

运筹学与优化方法

晁国清

计算机科学与技术学院



课程回顾

- 对偶分解 Dual Decomposition
- 乘子法 Method of Multipliers
- 交替方向乘子法 Alternating Direction
Method of Multipliers
- 常见的模式 Common Patterns
- 一致性 Consensus



课程内容

- 遗传算法概念
- 遗传算法的发展历史
- 基本遗传算法
- 遗传算法的模式定理
- 遗传算法的特点
- 遗传算法的应用
- 模拟退火算法



遗传算法概念

- 遗传算法 (Genetic Algorithm, GA) 是模拟生物在自然环境中的遗传和进化过程形成的一种自适应全局概率优化算法。
- 再具体些: 就是通过对个体染色体选择 (Selection)、交叉 (Crossover) 与变异 (Mutation) 等操作产生多样性的群体然后通过自然界的选择和代代更新使得最终的群体适应性 (Fitness) 最高
- 核心思想: 物竞天择, 适者生存
(“天”就代表适应度函数, Fitness Function)



遗传算法的发展历史

20世纪60年代，美国密执安大学的Holland教授及其学生们受到生物模拟技术的启发，创造出了一种基于生物遗传和进化机制的适合于复杂系统计算优化的自适应概率优化技术——遗传算法。下面是在遗传算法的发展进程中一些关键人物所作出的一些主要贡献。

➤ J. H. Holland

20世纪60年代，Holland认识到了生物的遗传和自然进化现象与人工自适应系统的相似关系，运用生物遗传和进化的思想来研究自然和人工自适应系统的生成以及它们与环境的关系，提出在研究和设计人工自适应系统时，可以借鉴生物遗传的机制，以群体的方法进行自适应搜索，并且充分认识到了交叉、变异等运算策略在自适应系统中的重要性。



遗传算法的发展历史

➤ J. D. Bagley

1967年，Holland的学生Bagley在其博士论文中首次提出了“遗传算法”一词，并发表了遗传算法应用方面的第一篇论文。他发展了复制、交叉、变异、显性、倒位等遗传算子，在个体编码上使用了双倍体的编码方法。这些都与目前遗传算法中所使用的算子和方法类似。他还敏锐地意识到了在遗传算法执行的不同阶段可以使用不同的选择率，这将有利于防止遗传算法的早熟现象，从而创立了自适应遗传算法的概念。

➤ J. H. Holland

20世纪70年代，Holland提出了遗传算法的基本定理——模式定理 (Schema Theorem)，奠定了遗传算法的理论基础。1975年，Holland出版了第一本系统论述遗传算法和人工自适应系统的专著《自然系统和人工系统的自适应性 (Adaptation in Natural and Artificial Systems) 》。



遗传算法的发展历史

➤ K. A. De Jong

1975年，De Jong在其博士论文中结合模式定理进行了大量的纯数值函数优化计算实验，树立了遗传算法的工作框架，得到了一些重要且具有指导意义的结论。他推荐了在大多数优化问题中都比较适用的遗传算法参数，还建立了著名的De Jong 五函数测试平台，定义了评价遗传算法性能的在线指标和离线指标。

➤ J. H. Holland

20世纪80年代，Holland实现了第一个基于遗传算法的机器学习系统——分类器系统，开创了基于遗传算法学习的新概念，为分类器系统构造出了一个完整的框架。



遗传算法的发展历史

➤ D. J. Goldberg

1989年，Goldberg出版了专著《搜索、优化和机器学习中的遗传算法》。该书系统地总结了遗传算法的主要研究成果，全面而完整地论述了遗传算法的基本原理及其应用。

➤ L. Davis

1991年，Davis编辑出版了《遗传算法手册》，书中包含了遗传算法在科学计算、工程技术和社会经济中的大量应用实例，该书为推广和普及遗传算法的应用起到了重要的指导作用。

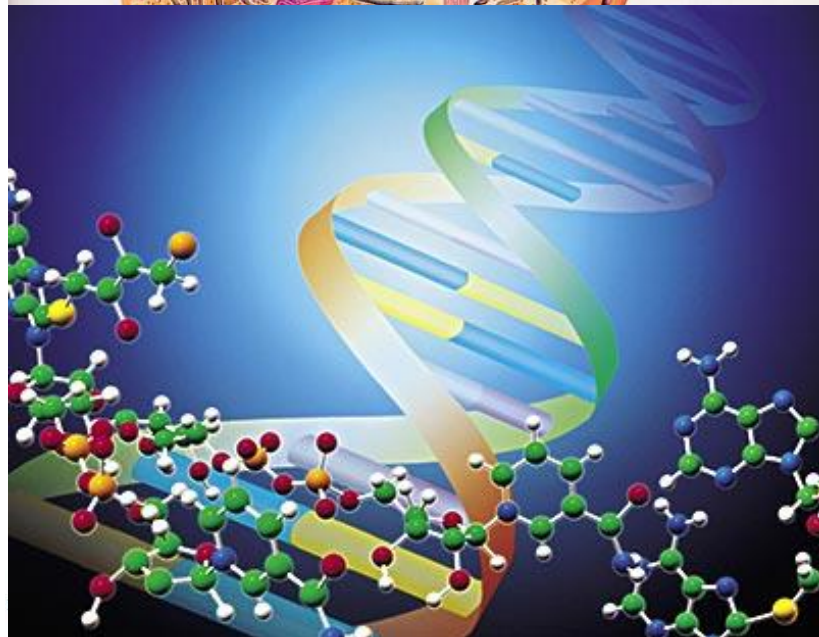
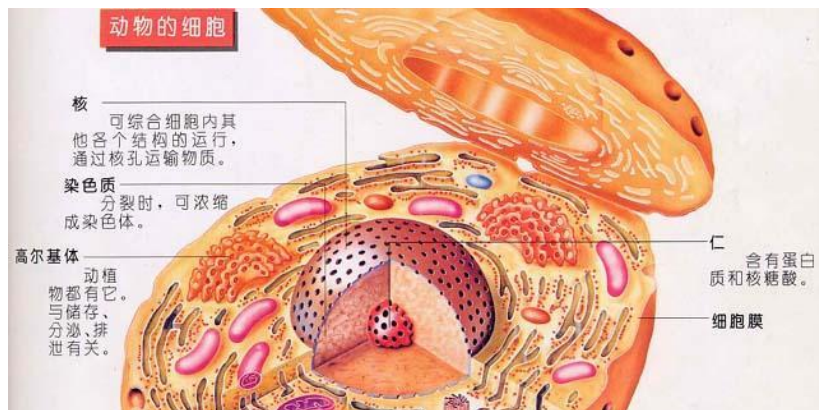
➤ J. R. Koza

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。Koza成功地将提出的遗传编程方法应用于人工智能、机器学习、符号处理等方面。



基本遗传算法

生物学基础：



细胞 (Cell)

染色体
(Chromosome)

脱氧核糖核酸
(Deoxyribonucleic Acid, DNA)

基因 (Gene)、
等位基因 (Allele)

基因型
(Genotype)

表现型
(Phenotype)

“种瓜得瓜，
种豆得豆”

复制 (Reproduction)

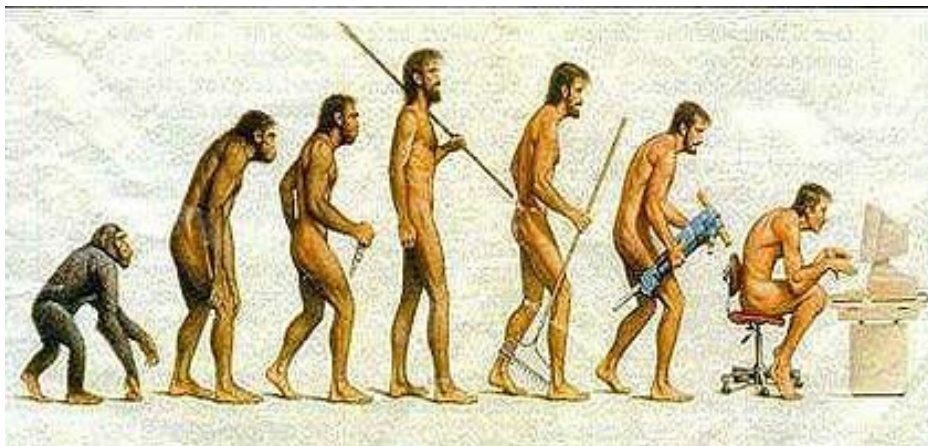
交叉 (Crossover)

变异 (Mutation)

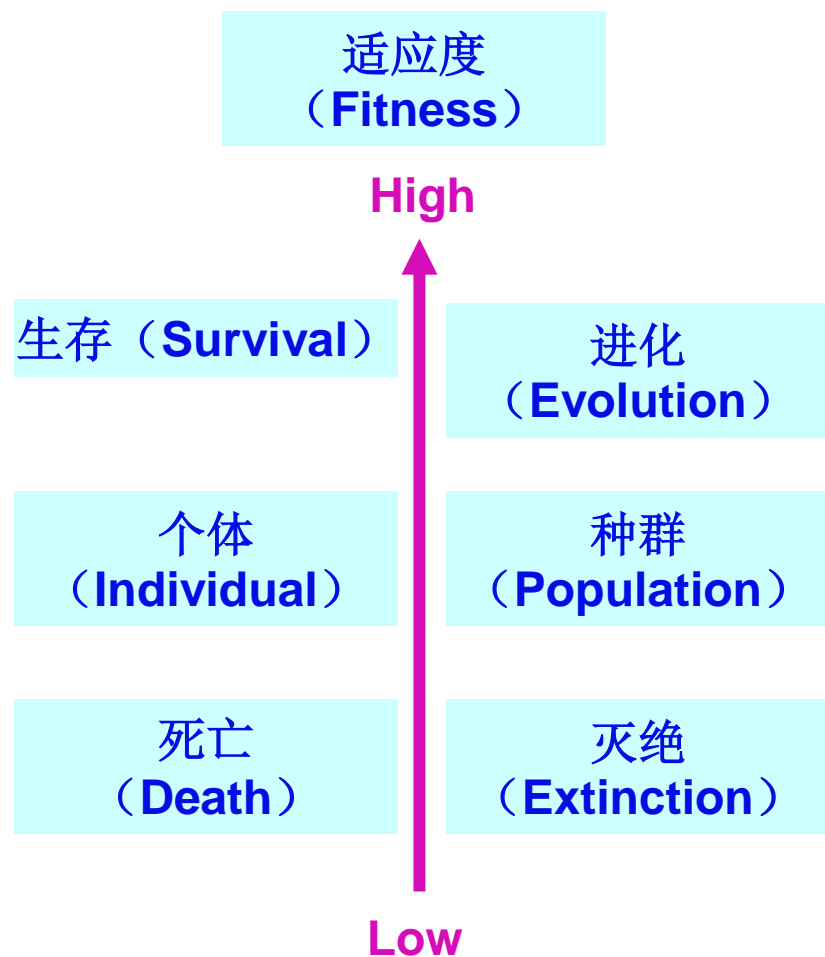
→ 对环境的适应性

基本遗传算法

生物学基础：

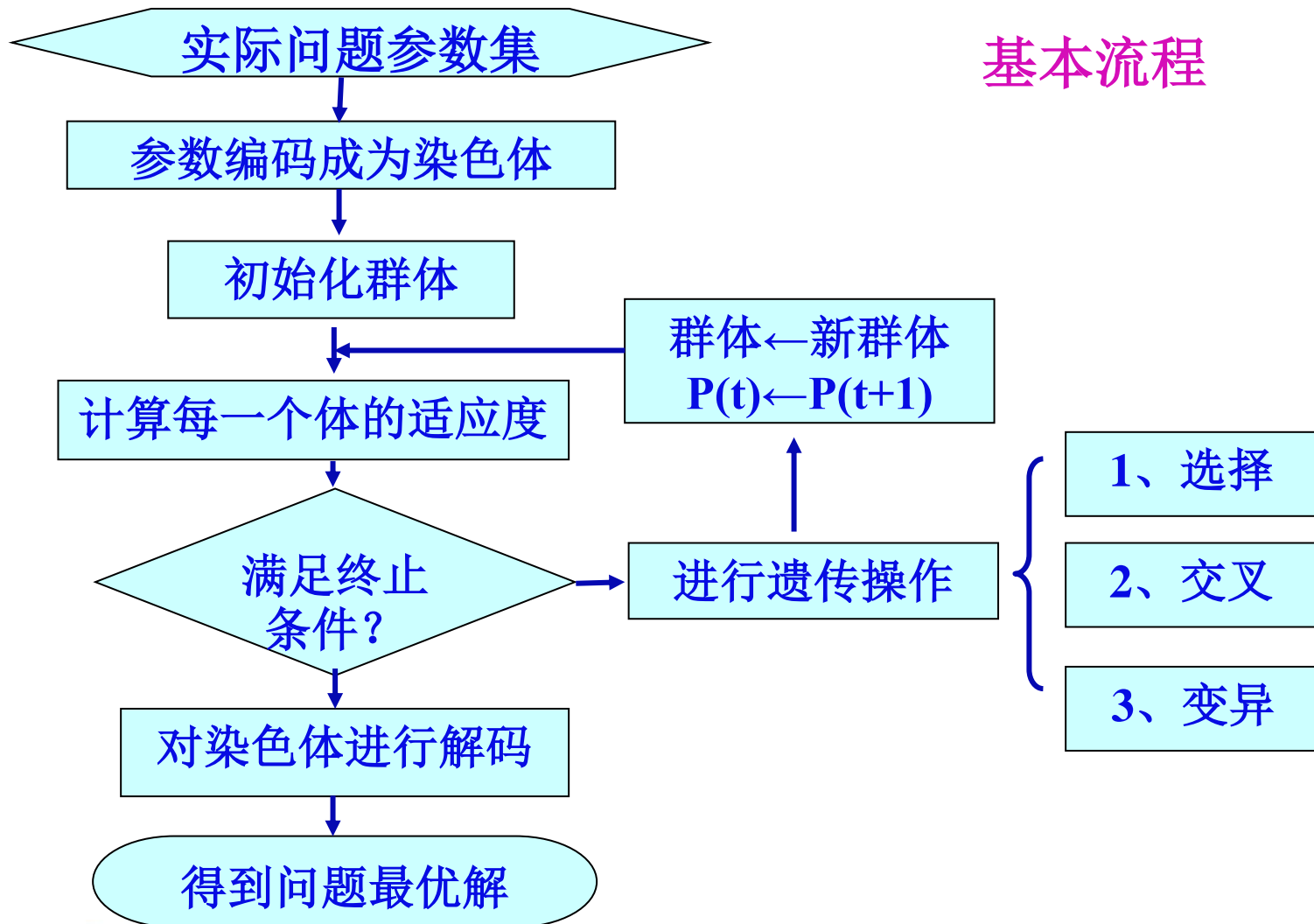


“物竞天择，适者生存”



基本遗传算法

基本流程



基本遗传算法

相关概念术语:

个体(Individual) 称 $S = \{0,1\}^l$ 为个体空间, 个体空间的元素 $a = a_0 a_1 \cdots a_{l-1} \in S$ 称为个体, 它是染色体带有特征的实体。分量 $a_i \in \{0,1\}$ 称为基因, 正整数 l 称为个体的基因长度。

群体(种群, Population) 称个体空间 S 中 N 个个体组成的一个子集(个体允许重复)称为一个群体, 记为 $A = (A_1, A_2, \cdots, A_N)$, 其中 $A_i (i = 1, 2, \cdots, N) \in S$, N 称为群体规模。

适应度(Fitness) 遗传算法会按与个体适应度成正比的概率决定当前群体中个体遗传到下一代群体的机会大小, 所以**要求非负**。在遗传算法中, 一般通过适应度函数 (Fitness function) 来衡量某一个体的适应度高低。

编码(Coding) 将一个待求解问题的实际可行解从其解空间转换到遗传算法所能处理的搜索空间 (即个体空间) 的过程, 称为编码。

基本遗传算法

解码(Decoding) 解码是将遗传算法所搜索到的最优个体的染色体转换成待求解问题的实际最优解的过程，即编码的逆过程。

选择操作(Selection) 根据各个个体的适应度，按照一定的规则，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。一般地，选择操作通过选择算子（Selection Operator）进行。

交叉操作(Crossover) 将群体 $P(t)$ 内的各个个体随机搭配成对，对每一对个体，以某个概率（称为交叉概率，Crossover Rate）遵循某一种规则交换它们之间的部分染色体。

变异操作(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）改变某一个或某一些基因座上的基因值为其他的等位基因。

基本遗传算法

应用步骤:

(1) 确定决策变量及各种约束条件, 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, 确定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。

编码是遗传算法解决问题的先决条件和关键步骤:

① 不仅决定个体基因的排列形式 (从而决定选择与繁殖等操作的作用方式), 而且也决定从搜索空间的基因型到解空间的表现型的解码方式 (从而决定对GA所获解的翻译与理解);

② 决定GA搜索的困难度与复杂性;

③ 决定对问题的求解精度。



基本遗传算法

应用步骤:

常用的编码方法主要有：二进制编码、浮点数编码等。二进制编码比浮点数编码搜索能力强，但浮点数编码比二进制编码在变异操作上能够保持更好的种群多样性。

基本遗传算法多采用二进制编码，将决策变量用二进制字符串表示，二进制编码串的长度由所求精度决定。然后将各决策变量的二进制编码串连接在一起，构成一个染色体。

例如：变量 x 的定义域为 $[-2,3]$ ，要求精度为 10^{-5} ，则需将 $[-2,3]$ 分成至少500 000个等长小区域，每个小区域用一个二进制串表示。则 $2^L = 500000$ ， $\log_2 500000 \approx 18.93$ ，向上取整，可得 $L = 19$ 。即可用19位二进制串 $a_{18}a_{17} \cdots a_0$ 表示。

基本遗传算法

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 的对应关系和转换方法。

例如: 对于二进制编码, 其解码过程如下: 若 X^* 的取值范围为 $[X_l, X_r]$, 参数的二进制编码码长为 L , 码串对应的十进制整数为 k , 则解码公式为:

$$k = \sum_{i=0}^{L-1} a_i 2^i, \quad X = \frac{(X_r - X_l)k}{2^L - 1} + X_l$$

式中, X_l, X_r 分别表示参数最小值和最大值, L 表示参数编码长度, k 表示二进制串对应的实数值。



基本遗传算法

应用步骤：

(5) 确定个体适应度的量化评价方法，就是确定出由目标函数值 $f(X)$ 到个体适应度 $Fit(f(X))$ 的转换规则。主要有三种方法：

①直接以待求解的目标函数为适应度函数

若目标函数为最大化问题，则 $Fit(f(X))=f(X)$

若目标函数为最小化问题，则 $Fit(f(X))=-f(X)$

优点：简单直观；

缺点：其一，可能不满足非负的要求；其二，某些代求解的函数值分布相差很大，由此得到的平均适应度可能不利于体现种群的平均性能

②界限构造法

若目标函数为最大化问题，则

$$Fit(f(X)) = \begin{cases} C_{min} + f(X), & f(X) + C_{min} > 0 \\ 0, & f(X) + C_{min} \leq 0 \end{cases}$$

C_{min} 是适当小的数，获取方法有：预先指定，进化到当前代为止的最小目标函数值，当前代或最近几代群体中的最小目标值



基本遗传算法

应用步骤:

若目标函数为最小化问题, 则

$$Fit(f(X)) = \begin{cases} C_{max} - f(X), & f(X) < C_{max} \\ 0, & f(X) \geq C_{max} \end{cases}$$

C_{max} 是适当大的数, 获取方法有: 预先指定, 进化到当前代为止的最大目标函数值, 当前代或最近几代群体中的最大目标值

③倒数法:

若目标函数为最大化问题, 则

$$Fit(f(X)) = \frac{1}{1 + c - f(X)}, \quad c \geq 0, c - f(x) \geq 0$$

若目标函数为最小化问题, 则

$$Fit(f(X)) = \frac{1}{1 + c + f(X)}, \quad c \geq 0, c + f(x) \geq 0$$

C 为目标函数界限的保守估计值



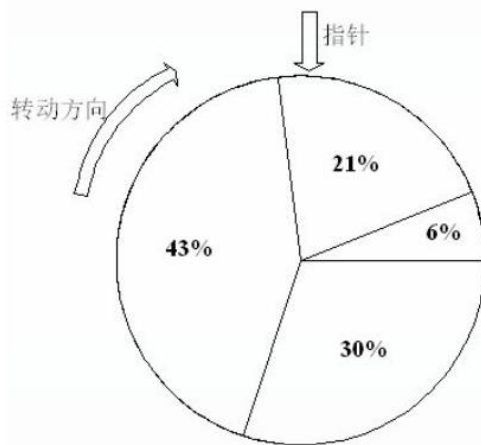
基本遗传算法

应用步骤:

(6) 确定各遗传操作方法

①**选择算子**: 比例选择算子是一种有退换的随机选择, 也叫做赌盘选择(Roulette Wheel Selection)。若某个个体 i , 其适应度为 f_i , 则其被选取的概率表示为 $p_i = \frac{f_i}{\sum_{i=1}^N f_i}$, M 表示个体数目。

按适应度大小切分区域大小, 即适应度越大的个体占比越大, 越有可能被选中, 同时由于是随机选取, 也保证了适应度小个体也有被选中的可能。



/* once of roulette wheel selection

* 输出参数:

* 选中的染色体

*/

procedure RWS

1 $m \leftarrow 0$;

2 $r \leftarrow \text{Random}(0,1)$; //0至1的随机数

3 **for** $i=1$ to N

4 $m \leftarrow m + P_i$;

5 **if** $r \leq m$

6 **return** i ;

7 **end if**

8 **end for**

end procedure

https://blog.csdn.net/qq_44185838



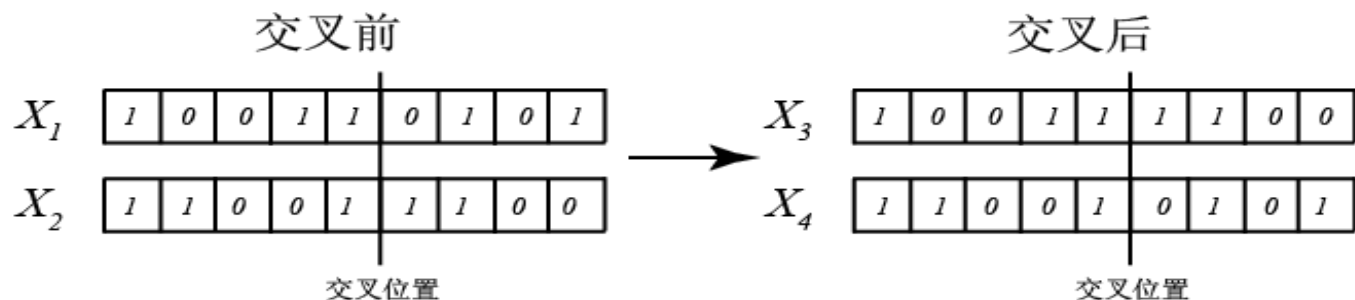
基本遗传算法

应用步骤:

②交叉率和交叉算子: 交叉, 也叫基因重组(Recombination), 是遗传算法获取新的优良个体的最重要手段, 决定了遗传算法的全局搜索能力。

交叉率 p_c 就是对随机配对的个体在交叉点进行交叉的概率。较大的交叉率使各代充分交叉, 但群体中的优良模式更可能遭到破坏, 以致产生较大的代沟, 从而使搜索走向随机化; 交叉率越小, 产生的代沟越小, 就会使得更多的个体直接复制到下一代, 遗传搜索可能陷入停滞状态, 一般取值0.4~0.99。

对于二进制编码, 常用的交叉方法有: 单点交叉、多点交叉和均匀交叉等。一个单点交叉的例子如下图所示。



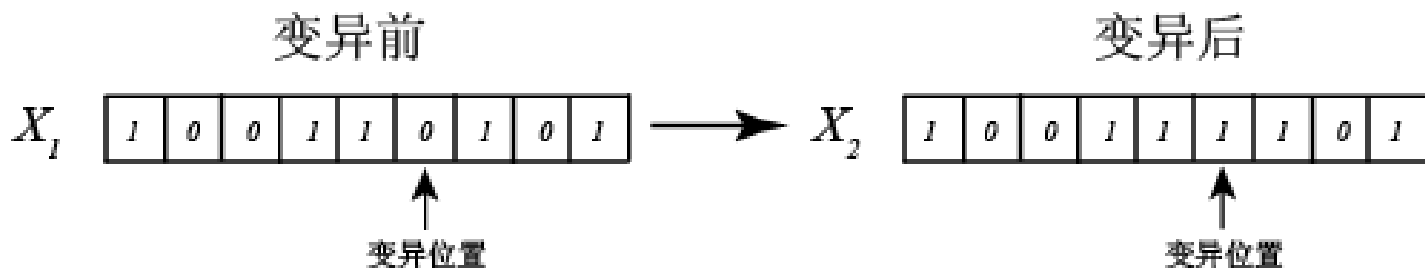
基本遗传算法

应用步骤:

③变异率和变异算子: 变异本身是一种局部随机搜索, 使遗传算法具有局部的随机搜索能力; 同时使得遗传算法保持种群的多样性, 以防止出现非成熟收敛。

变异率 p_m 一般可取0.001~0.1。变异率不能取得太大, 如果大于0.5, 遗传算法就退化为随机搜索, 而遗传算法的一些重要的数学特性和搜索能力也不复存在了。

常用的变异操作方法是二进制变异法



基本遗传算法

应用步骤:

(7) 确定遗传算法的有关运行参数, 包括群体规模 M (Population Size, 一般取值100~500)、迭代次数 T (一般取为100~500)、选择算子、交叉率、变异率等等。

(8) 初始化群体。

①初始群体一般随机产生

②初始值最好能在解空间中均匀采样 (收敛速度比较快)

③对于非二进制编码, 还要考虑所生成的染色体是否在可行区域内。

(9) 计算群体中个体解码后的适应值。

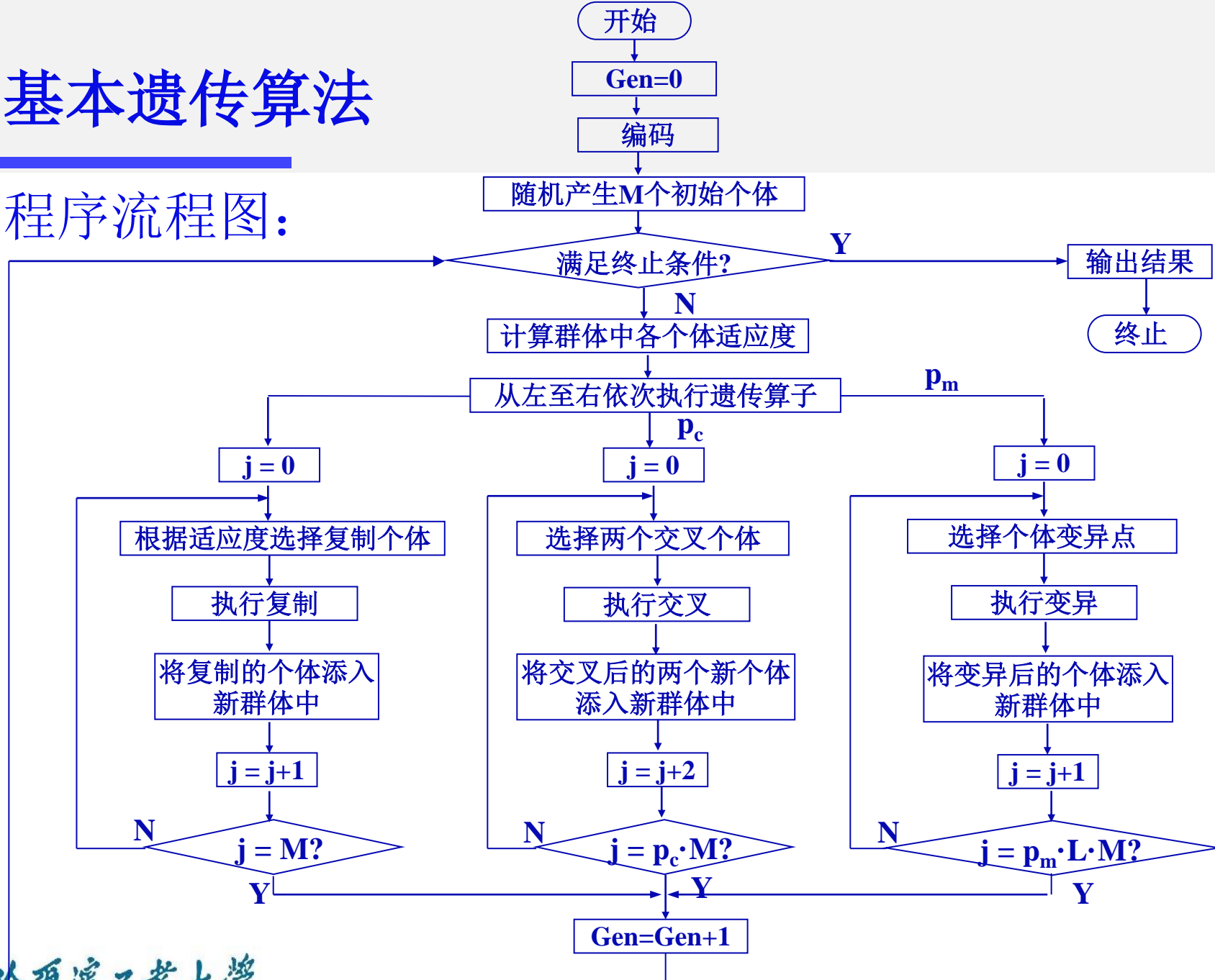
(10) 按照遗传策略, 运用所选定的选择、交叉和变异算子作用于群体, 生成下一代群体。

(11) 判断群体性能是否满足某一指标或是否完成预定迭代次数, 不满足则返回 (9)。



基本遗传算法

程序流程图:



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，那些低阶、定义长度短、超过群体平均适应值的模式的生存数量，将随着迭代次数的增加以指数规律增长。



遗传算法的特点

- 遗传算法以决策变量的编码作为运算对象。传统的优化算法通常利用决策变量本身作为运算对象。以决策变量编码作为运算对象，便于引入和应用遗传操作算子。
- 遗传算法直接以目标函数值作为搜索信息。传统的优化算法往往不只需要目标函数值，还需要目标函数的导数等其他信息。这样对许多目标函数无法求导或很难求导的函数而言，遗传算法就比较方便。



遗传算法的特点

- 遗传算法同时进行解空间的多点搜索。传统的优化算法通常从解空间的一个初始点开始搜索，容易陷入局部极值点。遗传算法进行群体搜索，并在搜索过程中引入遗传运算，使群体可不断进化。该特性也可称为隐含并行性。
- 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、变异等运算都以概率方式进行，从而增加了搜索过程的灵活性。



遗传算法的应用

遗传算法提供了一种求解复杂系统优化问题的通用框架，它不依赖于问题的具体领域，对问题的种类有很强的鲁棒性，所以广泛应用于很多学科。下面列举一些遗传算法的主要应用领域。

- **函数优化**。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。对于一些非线性、多模型、多目标的函数优化问题，用其他优化方法较难求解，而遗传算法却可以方便地得到较好的结果。
- **组合优化**。遗传算法是寻求组合优化问题(旅行商问题、背包问题、装箱问题)满意解的最佳工具之一，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。
- **生产调度问题**。生产调度问题在很多情况下所建立起来的数学模型难以精确求解，即使经过一些简化之后可以进行求解，也会因简化得太多而使求解结果与实际相差太远。遗传算法成为解决复杂调度问题的有效工具。



遗传算法的应用

- **自动控制**。遗传算法已经在自动控制领域中得到了很好的应用，例如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。
- **机器人学**。机器人是一类复杂的难以精确建模的人工系统，而遗传算法的起源就来自于对人工自适应系统的研究，所以机器人学自然成为遗传算法的一个重要应用领域。
- **图像处理**。图像处理过程如扫描、特征提取、图像分割等不可避免存在一些误差，会影响图像处理的效果。如何使误差最小是使计算机视觉达到实用化的重要要求，遗传算法正好适用。



遗传算法的应用

- **人工生命**。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。自组织能力和自学习能力是人工生命的两大重要特征。人工生命与遗传算法有着密切的关系，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。
- **遗传编程**。Koza发展了遗传编程的概念，他使用了以LISP语言所表示的编码方法，基于对一种树形结构所进行的遗传操作来自动生成计算机程序。
- **机器学习**。基于遗传算法的机器学习可用于调整人工神经网络的连接权，也可以用于人工神经网络的网络结构优化设计。



模拟退火算法

模拟退火算法(simulated annealing, 简称SA)的思想最早是由Metropolis等(1953)提出的, 1983年Kirkpatrick等将其用于组合优化。SA算法是基于Monte Carlo迭代求解策略的一种随机寻优算法, 其出发点是基于物理中固体物质的退火过程与一般组合优化问题之间的相似性。

模拟退火算法在某一初温下, 伴随温度参数的不断下降, 结合概率突跳特性在解空间中随机寻找目标函数的全局最优解, 即在局部最优解能概率性地跳出并最终趋于全局最优解。模拟退火算法是一种通用的优化算法, 目前已在工程中得到了广泛应用。



模拟退火算法

1. 物理退火过程和Metropolis准则

简单而言，物理退火过程由以下三部分组成：

- (1)加温过程。其目的是增强粒子的热运动，使其偏离平衡位置。当温度足够高时，固体将溶解为液体，从而消除系统原先可能存在的非均匀态，使随后进行的冷却过程以某一平衡态为起点。溶解过程与系统的熵增过程联系，系统能量也随温度的升高而增大。
- (2)等温过程。物理学的知识告诉我们，对于与周围环境交换热量而温度不变的封闭系统，系统状态的自发变化总是朝自由能减少的方向进行，当自由能达到最小时，系统达到平衡态。
- (3)冷却过程。目的是使粒子的热运动减弱并渐趋有序，系统能量逐渐下降，从而得到低能的晶体结构。



模拟退火算法

Metropolis等在1953年提出了重要性采样法，即以概率接受新状态。具体而言，在温度 t ，由当前状态 i 产生新状态 j ，两者的能量分别为 E_i 和 E_j ，若 $E_j < E_i$ ，则接受新状态 j 为当前状态；否则，若概率 $p_r = \exp - \frac{E_j - E_i}{kt}$ 大于 $[0,1)$ 区间内的随机数，则仍旧接受新状态为 j 当前状态，若不成立则保留 i 为当前状态，其中 k 为Boltzmann常数。

这种重要性采样过程在高温下可接受与当前状态能量差较大的新状态，而在低温下基本只接受与当前能量差较小的新状态，而且当温度趋于零时，就不能接受比当前状态能量高的新状态。这种接受准则通常称为Metropolis准则。



模拟退火算法

2. 模拟退火算法的基本思想和步骤

1983年Kirkpatrick等意识到组合优化与物理退火的相似性，并受到Metropolis准则的启迪，提出了模拟退火算法。模拟退火算法是基于Monte Carlo 迭代求解策略的一种随机寻优算法，其出发点是基于物理退火过程与组合优化之间的相似性，SA由某一较高初温开始，利用具有概率突跳特性的Metropolis抽样策略在解空间中进行随机搜索，伴随温度的不断下降重复抽样过程，最终得到问题的全局最优解。



模拟退火算法

标准模拟退火算法的一般步骤可描述如下：

(1) 给定初温 $t = t_0$, 随机产生初始状态 $s = s_0$, 令 $k = 0$;

(2) Repeat:

1) Repeat

产生新状态 $s_j = Genete(s)$;

if $\min \left\{ 1, \exp \left[- \left(C(s_j) - C(s) \right) \right] \right\} \geq random[0,1]$

$s = s_j$

Until 抽样稳定准则满足;

2) 退温 $t_{k+1} = update(t_k)$, 并令 $k = k + 1$;

Until 算法终止准则满足;

(3) 输出算法搜索结果



模拟退火算法

3. 模拟退火算法关键参数和操作的设定

从算法流程上看，模拟退火算法包括三函数两准则，即状态产生函数、状态接受函数、温度更新函数、内循环终止准则和外循环终止准则，这些环节的设计将决定SA算法的优化性能。此外，初温的选择对SA算法性能也有很大影响。

(1) 状态产生函数

设计状态产生函数（邻域函数）的出发点应该是尽可能保证产生的候选解遍布全部的解空间。通常，状态产生函数由两部分组成，即产生候选解的方式和候选解产生的概率分布。

(2) 状态接受函数

状态接受函数一般以概率的方式给出，不同接受函数的差别主要在于接受概率的形式不同。设计状态接受概率，应该遵循以下原则：



模拟退火算法

- ①在固定温度下，接受使目标函数值下降的候选解的概率要大于使目标值上升的候选解的概率；
- ②随温度的下降，接受使目标函数值上升的解的概率要逐渐减小；
- ③当温度趋于零时，只能接受目标函数值下降的解。

状态接受函数的引入是SA算法实现全局搜索的最关键的因素，SA算法中通常采用 $\min[1, \exp(-\Delta C/t)]$ 作为状态接受函数。

(3)初温

初始温度、温度更新函数、内循环终止准则和外循环终止准则通常被称为退火历程(annealing schedule)。实验表明，初温越大，获得高质量解的几率越大，但花费的计算时间将增加。因此，初温的确定应折中考虑优化质量和优化效率，常用方法包括：



模拟退火算法

- ①均匀抽样一组状态，以各状态目标值的方差为初温。
- ②随机产生一组状态，确定两两状态间的最大目标值差 $|\Delta_{max}|$ ，然后依据差值，利用一定的函数确定初温。譬如 $t_0 = -\Delta / \ln p_r$ ，其中 p_r 为初始接受概率。
- ③利用经验公式给出。

(4)温度更新函数

温度更新函数，即温度的下降方式，用于在外循环中修改温度值。

目前，最常用的温度更新函数为指数退温函数，即 $t_{k+1} = \lambda t_k$ 其中 $0 < \lambda < 1$ 且其大小可以不断变化。



模拟退火算法

(5)内循环终止准则

内循环终止准则，或称Metropolis抽样稳定准则，用于决定在各温度下产生候选解的数目。在非时齐SA算法理论中，由于在每个温度下只产生一个或少量候选解，所以不存在选择内循环终止准则的问题。而在时齐SA算法理论中，收敛条件要求在每个温度下产生候选解的数目趋于无穷大，以使相应的马氏链达到平稳概率分布，显然在实际应用算法时这是无法实现的。

常用的抽样准则包括：

- ①检验目标函数的均值是否稳定；
- ②连续若干步的目标值变化较小；
- ③按一定的步数抽样。



模拟退火算法

(6)外循环终止准则

外循环终止准则，即算法终止准则，用于决定算法何时结束。设置温度终值是一种简单的方法。SA算法的收敛性理论中要求温度终值趋于零，这显然不合实际。

通常的做法是：

- ①设置终止温度的阈值；
- ②设置外循环迭代次数；
- ③算法收敛到的最优值连续若干步保持不变；
- ④检验系统熵是否稳定。

参考文献

数模智能算法之遗传算法 老教练

运筹学与优化方法 第11章 P269–281

A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications (2021) 80:8091–8126

