

# 第一章 现代优化算法

现代优化算法是上世纪 80 年代初兴起的启发式算法，主要用于解决大量的实际应用问题。这些算法大致分为如下的几大类。

进化类算法：遗传算法、差分进化算法、免疫算法等。

群体智能算法：蚁群算法、粒子群算法、鱼群算法、蜂群算法、猫群算法、狼群算法、鸡群算法、雁群算法、蝙蝠算法、布谷鸟算法、果蝇算法、蛙跳算法、细菌觅食算法、萤火虫算法、烟花算法、杂草算法、头脑风暴算法等。

模拟退火算法、禁忌搜索算法、神经网络算法等。

目前，这些算法在理论和实际应用方面得到了较大的发展。无论这些算法是怎样产生的，它们有一个共同的目标：求 NP-hard 组合优化问题的全局最优解。

这些算法在人工智能的众多领域便得到了广泛应用。例如，机器学习、聚类、控制（如煤气管道控制）、规划（如生产任务规划）、设计（如通信网络设计、布局设计）、调度（如作业车间调度、机器调度、运输问题）、配置（机器配置、分配问题）、组合优化（如 TSP、背包问题）、函数的最大值以及图像处理和信号处理等领域。

## 1.1 遗传算法

早在 50 年代，一些生物学家开始研究运用数字计算机模拟生物的自然遗传与自然进化过程；1963 年，德国柏林技术大学的 I. Rechenberg 和 H. P. Schwefel，做风洞实验时，产生了进化策略的初步思想；60 年代，L. J. Fogel 在设计有限态自动机时提出进化规划的思想。1966 年 Fogel 等出版了《基于模拟进化的人工智能》，系统阐述了进化规划的思想。60 年代中期，美国 Michigan 大学的 J. H. Holland 教授提出借鉴生物自然遗传的基本原理用于自然和人工系统的自适应行为研究和串编码技术；1967 年，他的学生 J. D. Bagley 在博士论文中首次提出“遗传算法 (Genetic Algorithms)”一词；1975 年，Holland 出版了专著《Adaptation in Natural and Artificial Systems》，标志遗传算法的诞生。

70 年代初，Holland 提出了“模式定理” (Schema Theorem)，一般认为是“遗传算法的基本定理”，从而奠定了遗传算法研究的理论基础；1985 年，在美国召开了第一届遗传算法国际会议，并且成立了国际遗传算法学会 (ISGA, International Society of Genetic Algorithms)；1988 年，Holland 的学生 D. J. Goldberg 出版了《Genetic Algorithms in Search, Optimization, and Machine Learning》，对遗传算法及其应用作了全面而系统的论述；1991 年，L. Davis 编辑出版了《Handbook of genetic algorithms》，其中包括了遗传算法在工程技术和社会生活中大量的应用实例。

遗传算法 (Genetic Algorithms, GA) 是一种基于自然选择原理和自然遗传机制的搜索 (寻优) 算法，它是模拟自然界中的生命进化机制，在人工系统中实现特定目标的优化。

遗传算法的基本思想来源于 Darwin 的进化论和 Mendel 的遗传学说，其基本的观点为每一物种在不断的发展过程中都是越来越适应环境。物种的每个个体的基本特征被后代所继承，但后代又不完全同于父代，这些新的变化，若适应环境，则被保留下来。在某一环境中也是那些更能适应环境的个体特征能被保留下来，这就是适者生存。因此，生物一代比一代优。

生物虽然一代比一代优，但并不是说后一代与前一代没有任何的关系。生物的后一代总是或多或少的继承了前一代的一些特性，这就叫遗传。而后一代又不完全像前一代，这叫变异。生物在进化的过程中既有遗传，又有变异，生物就是在这样的遗传、变异的作用那个下，一代一代的繁衍下去，而且得到的是一代比一代优。

### 1.1.1 基本概念

**个体** 模拟生物个体而对问题中的对象（一般就是问题的解）的一种称呼，一个个体也就是搜索空间中的一个点。

**种群** 模拟生物种群而由若干个体组成的群体, 它一般是整个搜索空间的一个很小的子集。

**适应度** 借鉴生物个体对环境的适应程度, 而对问题中的个体对象所设计的表征其优劣的一种测度。

**适应度函数** 问题中的全体个体与其适应度之间的一个对应关系。它一般是一个实值函数。该函数就是遗传算法中指导搜索的评价函数。

**染色体** 问题中个体的某种字符串形式的编码表示。

**基因** 字符串中的字符也就称为基因。

例如，个体 9，染色体为：1001，染色体长度  $l = 4$ 。

**进化** 生物在其延续生存的过程中，逐渐适应其生存环境，使得其品质不断得到改良，这种生命现象称为进化。生物的进化是以种群的形式进行的。

## 1.1.2 遗传算法和数学优化

遗传学	遗传算法	数学
个体	要处理的基本对象	可行解
群体	个体的集合	一组可行解
染色体	个体的表现形式	可行解的编码
基因	染色体中的元素	编码中的元素
基因位	某一基因在染色体中的位置	元素在编码中的位置
适应值	个体对于环境的适应程度	可行解所对应的适应函数值
种群	被选定的一组染色体	根据入选概率定出的一组可行解
选择	优胜劣汰	保留大的可行解，去掉小的可行解
交叉	一组染色体上对应基因段的交换	根据交叉原则产生的一组新解
交叉概率	染色体对应基因段的交换概率	$[0, 1]$ 上的一个值，一般为 $0.65 \sim 0.9$
变异	染色体上基因变化	编码的某些元素被改变
变异概率	染色体基因变化的概率	$(0, 1)$ 上的一个值，一般为 $0.001 \sim 0.01$
进化	一代又一代的优化	目标函数取到更优值

### 1.1.3 遗传操作

亦称遗传算子，就是关于染色体的运算。遗传算法中有三种遗传操作：

#### 1. 选择-复制：

选择是按照一定概率随机地选择两对个体进行繁殖。遗传算法在很大程度上是一种偶然性现象，随机过程在其中起重要作用。一般而言，选择的过程是一种基于适应度的优胜劣汰的过程。

复制是从一个旧种群中选择生命力强的个体位串产生新种群的过程根据位串的适应度值拷贝位串意味着，具有较高适应度值的位串更有可能在下一代中产生一个或多个后代。显然，这个操作是模仿自然选择现象，将达尔文的适者生存理论应用于位串的复制，适应度值是该位串被复制或被淘汰的决定因素。

#### 2. 交叉

有性生殖生物在繁殖下一代时，两个同源染色体通过交叉而重组，亦即在两个染色体的某一相同位置处 DNA 被切断，其前后两串分别交叉组合形成两个新的染色体，该过程又称为杂交。

交叉即互换两个染色体某些位上的基因。例如，设染色体

$$s_1 = 0100\ 1011, s_2 = 1001\ 0101$$

交换其后 4 位基因，可得：

$$s'_1 = 0100\ 0101, s'_2 = 1001\ 1011$$

可以看做是原染色体  $s_1$  和  $s_2$  的子代染色体。

#### 3. 变异

在细胞进行复制时可能以很小的概率产生某些复制差错，从而使 DNA 发生某种变异，产生出新的染色体，这些新的染色体表现出新的性状。

变异即改变染色体某个(些)位上的基因。例如，设染色体

$$s = 11001101$$

若其第三位上发生变异，则  $s$  变为：

$$s' = 11101101$$

$s'$  也可以看做是原染色体  $s$  的子代染色体。

**例题 1.1** 模仿生物遗传的方式求  $y = x^2$  的最小值。

步骤 1. 任取一组值  $x$  记为第一代：

$$x = 2, 5, -4, 8, -10, 12$$

对用的函数值分别为：

$$y = 4, 25, 16, 64, 100, 144$$

步骤 2. 构造染色体，若用 5 位二进制表示，第一位表示正负号：0 表示正、1 表示负。

$x$	2	5	-4	8	-10	12
第一代	00010	00101	10100	01000	11010	01100

步骤 3. 根据染色体进行遗传与变异，变异得到第二代。变异的概率很小，一般变异的概率 8%，即每一位的 0 或 1 都有 8% 的可能编码 1 或者 0。

第二代	00001	00110	10100	01000	11000	01110
$x$	1	4	-4	8	-8	16
$y$	1	16	16	64	64	169

步骤 4. 继续变异得到第三代。

第三代	00000	00101	10100	01010	11010	01100
$x$	0	5	-4	10	-10	12
$y$	0	25	16	100	100	144

已经得到了函数的最小值点  $x = 0$ ，结束。

### 1.1.4 遗传算法的描述

遗传算法的实质是通过群体搜索技术，根据适者生存的原则逐代进化，最终得到最优解或准最优解。其大致的过程为：首先生成初始种群，接着求每个体的适应度。根据适者生存的原则选择优良个体并两两配对，通过随机交叉、变异等步骤生成下一代群体。循环以上过程使群体逐代进化，直到满足条件为止，具体过程如下图所示。

```
Begin
1. 生成初始种群，选择适当的编码方式；
2. 通过计算群体中各个体的适应度对群体进行评价；
3. While 未达到要求的目标 do
    begin
        1) 选择繁殖下一代群体的各个体
        2) 执行杂交操作
        3) 执行变异操作
        4) 对群体进行评价
    end
end
```

### 1.1.5 遗传算法的 Matlab 求解

遗传算法函数 ga 调用格式如下：

$$[x, fval] = ga(fun, nvars, A, b, Aeq, beq, lb, ub, nonlcon, options)$$

其中：fun 是目标函数，nvars 是目标函数中独立变量的个数，options 是一个包含遗传算法选项参数的数据结构，其它参数的含义与非线性规划 fmincon 中的参数相同。

返回值 x 为所求函数的局部极小点，这里 x 为行向量，fval 为目标函数的极小值。

**例题 1.2** 求解优化问题：

$$\begin{aligned} \max \quad & f(x) = 2x_1 + 3x_1^2 + 3x_2 + x_2^2 + x_3 \\ \text{s.t.} \quad & \begin{cases} x_1 + 2x_1^2 + x_2 + 2x_2^2 + x_3 \leq 10 \\ x_1 + x_1^2 + x_2 + x_2^2 - x_3 \leq 50 \\ 2x_1 + x_1^2 + 2x_2 + x_3 \leq 40 \\ x_1^2 + x_3 = 2 \\ x_1 + 2x_2 \geq 1 \\ x_1 \geq 0, x_2, x_3 \end{cases} \end{aligned}$$

调用 ga 命令的 Matlab 程序如下。

```
clc
clear
A = [-1 -2 0; -1 0 0];
b = [-1; 0];          % 两个线性约束条件
[x, y] = ga(@obj, 3, A, b, [], [], [], [], @constr); % 调用遗传算法函数
x
y=-y

function y = obj(x);    % 定义目标函数，其中 x 为行向量
c1 = [2 3 1];
c2 = [3 1 0];
y = c1 * x' + c2 * x' .^2;
y=-y;                  % 目标函数极小化
end

function [f, g] = constr(x); % 定义约束条件函数
f = [x(1) + 2 * x(1)^2 + x(2) + 2 * x(2)^2 + x(3) - 10
      x(1) + x(1)^2 + x(2) + x(2)^2 - x(3) - 50
      2 * x(1) + x(1)^2 + 2 * x(2) + x(3) - 40];
g = x(1)^2 + x(3) - 2;
end
```

计算结果：x = [1.9828 0.8015 -1.9316]，fval = 16.8761。



### 1.1.6 遗传算法的讨论

#### 关于编码

由于遗传算法是模拟生物的遗传与变异机制而设计的，而生物的遗传与变异是通过染色体来进行，所以遗传算法必须编码。不同的编码方法就得到不同形式的遗传算法。

例如，上面这个例子是用 5 位编码，也可以采用 6 位编码、7 位或 100 位编码。问题是采用什么编码好呢？这个问题到现在为止还没有答案，很多学者就是研究对各种各样的编码方法进行研究，看哪种编码更好。

#### 杂交和变异方式

遗传算法的两个基本运算就是杂交和变异。因此，不同的选择遗传杂交方式、不同的变异方式就会得到不同的遗传算法。

一种杂交方式是每个个体都参与杂交，也可以让好的个体多杂交，淘汰那些差的个体。很自然的一个问题是：如何区分好的个体和差的个体？

如上例求函数的最小值问题，第一代染色体编码及函数值如下表所示：

$x$	2	5	-4	8	-10	12
第一代	00010	00101	10100	01000	11010	01100
$y$	4	25	16	64	100	144

依函数值越小越好的原则可得：最好的点是 2，最差的点是 12。因此，淘汰 12，而让 2 这个点多杂交一次。进而有如下的染色体：

$x'$	2	5	-4	8	-10	2
杂交	00010	00101	10100	01000	11010	00010
$y'$	4	25	16	64	100	4

也可以删除两个最差，用两个好一点的取代。因此，淘汰 12 和 -10，而让 2 这个点多杂交两次。进而有如下的染色体：

$x'$	2	5	-4	8	2	2
杂交	00010	00101	10100	01000	00010	00010
$y'$	4	25	16	64	4	4

到底淘汰几个最差的？又用哪些好的取代差的，这些数就叫控制参数。选择不同的控制参数，得到的算法就不同。

上面只是从不同的编码方法与不同的杂交方式变形出各种各样的遗传算法。同样，如在进行变异操作时，选多不同的变异方式也得到不同的算法。有选 8% 的个体进行变异，也有选 5% 的个数进行变异，这个百分数通常称为变异率。

#### 关于群体规模

为便于计算，每一代群体的个体数目都取相等。群体规模越大、越容易找到最优解，但由于受到计算机的运算能力的限制，群体规模越大，计算所需要的时间也相应地增加。进化终止条件可以设定到某一代进化结束，也可以根据找出近似最优解是否满足精度要求来确定。

### 1.1.7 遗传算法的主要特点

1. 遗传算法直接在解空间搜索，而不像图搜索那样在问题空间搜索。
2. 遗传算法随机地始于搜索空间的一个点集，而不像图搜索那样固定地始于搜索空间的初始节点或终止节点。从这个角度看，遗传算法是一种随机搜索算法。
3. 遗传算法总是在寻找优解，而不像图搜索那样并非总是要求优解。从这个角度看，遗传算法又是一种优化搜索算法。
4. 遗传算法的搜索过程是从空间的一个点集到另一个点集的搜索，而不像图搜索那样一般是从空间的一个点到另一个点地搜索。因而它实际是一种并行搜索，适合大规模并行计算，而且这种种群到种群的搜索有能力跳出局部最优解。
5. 遗传算法的适应性强，除需知适应度函数外，几乎不需要其他的先验知识。<sup>3</sup>
6. 遗传算法长于全局搜索，它不要求函数连续，也不受搜索空间的限制，能以很大的概率从离散的、多极值的、含有噪声的高维问题中找到全局最优解。

### 1.1.8 遗传算法的收敛性

遗传算法通过对这些操作的适当设计和运行，可以实现兼顾全局搜索和局部搜索的所谓均衡搜索。遗传算法虽然可以实现均衡的搜索，并且在许多复杂问题的求解中往往能得到满意的结果，但是该算法的全局优化收敛性的理论分析尚待解决。目前普遍认为，标准遗传算法并不保证全局最优收敛。但是，在一定的约束条件下，遗传算法可以实现这一点。

#### 定理 1.1

如果变异概率为  $p_m \in [0, 1]$  同时采用比例选择法（按个体适应度占群体适应度的比例进行复制），则标准遗传算法的变换矩阵  $P$  是正定的。

标准遗传算法不能收敛至全局最优解。但对标准遗传算法作一些改进，就能够保证其收敛性。

最佳个体保存方法的思想是把群体中适应度最高的个体不进行配对交叉而直接复制到下一代中。设时间  $t$  (第  $t$  代) 时，群体中  $a^*(t)$  为最佳个体，并设  $A(t+1)$  为新一代群体，若  $A(t+1)$  中不存在  $a^*(t)$ ，则把  $a^*(t)$  作为  $A(t+1)$  中的第  $n+1$  个个体（其中， $n$  为群体大小）。

#### 定理 1.2

具有定理 1 所示参数，且在选择前保留当前最优值的遗传算法最终能收敛到全局最优解。

最佳个体保存方法的优点：确保进化过程中某一代的最优解不被交叉和变异操作所破坏。

最佳个体保存方法的缺点：局部最优个体的遗传基因会急速增加而使进化有可能限于局部解，即该方法的全局搜索能力差，它更适合单峰性质的搜索空间搜索，而不是多峰性质的空间搜索。

## Bibliography

- [1] 包子阳、余继周、杨杉著，智能优化算法及其 MATLAB 实例 [M]，电子工业出版社。