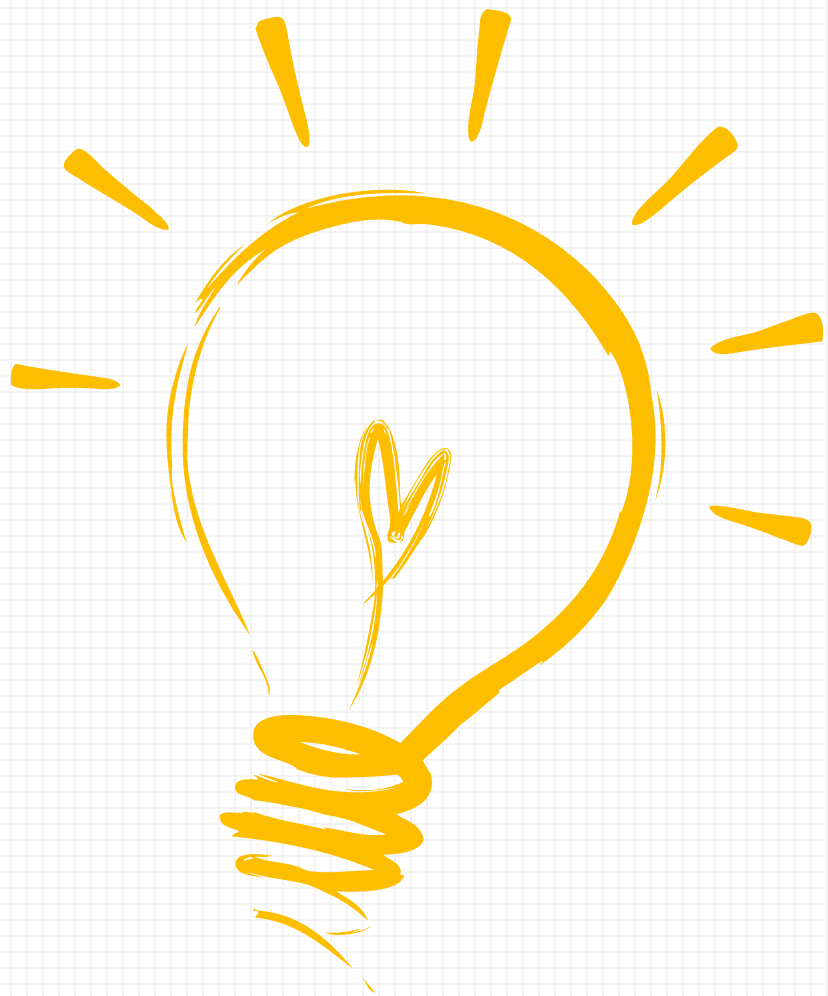


CONTENTS



- 01 算法背景**
- 02 算法原理**
- 03 算法分析**
- 04 算法拓展**
- 05 案例实操**

di

第

yi

一

zhang

章

jie

节

算法背景

1.1 背景

遗传算法（Genetic Algorithm, GA）是模拟生物在自然环境中的遗传和进化的过程而形成的自适应全局优化搜索算法。

遗传算法借鉴了达尔文的进化论和孟德尔的遗传学说。其本质是一种并行、高效、全局搜索的方法，它能在搜索过程中自动获取和积累有关搜索空间的知识，并自适应地控制搜索过程以求得最优解。遗传算法操作使用“适者生存”的原则，在潜在的解决方案种群中逐次产生一个近似最优的方案。在遗传算法的每一代中，根据个体在问题域中的适应度值和从自然遗传学中借鉴来的再造方法进行个体选择，产生一个新的近似解。这个过程导致种群中个体的进化，得到的新个体比原个体更能适应环境，就像自然界中的改造一样。

di

第

yi

二

zhang

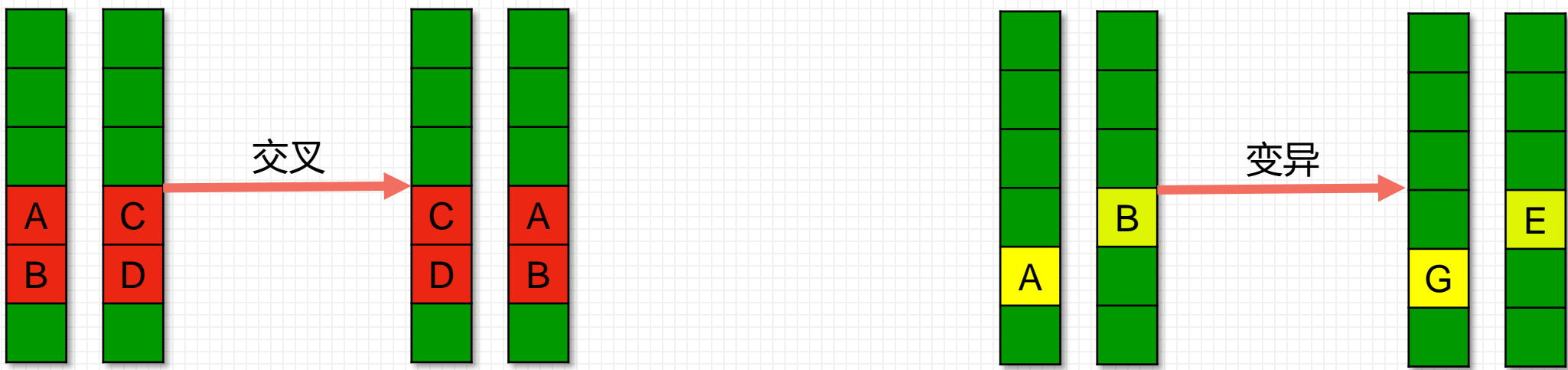
章

jie

节

算法原理

2.1 基本原理

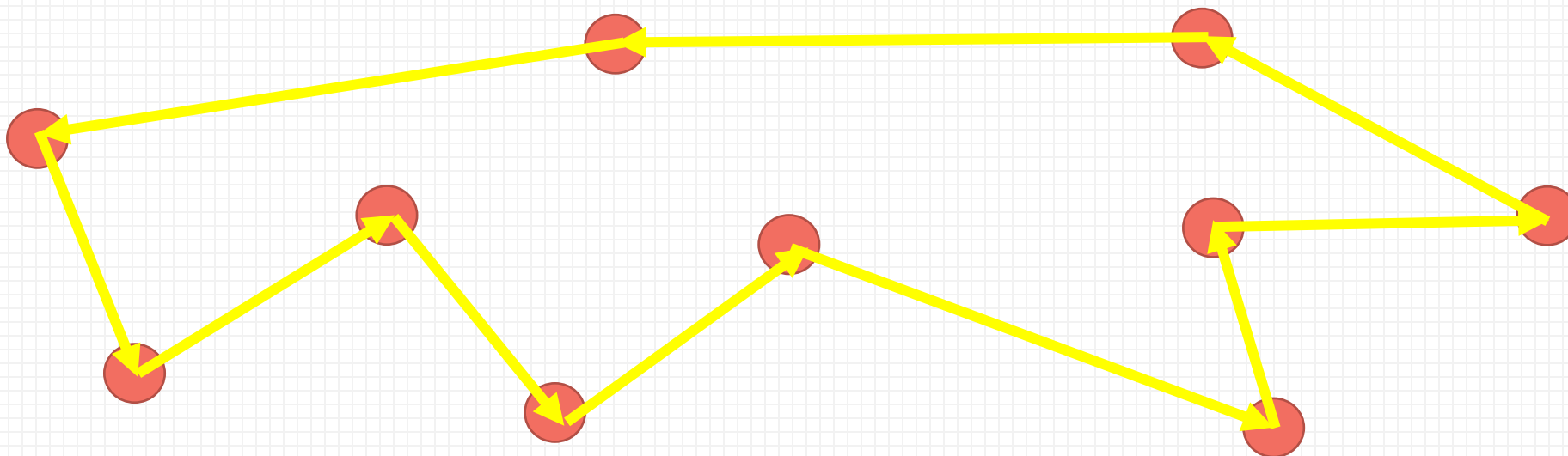


对每个个体都进行上述的进化，然后择优录取。

2.1 基本原理-编码

编码： 将问题的可行解，抽象编码为适用于遗传算法的形式。

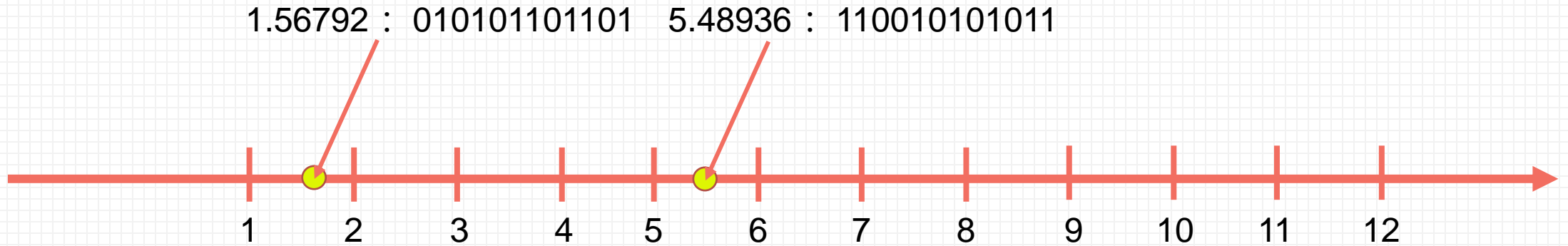
- 二进制编码，将数值转为二进制串，用以求解问题。
- TSP问题，比如10个城市，某个解可以表示为[3,2,1,4,5,7,6,8,9,0]。
- 根据不同的问题，进行不同的抽象实现。



2.1 基本原理-编码

二进制编码：用一个二进制串**表示**这个十进制数值。

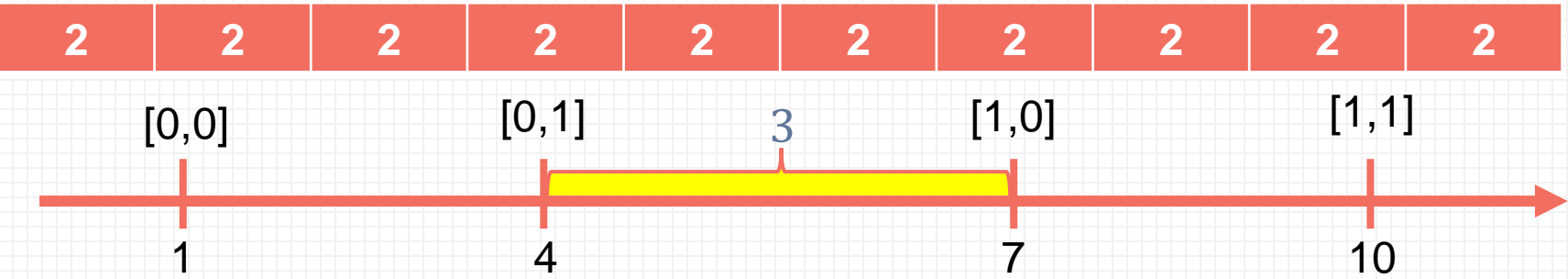
- 给定数值解的区间范围：[1,10]。
- 给定精度：1e-5，两个数值解的间隔。
- 进行编码：为每个数值解**分配**一个独一无二的二进制串。



2.1 基本原理-编码

要想实现每个数值都能分配一个独一无二的串，那么串所能表示的数值个数就要大于等于数值解的个数。

一个长度为 n 的串，能表示多少个数呢？ 2^n 。



区间范围为 $[1,10]$ ，长度为 2 的串提供的精度为多少？对应精度为 $\frac{10-1}{2^2-1} = 3$ 。

于是数值**区间长度为L**，精度为E的条件下，二进制串的长度 n ，三者关系为 $\frac{L}{2^{n-1}} \leq E$

要想达到 $1e-5$ 的精度，对于**区间长度为10**，串的长度 n 应该满足 $\frac{10}{2^{n-1}} \leq 0.00001 \rightarrow n \geq 20$ 。

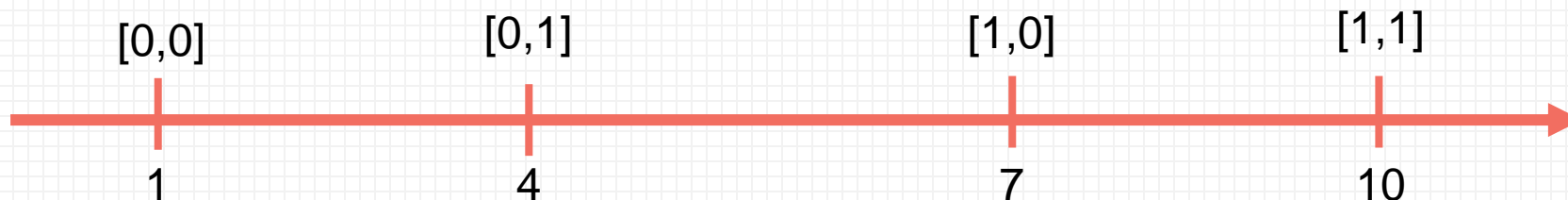
$n = 20$ 时，串提供的精度约为0.00000954。

2.1 基本原理-解码

二进制串的索引：就是在问当前串是第几个串。可以使用二进制转十进制得到。

以[1,0]为例，其十进制转换结果为 $0*2^0 + 1 * 2^1 = 2$,

对应表示的数值为： $7 = 1 + 2 * 3$



以[1,1]为例，其十进制转换结果为 $1*2^0 + 1 * 2^1 = 3$.

对应表示的数值为： $10 = 1 + 3 * 3$

一般的，区间范围为 $[a,b]$ ，区间长度为 L ，即 $L = b - a$ ，串长为 n ，当前串对应十进制为 T ，则该串对应实值解为：

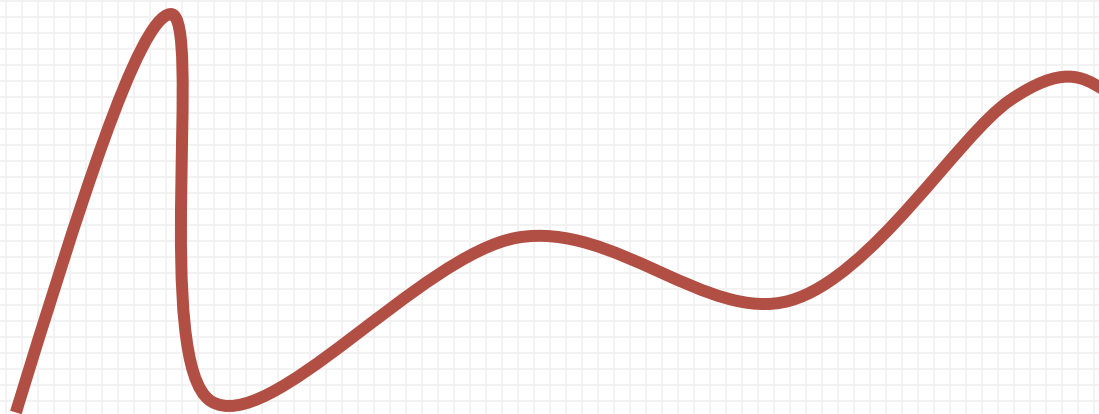
$$X = a + T * \frac{b-a}{2^n - 1}$$

2.1 基本原理-复制交叉

个体进行复制，以概率 p_c 进行基因的交叉，注意**复制交叉方式多种多样**。

- 复制

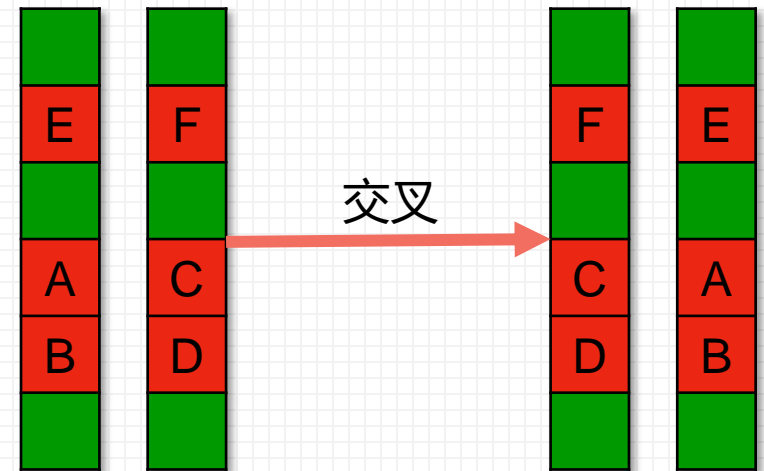
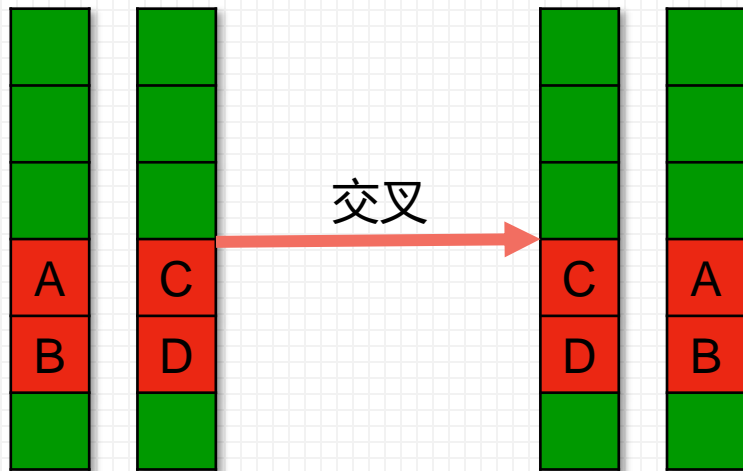
- 将个体适应度大小映射为概率进行复制：适应度高的个体有更大概率复制，且复制的份数越多-**轮盘赌法**。
- 对适应度高的前N/4的个体进行复制，然后用这些个体把后N/4个体替换掉- **精英产生精英**。
- 一定是将当前个体的复制体将下一个个体替换掉吗？随机可以吗？YES!
- 一定只能把“坏解”替换掉吗？把随机某个适应度高的解替换掉呢？YES!



2.1 基本原理-复制交叉

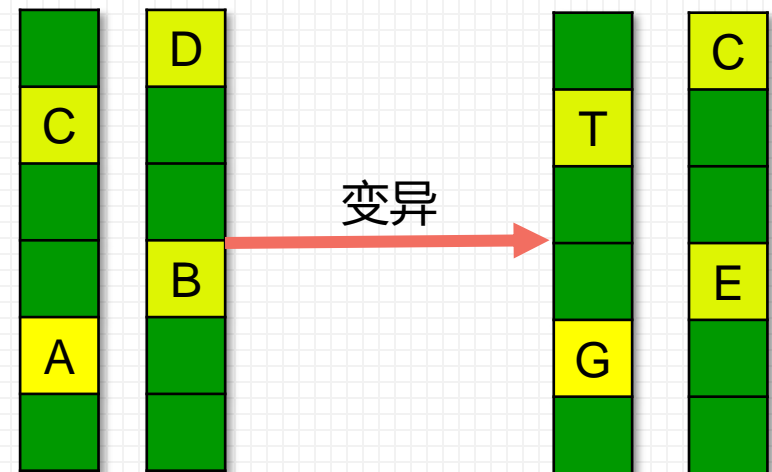
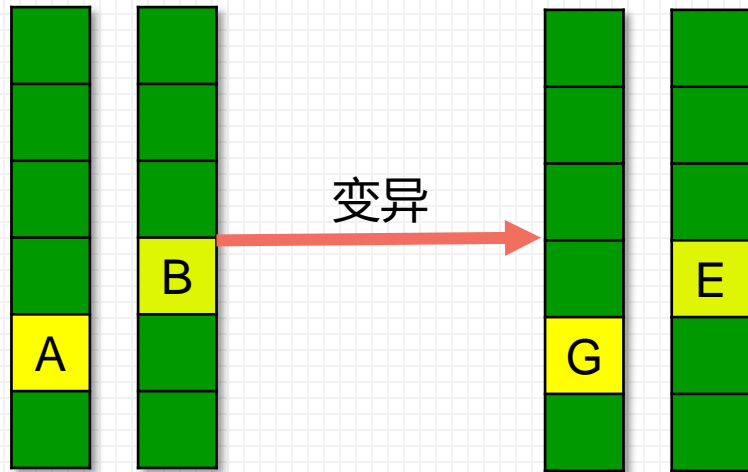
- 交叉

- 按顺序，两两个体之间按概率交叉。如1和2,2和3等。或者1和2,2和4等，或者1和4这样。
- 必须两两交叉？3个可以吗？YES!。5个？YES!
- 对适应度高的前N/2个体、甚至N/4的个体之间相互交叉？YES!
- 一定是按顺序交叉吗？对每个个体随机从前N/2中选一个个体交叉？YES!
- 一定是只有一段交叉？多段呢？YES!



2.1 基本原理-变异

- 变异
 - 每个个体都进行变异。
 - 只对适应度低的后N/4的，或者后N/2个个体变异？ YES!
 - 必须都变异吗？ 按适应度大小映射为概率变异？ YES!
 - 一定是只有一个位点变异？ 多个位点呢？ YES!
- 还有一个绝妙的策略，但这里地方太小，写不下了（狗头）。



2.2 算法实现

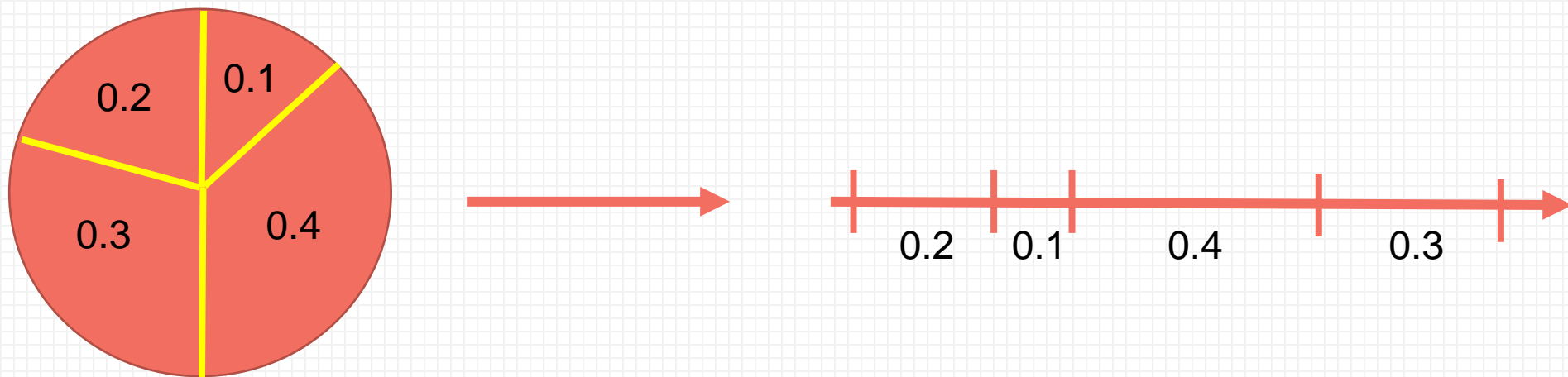
复制：按适应度大小映射为概率，进行轮盘赌复制。

交叉：1和2,3和4，以一定概率决定是否交叉。若交叉，则二者选择随机一个段进行交叉。

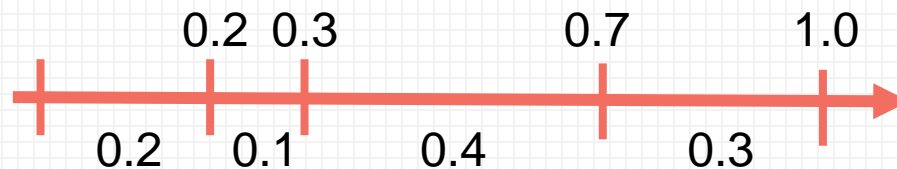
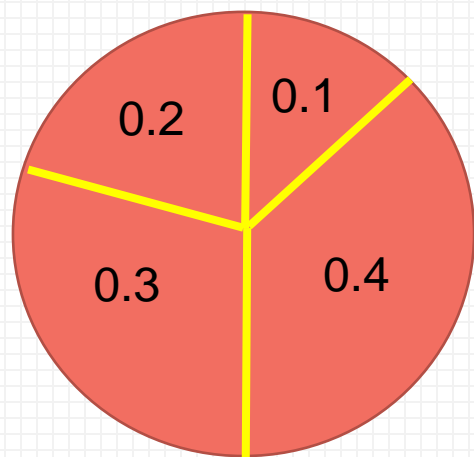
变异：按照一定概率决定该个体是否变异，若变异，随机选择一个位点进行变异：按位取反。

轮盘赌基本思想：适应度越高的解，按道理越应该高概率的进行复制，且复制的份数应该越多。

- 对于个体 x_i ，计算对应适应度 $f(x_i)$ - $\rightarrow p_i = f(x_i) / \sum f(x_i)$



2.2 算法实现

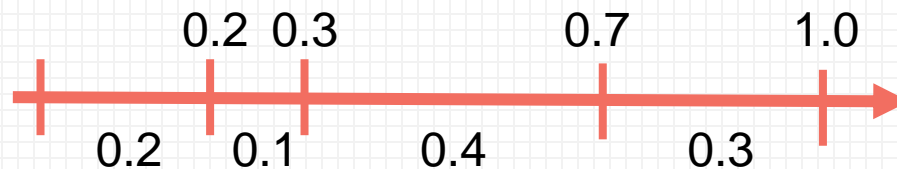
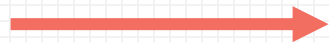
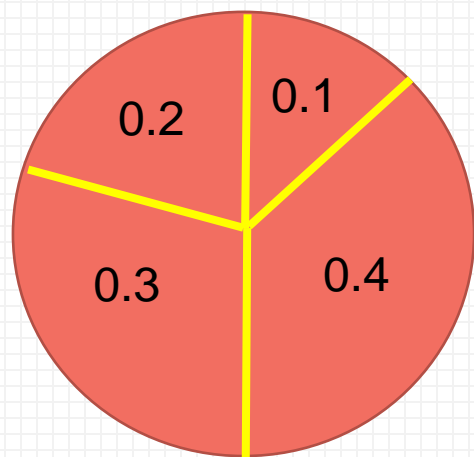


Rand = 0 ~ 0.2 : 表示对 x1 复制
 Rand = 0.2~0.3: 表示对 x2 复制
 Rand = 0.3~0.7 : 表示对 x3 复制
 Rand = 0.7~1.0: 表示对 x4 复制

0.2 = 0.2
 0.3 = 0.2 + 0.1
 0.7 = 0.3 + 0.4
 1.0 = 0.7 + 0.3

$p_i = f(x_i) / \sum f(x_i) // 0.2 \ 0.1 \ 0.4 \ 0.3$
 $Cs = \text{cumsum}(p_i) // 0.2 \ 0.3 \ 0.7 \ 1.0$
 $i = 1, j = 1$
 $R = \text{sort}(\text{rand}(4)) // 0.1 \ 0.5 \ 0.6 \ 0.8$
 While $j \leq n$
 if $R(j) < Cs(i)$
 $\text{newx}(j) = \text{oldx}(i)$
 $j = j + 1$
 else
 $i = i + 1$

2.2 算法实现



$i = 1, j = 1 : Cs = 0.2, R = 0.1 \text{ newx}(1) = \text{oldx}(1)$
 $l = 1, j = 2 : Cs = 0.2, R = 0.5$ -----
 $l = 2, j = 2 : Cs = 0.3, R = 0.5$ -----
 $l = 3, j = 2 : Cs = 0.7, R = 0.5 \text{ newx}(2) = \text{oldx}(3)$

·
·
·

相当于把原来乱序的判断做一个顺序的调整，这样可以使得

$$R < 0.7 \Leftrightarrow 0.3 < R < 0.7$$

$$p_i = f(x_i) / \sum f(x_i) // 0.2 \ 0.1 \ 0.4 \ 0.3$$

$$Cs = \text{cumsum}(p_i) // 0.2 \ 0.3 \ 0.7 \ 1.0$$

$i = 1, j = 1$

$R = \text{sort}(\text{rand}(4)) // 0.1 \ 0.5 \ 0.6 \ 0.8$

While $j \leq n$

if $R(j) < Cs(i)$

$\text{newx}(j) = \text{oldx}(i)$

$j = j + 1$

else

$i = i + 1$

di

第

san

三

zhang

章

jie

节

算法分析

3.1 算法分析

优点:

- 1) 参数少, 理论优势。
- 2) 变异机制赋予了群体跳出局部极值的能力等。

缺点:

- 1) 容易陷入局部最优、算法实现较为繁琐等。

di

第

si

四

zhang

章

jie

节

算法拓展

4.1 算法拓展

总是有同学问，XX算法你会吗？在这里我想和大家说一下：这类启发式算法的背后思想都是一样的，基本思想就是择优进化，好的根据**一定策略**实现更好，差的根据**一定策略**向好的发展。哪怕下一次产生的解不好，大不了不更新就可以了，如果更好，就更新。这就会有一个**保底机制**。

无论是蚁群算法，粒子群算法，免疫算法，鱼群算法，蝙蝠算法以及本次课程的遗传算法，算法思想内涵是一致的：

保底机制 + 更新策略 +



= 寻得优解。

为了说明这个观点，在这里使用MATLAB实现一个函数，为大家做演示。

代码的思想非常简单，随机生成一个初始解，每次循环再随机生成一个解，如果新的解比旧解好，就更新（择优进化），否则不更新。

我愿称之为随机大法(Stochastic Big)，简称SB算法。

di

第

wu

五

zhang

章

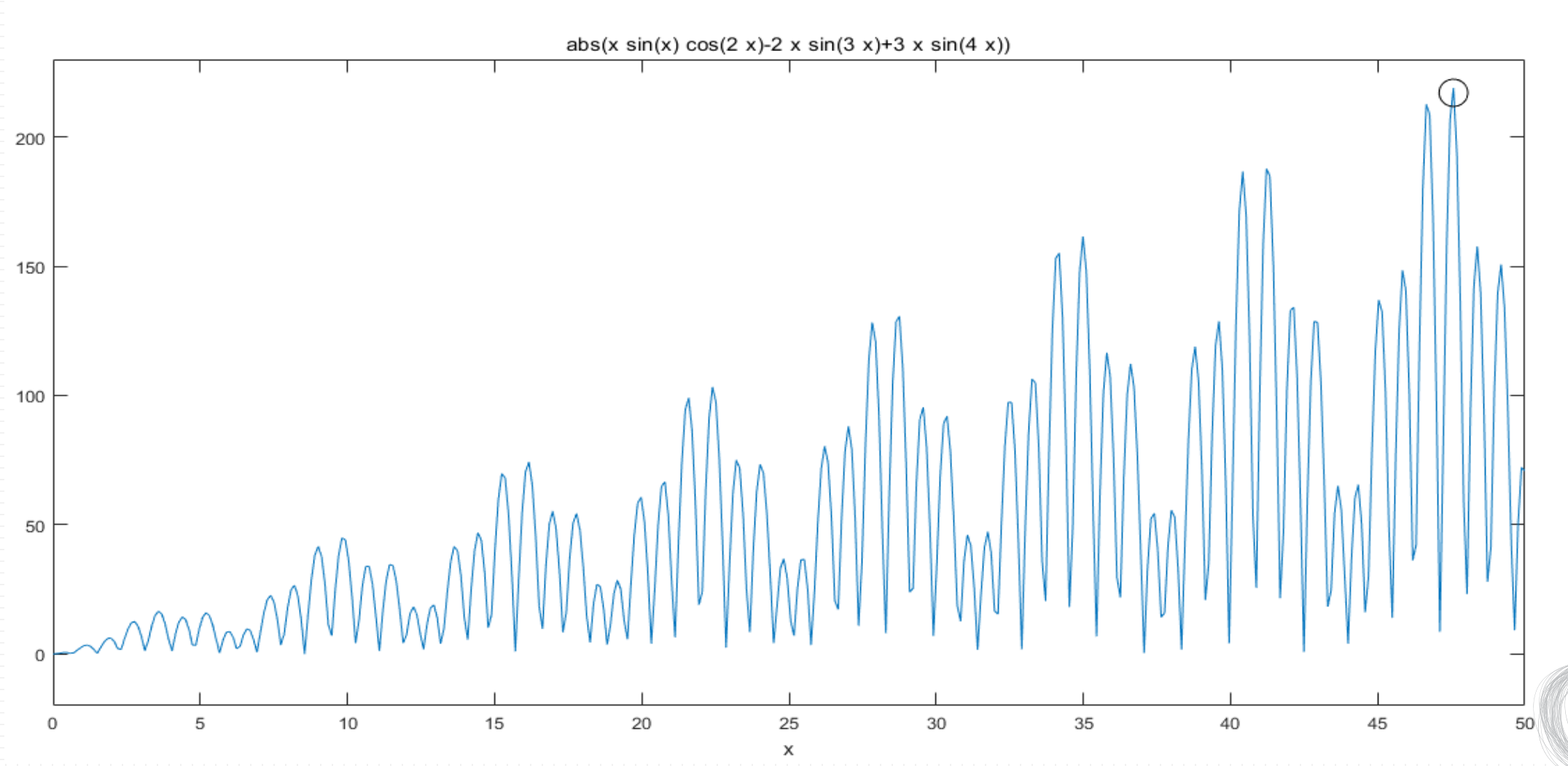
jie

节

案例实操

5.1 案例实操

1: 求解 $f(x) = \text{abs}(x \sin(x)\cos(2x) - 2x\sin(3x) + 3x \sin(4x))$ 在 $[0,50]$ 的最大值。



5.2 广告时间

案例代码见评论区自己下载或者联系我QQ索要。

其他课程:

基于Python实现网络爬虫: <https://www.bilibili.com/video/BV1WV411U7LQ>

通俗易懂关联规则: <https://www.bilibili.com/video/BV13f4y1k7x6>

通俗易懂K均值聚类: <https://www.bilibili.com/video/BV16v4y1Z7xJ>

通俗易懂ID3分类: <https://www.bilibili.com/video/BV1kg411u7RP>

通俗易懂随机森林: <https://www.bilibili.com/video/BV18U4y1E7jQ>

同时如果有一些其他问题, 可以联系我

QQ : 1366420642, Q群: 1019030249

欢迎大佬萌新加入

参 考 资 料

暂无。

THANKS

谢谢观看

