运筹与管理科学丛书 3

蚁群优化算法

马良 朱刚 宁爱兵 著

科学出版社
www.sciencep.com

运筹与管理科学丛书 3

蚁群优化算法

马 良 朱 刚 宁爱兵 著

科 学 出 版 社

北 京

内 容 简 介

本书围绕蚁群算法这一来自昆虫世界的优化思想，对其基本原理、核心步骤及其在最优化相关领域的实现作了详细介绍. 主要内容包括蚁群算法基本原理、蚁群算法在 TSP 及其扩展问题求解中的应用、蚁群算法在 VRP 及其扩展问题求解中的应用、蚁群算法在最优树问题求解中的应用、蚁群算法在整数规划问题求解中的应用、一般连续优化问题的蚁群算法以及多目标蚁群算法等. 书中还给出了一些主要算法的 Delphi 程序实现源代码，可供参考或修改使用.

本书可供运筹学、管理科学、系统工程、计算机科学等有关专业的高校师生、科研人员和工程技术人员阅读参考.

图书在版编目（CIP）数据

蚁群优化算法 / 马良，朱刚，宁爱兵著. 北京：科学出版社，2008.2
（运筹与管理科学丛书；3）
ISBN 978-7-03-020568-1

Ⅰ.蚁… Ⅱ.①马… ②朱… ③宁… Ⅲ.智能控制-算法 Ⅳ.TP273

中国版本图书馆 CIP 数据核字（2007）第 189936 号

责任编辑：赵彦超 / 责任校对：陈玉凤
责任印制：赵德静 / 封面设计：陈 敬

科学出版社出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com
新蕾印刷厂印刷
科学出版社发行 各地新华书店经销

*

2008 年 2 月第 一 版 开本：B5(720×1000)
2008 年 2 月第一次印刷 印张：16 1/2
印数：1—3 000 字数：310 000

定价：43.00 元

《运筹与管理科学丛书》序

运筹学是运用数学方法来刻画、分析以及求解决策问题的科学. 运筹学的例子在我国古已有之，春秋战国时期著名军事家孙膑为田忌赛马所设计的排序就是一个很好的代表. 运筹的重要性同样在很早就被人们所认识，汉高祖刘邦在称赞张良时就说道："运筹帷幄之中，决胜千里之外."

运筹学作为一门学科兴起于第二次世界大战期间，源于对军事行动的研究. 运筹学的英文名字 Operational Research，诞生于 1937 年. 运筹学发展迅速，目前已有众多的分支，如线性规划、非线性规划、整数规划、网络规划、图论、组合优化、非光滑优化、锥优化、多目标规划、动态规划、随机规划、决策分析、排队论、对策论、物流、风险管理等.

我国的运筹学研究始于 20 世纪 50 年代，经过半个世纪的发展，运筹学队伍已具相当大的规模. 运筹学的理论和方法在国防、经济、金融、工程、管理等许多重要领域有着广泛应用，运筹学成果的应用也常常能带来巨大的经济和社会效益. 由于在我国经济快速增长的过程中涌现出了大量迫切需要解决的运筹学问题，因而进一步提高我国运筹学的研究水平、促进运筹学成果的应用和转化、加快运筹学领域优秀青年人才的培养是当今我们面临的十分重要、光荣、同时也是十分艰巨的任务. 我相信，《运筹与管理科学丛书》能在这些方面有所作为.

《运筹与管理科学丛书》可作为运筹学、管理科学、应用数学、系统科学、计算机科学等有关专业的高校师生、科研人员、工程技术人员的参考书，同时也可作为相关专业的高年级本科生和研究生的教材或教学参考书. 希望该丛书能越办越好，为我国运筹学和管理科学的发展做出贡献.

袁亚湘

2007 年 9 月

前　言

齐心协力搬运食物，是人们生活中见得最多的蚂蚁行为. 据说蚂蚁很爱卫生，经常对其巢穴进行大扫除，将垃圾堆在一起，然后拉到巢外. 每个蚂蚁只需对自己周围的环境作出适当的反应，整个群体就能完成一件在旁观者看来似乎是非常复杂的任务. 个体的行为简单、盲目而且带有随机性，整体的行为却连贯、流畅与一致. 蚂蚁的群体合作精神令人钦佩，它们的寻食、御敌、筑巢之精巧又令人惊叹.

于是，以蚂蚁和其他群居性动物为模型，计算机科学家开发了相互合作以解决复杂问题的软件代替人，如在繁忙的通信网络中重新安排通信源. 受蚂蚁分工合作（蚁后管生男育女、工蚁管干活、兵蚁管保卫）特点的启迪，人们设计了求解任务分配问题的蚁群算法，并应用于工厂中的汽车喷漆问题，从而提高了整体生产率. 有人以蚂蚁群体为蓝本设计出几个机器人共同推盒子的算法. 另外，一些公司开始研究人工蚂蚁，并用于管理公司的电话网，以及对用户记账收费等工作.

人类对蚂蚁观察和研究后发现，蚂蚁有能力在没有任何可见提示下找出从其窝巢至食物源的最短路径，并且能随环境的变化而变化，适应性地搜索新的路径，产生新的选择. 这种自催化行为的本质思想是一种正反馈机制，因此，有人将蚂蚁王国理解成一种增强型学习系统.

群体中的每一个蚂蚁看来都有它自己的安排和计划，但是，这些蚂蚁作为一个整体是有高度组织性的，把所有个体的活动综合成一个天衣无缝的整体似乎并不需要任何监督. 事实上，研究群居性昆虫行为的科学家们发现，昆虫群落一级上的合作基本上是自组织的：在许多场合中，个体之间的相互作用产生协调一致的行为. 尽管这些相互作用可能很简单（例如，一只蚂蚁也就是紧跟着另一只蚂蚁留下来的轨迹路线而已），但是它们合起来却可以解决棘手的问题（例如，从通往一个食物源的无数条可能路径中找出最短的一条），从一群群居性生物中产生出来的这样一种集体行为就被称为“群集智能”（swarm intelligence）.

蚁群算法这种来自生物界的随机搜索寻优方法目前已在许多方面表现出相当好的性能，其求解问题的领域也在进一步扩大. 本书对蚁群算法的阐述将主要聚焦在最优化领域（包括组合优化和连续优化），尤其是一些组合优化. 该分支中的许多问题都是迄今为止仍悬而未决的著名难题，具有极大的挑战性，诸如旅行商问题、度约束最小树问题、二次分配问题、图着色问题等所谓的 NP 难题. 鉴于计算困难是这类问题

的固有性质，因此，目前尚无法用有效算法精确求解. 但这些问题在现实领域中有着广泛的应用，因而寻找其实际而有效的算法就显得颇为重要. 近年来，一系列来自自然界的进化型算法被相继引入，其思想吸收了许多看似无关的其他学科中的概念和方法，典型的有模拟退火算法、遗传算法、禁忌搜索法、蚁群算法等. 本书主要就蚁群算法这种新的仿生类算法思想，对一系列不同的优化问题设计了相应的求解策略并在计算机上予以实现，获得了满意的效果.

结群而居的昆虫，如蚂蚁、蜜蜂等很早就引起了博物学家和艺术家们的极大兴趣，比利时诗人 Maurice Maeterlinck 就曾写道："是什么东西在支配着它们？是什么东西在维持秩序、预见未来、制定计划并保持平衡？……" 这的确是一些令人困惑不解的问题.

自然界的蚁群、鸟群、鱼群、羊群、牛群、蜂群等，其实时时刻刻都在给予我们以某种启示，只不过我们常常忽略了大自然对我们的最大恩赐！……

本书的出版得到了上海市高校青年科学基金（No.98QN28）、上海市曙光计划（No.2000SG30）、国家自然科学基金（No.70471065）、上海市重点学科建设（No.T0502）等项目的资助，在此谨致谢意.

同时，感谢所有被本书直接或间接引用其文献资料的同行学者.

感谢作者的研究生崔雪丽（博士后）、张瑾（博士）、金慧敏（硕士）、廖飞雄（硕士）、王洪刚（硕士）等人所做的大量工作.

作　者

2007 年 7 月 5 日

目　录

第 1 章　引　论

1.1　组合优化与计算复杂性

1.1.1　最优化问题与局部搜索

最优化问题似乎自然地分为两类：一类是连续变量的问题，另一类是离散变量的问题.

在连续变量的问题中，一般是求一组实数，或者是一个函数，因而常常被称为函数优化，一般可分为线性规划和非线性规划；在组合问题里，是从一个无限集或者可数无限集里寻找一个对象 —— 典型的是一个整数、一个集合、一个排列，或者一个图，这种离散变量的问题，往往被称为组合优化.

一般地，这两类问题有相当不同的特色，并且求解它们的方法也很不同.

在组合优化的研究中，从某种意义上说，我们是从它与连续优化间的分界线入手的.

定义 1.1　一个最优化问题的一个实例 (或例子) 是一对元素 (F,f)，其中 F 是一个集合或可行点的定义域，f 是费用函数 (目标函数) 或映射

$$f:F\to R^1.$$

问题是求一个 $x\in F$，使 $\forall y\in F$，有

$$f(x)\leqslant f(y)\quad (\text{不失一般性, 假定为最小化}).$$

这样一个点 x 称为给定实例的整体 (或全局) 最优解，或者在不引起混淆的情况下，简称为最优解.

定义 1.2　一个最优化问题，就是它的一些实例的集合 I.

非形式地说，在一个实例里，我们有输入数据以及用于求解的足够信息. 一个问题就是实例的总体，通常这些实例是用类似的方式产生的.

对某些问题的实例，求其整体最优解可能存在不可克服的困难，但是求一个解 x，使得在邻域 $N(x)$ 里存在没有比 x 更好的解是可能的. 在这个意义上，我们可以说这样的解 x 是最好的.

定义 1.3　给定一个最优化问题实例 (F,f) 的一个邻域 N，一个可行解 $x\in F$，若 $\forall g\in N(x)$，有 $f(x)\leqslant f(g)$，称 x 为关于 N 的局部最优解.

邻域函数是优化中的一个重要概念, 其作用是指导如何由一个 (组) 解来产生一个 (组) 新的解. 邻域函数的设计往往依赖于问题的特性和解的表达方式 (编码). 由于优化状态表征方式的不同, 函数优化和组合优化中的邻域函数的具体方式将明显存在差异.

函数优化中的邻域函数是在距离空间中通过附加扰动构造而成, 如

$$x' = x + \eta \cdot \zeta,$$

其中 x' 为新解, x 为旧解, η 为尺度参数, ζ 为满足某种概率分布的随机数或白噪声或混沌序列或梯度信息等. 显然, 采用不同的概率分布 (如高斯分布、柯西分布、均匀分布等) 或下降策略, 将实现不同性质的状态转移.

在组合优化中, 邻域函数也是基于在一点附近搜索另一个使目标函数下降的点的基本思想. 但上述距离邻域的概念已不再适用, 因此需对其进行重新定义.

定义 1.4　对于组合优化问题 (D, F, f), 其中 D 为所有解构成的状态空间, F 为 D 上的可行域, f 为目标函数, 则一个邻域函数可定义为一种映射, 即

$$N : x \in D \rightarrow N(x) \in 2^D,$$

其中, 2^D 表示 D 所有子集的集合, $N(x)$ 为 x 的邻域.

局部搜索法是基于贪婪思想, 利用邻域函数进行搜索的. 它通常可描述为：从一个初始解出发, 利用邻域函数持续地在当前解的邻域中搜索比它好的解. 若能找到如此的解, 则使之成为新的当前解, 然后重复上述过程; 否则结束搜索过程, 并以当前解作为最终解. 在搜索过程中, 始终向着离目标最接近的方向搜索.

局部搜索法可能落入局部最优点, 或由于步长和初始点选择不好而错过最优点, 在实际应用中可采用相应的修正方法. 如对于局部最优问题：每次并不一定选择邻域内最优的点, 而是依据一定的概率, 从邻域内选择一个点, 目标函数好的点, 被选中的概率也大, 而目标函数差的点, 被选中的概率则较小.

对于步长问题, 可以采用变步长方法, 在接近最优点附近, 采用某种策略改变步长.

对于起始点问题, 可以采用随机生成的一些初始点, 从每个初始点出发进行搜索, 找到各自的最优解, 再从这些最优解中选择一个最好的结果作为最终的结果.

在许多优化算法中, 尤其是那些实用算法, 局部搜索法 (或其修改形式) 往往作为一种改进策略被嵌入算法中, 从而提高算法的效果.

1.1.2　计算复杂性

组合优化研究离散现象中所出现的优化问题、性质与算法, 在工程技术、经济管理、计算机技术等方面有着广泛的应用.

组合优化问题一般是一个极小化 (或极大化) 问题, 它由下面三部分组成：

(1) 实例集合;

(2) 对每一个实例 I, 有一个有穷的可行解集合 $S(I)$;

(3) 目标函数 f, 它对每一个实例 I 和每一个可行解 $\sigma \in S(I)$, 赋以一个有理数 $f(I,\sigma)$. 如果 π 是极小化 (或极大化) 问题, 则实例 I 的最优解为这样一个可行解 $\sigma^* \in S(I)$, 使得对于所有 $\sigma \in S(I)$, 都有

$$f\left(I,\sigma^*\right) \leqslant f\left(I,\sigma\right) \quad (\text{或} f\left(I,\sigma^*\right) \geqslant f\left(I,\sigma\right)).$$

组合优化的主要内容是研究如何寻找适合解决某个实际离散问题的算法. 这里, 算法是指在有限步骤内求解某一问题所使用的一组定义明确的规则. 一个算法应该具有以下五个重要特征：

(1) 有限性：一个算法必须保证执行有限步之后结束;

(2) 确切性：算法的每一步骤必须有确切的定义;

(3) 输入：一个算法有 0 个或多个输入, 以刻画运算对象的初始情况, 所谓 0 个输入是指算法本身限定了初始条件;

(4) 输出：一个算法有一个或多个输出, 以反映对输入数据加工后的结果. 没有输出的算法是毫无意义的;

(5) 可行性：算法原则上能够精确的运行.

衡量一个算法是否优良的标准, 是看这个算法解决问题所花费的时间和空间有多大. 算法对时间和空间的需求量称为算法的时间复杂性和空间复杂性. 所谓的“计算复杂性”, 通俗地说, 就是用计算机求解问题的难易程度.

若一个算法的复杂性 $f(n)$ 是问题规模 n 的多项式函数, 则称这个算法是有效的或有多项式界的, 或简称该算法是“好”算法或多项式算法. 那些算不上是“好”算法的典型代表之一是指数算法, 即复杂性 $f(n)$ 是 n 的指数函数的算法.

算法的复杂性对计算机的求解能力有着重大影响, 甚至起着决定性的作用. 若要依据难度去研究各种计算问题之间的联系, 按复杂性把问题分成不同的类, 那么首先需要一个计算模型, 用以说明哪种操作或步骤是许可的, 以及它们的代价有多大. 常用的计算模型有图灵机、随机存取机、组合线路等. 通过这些计算模型可以研究问题复杂性的上界和下界, 或寻求最佳算法.

图灵机是一种抽象的计算模型, 由英国数学家图灵 (Turing) 于 1936 年提出. 由于图灵机在计算能力上等价于数字计算机, 故利用图灵机可以研究计算机的能力和局限性. 一台多带图灵机由一个有限状态控制器和 k 条读写带 $(k \geqslant 1)$ 组成. 这些读写带的右端无限长, 每条带都从左到右划分为方格, 每个方格可以存放一个带符号, 且带符号的总数是有限的. 每条带上都有一个由有限状态控制器操纵的读写头

或称为带头, 可以对这 k 条带进行读写操作. 有限状态控制器在某一时刻处于某种状态, 且状态总数是有限的.

多带图灵机如图 1.1 所示.

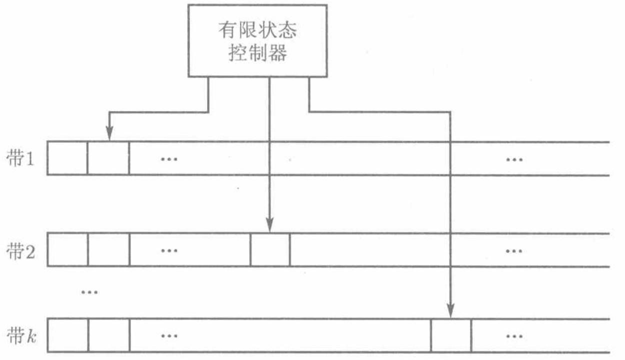

图 1.1　多带图灵机

根据有限状态控制器的当前状态及每个读写头读到的带符号, 图灵机的一个计算步实现下面三个操作之一或全部:

(1) 改变有限状态控制器中的状态;

(2) 消除当前读写头下的方格中原有带符号并写上新的带符号;

(3) 独立地将任何一个或所有读写头, 向左移动一个方格 (L) 或向右移动一个方格 (R) 或停在当前单元不动 (S).

形式上, 一个图灵机可记成一个七元组: $M = (Q, T, I, \delta, b, q_0, q_j)$, 其中,

Q 是有限状态的集合;

T 是有限带符号的集合;

I 是输入符号的集合, $I \subset T$;

b 是唯一的空白符, $b \in T - I$;

q_0 是初始状态;

q_j 是终结 (或接受) 状态;

δ 是下一动作函数, 是从 $Q \times T^k$ 到 $Q \times (T \times \{L, R, S\})^k$ 的映射.

若用一台图灵机来识别语言, 图灵机的带符号集 T 应当包括这个语言的字母表中的全体符号和一个空白符 b, 也许还有其他符号. 开始时, 第一条带上放有一个输入符号串, 从最左的方格起每格放一个输入符号, 这条带上其余方格都是空白. 其他各带上也全是空白. 所有读写头都处在各带左端第一个方格上. 当且仅当图灵机从指定的初始状态 q_0 开始, 经过一系列计算步后, 最终进入终止状态 (或接受状态) q_j 时, 称图灵机接受这个输入符号串. 这台图灵机所能接受的所有输入符号串的集合, 称作这台图灵机识别的一个语言.

图灵机 M 的时间复杂性 $T(n)$ 是它处理所有长度为 n 的输入所需的最大计算步数. 如果对某个长度为 n 的输入, 图灵机不停机, $T(n)$ 对这个 n 值无定义.

图灵机的空间复杂性 $S(n)$ 是它处理所有长度为 n 的输入时, 在 k 条带上所使用过的方格数的总和. 如果某个读写头无限地向右移动而不停机, $S(n)$ 也无定义.

许多算法都是多项式时间算法, 即对规模为 n 的输入, 算法在最坏情况下的计算时间为 $O(n^k)$, k 为一个常数. 但是有一些问题, 虽然可以用计算机求解, 但是对任意常数 k, 它们都不能在 $O(n^k)$的时间内得到解答. 在图灵机计算模型下, 这类问题的计算复杂性至今未知. 为研究这类问题的计算复杂性, 人们提出了一个能力更强的计算模型, 即非确定性图灵机计算模型, 简记为 NDTM (non-deterministic Turning machine). 在这个计算模型下, 许多问题就可以在多项式时间内求解.

确定性图灵机, 简记为 DTM (deterministic Turing machine), 其移动函数 δ 是单值的, 即对于 $Q \times T^k$ 中的每个值, 当它属于 δ 的定义域时, $Q \times (T \times \{L, R, S\})^k$ 中只有唯一的值与之对应.

一个 k 带的非确定性图灵机 M 也是一个 7 元组: $(Q, T, I, \delta, b, q_0, q_j)$. 与确定性图灵机不同的是, 非确定性图灵机允许 δ 具有不确定性, 即对于 $Q \times T^k$ 中的每一个值 $(q : x_1, x_2, \cdots, x_k)$, 当它属于 δ 的定义域时, $Q \times (T \times \{L, R, S\})^k$ 中有唯一的一个子集 $\delta(q : x_1, x_2, \cdots, x_k)$ 与之对应. 可以在 $\delta(q : x_1, x_2, \cdots, x_k)$ 中随意选定一个值作为它的函数值, 这个不确定的函数 δ 仍称为移动函数.

在采用图灵机作为标准的计算工具的情况下, 可以形式化地定义如下几类计算问题:

(1) P(polynomial) 类问题. 由确定型图灵机 (DTM) 在多项式时间内可解的一切判定问题所组成的集合 (所谓判定问题是指给定问题实例及整数 L, 问是否存在目标值不超过 L 的可行解).

(2) NP (non-deterministic-polynomial) 类问题. 由非确定型图灵机 (NDTM) 在多项式时间内可计算的判定问题所组成的集合.

由于确定型图灵机在任一状态一次只能做一种运算, 而非确定型图灵机在同一时刻里可以同时做多种运算, 即每步有多个结果. 因此 NP 问题的含义是: 用多台处理机并行计算, 这个问题存在着多项式算法, 而改用单台处理时不一定能找到多项式算法. 因此, 确定性图灵机可看作是非确定图灵机的特例, 由此, 可在多项式时间内被确定性图灵机接受的语言也可在多项式时间内被非确定性图灵机接受, 故 $\mathrm{P} \subset \mathrm{NP}$.

(3) NP 完全问题. 如果判定问题 $\pi \in \mathrm{NP}$, 并且对所有其他判定问题 $\pi' \in \mathrm{NP}$, 都有 π' 多项式变换到 π, 则称判定问题 π 是 NP 完全的 (NP-complete).

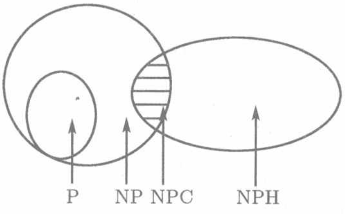

图 1.2 四类问题的关系

所有的 NP 完全问题组成的集合称为 NP 完备类, 这些问题在算法复杂性上是等价的, 它们构成一个等价类, 记为 NPC.

(4) NP 难题 (NP-hard). 若所有的 NP 问题都可以转换为 π, 则称 π 为 NP 难题, 简记为 NPH. π 是 NP 难题并不要求 π 属于 NP 类.

上述四类问题的关系可用图 1.2 表示.

1.1.3 典型组合优化难题

组合优化问题常常涉及排序、分类、筛选等, 是运筹学的一个重要分支. 典型的组合优化难题有旅行商问题 (traveling salesman problem, TSP)、图着色问题 (graph coloring problem, GCP)、工件排序问题 (job-shop scheduling problem, JSP)、二次分配问题 (quadratic assignment problem, QAP)、度约束最小树问题 (degree-constrained minimum spanning tree problem, DCMSTP) 等.

1. 旅行商问题

旅行商问题 (简记 TSP), 又称旅行推销员问题、货郎担问题等, 是指有一货物推销员从城市 1 出发到城市 2, 3, $\cdots$, n 去推销货物, 然后回到城市 1, 问应怎样选择一条总行程最短的路线 (各城市间距离 d_{ij} 为已知). 由于推销员的每条路线可用以 1 开始的一个排列来表示, 因此所有可能的路线最多有 $(n-1)!/2$ 条. 这样, 假如用穷举法来解决此问题, 那么即使 n 不太大, 也是很难求解的.

2. 图着色问题

图着色问题 (简记 GCP) 要求对给定图 G 找出最少的顶点着色数, 使得图中任何两个关联顶点都具有不同的颜色.

记图的最大顶点度数为 Δ, 则图的最小着色数以 $\Delta+1$ 为上界.

3. 工件排序问题

在工件排序问题 (简记 JSP) 中, 有 n 个相互独立的任务 $J_1, J_2, \cdots, J_n$, 所需加工时间分别为 $T_1, T_2, \cdots, T_n$, 并均可由 m 台机器 $M_1, M_2, \cdots, M_m$ 中的任一台完成, 且每台机器一次仅可完成一项任务, 现要找最优任务安排, 使得完成所有任务的时间最少.

当 m 不超过 2 时, 该问题可用简单有效的方法进行求解. 但一般情形下, 此问题仍是一个难以处理的 NP 难题.

4. 二次分配问题

二次分配问题 (简记 QAP) 的原始提法为: 已知有 n 个位置点和 n 家工厂, 各

位置点之间的距离矩阵设为

$$D = [d_{ij}]_{n\times n}.$$

各工厂之间的运输量矩阵为

$$F = [f_{ij}]_{n\times n}.$$

现要将这 n 家工厂建造在这 n 个位置点上, 使得总费用最小. 其中, 工厂 i 建造在位置点 k 且工厂 j 建造在位置点 l 的费用表示为 $f_{ij}\cdot d_{kl}$. 该问题由于目标函数的非线性而变得异常困难.

5. *度约束最小树问题*

度约束最小树问题 (简记 DCMSTP) 源于网络优化中的最小生成树问题. 其不同点在于对生成树的各顶点度数加上了一定的限制条件, 即不得超过预先给定的数值, 组合含义是从所有的生成树中找出顶点度符合约束条件且权总数最小的生成树.

该问题的求解难度随各顶点度约束的不同而不同: 当约束至少为 $n-1$ 时, 即为一般的最小生成树问题; 当约束为 2 时, 即为著名的 TSP 问题.

以上这些问题都是所谓的 NP 难题, 已经发现和证明的等价问题目前达数百个之多, 它们抵御了两代数学家们的顽强攻击. 目前, 人们普遍认为: 这些 NP 难题不能用任何已知的多项式算法求解; 若任何一个有多项式算法, 则所有这些问题都有多项式算法. 因此, 许多人猜测任何 NP 难题都没有多项式算法, 但至今无人证明.

1.2 来自自然界的几类优化方法

1.2.1 概述

复杂系统具有严重的不确定性, 环境、信息以及任务的复杂性使得传统的基于数学模型的控制方法难以奏效. 近年来, 人们在神经网络、人工智能、模糊逻辑和进化计算方面的最新研究成果已成为复杂非线性系统建模、控制和优化的主要工具. 随着各方面研究的深入, 模糊系统、神经网络和进化计算在这方面应用所取得的成效比较显著, 特别是这些技术的相互交叉与结合所产生的系统比单一技术所产生的系统更为有效.

对低级动物而言, 其生存、繁衍是一种智能. 为了生存, 它必须表现出某种适当的行为, 如觅食、避免危险、占领一定的地域、吸引异性以及生育和照料后代等. 因此, 从个体的角度看, 生物智能是动物为达到某种目标而产生正确行为的生理机制. 于是, 智能水平高的个体比智能水平低的个体更容易找到食物, 也更知道用伪装

的办法来减少危险. 此外, 在自然界, 生物智能还表现在生物的群体行为. 大多数动物是以一定数量的个体组合起来行动的, 这种组合使生物增强了感觉危险存在的能力和抵御外来侵犯的能力. 除了生存、繁衍等智能行为外, 自然界还存在更高层次的生物智能. 自然界智能水平最高的生物就是人类自身, 不但具有很强的生存能力, 而且具有感受复杂环境、识别物体、表达和获取知识以及进行复杂的思维推理和判断的能力.

参照生物智能, 人们将广义的智能定义为：智能是个体或群体在不确定的动态环境中作出适当反应的能力, 这种反应必须有助于它 (们) 实现其最终的行为目标.

根据智能的定义, 不难发现, 除生物系统外, 许多机器系统也表现出一定的智能行为. 因此, 除了生物智能外, 还存在人工智能 (AI). 人工智能到目前为止尚无统一的定义, 其创始人之一 Simon 认为：人工智能的研究目的是学会怎样编制计算机程序来完成智能的行为, 并认识人类是如何完成这些智能行为的. 另一创始人 Minsky 则认为：人工智能的研究, 一方面是帮助人思考, 另一方面使计算机更加有用. 人工智能的权威 Feigenbaum 也指出：只告诉计算机做什么, 而不需告诉它怎么做, 计算机就能完成工作, 便可以说它有智能了.

很明显, 虽然对人工智能的说法各不相同, 共同的认识是, 人工智能系统必须具备推理、学习和联想三大功能.

计算智能系统是在神经网络、模糊系统、进化计算三个分支发展相对成熟的基础上, 通过相互之间的有机融合而形成的新的科学方法, 也是智能理论和技术发展的崭新阶段. 这些不同的成员方法从表面上看各不相同, 但实际上它们是紧密相关、互为补充和促进的. 按照 Bezdek 的观点, 计算智能是基于操作者提供的数据, 而传统人工智能是基于 “知识”. 于是, 计算智能系统可定义为：当一个系统仅仅处理底层的数据, 具有模式识别的部分, 并且不使用 AI 意义中的知识, 那么这个系统便是计算智能系统. 这样一个系统表现出如下特点：

(1) 具有计算的适应性;

(2) 具有计算误差的容忍度;

(3) 接近人处理问题的速度;

(4) 近似人的误差率.

近年来, 特别引人注目的是一类来自自然界的进化型算法, 其思想吸收了许多看似无关的其他学科中的概念和方法, 最典型的有遗传算法、模拟退火算法、禁忌搜索法、人工神经网络、蚁群系统、微粒群算法等, 以及由这些算法与其他算法相结合而形成的一些混合型方法. 这些方法构成了当前优化范畴中一组来自跨学科领域的别具特色的寻优策略, 由于它们往往具备跳出局部极值点的潜在能力以及广泛的适应性, 因而受到了各个学科分支的关注和重视.