### 遗传算法GA

**01 什么是遗传算法？**

遗传算法（Genetic Algorithm, GA）是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，不需要确定的规则就能自动获取和指导优化的搜索空间，自适应地调整搜索方向。

遗传算法以一种群体中的所有个体为对象，并利用随机化技术指导对一个被编码的参数空间进行高效搜索。其中，选择、交叉和变异构成了遗传算法的遗传操作；参数编码、初始群体的设定、适应度函数的设计、遗传操作设计、控制参数设定五个要素组成了遗传算法的核心内容。

**02 执行过程**

遗传算法是从代表问题可能潜在的解集的一个种群（population）开始的，而一个种群则由经过基因（gene）编码的一定数目的个体(individual)组成。每个个体实际上是染色体(chromosome)带有特征的实体。

染色体作为遗传物质的主要载体，即多个基因的集合，其内部表现（即基因型）是某种基因组合，它决定了个体的形状的外部表现，如黑头发的特征是由染色体中控制这一特征的某种基因组合决定的。因此，在一开始需要实现从表现型到基因型的映射即编码工作。由于仿照基因编码的工作很复杂，我们往往进行简化，如二进制编码。

初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代（generation）演化产生出越来越好的近似解，在每一代，根据问题域中个体的适应度（fitness）大小选择（selection）个体，并借助于自然遗传学的遗传算子（genetic operators）进行组合交叉（crossover）和变异（mutation），产生出代表新的解集的种群。

这个过程将导致种群像自然进化一样的后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码（decoding），可以作为问题近似最优解。

PS生物学术语:

基因型(genotype)：性状染色体的内部表现；

表现型(phenotype)：染色体决定的性状的外部表现，或者说，根据基因型形成的个体的外部表现；

进化(evolution)：种群逐渐适应生存环境，品质不断得到改良。生物的进化是以种群的形式进行的。

适应度(fitness)：度量某个物种对于生存环境的适应程度。

选择(selection)：以一定的概率从种群中选择若干个个体。一般，选择过程是一种基于适应度的优胜劣汰的过程。

复制(reproduction)：细胞分裂时，遗传物质DNA通过复制而转移到新产生的细胞中，新细胞就继承了旧细胞的基因。

交叉(crossover)：两个染色体的某一相同位置处DNA被切断，前后两串分别交叉组合形成两个新的染色体。也称基因重组或杂交；

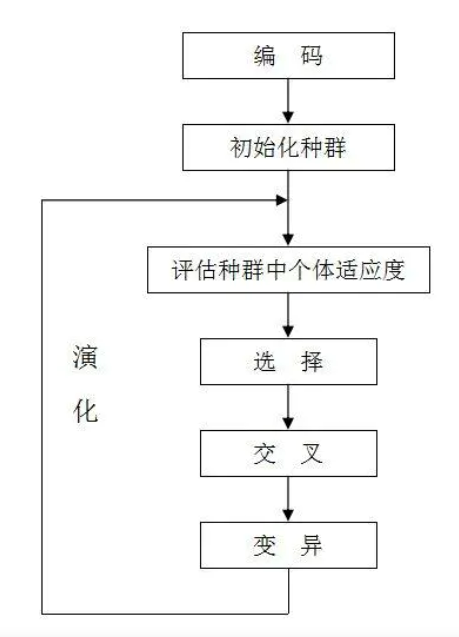
变异(mutation)：复制时可能（很小的概率）产生某些复制差错，变异产生新的染色体，表现出新的性状。

编码(coding)：DNA中遗传信息在一个长链上按一定的模式排列。遗传编码可看作从表现型到基因型的映射。

解码(decoding)：基因型到表现型的映射。

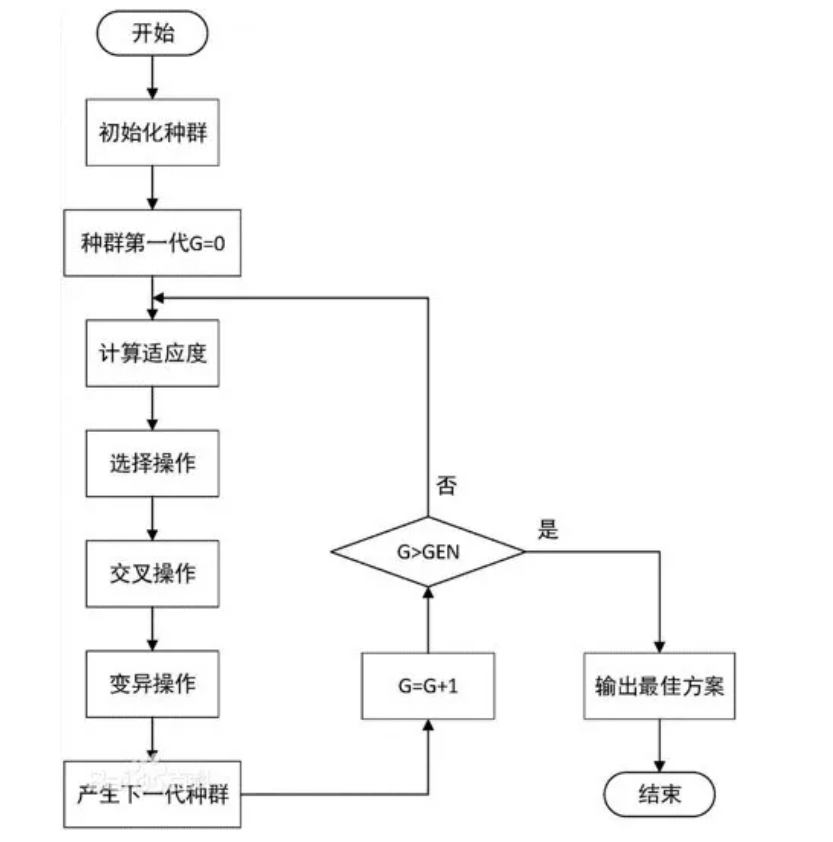
个体（individual）：指染色体带有特征的实体；

种群（population）：个体的集合，该集合内个体数称为种群



遗传算法的一般步骤：

1. 随机产生种群。
2. 根据策略判断个体的适应度，是否符合优化准则（迭代停止条件，如迭代代数限制，见下图），若符合，输出最佳个体及其最优解，结束。否则，进行下一步。
3. 依据适应度选择父母，适应度高的个体被选中的概率高，适应度低的个体被淘汰。
4. 用父母的染色体按照一定的方法进行交叉，生成子代。
5. 对子代染色体进行变异。
6. 由交叉和变异产生新一代种群，返回步骤2，直到最优解产生。



**03 问题引入**

“袋鼠跳”

爬山算法：一只袋鼠朝着比现在高的地方跳去。它找到了不远处的最高的山峰。但是这座山不一定是最高峰。这就是爬山算法，它不能保证局部最优值就是全局最优值。

模拟退火：袋鼠喝醉了。它随机地跳了很长时间。这期间，它可能走向高处，也可能踏入平地。但是，它渐渐清醒了并朝最高峰跳去。这就是模拟退火算法。

遗传算法：有很多袋鼠，它们降落到喜玛拉雅山脉的任意地方。这些袋鼠并不知道它们的任务是寻找珠穆朗玛峰。但每过几年，就在一些海拔高度较低的地方射杀一些袋鼠。于是，不断有袋鼠死于海拔较低的地方，而越是在海拔高的袋鼠越是能活得更久，也越有机会生儿育女。就这样经过许多年，这些袋鼠们竟然都不自觉地聚拢到了一个个的山峰上，可是在所有的袋鼠中，只有聚拢到珠穆朗玛峰的袋鼠被带回了美丽的澳洲。

**04 大体实现过程**

①首先寻找一种对问题潜在解进行“数字化”编码的方案。（建立表现型和基因型的映射关系）

编码是应用遗传算法时要解决的首要问题，也是设计遗传算法时的一个关键步骤。编码方法影响到交叉算子、变异算子等遗传算子的运算方法，在很大程度上决定了遗传进化的效率。迄今为止人们已经提出了许多种不同的编码方法。总的来说，这些编码方法可以分为三大类：二进制编码法、浮点编码法、符号编码法。；

二进制编码的缺点是：对于一些连续函数的优化问题，由于其随机性使得其局部搜索能力较差，如对于一些高精度的问题（如求连续函数的最大值），当解迫近于最优解后，由于其变异后表现型变化很大，不连续，所以会远离最优解，达不到稳定。一定长度的二进制编码序列，只能表示一定精度的浮点数。假如我们要求解精确到六位小数，若区间长度为 [-1,2] ,为了保证精度要求，至少要把区间分为3 × 10^6等份，由于2^21 = 2097152 < 3\*10^6 < 2^22 = 4194304，故此时需要至少22位编码。

浮点法，是指个体的每个基因值用某一范围内的一个浮点数来表示。在浮点数编码方法中，必须保证基因值在给定的区间限制范围内，遗传算法中所使用的交叉、变异等遗传算子也必须保证其运算结果所产生的新个体的基因值也在这个区间限制范围内。如1.2-3.2-5.3-7.2-1.4-9.7

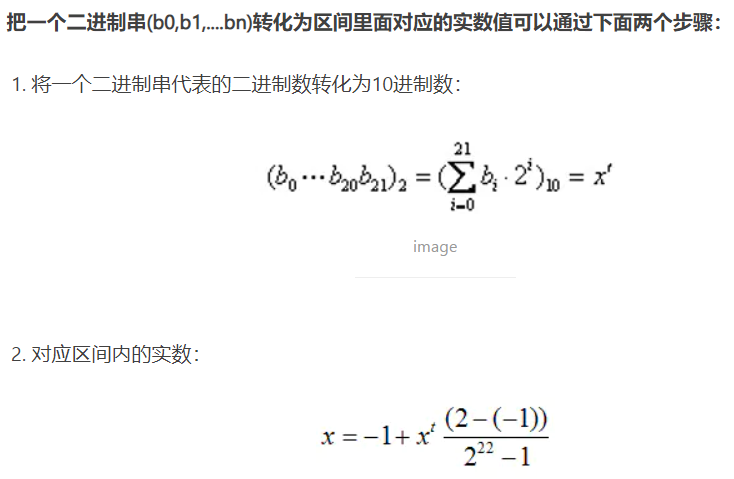
浮点数编码法，适用于在遗传算法中表示范围较大的数；适用于精度要求较高的遗传算法；便于设计针对问题的专门知识的知识型遗传算子；便于处理复杂的决策变量约束条件

符号编码法是指个体染色体编码串中的基因值取自一个无数值含义、而只有代码含义的符号集如｛A,B,C…｝，优点主要是便于在遗传算法中利用所求解问题的专门知识。

②随机初始化一个种群（那么第一批袋鼠就被随意地分散在山脉上），种群里面的个体就是这些数字化的编码。

③接下来，通过适当的解码过程之后（得到袋鼠的位置坐标）。

如何利用上面的编码来为我们的袋鼠染色体编码呢？这里的**表现型可以理解为个体特征**（比如身高、体重、毛色等等）。那么，在此问题下，我们关心的个体特征就是：袋鼠的位置坐标（因为我们要把海拔低的袋鼠给杀掉）。无论袋鼠长什么样，爱吃什么。我们关心的始终是袋鼠在哪里，并且只要知道了袋鼠的位置坐标（位置坐标就是相应的染色体编码，可以通过解码得出）



如一个二进制串(1000101110110101000111)2通过上面换算以后，表示实数值0.637197。实际上，编码的方式千奇百怪，层出不穷，每个问题可能采用的编码方式都不一样。

④用适应性函数对每一个基因个体作一次适应度评估（袋鼠爬得越高当然就越好，所以适应度相应越高）。

适应度函数也称评价函数，是根据目标函数确定的用于区分群体中个体好坏的标准。**适应度函数总是非负的，而目标函数可能有正有负，故需要在目标函数与适应度函数之间进行变换**。在本例的袋鼠跳中，我们只关心袋鼠的海拔高度，以此来判断是否该射杀该袋鼠。这样一来，该函数就非常简单了。只要输入袋鼠的位置坐标，在通过相应查找运算，返回袋鼠当前位置的海拔高度就行。

评价个体适应度的一般过程为：a对个体编码串进行解码处理后，可得到个体的表现型; b由个体的表现型可计算出对应个体的目标函数值; c 根据最优化问题的类型，由目标函数值按一定的转换规则求出个体的适应度。

⑤用选择函数按照某种规定择优选择（每隔一段时间，射杀一些所在海拔较低的袋鼠，以保证袋鼠总体数目持平。

遗传算法中的选择操作就是用来确定如何从父代群体中按某种方法选取那些个体，以便遗传到下一代群体。选择操作用来确定重组或交叉个体，以及被选个体将产生多少个子代个体。前面说了，我们希望海拔高的袋鼠存活下来，并尽可能繁衍更多的后代。但我们都知道，在自然界中，适应度高的袋鼠越能繁衍后代，但这也是从概率上说的而已。毕竟有些适应度低的袋鼠也可能逃过我们的眼睛。那么，怎么建立这种概率关系呢？

**下面介绍几种常用的选择算子：**

轮盘赌选择（Roulette Wheel Selection）：是一种回放式随机采样方法。每个个体进入下一代的概率等于它的适应度值与整个种群中个体适应度值和的比例。选择误差较大。

随机竞争选择（Stochastic Tournament）：每次按轮盘赌选择一对个体，然后让这两个个体进行竞争，适应度高的被选中，如此反复，直到选满为止。

最佳保留选择：首先按轮盘赌选择方法执行遗传算法的选择操作，然后将当前群体中适应度最高的个体结构完整地复制到下一代群体中。

无回放余数随机选择：可确保适应度比平均适应度大的一些个体能够被遗传到下一代群体中，因而选择误差比较小。

均匀排序：对群体中的所有个体按期适应度大小进行排序，基于这个排序来分配各个个体被选中的概率。

最佳保存策略：当前群体中适应度最高的个体不参与交叉运算和变异运算，而是用它来代替掉本代群体中经过交叉、变异等操作后所产生的适应度最低的个体.

随机联赛选择：每次选取几个个体中适应度最高的一个个体遗传到下一代群体中.

排挤选择：新生成的子代将代替或排挤相似的旧父代个体，提高群体的多样性。

......

⑥交叉

遗传算法的交叉操作，是指对两个相互配对的染色体按某种方式相互交换其部分基因，从而形成两个新的个体。

适用于二进制编码个体或浮点数编码个体的交叉算子：

单点交叉（One-point Crossover）：指在个体编码串中只随机设置一个交叉点，然后再该点相互交换两个配对个体的部分染色体。

 两点交叉（Two-point Crossover）：在个体编码串中随机设置了两个交叉点，然后再进行部分基因交换。

均匀交叉（也称一致交叉，Uniform Crossover）：两个配对个体的每个基因座上的基因都以相同的交叉概率进行交换，从而形成两个新个体

算术交叉（Arithmetic Crossover）：由两个个体的线性组合而产生出两个新的个体。该操作对象一般是由浮点数编码表示的个体。

但考虑到“袋鼠跳”问题的具体情况――袋鼠的个体特征仅仅表现为它所处的位置。可以想象，同一个位置的袋鼠的基因是完全相同的，而两条相同的基因进行交叉后，相当于什么都没有做，所以我们不打算在这个例子里面使用交叉这一个遗传操作步骤。（当然硬要这个操作步骤也不是不行的，你可以把两只异地的袋鼠捉到一起，让它们交配，然后产生子代，再把它们送到它们应该到的地方。）

⑦让个体基因变异（让袋鼠随机地跳一跳）。

遗传算法中的变异运算，是指将个体染色体编码串中的某些基因座上的基因值用该基因座上的其它等位基因来替换，从而形成新的个体。

基本位变异（Simple Mutation）：对个体编码串中以变异概率、随机指定的某一位或某几位仅因座上的值做变异运算

均匀变异（Uniform Mutation）：分别用符合某一范围内均匀分布的随机数，以某一较小的概率来替换个体编码串中各个基因座上的原有基因值。（特别适用于在算法的初级运行阶段）

边界变异（Boundary Mutation）：随机的取基因座上的两个对应边界基因值之一去替代原有基因值。特别适用于最优点位于或接近于可行解的边界时的一类问题

非均匀变异：对原有的基因值做一随机扰动，以扰动后的结果作为变异后的新基因值。对每个基因座都以相同的概率进行变异运算之后，相当于整个解向量在解空间中作了一次轻微的变动。

⑧然后产生子代（希望存活下来的袋鼠是多产的，并在那里生儿育女）。

⑨判断是否停止迭代