \quad (约定\ \pi_{n+1}=\pi_1)$$

是货郎担问题的目标函数．TSP 的目标是使路径长度最短，即使目标函数 $f(\pi)$ 最小．

下面给出组合优化问题的定义．

定义 1.1　组合优化问题是在给定的约束条件下，求目标函数最优值（最小值或最大值）的问题．组合优化问题的一个实例可以表示为一个对偶 (S,f)，其中解空间 S 为可行解集，目标函数 f 是一个映射，定义为

$$f:S\rightarrow R.$$

求目标函数最小值的问题称为最小化问题，记为

$$\min f(i),\quad i\in S; \tag{1.1.1}$$

求目标函数最大值的问题称为最大化问题，记为

$$\max f(i),\quad i\in S. \tag{1.1.2}$$

显然，只要改变目标函数的符号，最小化问题与最大化问题就可以等价转换．

使目标函数取最优值的解称为最优解（整体最优解），定义为：

定义 1.2　设 (S,f) 是组合优化问题的一个实例，$i_{\mathrm{opt}}\in S$．称 i_{opt} 为最小化问题

$$\min f(i),\quad i\in S$$

的最优解，若

$$f(i_{\mathrm{opt}})\leqslant f(i),\ 对所有\ i\in S\ 成立.$$

在组合优化问题的近似算法中，还要涉及邻域和局部最优解的概念，我们将在 § 3 中定义．

1.2　几个组合优化问题的数学描述

例 1.2　0-1 背包问题

一个旅行者要从 n 种物品中选取 b 公斤重的物品，每种物品至多选一件．问这个旅行者应该怎样选取，使所选物品的总价值最

大.

设第 i 种物品的重量为 w_i，价值为 c_i，$i=1,2,\cdots,n$. 则问题是

$$\max z = \sum_{i=1}^{n} c_i x_i,$$

$$\text{s.t} \sum_{i=1}^{n} w_i x_i \leqslant b,$$

$$x_i = \begin{cases} 0 & \text{若第 } i \text{ 种物品未选上,} \\ 1 & \text{否则.} \end{cases}$$

s.t 右边的式子是约束条件(下同).

例 1.3 最大截问题

设 $G=(V,E)$ 是一个无向图，现把顶点集 V 分划为两个子集 V_0 和 $V_1=V\backslash V_0$，使 G 中那些端点分别在 V_0 与 V_1 中的边集有最大的权和.

设 $\delta(V_0,V_1)$ 是一个分划，$w(\{u,v\})$ 表示边 $\{u,v\}$ 的权，则问题是

$$\max f(V_0,V_1) = \sum_{\{u,v\}\in\delta(V_0,V_1)} w(\{u,v\}),$$

$$\text{s.t} \quad \delta(V_0,V_1) = \{\{u,v\}\in E \mid u\in V_0 \wedge v\in V_1\}.$$

例 1.4 图着色问题

设 G 是一个无环图，对 G 的每个顶点着色，使任意两个相邻的顶点都有不同的颜色，要求所用颜色数最少.

图 $G=(V,E)$ 中着同一种颜色的顶点集是 G 的一个独立集. 若 G 的顶点集 V 可划分为 k 个独立集 $V_1,V_2,\cdots,V_k$，则 G 是 k 可着色的. 因此图着色问题等价于找一个映射 $f:V\to\{1,2,\cdots,k\}$. 设 Δ 是 G 的最大度数，$\chi(G)$ 是 G 的色数(使 G 最优着色的颜色数)，则问题是

$$\min \ k, \chi(G) \leqslant k \leqslant \Delta + 1,$$

$$\text{s.t} \quad \forall u,v\in V:\{u,v\}\in E \Longrightarrow f(u) \neq f(v).$$

§2 计算复杂性与 NP 完全问题

算法可解问题在实践中不一定是可解的，因为求解该问题的算法可能需要极长的运行时间与极大的存贮空间，以至根本不可能在现有计算机上实现．算法对时间和空间的需要量称为算法的时间复杂性和空间复杂性．问题的时间复杂性是指求解该问题的所有算法中时间复杂性最小的算法的时间复杂性．类似地，可以定义问题的空间复杂性．按照计算复杂性理论研究问题求解的难易性,可把问题分为 P 类、NP 类和 NP 完全类[2]。

2.1 计算复杂性的基本概念

算法或问题的复杂性一般是问题规模 n 的函数，时间复杂性记为 $T(n)$，空间复杂性记为 $S(n)$． 货郎担问题中的城市数、0-1 背包问题中的物品数以及最大截问题和图着色问题中图的顶点数或边数都是刻划问题规模的特征数.

在算法分析和设计中,沿用实用性的复杂性概念,即把求解问题的关键操作,如加法、乘法、比较等运算指定为基本操作,然后把算法执行基本操作的次数定义为算法的时间复杂性；算法执行期间占用的存贮单元则定义为算法的空间复杂性．在组合优化问题中,比较和搜索常被指定为基本操作． 基本操作的执行次数随问题规模的增长以一定方式增长.

在分析复杂性时,可以求出算法的复杂性函数 $f(n)$，也可以方便地用复杂性函数主要项的阶 $O(f(n))$ 表示． 若算法 A 的时间复杂性是 $T_A(n)=O(p(n))$，$p(n)$ 是 n 的多项式函数，则称算法 A 为多项式时间算法，J. Edmonds 称之为好的算法． 时间复杂性不能囿于多项式时间的算法统称为指数时间算法．如对 n 个元素,二分搜索一个元素的时间复杂性是 $O(\log n)$， 堆排序的时间复杂性是 $O(n\log n)$，它们是好的算法． 而用动态规划解货郎担问题的时间复杂性是 $O(n^2 2^n)$，用回溯法解图 k 着色问题的

时间复杂性是 $O(nk^n)$[3]，它们是指数时间算法.

问题复杂性的形式定义可用图灵机（Turing Machine）计算模型给出. 如果一个问题有解它的多项式时间 DTM（确定型图灵机)程序,则称该问题属于P类. P类问题是指具有多项式时间算法的问题类. 如果一个问题有解它的多项式时间 NTM（非确定型图灵机)程序，则称该问题属于 NP 类. NP 类问题是指可在多项式时间里检验的问题类，至今尚未找到多项式时间算法. NP 完全问题是 NP 类中最难的问题. Cook 定理奠定了 NP 完全理论的基础,参见文献[1].

算法的时间复杂性对计算机的解题能力(速度和规模)有重大影响.以货郎担问题为例,如前所述,可能的路径数等于$\frac{(n-1)!}{2}$.

若以路径间的比较为基本操作，则需进行的基本操作数是

$$\frac{(n-1)!}{2}-1.$$

用运算能力为 1 M flops（每秒一百万次浮点运算）的计算机进行求解，在 $n=10$ 时只需 0.18 s. 而在 $n=20$ 时，需用 1929 年才能找到最优解. 再以 0-1 背包问题为例,物品数为 n 时有 2^n 个解(含不可行解),找出最优解需进行 2^n-1 次比较运算. $n=10$ 时只需 1 ms，而当 $n=60$ 时,需用 366 世纪!

表 1.1　时间复杂性对解题速度的影响

$T(n)$	解题速度		
	$n=10$	$n=30$	$n=60$
n	0.01ms	0.03 ms	0.06ms
n^2	0.1 ms	0.9 ms	3.6 ms
n^3	0.1 s	24.3s	13.0 min
2^n	1.0 ms	17.9min	366.0世纪
3^n	0.059s	6.5 年	1.3×10^{13}世纪

注. 表中数据是用 1 Mflops 计算机算出的.

表 1.1 和表 1.2 给出算法时间复杂性对解题速度和规模的影响. n 是问题规模.

表 1.2 时间复杂性对解题规模的影响

$T(n)$	解题规模		
	现有计算机	速度提高 100 倍	速度提高1000倍
n	N_1	$100N_1$	$1000N_1$
n^2	N_2	$10N_2$	$31.6N_2$
n^3	N_3	$2.5N_3$	$3.98N_3$
2^n	N_4	$N_4+6.64$	$N_4+9.97$
3^n	N_5	$N_5+4.19$	$N_5+6.29$

2.2 NP 完全问题

NP 完全问题是 NP 类中最难的问题，其涵义是只要有一个 NP 完全问题存在多项式时间算法，则整个 NP 类问题都存在多项式时间算法，也即证明了 P = NP 的命题[1].

现已证明的 NP 完全问题有上千个，但没有找到任一问题的多项式时间算法. 因此，计算科学家们大都认为 NP 完全问题不存在有效的多项式时间算法，即猜想 P ≠ NP，但未能证明.

上节列举的四个组合优化问题中，货郎担问题（注意：例 1.1 说明的是对称 TSP）是不属于 NP 完全的 NP 难题[2]，其余三个均是 NP 完全问题[3]. NP 难题与 NP 完全问题有同等的难度.

许多 NP 完全问题具有重大的实际意义，需要找出兼顾解的质量以及运行时间的较好算法. 一种方法是设计平均性态良好的概率算法，这种算法在多项式时间里几乎总是产生最优解，但在最坏情况下，仍然需要指数界的时间. 另一种方法是设计求近似最优解的近似算法，这种算法采用的策略和启发方式简单、直接，而且对于某些问题产生了意想不到的好结果. 下面介绍几种求解组合优化问题的近似算法.

2.3 NP 完全组合优化问题的近似算法

一、货郎担问题的近似算法

货郎担问题有很多近似算法，主要的方法有最近邻法、最近插入法、最小支撑树法及局部搜索法. §4 专门讨论局部搜索法，这里只介绍最近邻法.

算法 1.1　货郎担问题的近似算法 NN

(1) 任选一个城市 π_1 为出发点；

(2) 根据与出发城市距离最近的原则，在 π_1 以外的 $n-1$ 个城市中选取第二个经过城市 π_2；

(3) 再以 π_2 为出发城市，在尚未经过的城市中按最近邻原则选取 π_3，如此重复，直至所有城市都被经过，最后到达的城市是 π_n；

(4) 从 π_n 返回 π_1，构成一条路径.

用最近邻法得到的近似解，在最好情况下可能就是最优解；而在最坏情况下可能是最优解结果的好几倍(与城市数 n 有关).

令 π_{opt} 表示最优解，其对应路径长为 $|\pi_{opt}|$；π_{NN} 表示最近邻法得到的近似解，其对应路径长为 $|\pi_{NN}|$. 则在距离矩阵满足三角不等式时，有

$$2|\pi_{NN}| \leqslant (\lceil \log_2 n \rceil + 1)|\pi_{opt}|. \qquad (1.2.1)$$

特别地，对于任意的 $m \geqslant 3$，存在一个 $n = 2^m - 1$ 阶的矩阵 D，使最优路径长为 n，又使最坏路径长

$$|\pi_{NN}| > \frac{1}{3}\left(\log_2(n+1) + \frac{4}{3}\right)|\pi_{opt}| \qquad (1.2.2)$$

成立，即在距离矩阵不满足三角不等式(即非对称 TSP)时，最近邻法的最坏性态可能变得更差[1].

下面是满足(1.2.2)式关系的一个实例。

例 1.5　城市数 $n = 7$，距离矩阵是

$$D = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ \\ \end{array} \!\!\begin{array}{c} \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 1 \\ 1 & 0 & 1 & 2 & 2 & 3 & 3 \\ 1 & 1 & 0 & 1 & 2 & 3 & 3 \\ 2 & 2 & 1 & 0 & 1 & 2 & 2 \\ 2 & 2 & 2 & 1 & 0 & 1 & 1 \\ 2 & 3 & 3 & 2 & 1 & 0 & 1 \\ 1 & 3 & 3 & 2 & 1 & 1 & 0 \end{bmatrix} \\ \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \end{array},$$

则最优路径长 $|\pi_{\mathrm{opt}}| = n = 7$. 而从城市 1 出发，用最近邻法可能产生的最坏路径长

$$|\pi_{\mathrm{NN}}| = 11,$$

$$|\pi_{\mathrm{NN}}| > \frac{1}{3}\left(\log_2(n+1) + \frac{4}{3}\right)|\pi_{\mathrm{opt}}| = \frac{91}{9}.$$

见图 1.1 所示.

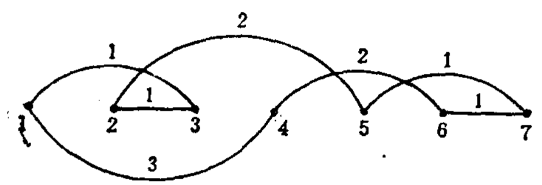

图 1.1 最近邻法产生的最坏路径

最近邻法在确定路径的过程中，共需进行 $\frac{1}{2}(n-1)(n-2)$ 次比较，因此其时间复杂性是 $O(n^2)$.

二、0-1 背包问题的近似算法

0-1 背包问题的近似算法基于贪心法. 贪心法是一种最直接的设计技术，能应用于多种问题. 这些问题的一般特征是：有 n 个输入和一组约束条件；满足约束条件的任一输入的子集都是可行解；要在可行解集中找出最优解.

贪心法的特点是"步步为营"，即每一步只在可选输入中选取一个满足约束条件的输入来构造一个可行解，实现某个优化测度（目标函数或别的测度）下的最优. 后继步对前趋步的结果进行扩