

第一篇 进化优化引论

第 1 章 绪 论

你且问走兽，走兽必指教你；又问空中的飞鸟，飞鸟必告诉你；或与地说话，地必指教你，海中的鱼也必向你说明。

约伯记 12:7-9

本书讨论求解优化问题的方法。特别地，我们¹ 讨论用于优化的进化算法。本书包括一些数学理论，但不应该把它看成是数学课本。它更多的是一本关于工程或应用于计算机科学的书。本书中所有优化方法都为了最终能用软件实现。本书旨在以最清楚但又严格的方式介绍进化优化算法，同时为读者提供足够先进的材料和参考文献，让他们准备好为进化算法添砖加瓦。

本章概览

本章在 1.1 节概述本书用到的术语。在 19 页的缩写词表可能对读者有用。1.2 节说明我为何决定撰写这本关于进化算法的书，希望它能达到的目标，以及与已有的进化算法专著的不同之处。1.3 节讨论读者需要先修的课程。1.4 节讨论布置习题的理念，以及如何得到习题答案。1.5 节总结本书用到的数学符号。读者在遇到不熟悉的符号时要记得回到这一节，在写家庭作业和做研究时也会用到这一节。1.6 节概述本书。1.7 节为采用本书开课的教师提供了一些重要的忠告。这一节还就各章的重要性为教师提出了一些建议。

1.1 术 语

一些作者称进化算法为进化计算。这样能强调进化算法需要在计算机上实施。但进化计算也可能指不用于优化的算法；例如，最初的遗传算法并不是用于优化本身，而是想用来研究自然选择的过程（参见第 3 章）。本书针对的是进化优化算法，它比进化计算更具体。

另外一些人称进化算法为基于种群的优化。它强调进化算法一般是让问题的候选解种群随着时间的进化以得到问题的更好的解。然而，许多进化算法每次迭代只有单个候

¹本书采用常见的做法，我们这个词一般代指第三人。有时候也用我们代指读者和作者。在其他时候，我们代表在进化算法和优化领域里的教师和研究人員。由上下文可以清楚区分其具体含义。对我们这个词的使用无须过分解读；它只是一种书写风格并非要显示权威。

解(例如,爬山法和进化策略).进化算法比基于种群的优化更一般化,因为进化算法还包括单一个体的算法.

一些作者称进化算法为计算机智能或计算智能.这样做常常是为了区分进化算法与专家系统,在传统上专家系统一直被称为人工智能.专家系统模仿演绎推理,进化算法则模仿归纳推理.进化算法有时候也被看成是人工智能的一种.计算机智能是比进化算法更一般的词,它包括像神经网络、模糊系统,以及人工生命这样的一些技术.这些技术可应用于优化之外的问题.所以,从不同的角度看,进化算法可能比计算机智能更一般化或更具体.

软计算是与进化算法有关的另一个词.软计算与硬计算相对.硬计算指的是准确的、精确的、数值上严格的计算.软计算指的是不太准确的计算,比如人们在日常生活中的表现.软计算的算法针对那些困难的、带噪声的、多峰的和多目标的问题,计算出总体上还不错(但不准确)的解.所以,进化算法是软计算的一个子集.

其他作者称进化算法为由自然启发的计算或仿生计算.像差分进化和分布估计算法这些进化算法可能并非源于自然.像进化策略和反向学习这些进化算法与自然过程联系甚微.进化算法比由自然启发的算法更一般化,因为进化算法包括非仿生算法.

进化算法还常常被称为机器学习.机器学习研究由经验学到的计算机算法.但这个领域经常包括很多不是进化算法的算法.一般认为机器学习比进化算法更广,包括强化学习、神经网络、分簇、支持向量机以及其他方法.

一些作者喜欢将进化算法称为启发式算法.启发式来自 $\eta\nu\rho\lambda\sigma\kappa\omega$ 这个希腊文的单词,在英语中它译作 *eurisko*.这个词的意思是找到(find)或发现(discovery).英语中的感叹词 *eureka* 也来自这个词,当我们有新的发现或解决了一个问题时,用 *eureka* 来表示胜利.启发式算法是利用经验法则或常识来解决问题的方法.我们通常不会期望启发式算法能找到问题的最佳答案,只会期望它能找到与最优解“足够接近”的解.元启发式这个词被用来描述一系列启发式算法.本书讨论的进化算法,如果不是全部,其中的绝大多数也都可以用不同方式在许多不同的选项和参数下实施.所以,它们都可以称为元启发式.例如,所有蚁群优化的一系列算法就可以称为蚁群元启发式.

大多数作者把进化算法与群智能算法区别开来.群智能算法以自然中的群(例如,蚁群或鸟群)为基础.蚁群优化(第 10 章)和粒子群优化(第 11 章)是两个突出的群智能算法,许多研究人员坚持认为它们不应该被归为进化算法.一些作者则认为群智能算法是进化算法的一个子集.例如,粒子群优化的一位发明者将它称为进化算法 [Shi and Eberhart, 1999].因为群智能算法与进化算法有相同的执行方式,即,每次迭代都改进问题的候选解的性能从而让解的种群进化,因此我们认为群智能算法是一种进化算法.

虽然术语不够确切且与上下文相关,但在本书中进化算法指的是经过多次迭代让问题的解进化的算法.按照进化算法的生物学基础,它的一次迭代通常被称为代.不过进化算法的这个简单定义并不完美,举例来说,这个定义暗示梯度下降法也是一个进化算法,但没有谁会认可这个说法.所以,在进化算法领域中术语的不统一会让人困惑.一个算法是进化算法如果它通常被认为是进化算法,我们采用这个戏谑的定义.这个循环定义一开

始有些麻烦,但一段时间之后,像我们这样在这个领域工作的人就习惯了.自然选择被定义为适者生存,而适者被定义为最有可能的存活者.

1.2 又一本关于进化算法的书

关于进化算法已有许多好书,这就提出了一个问题:为什么又要写一本关于进化算法的书呢?撰写本书的原因是想提供一种教学的方法、观点和素材,其他书都没有这些内容.特别地,作者希望本书能够提供:

- 一种直截了当自下而上的方法,它能协助读者清楚理解书中讲述的进化算法,这种理解在理论上也是严格的.许多书在讨论进化算法的变种时像菜谱一样完全没有理论上的支撑.其他书读起来更像研究专著而不是课本,学工程的普通学生不能完全理解.本书尝试通过易于实现的算法,连同一些严格的理论以及有关折中的讨论来获得一个平衡.
- 书中的简单例子让读者能够直观地理解进化算法的数学、方程和理论.很多书介绍进化算法的理论,然后给出的例子或问题却并不适合直观理解.其实,完全有可能给出一些只需要纸和笔就能求解的简单例子和问题.这些简单的问题能让学生更直接地看到理论在实践中的作用.
- 从作者的网址¹能得到书中所有例子的基于 MATLAB®的源代码.别的课本也提供源代码,但它们常常不完整或已经过时,这会令读者困惑.那个网址上还有作者的电邮地址,作者热烈欢迎读者的反馈、评论、改进意见和纠错.当然,网页地址会过时,本书中算法的伪代码清单比任何具体的软件清单更长久.注意,书中的例子和 MATLAB 代码并不是高效的或有竞争力的优化算法;只是为了让读者理解算法的基本概念.严肃的研究或应用应该只在最初的时候依赖样本代码.
- 本书包括在大多数课本中没有的理论以及最近开发的进化算法.它们包括进化算法的马尔可夫理论模型、进化算法的动态系统模型、人工蜂群算法、基于生物地理学的优化、反向学习、人工鱼群算法、混合蛙跳、细菌觅食优化和其他算法.这些主题是对现状的补充,把它们放入书中不是想要与其他书一分高下.不过,本书也不是要调查有关进化算法研究的各方面现状.本书旨在就进化算法研究的许多领域做一个高层次的概述以帮助读者全面理解进化算法,并让读者对现状有一个良好的定位以便从事进一步的研究.

1.3 先修课程

一般来说,如果修读这门课的学生不编写进化算法软件就会一无所获.所以,足够的编程技能应该是一个先决条件.我在大学里为电气与计算机工程的学生讲授这门课时,对

¹ 见 <http://academic.csuohio.edu/simond/EvolutionaryOptimization-> 如果地址变了,用网络搜索应该很容易找到.

先修课程并没有具体的要求；高年级本科生和研究生都可以选修。不过，我以为高年级本科生和研究生都是好的程序员。

本书采用在代数、几何、集合理论和微积分中用到的标准数学符号，假设读者对这些科目已经很熟悉。因此，要理解本书的另一个先决条件是具有高年级本科生的数学水平。1.5 节描述数学符号。如果读者能够理解其中的符号，就很有希望看懂本书余下部分的内容。

本书理论性的章节（第 4 章，7.6 节，第 13 章的大部分，以及一些零散章节）需要理解概率论和线性系统理论。除非学生已修过这两门科目的研究生课程，不然很难理解那些材料。面向本科生的课程也许应该略过那些内容。

1.4 家庭作业

每一章末尾的习题供教师和学生灵活使用。它包括书面练习和计算机练习。书面练习旨在强化学生对理论的掌握，加深学生对概念的直观理解，并增进学生的分析技能。计算机练习旨在帮助学生增强研究的能力，学习如何将理论应用于在工业界通常会遇到的那类问题。要想精通进化算法，这两类习题都很重要。书面练习与计算机练习并没有严格的区别，更多的是一种模糊的划分。也就是说，一些书面练习可能会用到计算机，而利用计算机的练习又需要一些分析。教师可以基于自己的兴趣来布置与进化算法有关的作业。基于项目的长达一学期的作业常常很有意义。例如，让学生利用书中的进化算法去解决一些实际的优化问题，用每章的进化算法求解问题，并在学期末比较这些进化算法的性能以及它们的差异。

教师从出版商那里可以得到书中所有习题的答案（书面练习和计算机练习）。我们鼓励任课教师与出版商联系以了解有关习题答案的更多信息。为维护家庭作业的诚信，习题答案只会提供给任课教师。

1.5 符 号

遗憾的是，英语中没有中性的第三人称单数的代词。所以，我们用“他”来指代第三人，无论是男性或女性。这个惯例让作者或读者尴尬，但它看起来是最令人满意的解决方案。下面描述在本书中用到的一些数学符号。

- 计算符号 $x \leftarrow y$ 表示将 y 赋值给变量 x 。例如，考虑下面的算法：

$a = x^2$ 的系数

$b = x^1$ 的系数

$c = x^0$ 的系数

$x^* \leftarrow (-b + \sqrt{b^2 - 4ac}) / (2a)$

算法中的前三行不是赋值语句；它们只是简单地描述或定义 a 、 b 和 c 的值。这三个参数可以由用户，或其他算法或过程设置。但最后一行是赋值语句，它表示将箭头右边的值写入 x^* 中。

- $df(\cdot)/dx$ 是 $f(\cdot)$ 关于 x 的全导数. 例如, 假设 $y = 2x$ 且 $f(x, y) = 2x + 3y$, 则 $f(x, y) = 8x$ 并且 $df(\cdot)/dx = 8$.
- $f_x(\cdot)$ 也记为 $\partial f(\cdot)/\partial x$, 是 $f(\cdot)$ 关于 x 的偏导数. 例如, 仍然假设 $y = 2x$ 且 $f(x, y) = 2x + 3y$. 则 $f_x(x, y) = 2$.
- $\{x : x \in S\}$ 是所有属于集合 S 的 x 的集合. 类似的符号用来表示满足其他具体条件的 x 的值. 例如, $\{x : x^2 = 4\}$ 与 $\{x : x \in \{-2, +2\}\}$ 相同, 也和 $\{-2, +2\}$ 一样.
- $[a, b]$ 是 a 和 b 之间的闭区间, 它意味着 $\{x : a \leq x \leq b\}$. 根据上下文可以确定它是一个整数集或是一个实数集.
- (a, b) 是 a 和 b 之间的开区间, 它表示 $\{x : a < x < b\}$. 根据上下文可以确定这个集合是一个整数集或是一个实数集.
- 如果从上下文知道非负数 $i \in S$, 那么 $\{x_i\}$ 就是 $\{x_i : i \in S\}$ 的简写形式. 例如, 如果非负数 $i \in [1, N]$, 则 $\{x_i\} = \{x_1, x_2, \dots, x_N\}$.
- $S_1 \cup S_2$ 是属于集合 S_1 或集合 S_2 的所有 x 的集合. 例如, 如果 $S_1 = \{1, 2, 3\}$ 且 $S_2 = \{7, 8\}$, 则 $S_1 \cup S_2 = \{1, 2, 3, 7, 8\}$.
- $|S|$ 是集合 S 中元素的个数. 例如, 如果 $S = \{i : i \in \mathbb{N} \text{ 且 } i \in [4, 8]\}$, 则 $|S| = 5$. 如果 $S = \{3, 19, \pi, \sqrt{2}\}$, 则 $|S| = 4$. 如果 $S = \{\alpha : 1 < \alpha < 3\}$, 则 $|S| = \infty$.
- $\emptyset$ 是空集. $|\emptyset| = 0$.
- $x \bmod y$ 是 x 被 y 除之后的余数. 例如, $8 \bmod 3 = 2$.
- $\lceil x \rceil$ 是 x 的天花板; 即, 大于或等于 x 的最小整数. 例如, $\lceil 3.9 \rceil = 4$, $\lceil 5 \rceil = 5$.
- $\lfloor x \rfloor$ 是 x 的地板; 即, 小于或等于 x 的最大整数. 例如, $\lfloor 3.9 \rfloor = 3$, $\lfloor 5 \rfloor = 5$.
- $\min_x f(x)$ 表示求使 $f(x)$ 最小的 x 的值的问題. 它也用来表示 $f(x)$ 的最小值. 例如, 假设 $f(x) = (x - 1)^2$. 我们可以利用微积分或绘制函数 $f(x)$ 的图形看出 $f(x)$ 的最小值来求解问题 $\min_x f(x)$. 我们发现对这个例子 $\min_x f(x) = 0$. 对 $\max_x f(x)$ 有类似的定义.
- $\operatorname{argmin}_x f(x)$ 是使 $f(x)$ 达到最小的 x 的值. 例如, 仍然假设 $f(x) = (x - 1)^2$. 当 $x = 1$ 时 $f(x)$ 达到最小值 0, 所以对这个例子 $\operatorname{argmin}_x f(x) = 1$. 对 $\operatorname{argmax}_x f(x)$ 有类似的定义.
- $\mathbb{R}^s$ 是有 s 个元素的所有实向量的集合. 根据上下文它可以表示列向量或行向量.
- $\mathbb{R}^{s \times p}$ 是所有 $s \times p$ 实矩阵的集合.
- $\{y_k\}_{k=L}^U$ 是整数 k 从 L 变到 U 时所有 y_k 的集合. 例如, $\{y_k\}_{k=2}^5 = \{y_2, y_3, y_4, y_5\}$.
- $\{y_k\}$ 是所有 y_k 的集合, 这里的整数 k 的下限和上限根据上下文确定. 例如, 假设根据上下文有 3 个值: y_1, y_2 和 y_3 . 则 $\{y_k\} = \{y_1, y_2, y_3\}$.
- $\exists$ 的意思是“存在”, $\nexists$ 的意思是“不存在”. 例如, 如果 $Y = \{6, 1, 9\}$, 则 $\exists y < 2 : y \in Y$. 但是 $\nexists y > 10 : y \in Y$.
- $A \implies B$ 的意思是 A 意味着 B . 例如, $(x > 10) \implies (x > 5)$.
- $\mathbf{I}$ 是单位矩阵. 它的阶数根据上下文确定.

更多的符号参见第 19 页的缩写词列表.

1.6 本书的大纲

本书分为 5 篇¹.

1. 第一篇包括这个绪论以及介绍优化相关内容的另一章. 它介绍不同类型的优化问题, 简单但有效的爬山法, 并讨论和总结构成算法智能的要素.
2. 第二篇讨论公认的四种经典进化算法:

- 遗传算法;
- 进化规划;
- 进化策略;
- 遗传规划.

第二篇还会用一章来讨论遗传算法的数学分析方法. 第二篇的最末一章讨论多种算法的一些变种, 这些变种可用于上述四种经典算法. 同样也可以用于在下一部分中讲到的较新的进化算法.

3. 第三篇讨论一些较新的进化算法. 其中有一些可以追溯到 20 世纪 80 年代, 其实并不是最新的, 但其余的是在 21 世纪的前十年才出现的.
4. 第四篇讨论特殊类型的优化问题, 并说明如何修改前面几章中的进化算法来解决这些问题. 特殊类型包括:

- 组合问题, 它们的可行域由整数组成;
- 约束问题, 它们的可行域限定为已知的集合;
- 多目标问题, 想要同时最小化多个目标; 以及
- 有噪声的问题或昂贵适应度函数, 对于这些问题, 或者很难准确地得到候选解的性能, 或者评价候选解性能的计算量很大.

5. 第五篇包括几个附录, 用来讨论更重要或更有趣的问题.

- 附录 A 为进化算法的学生和研究人员提供各种实际的建议.
- 附录 B 讨论没有免费午餐定理, 这个定理告诉我们, 平均而言, 所有优化算法的表现都相同. 此附录还讨论如何利用统计学来评价进化算法之间的差别.
- 附录 C 列出一些标准的基准函数, 可以用它们来比较不同进化算法的性能.

1.7 基于本书的课程

基于本书的课程都应该从第 1 章和第 2 章开始, 第 2 章概述优化问题. 在那之后, 可以根据教师的偏好和兴趣决定以何种顺序学习余下的各章. 显然, 应该先学习遗传算法 (第 3 章), 然后再学习它的数学模型 (第 4 章). 同样, 在第四篇之前至少需要仔细学习第二篇或第三篇中的一章 (也就是说, 至少学习一个具体的进化算法).

大多数课程至少会覆盖第 3 章和第 5~7 章的内容, 这样学生能了解经典进化算法的背景. 如果学生有足够的数学功底并且也有时间, 就应该在某个时候学习第 4 章. 对研究

¹原书为“6 篇”. ——译者注.

生来说,第4章更重要.因为能帮助他们明白进化算法不仅是一个定性的学科,而且还有一些理论基础.现今有关进化算法的研究大多只是对算法做一些较小的调整完全没有数学上的支撑.许多进化算法从业者只关心得到好的结果,但研究学术的人需要参与到理论和实践中去.

根据教师或学生的具体兴趣学习第三篇和第四篇的各章.

附录没有包含在本书的主要部分中,因为它们并不是关于进化算法本身的内容,但是不应该低估附录的重要性.特别地,附录B和附录C中的材料至关重要,每一门进化算法的课程都应该包含它们.我推荐紧接在第二篇或第三篇的首章之后仔细讨论这两个附录.

综合上面的建议,在这里提出研究生一学期课程的大纲.

- 第1章和第2章.
- 第3章.
- 附录B和附录C.
- 第4~8章.对于大多数本科生和短课程,我建议跳过第4章.
- 第三篇中的几章,根据教师的偏好确定.冒着与读者开打“进化算法论战”的风险,我斗胆声称蚁群优化、粒子群优化和差分进化是最重要的“其他”进化算法,因此教师至少应该讲授第10~12章.
- 根据教师的偏好和可用的时间确定第四篇中的几章.

第 2 章 优 化

优化渗透进我们的所做所为并驱动着工程的方方面面。

丹尼斯·伯恩斯坦 (Dennis Bernstein) [Bernstein, 2006]

正如上面的引用所述, 优化几乎成为我们所做的每件事的一部分. 人员安排需要优化, 教学风格需要优化, 经济系统需要优化, 博弈策略需要优化, 生物系统需要优化, 卫生保健系统也需要优化. 优化是一个令人着迷的研究领域, 不仅因为它的算法和理论也因为它具有普遍的适用性.

本章概览

本章简要概述优化 (2.1 节), 包括带约束的优化 (2.2 节), 多目标优化问题 (2.3 节), 以及有多个解的优化问题 (2.4 节). 我们这本书的绝大部分都聚焦在连续优化问题, 也就是说, 问题中的独立变量可以连续地变化. 如果问题中的独立变量被限制在一个有限集中, 就是组合问题, 组合问题也颇有趣, 我们会在 2.5 节介绍. 在 2.6 节, 我们描述被称为爬山法的一个简单通用的优化算法, 并讨论它的一些变种. 在 2.7 节我们讨论几个与智能的本质相关的概念, 并展示它们如何与后面几章描述的进化优化算法相关联.

2.1 无约束优化

优化几乎适用于生活中的所有领域. 从食蚁兽的育种到受精卵的研究, 优化算法可以应用于一切事物. 进化算法的应用只会受到工程师的想象力的限制, 只有想不到没有做不到, 这就是为什么进化算法在过去几十年能得到广泛研究和应用的原因.

例如, 在工程上用进化算法找出机器人在执行某个任务时最好的轨迹. 假设在你的工厂里有一个机器人, 你希望它能尽快完成任务, 或用尽可能少的能量完成任务. 怎样才能知道对机器人来说最佳的路径? 有太多可能的路径, 很难找出最好的那一条. 但进化算法能较容易地完成这个任务, 至少能找到一个好的解 (如果不是最好的). 机器人有很强的非线性的性质, 因此机器人优化问题的搜索空间会有很多波峰和波谷, 在下面的简单例子中我们将会看到这种情况. 对于机器人问题而言, 情况会更糟, 因为是在多维空间 (而不是简单的三维空间) 中的波峰和波谷. 机器人优化问题的复杂度很自然地让它们成为进化算法的目标. 这个思路对静态机器人 (如, 机械手) 和移动机器人都适用.