模式识别的理论与方法 Pattern Recognition

裴继红

Chapter 7 Stochastic Methods Part 2

(Sections 7.5-7.7)

本讲内容

- 进化方法
 - 遗传算法(Genetic Algorithms)
 - 遗传编程(Genetic Programming)
- · 粒子群 (PSO) 优化方法
- 蚁群优化算法
- 0 0 0 0 0



遗传算法的概述

- 1. 遗传算法起源于上世纪**60**年代对自然和人工自适应系统的研究,是模拟生物在自然环境中的遗传和进化过程而形成的一种<mark>自适应全局</mark>优化概率搜索算法。
- 2. 由美国密执安大学的Holland教授最早提出。
- 3. 上世纪70年代,De Jong基于遗传算法的思想在计算机上进行了大量的纯数值函数优化计算实验
- 4. 上世纪80年代Goldberg进行归纳总结,形成了遗传算法的基本框架
- **5. 1991**年,**Davis**编辑出版了《遗传算法手册》,书中包含了大量应用实例。该书为推广和普及遗传算法起到了重要的指导作用。
- 6. 1992年,Koza将遗传算法应用于计算机程序的优化设计及自动生成, 提出了遗传编程的概念。并成功应用于人工智能、机器学习、符号 处理等方面。



遗传算法的模式定理

• Holland提出的**模式定理**(schema theorem)是遗传算法的基本原理, 从进化动力学的角度提供了能够较好地解释遗传算法机理的一种数学 工具,同时也是编码策略、遗传策略等分析的基础。

• 模式定理:

▶ 在选择、交叉、变异算子的作用下,那些超过群体平均适应值的低阶的、长度短的模式生存数量,将随着迭代次数的增加以指数规律增长。

- Holland's schema theorem, also called the fundamental theorem of genetic algorithms.
- It says that short, low-order schemata with above-average fitness increase exponentially in successive generations.
- A <u>schema</u> is a template that identifies a <u>subset</u> of strings with similarities at certain string positions.

遗传算法的应用领域

- > 遗传算法提供了一种求解复杂系统优化问题的通用框架。
 - 1. 函数优化

是遗传算法的经典应用领域,也是对遗传算法进行性能测试评价的常用算例。

2. 组合优化

实践证明,遗传算法对于组合优化问题中的 NP 完全问题非常有效

3. 生产调度

遗传算法现在已经成为解决复杂调度问题的有效工具

4. 自动控制

基于遗传算法的模糊控制器的优化设计、参数辨识、模糊控制规则的学习、人工神经网络的结构优化设计和权值学习等

5. 机器人

遗传算法的起源就来自于对人工自适应系统的模拟,机器人学自然成为遗传算法的一个重要应用领域

6. 人工生命

是用计算机、机械等人工媒体模拟或构造出的具有自然生物系统特有行为的人造系统。基于遗传算法的进化模型是研究人工生命现象的重要理论基础

7. 模式识别、图像处理、机器学习中的优化问题 等等



遗传算法的特点

- 1. 遗传算法以决策变量的编码作为运算对象。
 - ▶ 传统优化算法往往直接利用变量数值本身进行优化计算,
 - 遗传算法是以决策变量某种形式的编码为运算对象,可以方便 地引入和应用遗传操作算子。
- 2. 遗传算法直接以目标函数值作为搜索指导信息。
 - ▶ 传统的优化算法往往不仅需要目标函数值,还需要目标函数的 导数等信息。
 - 对目标函数无法求导或很难求导的函数,用遗传算法进行优化 比较方便。



遗传算法的特点

- 3. 遗传算法同时进行解空间的多点搜索。
 - 传统的优化算法往往从解空间的一个初始点开始搜索,容易陷入局部极值点,且效率不高。
 - 遗传算法进行群体搜索,在搜索的过程中引入遗传运算,使群体可以不断进化。遗传算法具有其特有的隐含并行性。
- 4. 遗传算法使用概率搜索技术。
 - ▶ 遗传算法属于一种自适应概率搜索技术,其选择、交叉、变异等运算都是以一种概率的方式来进行的,从而增加了其搜索过程的灵活性。
 - ➤ 实践和理论都已证明了在一定条件下遗传算法总是以概率1收敛 于问题的最优解。





• 求函数最大值的优化问题,一般可描述为下述数学规划模型:

$$\begin{cases} \max & f(\mathbf{X}) \\ s.t. & \mathbf{X} \in \mathbf{R} \\ \mathbf{P} \subseteq \mathbf{U} \end{cases}$$



- □ 遗传算法中,将 n 维模式解向量用n个记号 X_i (i=1,2,...n)所组成的符号串 $X=X_1X_2...X_n$ 来表示. 把每一个 X_i 看作一个遗传基因,它的所有可能取值称为等位基因,这样,X就可看作是由 n 个遗传基因所组成的一个染色体。
 - ▶ 染色体的长度可以是固定的,也可以是变化的。
 - ▶ 等位基因可以是一组整数,也可以是某一范围内的实数值,或者是记号。
 - ▶ 最简单的等位基因是由 0 和 1 这两个整数组成的,相应的染色体就可表示为一个二进制符号串
- □ 这种编码所形成的排列形式 X 是个体的基因型,与它对应的 X 值是个体的表现型。
 - \triangleright 染色体 X 也称为个体 X,对于每一个个体 X,要按照一定的规则确定出其适应度。
 - ▶ 个体的适应度与其对应的个体表现型**X**的目标函数值相关联,**X**越接近于目标函数的最优点,其适应度越大;反之,其适应度越小。

- □ 遗传算法中,决策变量 *X* 组成了问题的解空间。
 - ▶ 对问题最优解的搜索是通过对染色体 X 的搜索过程来进行的,从而由所有的染色体 X 就组成了问题的搜索空间。
- □ 生物的进化是以群体为主体的。
 - ▶ 与此相对应,遗传算法的运算对象是由 M 个个体所组成的集合,称为群体。
- □ 与生物一代一代的自然进化过程相似,遗传算法的运算过程也是一个 反复迭代过程,第 t 代群体记做 P(t), 经过一代遗传和进化后,得到第 t+1代群体,它们也是由多个个体组成的集合,记做P(t+1)。
 - ▶ 这个群体不断地经过遗传和进化操作,并且每次都按照优胜劣汰的规则将适应度较高的个的个体更多地遗传到下一代,这样最终在群体中将会得到一个优良的个体X,它所对应的表现型X将达到或接近于问题的最优解



- □ 生物的进化过程主要是通过染色体之间的交叉、变异来完成的。
 - 遗传算法中最优解的搜索过程也模仿生物的这个进化过程,使用所谓的遗传算子(genetic operators)作用于群体 P(t) 中,进行下述遗传操作,从而得到新一代群体P(t+1)。

➤ 选择(selection)

- 根据各个个体的适应度,按照一定的规则或方法,从第 t 代群体 P(t) 中选择出一些优良的个体遗传到下一代群体 P(t+1) 中。

➤ 交叉(crossover)

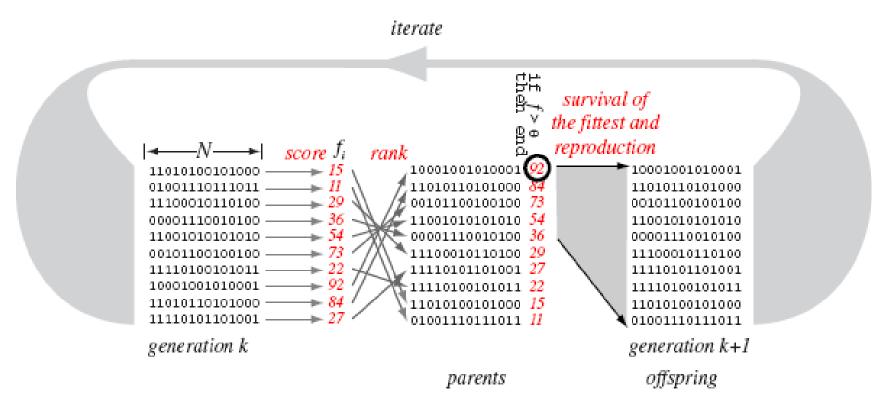
- 将群体 P(t) 内的各个个体随机搭配成对,对每一个个体,以某个概率 (称为交叉概率, crossover rate) 交换它们之间的部分染色体。

> 变异(mutation)

- 对群体*P(t)*中的每一个个体,以某一概率 (称为变异概率,mutation rate) 改变某一个或一些基因位置上基因值,使其为其它的等位基因。



遗传算法优化过程示例





遗传算子-1: 交叉算子

1. 单点交叉算子:

| Initial Strings | Crossover Mask | Offspring |
|---------------------|----------------|-------------|
| <u>11101</u> 001000 | 11111000000 | 11101010101 |
| 00001 <u>010101</u> | | 00001001000 |



遗传算子-2、3:变异算子、复制算子

2. 单点变异算子:

00001010101 — 01001010101

Select second bit.

3. 复制算子 染色体不加改变地传到子代



遗传算法解优化问题举例

• 感知器问题的解举例



基本遗传算法及其设计

- 基本遗传算法(Simple Genetic Algorithms, 简称SGA) 是一种统一的最基本的遗传算法,
- 它只使用选择、交叉、变异这三种基本遗传算子
- 其遗传进化操作过程简单,容易理解,是其他一些遗传算法的雏形和基础,
- 它不仅给各种遗传算法提供了一个基本框架,同时也具有一定的应用价值



基本遗传算法的构成要素

① 染色体编码方法。

基本遗传算法使用固定长度的二进制符号串来表示群体中的个体,其等位基因是由二值符号集{0,1}所组成的。初始群体中各个个体的基因值可用均匀分布的随机数来生成。

② 个体适应度评价。

基本遗传算法按与个体适应度成正比的概率来决定当前群体中每个个体遗传到下一代群体中的机会多少。为正确计算这个概率,这里要求所有个体的适应度必须为正数或零。

③遗传算子。

基本遗传算法使用下述三种遗传算子:选择运算使用比例选择算子,交叉运算使用单点交叉算子,变异运算使用基本位变异算子或均匀变异算子。

基本遗传算法的运行参数

- 基本遗传算法有下述4个运行参数需要提前设定:
 - 群体大小M, 即群体中所含个体数目, 一般取为 20~100;
 - 遗传运算的终止进化代数T,一般取为100~500;
 - $交叉概率P_c, 一般取为0.4~0.99;$
 - $变异概率 P_m, 一般取为0.0001~0.1$ 。



基本遗传算法的形式化定义

• 基本遗传算法可定义为一个8元组:

$$SGA = (C, E, P_0, M, \Phi, \Gamma, \Psi, T)$$

式中,

- **C---**个体的编码方法;
- E---个体适应度评价函数;
- *P₀*---初始群体;
- M---群体大小;
- **Φ---**选择算子;
- *[*---交叉算子;
- T---遗传运算终止条件。



基本遗传算法的应用步骤

- ▶ 遗传算法提供了一种求解复杂系统优化问题的通用框架。 对于具体问题,可按下述步骤来构造:
 - ①确定决策变量及其各种约束条件,即确定出个体的表现型**X**和问题的解空间;
 - ②建立优化模型,即描述出目标函数的类型及其数学描述形式或量 化方法
 - ③确定表示可行解的染色体编码方法,即确定出个体的基因型**X**及遗传算法的搜索空间;



基本遗传算法的应用步骤

- ④确定解码方法,即确定出由个体基因型**X**到个体表现型**X**的对应关系或转换方法;
- ⑤确定个体适应度的量化评价方法,即确定出由目标函数值 f(X) 到个体适应度的转换规则;
- ⑥设计遗传算子,即确定出选择运算、交叉运算、变异运算等遗传算子的具体操作方法;
- ⑦确定遗传算法的有关运行参数,即确定出遗传算法的 M, T, p_c, p_m 等参数。





1. 个体适应度评价

- 在遗传算法中,以个体适应度的大小来确定该个体被遗传到下一代群体中的概率。
- 个体适应度越大,该个体被遗传到下一代的概率也越大;反之, 个体的适应度越小,该个体被遗传到下一代的概率也越小。
- ➤ 基本遗传算法使用比例选择算子来确定群体中各个个体遗传到下一代群体中的数量。为正确计算不同情况下各个个体的遗传概率,要求所有个体的适应度必须为正数或0,不能是负数。
- 为满足适应度取非负值的要求,基本遗传算法一般采用下面方 法将目标函数值变换为个体的适应度。

$$F(X) = \begin{cases} f(X) + C_{\min} & \stackrel{\text{def}}{=} & f(X) + C_{\min} > 0 \text{ for } \\ 0 & \stackrel{\text{def}}{=} & f(X) + C_{\min} \leq 0 \end{cases}$$



2. 比例选择算子

- ➤ 比例选择实际上是一种随机选择,也叫做赌盘 (Roulette Wheel)选择,因为这种选择方式与赌博中的 赌盘操作原理非常相似。
- ▶ 比例选择算子的具体执行过程是:
 - ① 先计算出群体中所有个体的适应度之和;
 - ② 其次计算出每个个体的相对适应度的大小,此值即为各个个体被遗传到下一代群体中的概率;
 - ③ 最后再使用模拟赌盘操作(即**0**到**1**之间的随机数)来确定 各个个体被选中的次数。



- 3. 单点交叉算子。单点交叉算子是最常用和最基本的交叉操作算子。单点交叉算子的具体执行过程如下:
 - ▶ 对群体中的个体进行两两随机配对;
 - 对每一对相互配对的个体,随机设置某一基因座之后的位置为 交叉点;
 - 对每一对相互配对的个体,依设定的交叉概率 在其交叉点处相互交换两个个体的部分染色体,从而产生出两个新个体。



- 4. 基本位变异算子。基本位变异算子的具体执行过程为:
 - 对个体的每一个基因座,依变异概率指定其为变异点;
 - 对每一个指定的变异点,对其基因值做取反运算或用其他等位基因值来代替,从而产生出一个新的个体。





粒子群优化算法PSO



蚁群算法

