

# 智能优化算法

刘勇 马良 张惠珍 魏欣 著

# 智能优化算法

刘勇 马良 张惠珍 魏欣 著



## 前 言

所以知之在人者谓之知，知有所合谓之智。

智所以能之在人者谓之能，能有所合谓之能。

——《荀子·正名》

近年来，智能优化算法得到了快速发展和广泛应用。出现许多有代表性的方法思想，如遗传算法、神经网络、蚁群优化算法、模拟退火算法、引力搜索算法、禁忌搜索算法、差分进化算法和文化算法等。这些算法为许多复杂困难问题的求解提供了可行有效的策略，已经受到越来越多的关注。此外，大数据和人工智能的兴起，也掀起了智能优化算法的研究热潮。

为进一步推动智能优化算法的发展，本书在全面细致分析的基础上，将我们团队多年来在该领域的相关成果进行了总结和提炼，同时也参考了国内外具有代表性的研究成果。全书主要分为三个部分：第一部分是基于生物学原理的优化算法，具体包括遗传算法、神经网络、蚁群优化算法、微粒群优化算法、人工蜂群优化算法、蝙蝠算法、萤火虫群优化算法、布谷鸟搜索算法、人工鱼群算法、细菌觅食优化算法、生物地理学优化算法和植物生长算法；第二部分是基于物理学原理的优化算法，具体包括模拟退火算法、引力搜索算法、混沌优化算法、随机分形搜索算法、光学优化算法和量子优化算法；第三部分是其他类型的优化算法，具体包括禁忌搜索算法、差分进化算法、和声搜索算法、大洪水算法、正弦余弦算法、竞争决策算法和文化算法。通过阐述这些算法的基本原理、数学模型、计算步骤，并给出部分理论分析，详细说明了各个算法的特征及其相关应用。

全书撰写耗时三年,整个过程也同时体现了智能优化算法所蕴含的进化过程。值此定稿之际,衷心感谢近些年来修读过《智能优化概论》课程的本科生、《进化计算》课程的硕士生以及《智能优化》课程的博士生,从他们那里得到许多有益的反馈。感谢被本书直接或者间接引用文献资料的同行学者。

本书所涉及的研究工作先后得到了教育部人文社科规划基金资助项目(No. 16YJA630037)、上海市“科技创新行动计划”软科学研究重点项目(No. 17692109400、No. 18692110500)、上海理工大学人文社科“攀登计划”项目(SK17PB04)、上海理工大学研究生课程建设项目(进化计算)上海市高原学科第二期建设(“管理科学与工程”)等项目的资助,在此一并致以谢意。

由于作者水平有限,书中不妥之处在所难免,恳请读者批评指正。

作者

2018年10月

# 目 录

前言 .....	001
<b>第一部分 基于生物学原理的优化算法 .....</b>	<b>001</b>
第一章 遗传算法 .....	003
第二章 神经网络 .....	021
第三章 蚁群优化算法 .....	033
第四章 微粒群优化算法 .....	051
第五章 人工蜂群优化算法 .....	065
第六章 蝙蝠算法 .....	076
第七章 萤火虫群优化算法 .....	090
第八章 布谷鸟搜索算法 .....	098
第九章 人工鱼群算法 .....	105
第十章 细菌觅食优化算法 .....	117
第十一章 生物地理学优化算法 .....	127
第十二章 模拟植物生长算法 .....	135
参考文献 .....	145
<b>第二部分 基于物理学原理的优化算法 .....</b>	<b>155</b>
第十三章 模拟退火算法 .....	157
第十四章 引力搜索算法 .....	170

第十五章 混沌优化算法 .....	184
第十六章 随机分形搜索算法 .....	194
第十七章 光学优化算法 .....	204
第十八章 量子优化算法 .....	215
参考文献 .....	224
<b>第三部分 其他类型的优化算法 .....</b>	<b>231</b>
第十九章 禁忌搜索算法 .....	233
第二十章 差分进化算法 .....	242
第二十一章 和声搜索算法 .....	253
第二十二章 大洪水算法 .....	262
第二十三章 正弦余弦算法 .....	270
第二十四章 竞争决策算法 .....	278
第二十五章 文化算法 .....	295
参考文献 .....	302
<b>附录 .....</b>	<b>311</b>
1. 蚁群优化算法主要程序(Delphi 代码) .....	313
2. 微粒群优化算法主要程序(Python 代码) .....	321
3. 人工蜂群优化算法主要程序(Java 代码) .....	324
4. 生物地理学优化算法主要程序(Matlab 代码) .....	336
5. 模拟退火算法主要程序(Python 代码) .....	345
6. 差分进化算法主要程序(Fortran 代码) .....	353
7. 正弦余弦算法主要程序(Matlab 代码) .....	363

## 第一部分 基于生物学原理的优化算法

有限的资源将所有的生命都逼上生存竞争的战场,只有受到自然偏爱的物种能够存活下来,在自然选择的法则下开始物种起源。

——《物种起源》

据估计,地球上的生物数量在 500 万到 1 万亿之间。种类繁多的生物都是通过自然选择过程进化而来的。许多生物个体的行为简单、盲目而且带有随机性,但是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,并且行为连贯、流畅与一致。基于生物学原理的优化算法是对各种生物在“优胜劣汰”进化过程中行为的模拟,能够解决许多传统方法难以解决的复杂问题,已经成为智能优化算法重要的组成部分。





# 第一章 遗传算法

遗传算法(Genetic algorithm, GA)是一种模拟生物在自然环境中遗传和进化的自适应全局优化算法,其基本思想遵循生物进化理论中“自然选择、适者生存”原则<sup>[1, 2]</sup>。遗传算法源于美国密歇根大学霍兰德在自然和人工自适应系统的研究,其学生巴格利在博士论文中首次提出了“Genetic algorithm”一词,采用了复制、交换和变异等进化操作。在1975年霍兰德正式出版了第一本遗传算法专著 *Adptation in Natural and Artificial Systems*,标志着遗传算法的诞生,而霍兰德被认为是遗传算法的创始人。

遗传算法自提出后,经历了很长时间的发展过程。直到1983年,遗传算法能够解决的最大问题是天然气管道的控制<sup>[3]</sup>。但随着研究的深入,遗传算法已经能够成功用于数值优化、模式识别、自动控制和生物工程等领域。可以说,遗传算法是应用最广泛和效果最显著的智能优化算法<sup>[4]</sup>。这里,我们将从算法原理、算法模型、算法分析、理论基础和应用案例等方面对遗传算法进行探讨。

## 1.1 算法原理

1831年,达尔文以博物学者身份进行了为期5年的环球科学考察,并进行了大量的人工育种研究,由此形成了生物进化的观点。1859年,达尔文发表了划时代巨著——《物种起源》,并提出了以自然选择、适者生存为基础的进化论。同时,华莱士也发表了和进化论观点几乎一致的《论变种无限地离开其原始模式的倾向》论文。进化论被恩格斯高度评价为“十九世纪自然科学的三大发现”之一。

根据达尔文的进化论,后代与亲代之间性状存在相似性,即性状可以从亲代传递给子代,这种生物现象称为遗传。“种瓜得瓜,种豆得豆”的现象就是遗传特征

的生动体现。染色体是细胞核中载有遗传信息的物质,在显微镜下呈圆柱状或杆状,主要由脱氧核糖核酸(DNA)和蛋白质组成,易被碱性染料染着色,因此而得名。基因是带有遗传信息的 DNA 片段,是控制生物性状的基本遗传单位。一条染色体上会有许多基因,而生物的各种性状几乎都是各种基因相互作用的结果。

生物在进化过程中,大都会经历繁殖、交叉、变异和选择四个基本阶段。繁殖是生物都有的基本现象,每个新个体都是其上一代经过繁殖所得来。通过繁殖,子代和亲代能够保证生物性状的相似性。而有性繁殖是被证明最有利于进化的繁殖方式<sup>[5]</sup>。生物在有性繁殖下一代时,两个同源染色体利用交叉而重新组合产生新染色体,即两个染色体的某些部分进行互换。变异是染色体上某些基因发生突变的现象,从而改变染色体的组成结构。变异表现为亲代与子代之间具有不相似的现象。突变在自然状态下可以产生,也可以人为地实现。前者称为自发突变,后者称为诱发突变。选择就是适应环境的物种会生存下来,而不适应环境的物种被淘汰,即“适者生存,优胜劣汰”。此外,也因为选择的作用,具有某些特性的个体会比其他个体更容易生存并繁殖。如果要衡量某个物种是否进化成功,评价标准就在于种群数量的多少。从这个角度看,1 000 份 DNA 拷贝永远都强过 100 份<sup>[6]</sup>。

生物进化就是从低级到高级、从简单到复杂的过程。在此过程中,物种不断完善和发展。生物进化过程可视为一种优化过程<sup>[4]</sup>,人类进化如图 1.1 所示。

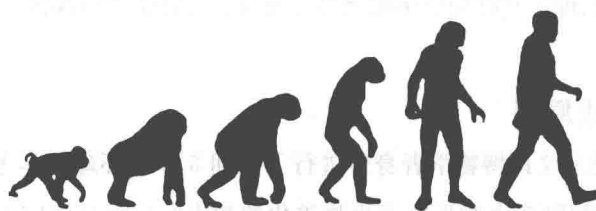


图 1.1 人类进化示意图

受到进化论的启发,同时结合 Fisher 于 1930 年出版的《自然选择的遗传理论》,霍兰德意识到可以通过计算机编程来模拟生物进化过程。模拟进化将在计算机中而不是生物体中进行,而且进化速度会比生物进化快得多。此外,有趣的是适应力较强的个体不仅会在它那一代为了生存而竞争,而且还会和他的“儿子”、“孙

子”和“重孙”等竞争<sup>[3]</sup>。

## 1.2 算法模型

遗传算法需要通过编码实现对个体的表示,并利用适应度函数对个体优劣进行评价,还要通过选择、交叉和变异等进化操作实现优化搜索。以下给出算法的具体实现模型。

### (1) 编码方法

在遗传算法中,如何表示待处理问题的解,即如何把问题的解空间映射到算法的搜索空间是算法设计的一个关键步骤。目前,可以利用编码将解通过染色体来表示。一个染色体由一个一定长度字符串表示。字符串的每一位对应一个基因。从生物学角度来看,编码相当于选择遗传物质<sup>[7]</sup>。在算法中,一个染色体可视为一个个体。而多个个体就可以组成算法的搜索群体。

遗传算法的编码方法有:二进制编码、自然数编码、实数编码和树型编码等。其中,最常见的是二进制编码。例如,一个长度为5的二进制串10100就可以表示一个个体;三个二进制串010、011、001就可以表示规模为3的群体。

对个体进行评价需要在问题的解空间中进行,即需要解码。例如,将二进制串10100表示成问题的解20,就是解码。实际上交替进行编码和解码是遗传算法的必备组成部分,实现了问题的解空间对算法搜索空间的相互转换。

### (2) 适应度函数

在遗传算法中,每个个体都有一个适应度函数值相对应。其优劣需要通过适应度函数值大小进行定量评价。个体越优,其适应度函数值越大。适应度函数是算法执行“适者生存、优胜劣汰”的依据,直接决定搜索群体的进化行为<sup>[4, 7]</sup>。

通常情况下,适应度函数要根据目标函数进行设置。令 $g(x)$ 表示目标函数,令 $G(x)$ 表示适应度函数。从目标函数 $g(x)$ 映射到适应度函数 $G(x)$ 的过程称为标定。基本标定方法如下<sup>[4]</sup>:

对于最大值优化问题,可直接将目标函数 $g(x)$ 设置为适应度函数 $G(x)$ ,即

$$G(x) = \max g(x) \quad (1.1)$$

对于最小值优化问题,可在目标函数  $g(x)$  前加一负号再将其设置为适应度函数  $G(x)$ ,即

$$G(x) = -\min g(x) \quad (1.2)$$

在遗传算法中,规定适应度函数值为正值,但是式(1.1)和(1.2)不能保证这一点,需要进一步转换,令  $F(x)$  表示转换后的适应度函数,具体方法如下<sup>[7]</sup>:

对于最大值优化问题,令

$$F(x) = \begin{cases} G(x) + C_{\min} & \text{当 } G(x) + C_{\min} > 0 \\ 0 & \text{否则} \end{cases} \quad (1.3)$$

其中,  $C_{\min}$  是足够小的常数。

对于最小值优化问题,令

$$F(x) = \begin{cases} C_{\max} - G(x) & \text{当 } C_{\max} > G(x) \\ 0 & \text{否则} \end{cases} \quad (1.4)$$

其中,  $C_{\max}$  是足够大的常数。

### (3) 选择操作

选择(也称为复制)就是从当前群体中选择适应度函数值大的个体,使这些优良个体有可能作为父代来繁殖下一代。选择操作直接体现了“适者生存、优胜劣汰”的原则。在该阶段,个体的适应度函数值越大,被选择作为父代的概率越大;个体的适应度函数值越小,被淘汰的概率越大。

实现选择操作的方法有很多,最基本的是霍兰德推荐的轮盘赌算法。计算每个个体被选择进入到下一代群体的概率,即

$$P_i = \frac{F_i}{\sum_{i=1}^N F_i} \quad (1.5)$$

其中,  $P_i$  表示第  $i$  个个体被选择的概率;  $F_i$  表示第  $i$  个个体的适应度函数值;  $N$  表示群体规模。

根据选择概率  $P_i$  将圆盘形赌轮分成  $N$  份,第  $i$  个扇形的中心角为  $2\pi P_i$ 。转

动轮盘一次,假设参考点落入第  $i$  个扇形中,就选择第  $i$  个个体<sup>[5]</sup>,如图 1.2 所示。上述过程可以采用计算机模拟来实现。

首先计算每个个体的累积概率,即

$$Q_i = \sum_{j=1}^i P_j \quad (1.6)$$

其中,  $Q_i$  表示第  $i$  个个体的累积概率,并规定  $P_0 = 0$ 。

然后随机产生在 0 到 1 之间服从均匀分布的数  $r$ ,当  $Q_{i-1} < r \leq Q_i$  时,则选择个体  $i$ 。最后重复上述过程  $N$  次,就可以选择  $N$  个个体。

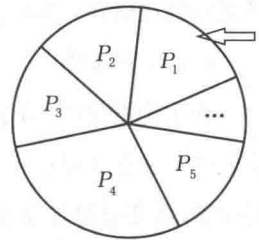


图 1.2 轮盘赌选择原理

#### (4) 交叉操作

在生物进化中,两个个体通过交叉互换染色体部分基因而重组产生新个体。在遗传算法中,交叉是产生新解的重要操作。要进行交叉操作,首先需要解决配对问题,采用随机配对是最基本的方法。

一般情况下,对二进制编码的个体采用点交叉方法。所谓点交叉就是在两个配对字符串随机选择一个或多个交叉点,互换部分子串从而产生新的字符串。如果只选择一个交叉点,这种方法称为单点交叉。以此类推,还有两点交叉以及多点交叉。图 1.3 和 1.4 给出了单点交叉和两点交叉示意图。

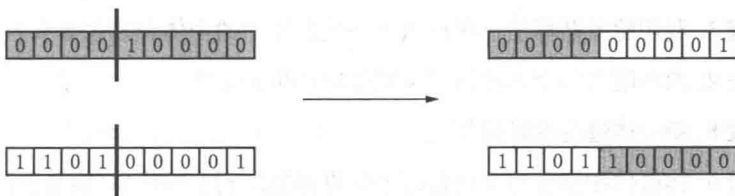


图 1.3 单点交叉

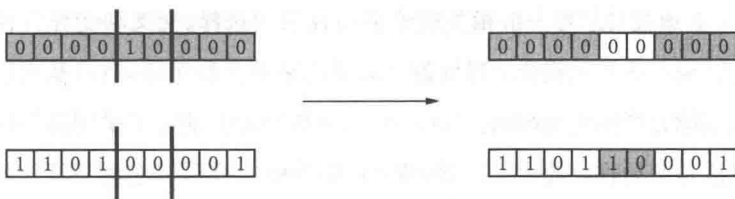


图 1.4 两点交叉

两个个体是否进行交叉操作由交叉概率决定。较大的交叉概率可以使遗传算法产生更多的新解,保持群体的多样性,并能防止算法早熟收敛。但是交叉概率过大,会使算法过多搜索不必要的解区域,消耗过多的计算时间。通常情况下,交叉概率的取值在 0.9 左右<sup>[4]</sup>。

#### (5) 变异操作

在生物进化中,由于偶然因素染色体某个(些)基因会发生突变,从而产生新的染色体。在遗传算法中,变异是产生新解的另一种操作。交叉操作相当于进行全局探索,而变异操作相当于进行局部开发。而全局探索和局部开发是智能优化算法必备的两种搜索能力<sup>[9, 10]</sup>。对二进制编码的染色体进行变异操作,等价于进行补运算,即将字符 0 变为 1,或者将字符 1 变为 0。

个体是否进行变异由变异概率决定。变异概率过低,部分有用的基因就难以进入染色体,不能有效提高算法解的质量;变异概率过大,子代较容易丧失父代优良的基因,导致算法失去从过去搜索经验进行的学习能力。一般情况下,变异概率的取值为 0.005 左右<sup>[4]</sup>。

综上所述,给出遗传算法的主要步骤:

**步骤 1.** 产生初始群体。

**步骤 2.** 计算每个个体适应度函数值。

**步骤 3.** 利用轮盘赌算法选择进入下一代群体中的个体。

**步骤 4.** 两两配对的个体进行交叉操作以产生新个体。

**步骤 5.** 新个体进行变异操作。

**步骤 6.** 将群体中迄今出现的最好个体直接复制到下一代中(精英保留策略)。

**步骤 7.** 反复执行步骤 2 到步骤 6,直到满足算法终止条件。

Rudolph 通过马尔可夫链相关理论证明仅采用选择、交叉和变异三种进化操作的遗传算法不能收敛到全局最优值。而采用精英保留策略,将对基本遗传算法的全局收敛能力产生重要影响。Rudolph 已经从理论上证明了采用精英保留策略的遗传算法是全局收敛的<sup>[11]</sup>。当然精英保留策略同样可以应用于其他智能优化算法。上述遗传算法的流程图如图 1.5 所示。

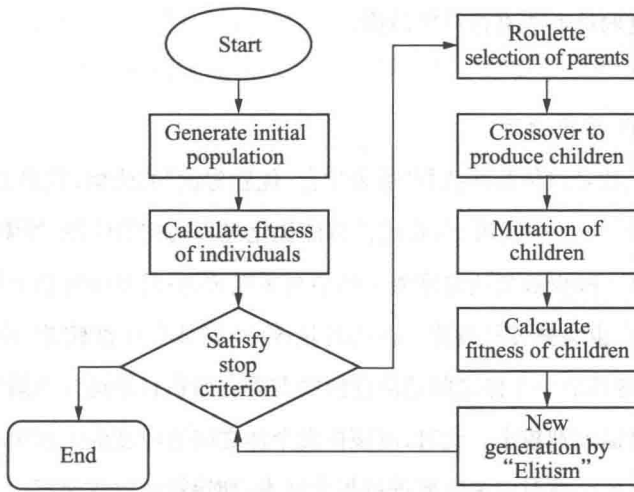


图 1.5 遗传算法流程图

### 1.3 算法分析

一个好的智能优化算法,其关键在于全局探索和局部开发能力的平衡。全局探索主要目的是对解空间进行更全面的搜索,希望发现更多的未知区域;而局部开发主要目的是对已知区域进行更为精细的搜索,希望获得质量更好的新解。实现算法全局探索和局部开发能力平衡的方法有很多种,需要根据算法自身的特点选择可行且有效的方法。

对遗传算法而言,可以通过设置选择压力来实现全局探索和局部开发能力的平衡。所谓选择压力,是指选择机制给适应度值低个体造成的生存压力。过大的选择压力,会使得群体中适应度值的个体快速被淘汰,降低群体的多样性,从而使算法的搜索空间急剧减小;过小的选择压力,会使得适应度值大的个体不容易产生更多的后代,从而使算法的优化速度大幅下降<sup>[8]</sup>。因此,在算法运行的初始阶段,设置较小的选择压力可以使算法具有较好的全局探索能力,能够进行比较全面的广域搜索;在的算法运行的后期,设置较大的选择压力使得算法具有较好的局部开发能力,能够对已有区域进行比较精细的局部搜索。

对选择压力的设置,可以从两个方面进行分析:适应度函数标定以及选择策

略。以下对这两种方法进行具体讨论。

### 1.3.1 适应度函数标定

适应度函数是遗传算法执行“适者生存、优胜劣汰”的依据,直接决定搜索群体的进化行为<sup>[4,7]</sup>。一般情况下,在遗传算法优化过程的初始阶段,搜索个体的适应度值差异显著。极少数高适应度值个体会被多次选择;低适应度值个体,尽管自身携带有效基因,也会被过早淘汰。在这种情况下,群体多样性较差,算法容易早熟收敛。此时,应该缩小个体之间适应度值的差距。而在算法运行的最后阶段,各个个体的适应度值差别较小。此时,应该扩大个体之间适应度值的差距,保证算法能够在高适应度值个体对应解区域进行集中搜索,加快算法收敛速度。

令  $F$  和  $H$  分别表示变换前和变换后的适应度函数,以下给出对适应度值进行尺度变换的常用方法<sup>[4, 7, 8]</sup>:

#### 1) 线性尺度变换法

线性变换采用如下方法:

$$H = aF + b \quad (1.7)$$

其中,参数  $a$  和  $b$  需要根据具体问题设置。

#### 2) $\sigma$ 截断法

$\sigma$  截断采用如下方法:

$$H = F + (\bar{F} - c\sigma) \quad (1.8)$$

其中, $\bar{F}$ 表示变换前的适应度函数值的平均值;参数  $c$  需要根据具体问题设置。

#### 3) 幂律尺度变换

幂律尺度变换采用如下方法:

$$H = F^\alpha \quad (1.9)$$

其中,当参数  $\alpha > 1$  时,选择压力增加;当参数  $\alpha < 1$  时,选择压力减小。

采用以上这些适应度函数设置方法是为调整群体中各个个体优劣程度的差



距,以体现“优胜劣汰”的原则。当需要选择更多的优秀个体时,增加选择压力,可以扩大个体间适应度函数值的差距<sup>[7]</sup>。以下给出基于选择策略的选择压力控制方法。

### 1.3.2 选择策略

在遗传算法中,一方面要保持搜索群体的多样性,设置低选择压力可选择多种类型的个体,加强对未知解区域的搜索,避免算法陷入局部极值,但算法优化速度会变得缓慢;另一方面要设置高选择压力可选择优良个体,实现优胜劣汰,可以加快优化速度但群体多样性会下降,会减小搜索到全局最优值概率。目前,可以通过选择策略调节选择压力和群体多样性之间的矛盾。

除了之前介绍的轮盘赌算法,选择策略还有分级选择法<sup>[7]</sup>、锦标赛选择法<sup>[4, 7]</sup>和 Boltzmann 选择法<sup>[7]</sup>等。以下给出这几种选择策略的实现方法。

#### 1) 分级选择法

假设群体规模为  $N$ ,所有个体按适应度值排名依次记为  $1, 2, \dots, i, \dots, N$ 。排名为  $i$  的个体被选择的概率  $prob(i)$ 可以通过线性函数进行设置:

$$prob(i) = q - (i-1)d \quad (1.10)$$

也可以通过非线性函数进行设置:

$$prob(i) = q(1-q)^{i-1} \quad (1.11)$$

其中,  $q$  表示最优个体被选择概率;  $d$  表示相邻个体被选择概率之差。此外,  $prob(i)$ 满足以下关系:

$$\sum_{i=1}^N prob(i) = 1 \quad (1.12)$$

#### 2) 锦标赛选择法

这种选择策略也基于排名的思想,但不进行显示排名。在当前群体中,每次随机选择  $k$  个个体,然后从这  $k$  个个体中选出适应度值最大的个体进入到下一代