人工智能

演化计算

中国科学院计算技术研究所
Institute Of Computing Technology Chinese Academy Of Sciences

罗平 luop@ict.ac.cn

模拟
与规律

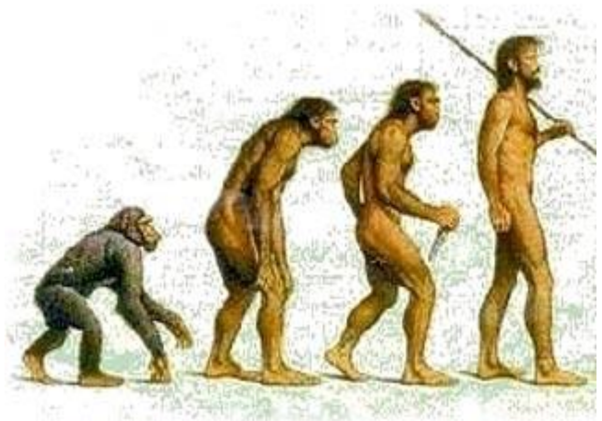
演化计算 概念

演化计算（Evolutionary Computation, EC）：

- **在基因和种群层次上模拟自然界生物进化过程与机制的问题求解技术和计算模型。**
- **思想源于生物遗传学和适者生存的自然规律**
- **基于达尔文（Darwin）的进化论和孟德尔（Mendel）的遗传变异理论**

典型代表：

- **遗传算法（Genetic Algorithm，GA）**
- **进化策略（Evolutionary Strategy,ES）**
- **进化规划（Evolutionary Programming,E**
- **遗传规划（Genetic Programming,GP）**



演 化 计 算

- 达尔文的自然选择学说是一种被人们广泛接受的生物进化学说：
  - 生物要生存下去，就必须进行生存斗争。
  - 具有**有利变异的个体容易存活**下来，并且有更多的机会将有利变异传给后代；具有**不利变异的个体就容易被淘汰**，产生后代的机会也少的多。
  - **适者生存，不适者淘汰**：自然选择。
  - **遗传和变异**是决定生物进化的内在因素。（相对稳定+新的物种）

演化计算及其生物学基础

演化计算：一种模拟自然界生物进化过程与机制进行问题求解的自组织、自适应的随机搜索技术。

- **演化规则：“物竞天择、适者生存”**
- **演化操作：**
  - **繁殖（Reproduction）**
  - **变异（Mutation）**
  - **竞争（Competition）**
  - **选择（Selection）**

遗传算法

- 遗传算法的基本思想是从初始种群出发，采用优胜劣汰、适者生存的自然法则选择个体，并通过杂交、变异来产生新一代种群，如此逐代进化，直到满足目标为止

- 基本概念：
  - **种群(Population)**：多个备选解的集合。
  - **个体(Individual)**：种群中的单个元素，通常由一个用于描述其基本遗传结构的数据结构来表示。例如，长度为L 的0、1串

遗传算法

- 基本概念:
  - **适应度（Fitness）函数：**用来对种群中各个个体的环境适应性进行度量的函数，函数值是遗传算法实现优胜劣汰的主要依据
  - **遗传操作（Genetic Operator）**：作用于种群而产生新的种群的操作。
    - ✓ **选择（Selection）**
    - ✓ **交叉（Cross-over）**
    - ✓ **变异（Mutation）**

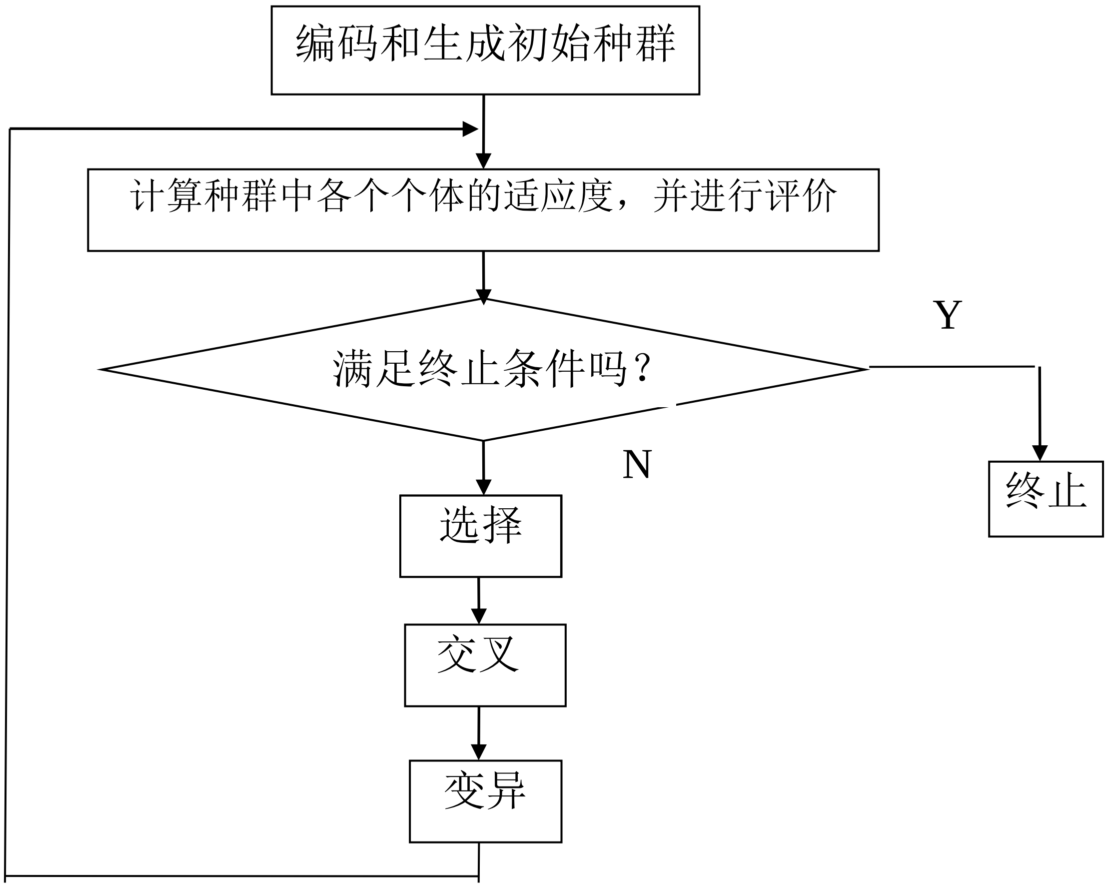
遗传算法

遗传算法主要由染色体编码、初始种群设定、适应度函数设定、遗传操作设计等几大部分所组成，

算法基本步骤：

1. 选择编码策略，将问题搜索空间中每个可能的点用相应的编码策略表示出来，即形成染色体；
2. **定义遗传策略，包括种群规模N，交叉、变异方法，以及选择概率Pr、交叉概率Pc、变异概率Pm等遗传参数；**
3. 令t=0，随机选择N个染色体初始化种群P(0)；
4. **定义适应度函数f；**
5. 计算P(t)中每个染色体的适应值；
6. t=t+1；
7. 运用选择算子，从P(t-1)中得到P(t)；
8. 对P(t)中的每个染色体，按概率Pc参与交叉；
9. 对染色体中的基因，以概率Pm参与变异运算；
10. 判断群体性能是否满足预先设定的终止标准，若不满足返回(5)。

遗传算法



遗传算法与生物进化之间对应关系

遗传算法	生物进化
适应函数	环境
适应函数值	适应性
适应函数值最大的解被保留的概率最大	适者生存
问题的一个解	个体
解的编码	染色体
编码的元素	基因
被选定的一组解	群体
根据适应函数选择的一组解（以编码形式表示）	种群
以一定的方式由双亲产生后代的过程	繁殖
编码的某些分量发生变化的过程	变异

遗传编码

- 二进制编码（Binary encoding）

　　二进制编码是将原问题的结构变换为染色体的位串结构。假设某一参数的取值范围是[A，B]，A<B。用长度为L的二进制编码串来表示该参数，将[A，B]等分成2^L-1个子部分，记每一个等分的长度为δ。

例：假设变量x的定义域为[5，10)，要求的计算精度为10^{-5}，则需要将[5，10)至少分为1000000个等长小区间，每个小区间用一个二进制串表示。于是，串长至少等于20，原因是：$524288=2^{19}<1000000<2^{20}=1048576$ 这样，对应于区间[5，10)内满足精度要求的每个值x，都可用一个20位编码的二进制串<b19,b18,...,b0>来表示。

- 优点：易于理解和实现，可表示的模式数最多
- 主要缺点：海明悬崖。 例如，7和8的二进制数分别为0111和1000，当算法从7改进到8时，就必须改变所有的位。

遗传编码

- 格雷编码（Gray encoding）

  要求两个连续整数的编码之间只能有一个码位不同，其余码位都是完全相同的。有效地解决了海明悬崖问题。

  - 基本原理：
    - 二进制码->格雷码（编码）：从最右边一位起，依次将每一位与左边一位异或(XOR)，作为对应格雷码该位的值，最左边一位不变；
    - 格雷码->〉二进制码（解码）：从左边第二位起，将每位与左边一位解码后的值异或，作为该位解码后的值，最左边一位依然不变。

遗传编码

- 符号编码（Symbol encoding）

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。

例如，对于TSP问题，采用符号编码方法，按一条回路中城市的次序进行编码，一般情况是从城市w_1开始，依次经过城市w_2，……，w_n，最后回到城市w_1，我们就有如下编码表示：

$$w_1, w_2, \ldots, w_n$$

由于是回路，记$w_{n+1} = w_1$。
它其实是1，……，n的一个循环排列。
要注意w_1，w_2，……，w_n是互不相同的。

适应度函数

- 适应度函数是一个用于对个体的适应性进行度量的函数。个体的适应度值越大，它被遗传到下一代种群中的概率越大

- 常用的适应度函数
  - **原始适应度函数: 直接将待求解问题的目标函数f(x)定义为遗传算法的适应度函数。**

    ✓ **例如：求最大值 $\max_{x\in[a,b]} f(x)$ 时，f(x)即为x的原始适应度函数。**

    ✓ **优点**：能够直接反映出待求解问题的最初求解目标，

    ✓ **缺点**：有可能出现适应度值为负的情况

适应度函数

- TSP的目标是路径总长度为最短，路径总长度可作为TSP问题的适应度函数：

$$f(w_1 w_2 \ldots w_n) = \frac{1}{\sum_{j=1}^{n} d(w_j, w_{j+1})}$$

适应度函数

- **标准适应度函数**

在遗传算法中，一般要求适应度函数非负，并其适应度值越大越好。这就往往需要对原始适应函数进行某种变换，将其转换为标准的度量方式，以满足进化操作的要求，这样所得到的适应度函数被称为标准适应度函数$f_{Normal}(x)$。

✓ **例：对极小化问题，其标准适应度函数可定义为**

$$f(x)=\begin{cases} f_{\max}(x)-f(x) & \text{当} f(x)<f_{\max}(x) \\ 0 & \text{否则} \end{cases}$$

其中，f_{max} (x)是原始适应函数f(x)的一个上界。如果f_{max} (x) 未知，则可用当前代或到目前为止各演化代中的f(x)的最大值来代替。 f_{max} (x) 是会随着进化代数的增加而不断变化的。

适应度函数

- ✓ **极大化问题**

  对极大化问题，其标准适应度函数可定义为

$$
f(x)=\begin{cases} f(x)-f_{\min}(x) & \text{当} f(x)>f_{\min}(x) \\ 0 & \text{否则} \end{cases}
$$

其中，$f_{min}(x)$是原始适应函数f(x)的一个下界。如果$f_{min}(x)$ 未知，则可用当前代或到目前为止各演化代中的f(x)的最小值来代替。

基本遗传操作

遗传算法中的基本遗传操作包括选择、交叉和变异。

- 选择(selection)操作：根据选择概率按某种策略从当前种群中挑选出一定数目的个体，使它们能够有更多的机会被遗传到下一代中
- 常用的选择策略：比例选择。

比例选择

基本思想是：各个个体被选中的概率与其适应度大小成正比。

基本遗传操作

- 轮盘赌选择

  个体被选中的概率取决于该个体的相对适应度。而相对适应度的定义为：

$$P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

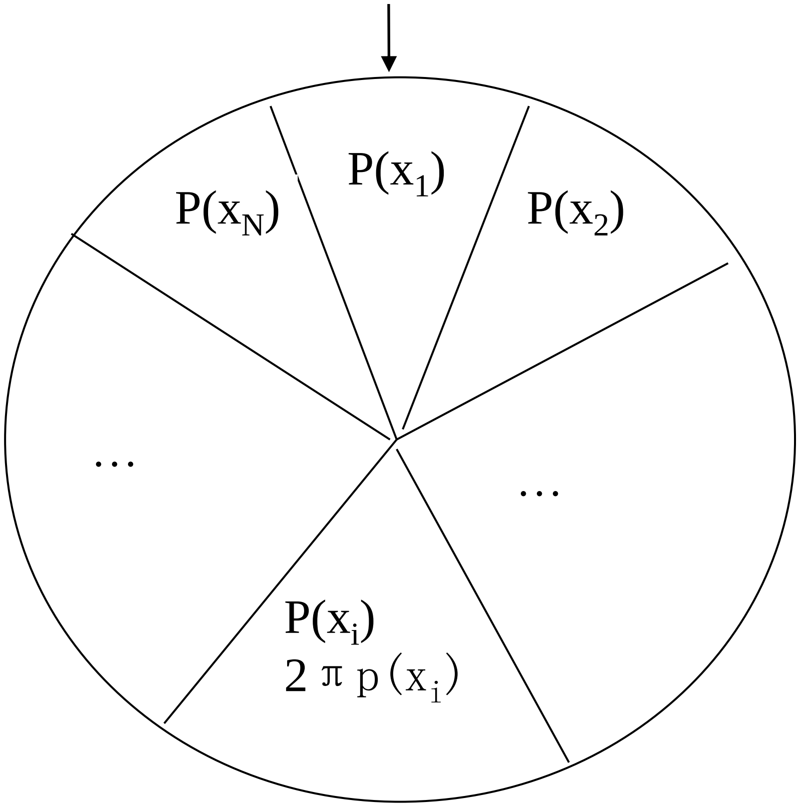
其中，$P(x_i)$是个体x_i的相对适应度，即个体x_i被选中的概率；$f(x_i)$是个体x_i的原始适应度。

轮盘赌选择算法的基本思想是：根据每个个体的选择概率$P(x_i)$将一个圆盘分成N个扇区，其中第i个扇区的中心角为：

$$2\pi\frac{f(x_i)}{\sum_{j=1}^{N} f_i(x_j)}=2\pi p(x_i)$$

并再设立一个固定指针。当进行选择时，可以假想转动圆盘，若圆盘静止时指针指向第i个扇区，则选择个体i。

基本遗传操作



从统计角度看，个体的适应度值越大，其对应的扇区的面积越大，被选中的可能性也越大。这种方法有点类似于发放奖品使用的轮盘，并带有某种赌博的意思，因此亦被称为轮盘赌选择。

基本遗传操作

- 交叉(crossover)操作：按照某种方式对选择的父代个体的染色体的部分基因进行交配重组，从而形成新的个体。

- 常见的交叉操作：

  ①二进制交叉：二进制编码情况下所采用的交叉操作，它主要包括：

  - **单点交叉**
  - **两点交叉**
  - **多点交叉**
  - **均匀交叉**

基本遗传操作

- 单点交叉

先在两个父代个体的编码串中**随机设定一个交叉点**，然后对这两个父代个体**交叉点前面或后面部分的基因进行交换**，并生成子代中的两个新的个体。

假设两个父代的个体串分别是：

$X=x_1\ x_2\ \dots\ x_k\ x_{k+1}\ \dots\ x_n$ $\qquad$ $Y=y_1\ y_2\ \dots\ y_k\ y_{k+1}\ \dots\ y_n$

随机选择第k位为交叉点，交叉后生成的两个新的个体是：

$X'=x_1\ x_2\ \dots\ x_k\ y_{k+1}\ \dots\ y_n$ $\qquad$ $Y'=y_1\ y_2\ \dots\ y_k\ x_{k+1}\ \dots\ x_n$

设有两个父代的个体串A=0 0 1 1 0 1 和B=1 1 0 0 1 0 ，若随机交叉点为4，则交叉后生成的两个新的个体是：

A'= 0 0 1 1 1 1

B'= 1 1 0 0 0 0

基本遗传操作

- 两点交叉

先在两个父代个体的编码串中**随机设定两个交叉点**，然后再**按这两个交叉点进行部分基因交换**，生成子代中的两个新的个体。

假设两个父代的个体串分别是：

$X=x_1\ x_2\ \dots\ x_i\ \dots\ x_j\ \dots\ x_n$　　　　$Y=y_1\ y_2\ \dots\ y_i\ \dots\ y_j\ \dots, y_n$

随机设定第i、j位为两个交叉点（其中i<j<n），交叉后生成的两个新的个体是：

$X'=x_1\ x_2\ \dots\ x_i\ y_{i+1}\ \dots\ y_j\ x_{j+1}\ \dots\ x_n$　　　　$Y'=y_1\ y_2\ \dots\ y_i\ x_{i+1}\ \dots\ x_j\ y_{j+1}\ \dots\ y_n$

例：设有两个父代的个体串A= 0 0 1 1 0 1 和B= 1 1 0 0 1 0，若随机交叉点为3和4，则交叉后的两个新的个体是：

A'= 0 0 1 0 1 1　　　　B'= 1 1 0 1 0 0

基本遗传操作

- 均匀交叉

先**随机生成一个与父串具有相同长度**的二进制串（交叉模版），然后**再利用该模版对两个父串进行交叉**，即将模版中1对应的位进行交换，而0对应的位不交换，依此生成子代中的两个新的个体。

事实上，这种方法对父串中的每一位都是以相同的概率随机进行交叉的。

例5.10 设有两个父代的个体串A=001101和B=110010，若随机生成的模版**T=010011**，则交叉后的两个新的个体是A’=011010和B’=100101。即

A：　0 0 1 1 0 1　　　B：　1 1 0 0 1 0

T：　0 1 0 0 1 1

A’：　0 1 1 1 1 0　　　B’：1 0 0 0 0 1

基本遗传操作

- 实值交叉

在实数编码情况下所采用的交叉操作，主要包括离散交叉和算术交叉

- **部分离散交叉**：先在两个父代个体的编码向量中随机选择一部分分量，然后对这部分分量进行交换，生成子代中的两个新的个体。
- **整体交叉**：对两个父代个体的编码向量中的所有分量，都以1/2的概率进行交换，从而生成子代中的两个新的个体。

假设两个父代个体的n维实向量分别是 $X=x_1x_2\ldots x_i\ldots x_k\ldots x_n$和 $Y=y_1y_2\ldots y_i\ldots y_k\ldots y_n$，若随机选择对第k个分量以后的所有分量进行交换，则生成的两个新的个体向量是：

$$X' = x_1\ x_2\ \ldots\ x_k\ y_{k+1}\ \ldots\ y_n \qquad Y' = y_1\ y_2\ \ldots\ y_k\ x_{k+1}\ \ldots\ x_n$$

基本遗传操作

- 变异(Mutation)操作：对选中个体的染色体中的某些基因进行变动，以形成新的个体。遗传算法中的变异操作增加了算法的局部随机搜索能力，从而可以维持种群的多样性。
  - **变异虽然可以带来群体的多样性，但因其具有很强的破坏性，因此一般通过一个很小的变异概率来控制它的使用。**
  - **根据个体编码方式的不同，变异操作可分为二进制变异和实值变异两种类型。**

① 二进制变异

先随机地产生一个变异位，然后将该变异位置上的基因值由“0”变为“1”，或由“1”变为“0”，产生一个新的个体。

例5.12 设变异前的个体为A=0 0 1 1 0 1，若随机产生的变异位置是2，则该个体的第2位由“0”变为“1”。

变异后的新的个体是A’= 0 1 1 1 0 1 。

基本遗传操作

②实值变异

用另外一个在规定范围内的随机实数去替换原变异位置上的基因值，产生一个新的个体。最常用的实值变异操作有：

- **基于次序的变异：** 先随机地产生两个变异位置，然后交换这两个变异位置上的基因。

  例: 设选中的个体向量D=20 12 16 19 21 30，若随机产生的两个变异位置分别时2和4，则变异后的新的个体向量是：

  D’= 20 19 16 12 21 30

基本遗传操作

- 精英主义 （Elitism）
  - 仅仅从产生的子代中选择基因去构造新的种群可能会丢失掉上一代种群中的很多信息。也就是说当利用交叉和变异产生新的一代时，我们有很大的可能把在某 个中间步骤中得到的最优解丢失。
  - 使用精英主义（Elitism）方法，在每一次产生新的一代时，我们首先把**当前最优解原封不动的复制到新的一代中**，其他步骤不变。这样任何时刻产生的一个最优解都可以存活到遗传算法结束。
  - 上述各种算子的实现是多种多样的，而且许多新的算子正在不断地提出，以改进GA某些性能。

遗传算法应用

例5.15 用遗传算法求函数 $f(x)=x^2$ 的最大值，其中x为[0，31]间的整数。

解：(1) 编码

由于x的定义域是区间[0，31]上的整数，由5位二进制数即可全部表示。因此，可采用二进制编码方法，其编码串的长度为5。

例如，用二进制串00000来表示x=0,11111来表示x=31等。其中的0和1为基因值。

(2) 生成初始种群

若假设给定的种群规模N=4，则可用4个随机生成的长度为5的二进制串作为初始种群。再假设随机生成的初始种群（即第0代种群）为：

$s_{01}=0\ 1\ 1\ 0\ 1$ $s_{02}=1\ 1\ 0\ 0\ 1$

$s_{03}=0\ 1\ 0\ 0\ 0$ $s_{04}=1\ 0\ 0\ 1\ 0$

遗传算法应用

(3) 计算适应度

要计算个体的适应度，首先应该定义适应度函数。由于本例是求f(x)的最大值，因此可直接用f(x)来作为适应度函数。即：

f(s)=f(x)

其中的二进制串s对应着变量x的值。根据此函数，初始种群中各个个体的适应值及其所占比例为。

编号	个体串（染色体）	x	适应值	百分比%	累计百分比%	选中次数
S_{01}	01101	13	169	14.44	14.44	1
S_{02}	11001	25	625	52.88	67.18	2
S_{03}	01000	8	64	5.41	72.59	0
S_{04}	10010	18	324	27.41	100	1

可以看出，在4个个体中S_{02}的适应值最大，是当前最佳个体。

遗传算法应用

(4) 选择操作

假设采用轮盘赌方式选择个体，且依次生成的4个随机数（相当于轮盘上指针所指的数）为0.85、0.32、0.12和0.46，经选择后得到的新的种群为：

S_{01}=10010

S_{02}=11001

S_{03}=01101

S_{04}=11001

```
14%              53%                     5%              27%
----------|----------------------------|------------|-*------------------------|
  个体1              个体2                      个体3   ^0.85      个体4
随机数为0.85落在了个体4的端内.本次选择了个体4.
```

其中，染色体11001在种群中出现了2次，而原染色体01000则因适应值太小而被淘汰

遗传算法应用

(5) 交叉

假设交叉概率P_i为50%，则种群中只有1/2的染色体参与交叉。若规定种群中的染色体按顺序两两配对交叉，且有S_{01}与S_{02}交叉，S_{03}与S_{04}不交叉，则交叉情况为：

编号	个体串（染色体）	交叉对象	交叉位	子代	适应值
S01	10010	**S02**	3	10001	289
S02	11001	**S01**	3	11010	676
S03	01101	**S04**	N	01101	169
S04	11001	**S03**	N	11001	625

可见，经交叉后得到的新的种群为：

S_{01}=10001

S_{02}=11010

S_{03}=01101

S_{04}=11001

遗传算法应用

(6) 变异

变异概率P_m一般都很小，假设本次循环中没有发生变异，则变异前的种群即为进化后所得到的第1代种群。即：

$S_{11}=10001$　　$S_{12}=11010$　　$S_{13}=01101$　　$S_{14}=11001$

然后，对第1代种群重复上述(4)-(6)的操作，选择情况如下：

编号	个体串（染色体）	x	适应值	百分比%	累计百分比%	选中次数
S_{11}	10001	27	289	16.43	16.437	1
S_{12}	11010	26	676	38.43	54.86	2
S_{13}	01101	13	169	9.61	64.47	0
S_{14}	11001	25	625	35.53	100	1

其中若假设按轮盘赌选择时依次生成的4个随机数为0.14、0.51、0.24和0.82，经选择后得到的新的种群为：

$S_{11}=10001$　　$S_{12}=11010$　　$S_{13}=11010$　　$S_{14}=11001$

可以看出，染色体11010被选择了2次，而原染色体01101则因适应值太小而被淘汰。

遗传算法应用

对第1代种群，其交叉情况：

编号	个体串（染色体）	交叉对象	交叉位	子代	适应值
S_{11}	10001	S12	3	10010	324
S_{12}	11010	S11	3	11001	625
S_{13}	11010	S14	2	11001	625
S_{14}	11001	S13	2	11010	675

可见，经杂交后得到的新的种群为：

S_{11}=10010 S_{12}=11001 S_{13}=11001 S_{14}=11010

可以看出，第3位基因均为0，已经不可能通过交配达到最优解。这种过早陷入局部最优解的现象称为早熟。为解决这一问题，需要采用变异操作。

遗传算法应用

对第1代种群，其变异情况：

编号	个体串（染色体）	是否变异	变异位	子代	适应值
S_{11}	10010	N		10010	324
S_{12}	11001	N		11001	625
S_{13}	11001	N		11001	625
S_{14}	11010	Y	3	11110	900

它是通过对S_{14}的第3位的变异来实现的。变异后所得到的第2代种群为：

S_{21}=10010　　S_{22}=11001　　S_{23}=11001　　S_{24}=11110

接着，再对第2代种群同样重复上述(4)-(6)的操作：

遗传算法应用

对第2代种群，同样重复上述(4)-(6)的操作。

编号	个体串（染色体）	x	适应值	百分比%	累计百分比%	选中次数
S21	10010	18	324	23.92	23.92	1
S22	11001	25	625	22.12	46.04	1
S23	11001	25	625	22.12	68.16	1
S24	11110	30	900	31.84	100	1

其中若假设按轮盘赌选择时依次生成的4个随机数为0.42、0.15、0.59和0.91，经选择后得到的新的种群为：

S_{21}=11001　　S_{22}=10010　　S_{23}=11001　　S_{24}=11110

遗传算法应用

对第2代种群，其交叉情况：

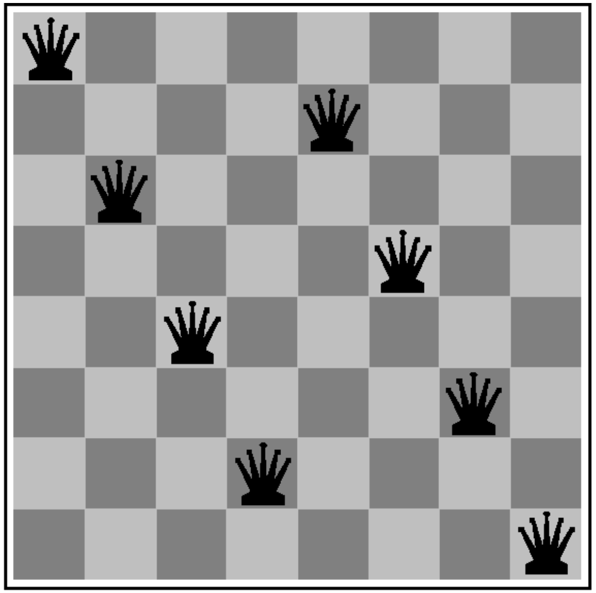
编号	个体串（染色体）	交叉对象	交叉位	子代	适应值
S_{21}	11001	S_{22}	3	11010	676
S_{22}	10010	S_{21}	3	10001	289
S_{23}	11001	S_{24}	4	11000	576
S_{24}	11110	S_{23}	4	11111	961

这时，函数的最大值已经出现，其对应的染色体为11111，经解码后可知问题的最优解是在点x=31处。

求解过程结束。

例子：8皇后问题

- 目标：任何一个皇后都不会攻击到其他的皇后（皇后可以攻击和它在同一行、同一列或同一对角线上的皇后）
- 适应度函数取作可以彼此攻击的皇后对的数目（忽略障碍）



8皇后的例子，其中每个状态用一个长度为8的字符串来表示，适应度函数取作不相互攻击的皇后对数目。

例子：8皇后问题

- 问题表示：从k个随机生成的状态（种群）开始
- 每个个体用一个有限长度的字符串表示-通常用0，1表示

每列八个皇后的位置：如

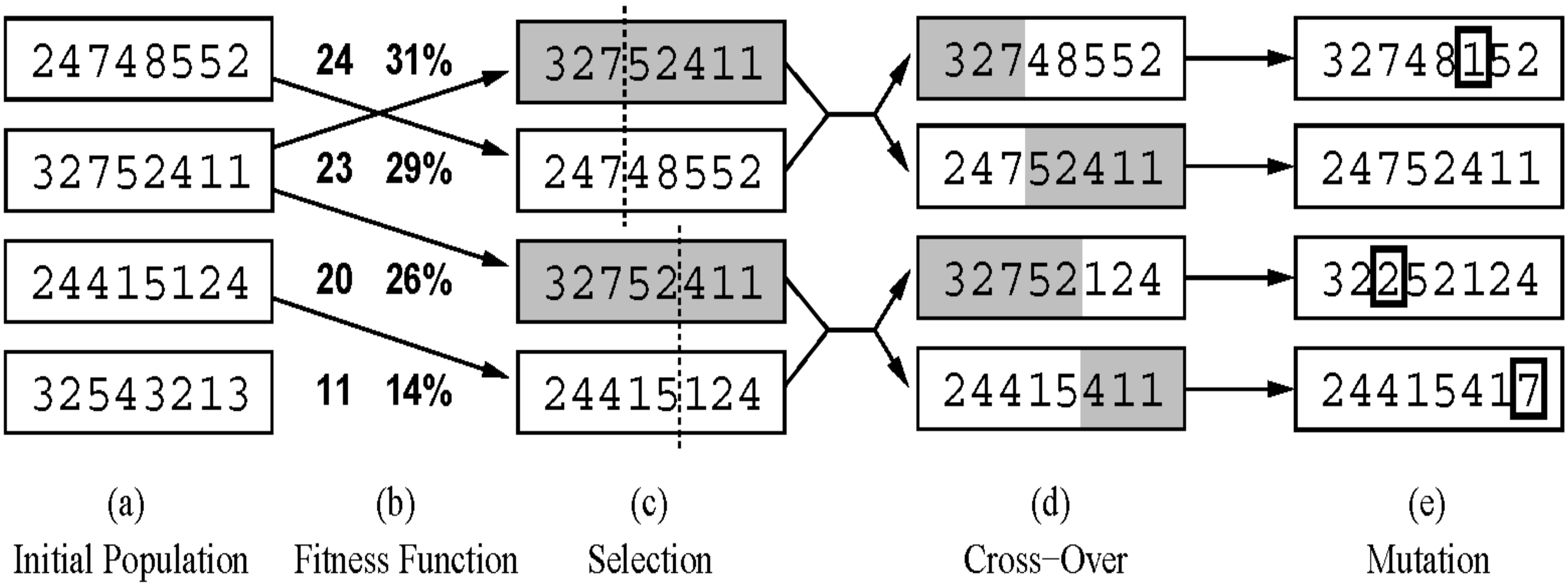
3 2 7 5 3 4 1 1

011 010 111 101 011 100 001 001

								8
		*						**7**
								6
			*					**5**
					*			**4**
*				*				**3**
	*							**2**
						*	*	**1**

例子：8皇后问题

- 通过把两个父状态结合来生成后继



遗传算法特点

- 自组织、自适应和自学习性—概率转移准则，非确定性规则
  - 确定进化方案后，算法将利用进化过程中得到的信息自行组织搜索；基于自然的选择策略，优胜劣汰；
  - 遗传算法很快就能找到良好的解，即使是在很复杂的解空间中
    - 采用随机方法进行最优解搜索，选择体现了向最优解迫近
    - 交叉体现了最优解的产生，变异体现了全局最优解的复盖
- 本质并行性---群体搜索
  - 算法本身非常适合大规模并行，各种群分别独立进化，不需要相互间交换信息；二是可以同时搜索解空间的多个区域并相互间交流信息，使得演化计算能以较少的计算获得较大的收益。

遗传算法特点

- 不需要其他知识，只需要影响搜索方向的目标函数和相应的适应度函数
  - 对待求解问题的指标函数没有什么特殊的要求，如不要求连续性、导数存在、单峰值等假设
  - 容易形成通用算法程序
  - 遗传算法不能解决那些“大海捞针”的问题，所谓“大海捞针”问题就是没有一个确切的适应度函数表征个体好坏的问题，遗传算法对这类问题无法找到收敛的路径。

遗传算法特点

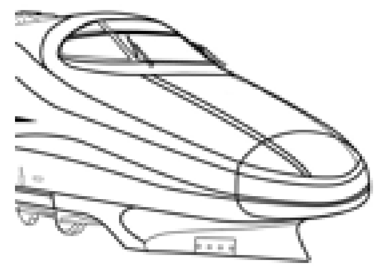
- 理论上证明算法的收敛性很困难
- 多用于解决实际问题
  - 汽车设计，包括材料选择、多目标汽车组件设计、减轻重量等。
  - 机电系统设计。
  - 分布计算机网络的拓扑结构。
  - 电路设计，此类用途的遗传算法叫做进化电路。
  - 移动通讯优化结构。
  - 煤气管道的最优控制、通信网络链接长度的优化问题、铁路运输计划优化、喷气式收音机涡轮机的设计、VLSI版面设计、键盘排列优化
  - 抓到老鼠的猫都是好猫

Examples of success

hard to apply traditional optimization methods
but easy to test a given solution

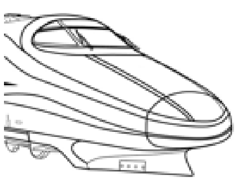
Representation:

parameterize

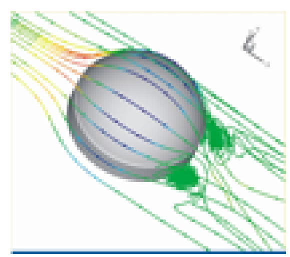


represented as a vector of parameters

Fitness:



x_i



$f(x_i)$

test by simulation/experiment

Examples of success

Series 700

Series N700

Technological overview of the next generation Shinkansen high-speed train Series N700

M. Ueno[1], S. Usui[1], H. Tanaka[1], A. Watanabe[2]

[1]*Central Japan Railway Company, Tokyo, Japan,* [2]*West Japan Railway Company, Osaka, Japan*

Abstract

In March 2005, Central Japan Railway Company (JR Central) has completed prototype

waves and other issues related to environmental compatibility such as external noise. To combat this, an aero double-wing-type has been adopted for nose shape (Fig. 3). This nose shape, which boasts the most appropriate aerodynamic performance, has been newly developed for railway rolling stock using the latest analytical technique (i.e. genetic algorithms) used to develop the main wings of airplanes. The shape resembles a bird in flight, suggesting a feeling

On the Tokaido Shinkansen line, Series N700 cars save 19% energy than Series 700 cars, despite a 30% increase in the output of their traction equipment for higher-speed operation (Fig. 4).

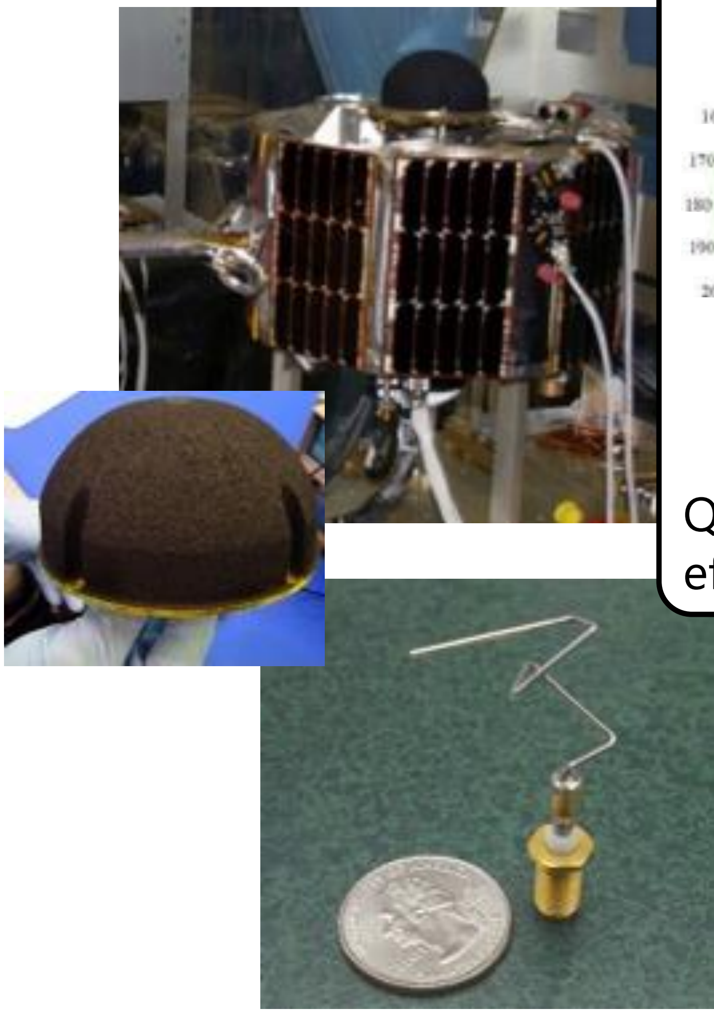
This is a result of adopting the aerodynamically excellent nose shape, reduced running resistance thanks to the drastically smoothened car body and under-floor equipment, effective

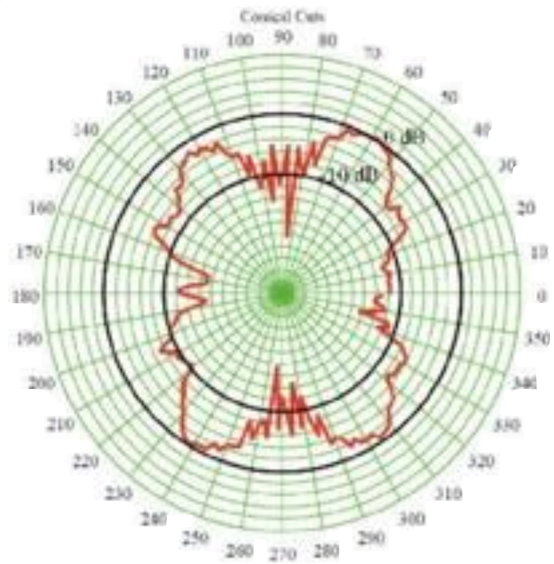
this nose ... has been newly developed ... using the latest analytical technique (i.e. **genetic algorithms**)

N700 cars save **19%** energy ... **30%** increase in the output... This is a result of adopting the ... nose shape

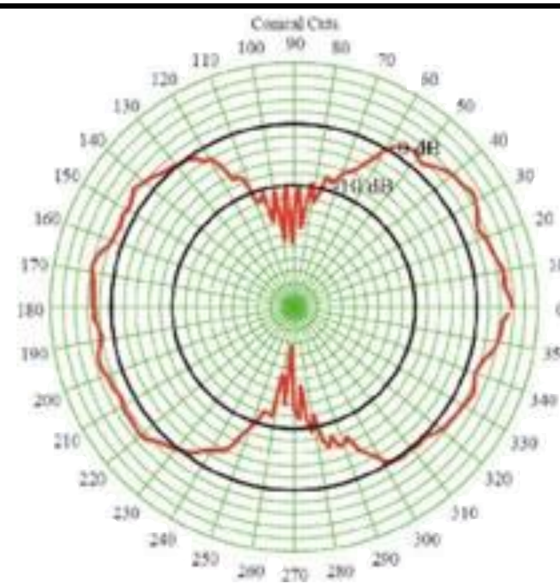
Examples of success

NASA ST5 satellite

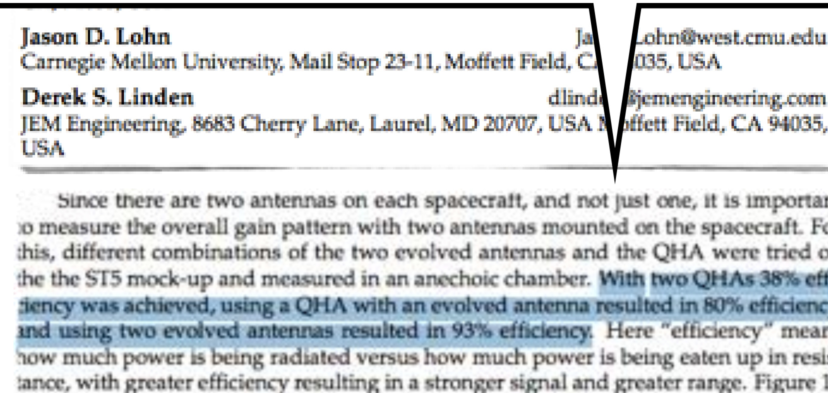




QHAs(人工设计) 38% efficiency



evolved antennas resulted in 93% efficiency



Jason D. Lohn Ja… Lohn@west.cmu.edu
Carnegie Mellon University, Mail Stop 23-11, Moffett Field, C… 035, USA
Derek S. Linden dlind…@jemengineering.com
JEM Engineering, 8683 Cherry Lane, Laurel, MD 20707, USA …ffett Field, CA 94035, USA

Since there are two antennas on each spacecraft, and not just one, it is importan… o measure the overall gain pattern with two antennas mounted on the spacecraft. Fo… this, different combinations of the two evolved antennas and the QHA were tried or… the the ST5 mock-up and measured in an anechoic chamber. With two QHAs 38% effi… iency was achieved, using a QHA with an evolved antenna resulted in 80% efficiency …nd using two evolved antennas resulted in 93% efficiency. Here "efficiency" means …ow much power is being radiated versus how much power is being eaten up in resis… tance, with greater efficiency resulting in a stronger signal and greater range. Figure 11

Other meta-heuristics