数学建模中的常用算法

成都信息工程学院
计算科学系
胡建成

jianchenghu@163.com

2006-8-6

数学建模竞赛网上资源

- CUMCM网站: http://mcm.edu.cn
- MCM和ICM网站: http://www.comap.com
- 中国数学建模: http://www.shumo.com
- 中科大数学建模网站: http://mcm.ustc.edu.cn
- MATLAB网站: http://www.mathworks.com
- GOOGLE大学

数学建模竞赛中的算法

- 93A 非线性交调的频率设计: 拟合、规划
- 93B 足球队排名次: 矩阵论、图论、层次分析法、整数规划
- 94A 逢山开路: 图论、插值、动态规划
- 94B 锁具装箱问题: 图论、组合数学
- 95A 飞行管理问题 : 非线性规划、线性规划
- 95B 天车与冶炼炉的作业调度: 非线性规划、动态规划、层次分析法、PETRI方法、图论方法、排队论方法

数学建模竞赛中的算法

- 96A 最优捕鱼策略：微分方程、积分、非线性规划
- 96B 节水洗衣机：非线性规划
- 97A 零件参数设计：微积分、非线性规划、随机模拟
- 97B 截断切割：组合优化、几何变换、枚举、蒙特卡罗、递归、最短路

数学建模竞赛中的算法

- **98A** 投资收益与风险：线性规划、非线性规划
- **98B** 灾情巡视：最小生成树、**Hamilton**圈、旅行商问题
- **99A** 自动化车床：积分、概率分布、随机模拟、分布拟合度检验
- **99B** 钻井布局：几何变换、枚举、最大完全子图、混合整数规划

数学建模竞赛中的算法

- 00A DNA分类：神经网络、最小二乘拟合、统计分类
- 00B 管道订购：最短路、二次规划
- 01A 血管的三维重建：数据挖掘、曲面重建与拟合
- 01B 公交车调度：非线性规划
- 02A 车灯光源优化设计：最优化
- 02B 彩票中的数学：概率与优化

数学建模常用数学软件

- MATLAB
- Maple
- Mathematica
- Lindo
- Lingo
- SAS
- SPSS
- C&C++
- Fortran
- Pascal

数学建模应掌握的十类算法

1. 蒙特卡罗方法(Monte-Carlo方法, MC)

该算法又称计算机随机性模拟方法，也称统计试验方法。这一方法源于美国在第一次世界大战进行的研制原子弹的“曼哈顿计划”。该计划的主持人之一、数学家冯·诺伊曼用驰名世界的赌城—摩纳哥的Monte Carlo—来命名这种方法。MC方法是一种基于“随机数”的计算方法，能够比较逼真地描述事物的特点及物理实验过程，解决一些数值方法难以解决的问题。MC方法的雏型可以追溯到十九世纪后期的蒲丰(Buffon)随机投针试验，即著名的蒲丰问题。 MC方法通过计算机仿真(模拟)解决问题，同时也可以通过模拟来检验自己模型的正确性，几乎是比赛时必用的方法。

数学建模应掌握的十类算法

- **97年的A题**，每个零件都有自己的标定值，也都有自己的容差等级，而求解最优的组合方案将要面对着的是一个极其复杂的公式和108种容差选取方案，根本不可能去求解析解，那如何去找到最优的方案呢？随机性模拟搜索最优方案就是其中的一种方法，在每个零件可行的区间中按照正态分布随机的选取一个标定值和选取一个容差值作为一种方案，然后通过蒙特卡罗算法仿真出大量的方案，从中选取一个最佳的。
- **02年的B题**，关于彩票第二问，要求设计一种更好的方案，首先方案的优劣取决于很多复杂的因素，同样不可能刻画出一个模型进行求解，只能靠随机仿真模拟。

数学建模应掌握的十类算法

2. 数据拟合、参数估计、插值等数据处理算法

比赛中通常会遇到大量的数据需要处理，而处理数据的关键就在于这些算法，通常使用MATLAB 作为工具。与图形处理有关的问题很多与拟合有关系。

- ✓ **98 年美国赛A 题，生物组织切片的三维插值处理。**
- ✓ **94 年A 题逢山开路，山体海拔高度的插值计算。**

此类问题在MATLAB中有很多函数可以调用，只有熟悉MATLAB，这些方法才能用好。

数学建模应掌握的十类算法

3. 规划类问题算法

此类问题主要有**线性规划、整数规划、多元规划、二次规划**等。竞赛中很多问题都和数学规划有关，可以说不少的模型都可以归结为一组不等式作为约束条件、几个函数表达式作为目标函数的问题，遇到这类问题，求解就是关键了。

- ✓ **98年B 题**，用很多不等式完全可以把问题刻画清楚。因此列举出规划后用Lindo、Lingo 等软件来进行解决比较方便，所以还需要熟悉这两个软件。

数学建模应掌握的十类算法

4. 图论问题

这类问题算法有很多，包括：**Dijkstra、Floyd、Prim、Bellman-Ford，最大流，二分匹配**等问题。

- ✓ **98 年B 题、00年B 题、95 年锁具装箱等问题体现了图论问题的重要性。**

数学建模应掌握的十类算法

5. 计算机算法设计中的问题

计算机算法设计包括很多内容：**动态规划、回溯搜索、分治算法、分枝定界**等计算机算法.

- ✓ **92 年B 题用分枝定界法**
- ✓ **97 年B 题是典型的动态规划问题**
- ✓ **98 年B 题体现了分治算法**

这方面问题和ACM 程序设计竞赛中的问题类似，可看一下与计算机算法有关的书。

数学建模应掌握的十类算法

6. 最优化理论的三大非经典算法:
模拟退火法(SA)、神经网络(NN)、遗传算法(GA)

近几年的赛题越来越复杂，很多问题没有什么很好的模型可以借鉴，于是这三类算法很多时候可以派上用场。

- ✓ **97年A 题用模拟退火算法**
- ✓ **00年B 题用神经网络分类算法**
- ✓ **01年B 题这种难题也可以使用神经网络**
- ✓ **美国89年A 题也和BP 算法有关系，当时是86年刚提出BP 算法，89年就考了，说明赛题可能是当今前沿科技的抽象体现。**
- ✓ **美国03年B 题伽马刀问题也是目前研究的课题，目前算法最佳的是遗传算法。**

数学建模应掌握的十类算法

7. 网格算法和穷举算法

网格算法和穷举法一样，只是网格法是连续问题的穷举。比如要求在*N* 个变量情况下的最优化问题，那么对这些变量可取的空间进行采点，比如在[*a b*] 区间内取*M* +1 个点，就是*a*，*a*+(*b*-*a*)/*M*，*a*+2(*b*-*a*)/*M*，…，*b*。那么这样循环就需要进行(*M* + 1)^*N* 次运算，所以计算量很大。

- ✓ **97 年A 题、99 年B 题都可以用网格法搜索**

这种方法最好在运算速度较快的计算机中进行，还有要用高级语言来做，最好不要用MATLAB 做网格，否则会算很久的。

数学建模应掌握的十类算法

8. 连续问题离散化的方法

很多问题都是实际来的，数据可以是连续的，而计算机只能处理离散的数据，因此需要将连续问题进行离散化处理后再用计算机求解。比如差分代替微分、求和代替积分等思想都是把连续问题离散化的常用方法。

数学建模应掌握的十类算法

9. 数值分析方法

数值分析研究各种求解数学问题的数值计算方法，特别是适合于计算机实现的方法与算法。它的主要内容包括**函数的数值逼近、数值微分与数值积分、非线性方程的数值解法、数值代数、常微分方程数值解**等。数值分析是计算数学的一个重要分支，把理论与计算紧密结合,是现代科学计算的基础 。MATLAB等数学软件中已经有很多数值分析的函数可以直接调用。

数学建模应掌握的十类算法

10. 图象处理算法

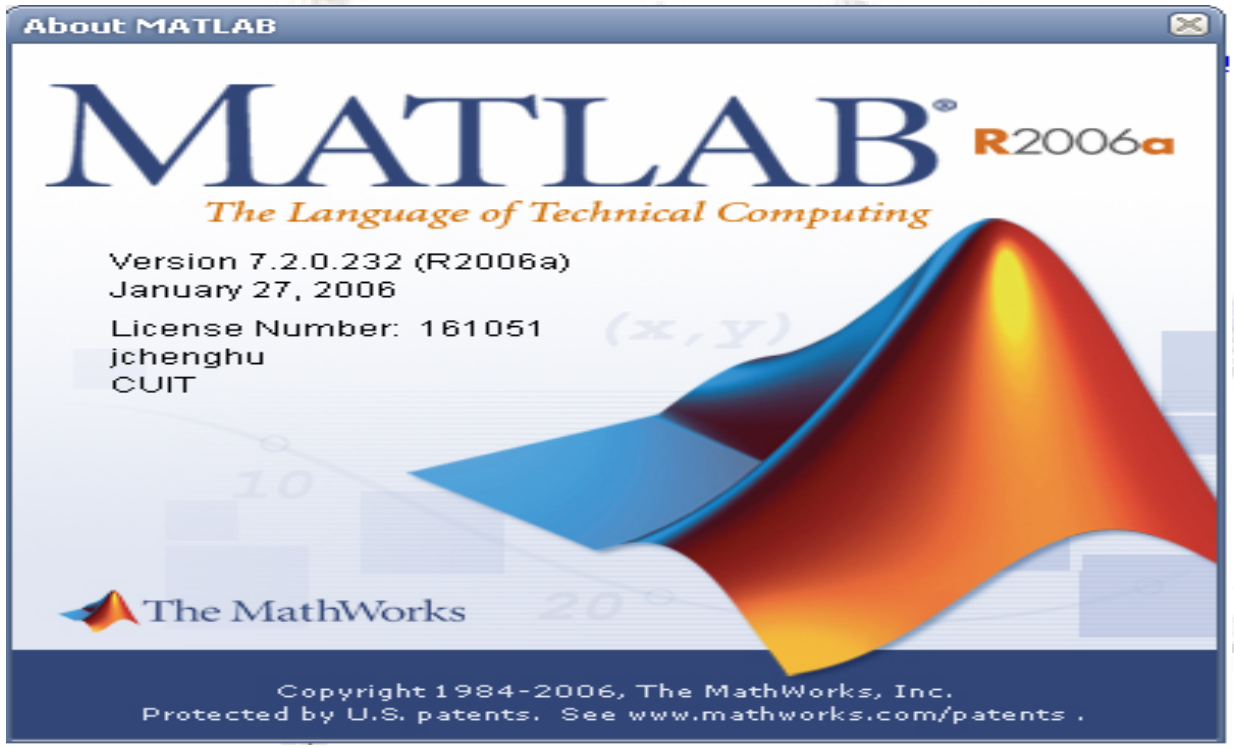
赛题中有一类问题与图形有关，即使问题与图形无关，论文中也会需要图片来说明问题，这些图形如何展示以及如何处理就是需要解决的问题，通常使用MATLAB 进行处理。

- ✓ **01年A 题中需要你会读BMP 图象**
- ✓ **98年美国A 题需要你知道三维插值计算**
- ✓ **03年B 题要求更高，不但需要编程计算还要进行处理**

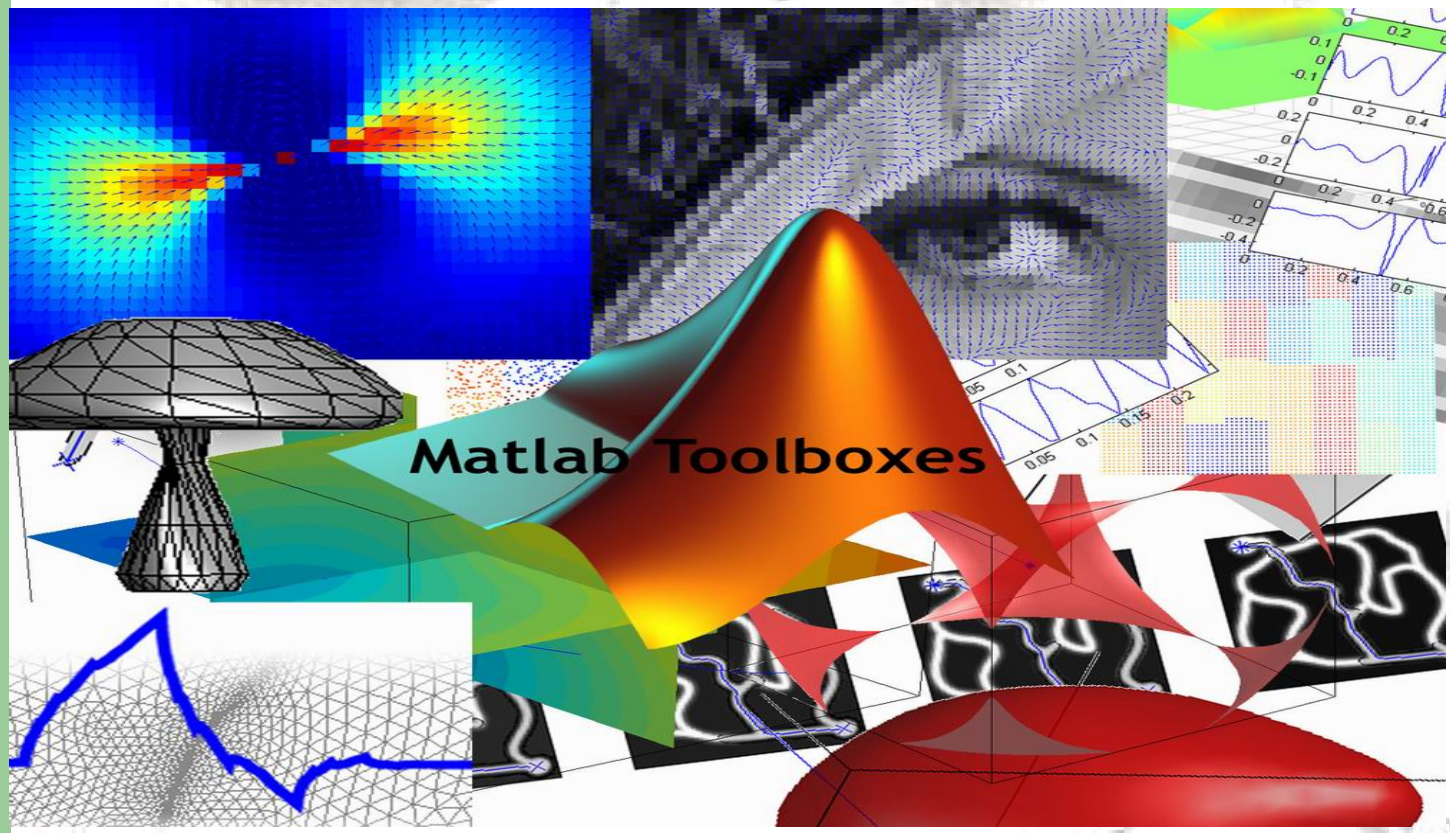
数模论文中也有很多图片需要展示，解决这类问题要熟悉MATLAB图形图像工具箱。

MATLAB=MATrix LABoratory

- Latest Version: **7.2.0.232 (R2006a)**
- ***Released: 01 Mar 2006***



MATLAB ToolBoxes (MATLAB®7.2)



MATLAB ToolBoxes(MATLAB®7.2)

- Math and Optimization
- **Optimization Toolbox**

  Solve standard and large-scale optimization problems

- ✓ **Symbolic Math Toolbox**

  Perform computations using symbolic mathematics and variable-precision arithmetic

- ✓ **Extended Symbolic Math Toolbox**

  Perform computations using symbolic mathematics and variable-precision arithmetic

- ✓ **Partial Differential Equation Toolbox**

  Solve and analyze partial differential equations

- ✓ **Genetic Algorithm and Direct Search Toolbox**

  Solve optimization problems using genetic and direct search algorithms

MATLAB ToolBoxes(MATLAB®7.2)

- Statistics and Data Analysis
- ✓ **Statistics Toolbox**

  Apply statistical algorithms and probability models
- ✓ **Neural Network Toolbox**

  Design and simulate neural networks
- ✓ **Curve Fitting Toolbox**

  Perform model fitting and analysis
- ✓ **Spline Toolbox**

  Create and manipulate spline approximation models of data
- ✓ **Model-Based Calibration Toolbox**

  Calibrate complex powertrain systems

MATLAB ToolBoxes(MATLAB®7.2)

- Control System Design and Analysis
  - ✓ **Control System Toolbox**
    Design and analyze control systems
  - ✓ **System Identification Toolbox**
    Create linear dynamic models from measured input-output data
  - ✓ **Fuzzy Logic Toolbox**
    Design and simulate fuzzy logic systems
  - ✓ **Robust Control Toolbox**
    Design robust controllers for plants with uncertain parameters and unmodeled dynamics
  - ✓ **Model Predictive Control Toolbox**
    Develop model predictive controllers in MATLAB and Simulink

MATLAB ToolBoxes(MATLAB®7.2)

- Signal Processing and Communications

✓ **Signal Processing Toolbox**

Perform signal processing, analysis, and algorithm development

✓ **Communications Toolbox**

Design and analyze algorithms for the physical layer of communication systems

✓ **Filter Design Toolbox**

Design and analyze fixed-point, adaptive, and multirate filters

✓ **Wavelet Toolbox**

Analyze and synthesize signals and images using wavelet techniques

✓ **Fixed-Point Toolbox**

Design and verify fixed-point algorithms and analyze fixed-point data

MATLAB ToolBoxes(MATLAB®7.2)

- Image Processing
- ✓ **Image Processing Toolbox**

  Perform image processing, analysis, and algorithm development
- ✓ **Image Acquisition Toolbox**

  Acquire images and video from industry-standard hardware
- ✓ **Mapping Toolbox**

  Analyze and visualize geographic information

MATLAB ToolBoxes(MATLAB®7.2)

- Financial Modeling and Analysis
  - ✓ **Financial Toolbox**
    Analyze financial data and develop financial algorithms
  - ✓ **Financial Derivatives Toolbox**
    Model and analyze equity and fixed-income derivatives
  - ✓ **GARCH Toolbox**
    Analyze financial volatility using univariate GARCH models
  - ✓ **Datafeed Toolbox**
    Acquire financial data from data service providers
  - ✓ **Fixed-Income Toolbox**
    Model and analyze fixed-income securities

遗传算法(Genetic Algorithm, GA)

- 进化算法(Evolutionary Algorithm)

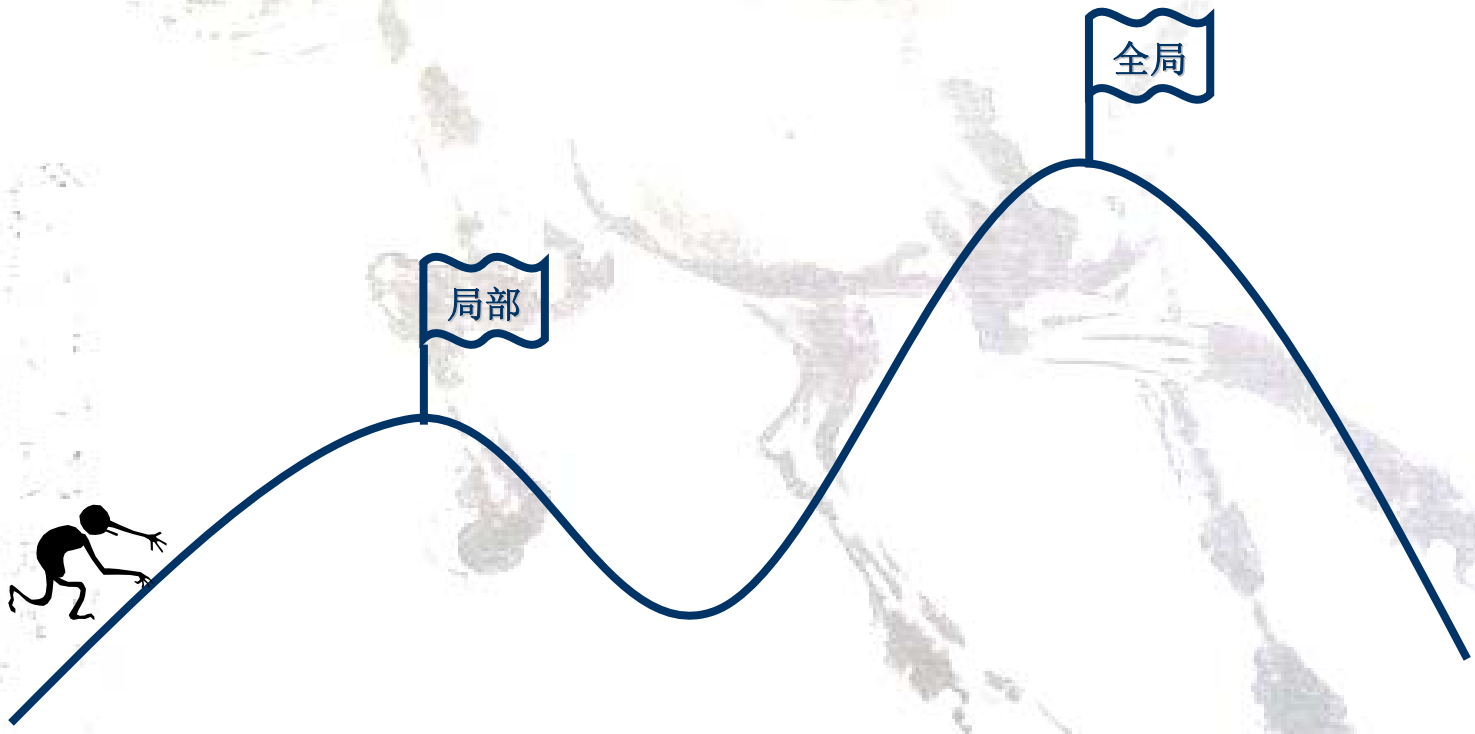
遗传算法(Genetic Algorithm, GA)

- Darwin(1859): “物竞天择，适者生存”
- John Holland (university of Michigan, 1975)
  《**Adaptation in Natural and Artificial System**》
- 遗传算法作为一种有效的工具，已广泛地应用于最优化问题求解之中。
- 遗传算法是一种基于自然群体遗传进化机制的自适应全局优化概率搜索算法。它摒弃了传统的搜索方式，模拟自然界生物进化过程，采用人工的方式对目标空间进行随机化搜索。

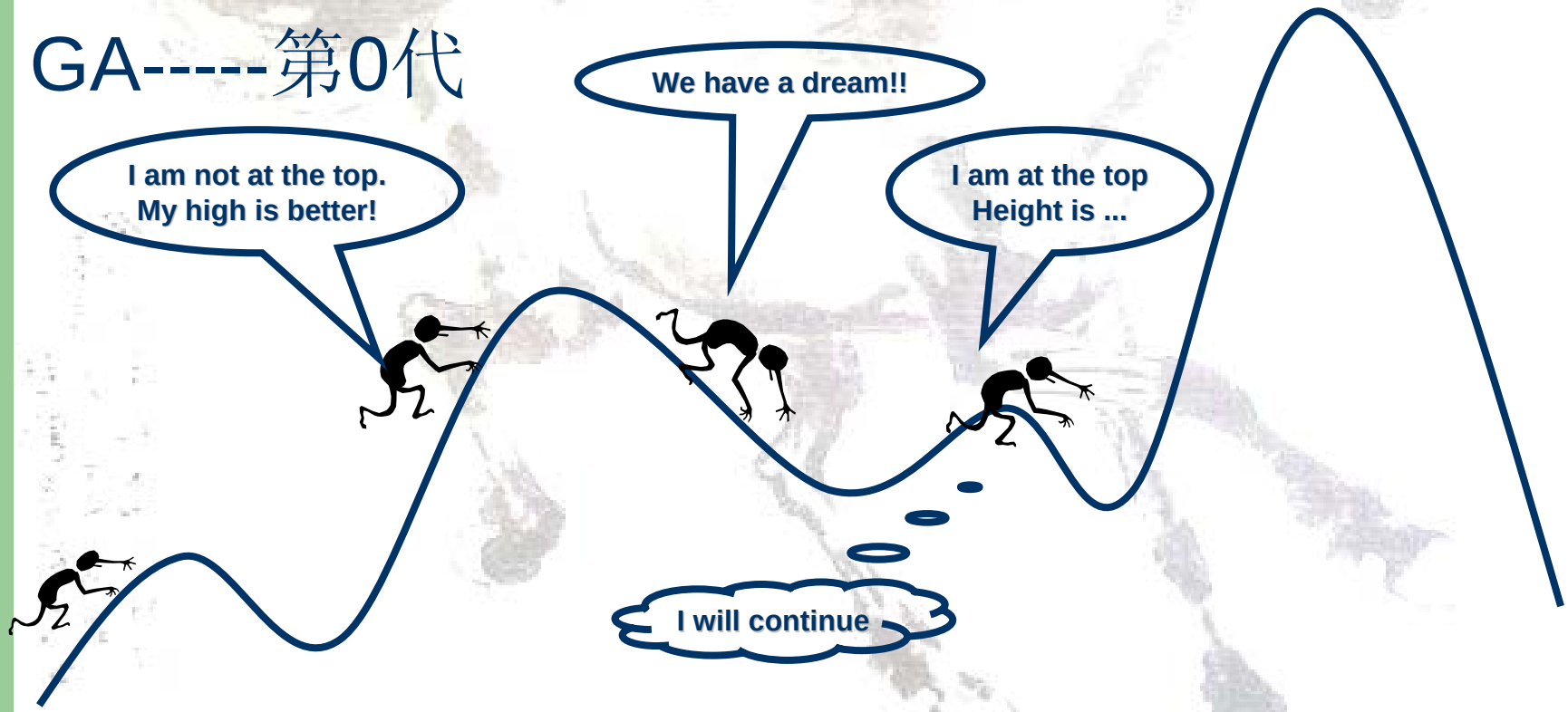
遗传算法(Genetic Algorithm, GA)

遗传算法是从代表问题可能潜在解集的一个种群(population)开始的，而一个种群则由经过基因(gene) 编码(coding)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。为了简化，往往采用二进制编码。对种群反复进行选择(selection)、交叉(crossover)、变异(mutation)操作，估计各个体的适应值(fitness)，根据"适者生存、优胜劣汰"的进化规则，产生最好的种群，使适应性好的个体比适应性差的个体有更多的繁殖机会。最后把末代种群中的最优个体经过解码(decoding),可以获得满足要求的最优解。

遗传算法(Genetic Algorithm, GA)

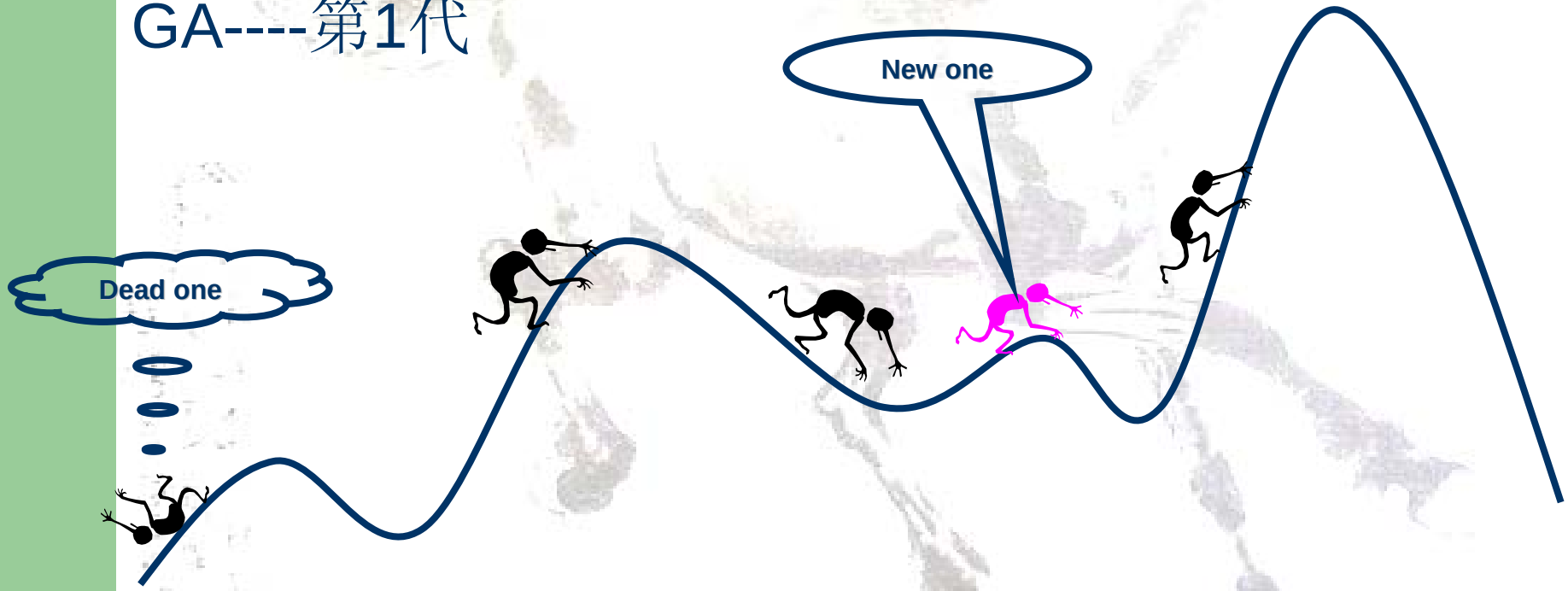


遗传算法(Genetic Algorithm, GA)



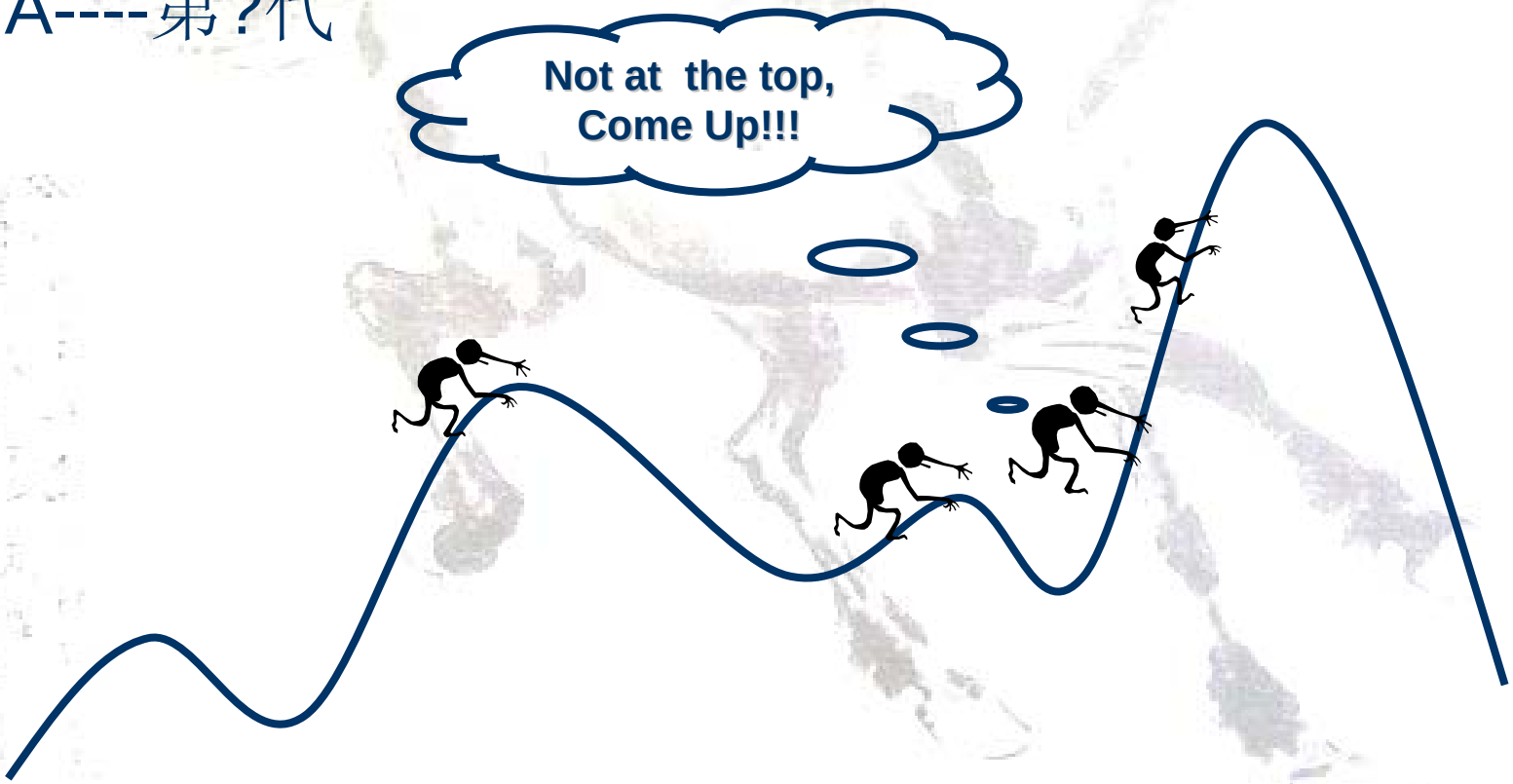
遗传算法(Genetic Algorithm, GA)

GA----第1代



遗传算法(Genetic Algorithm, GA)

GA----第?代



遗传算法(Genetic Algorithm, GA)

GA----第N代

遗传算法(Genetic Algorithm, GA)

适者生存(Survival of the Fittest)

- GA主要采用的进化规则是“适者生存”
- 较好的解保留，较差的解淘汰

生物进化与遗传算法对应关系

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

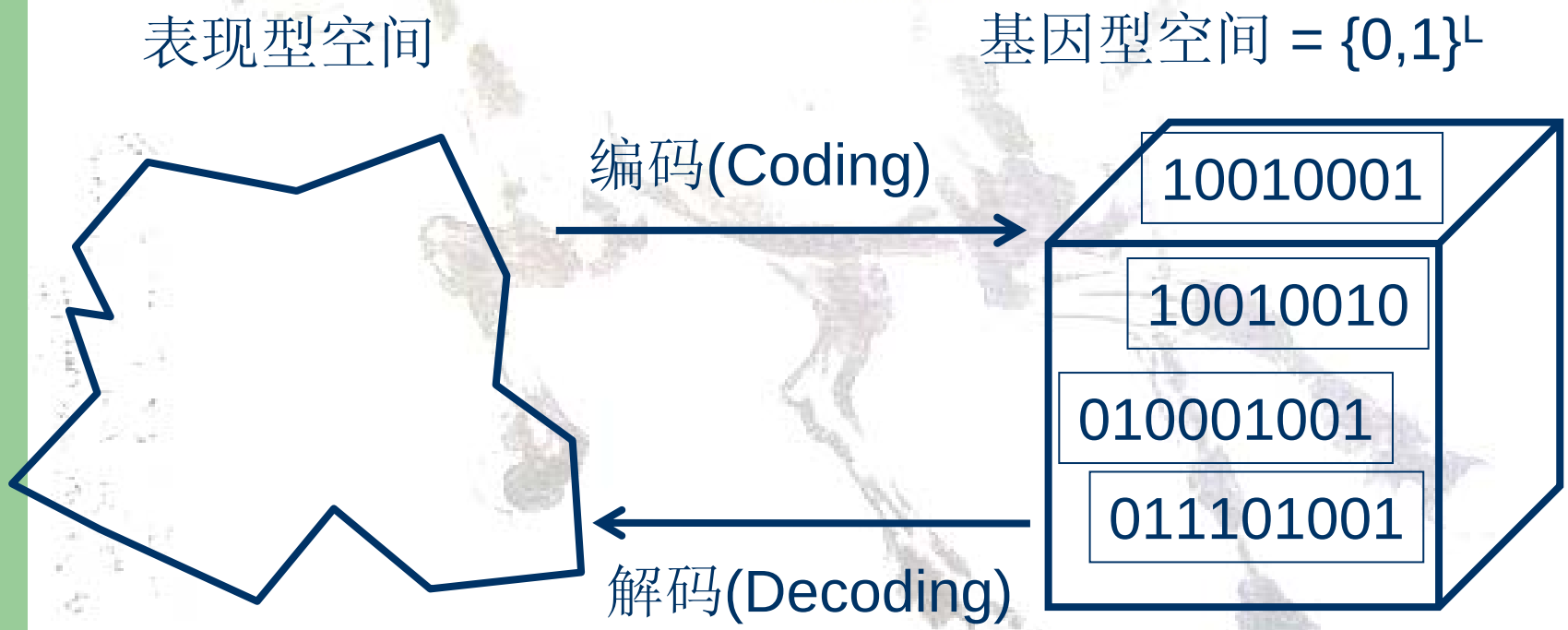
遗传算法的基本操作

- **选择(selection)**:
  根据各个个体的适应值，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P_c(称为交叉概率，crossvoer 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P_m(称为变异概率，mutation rate)改变某一个或一些基因座上基因值为其它的等位基因。

如何设计遗传算法

- 如何进行编码?
- 如何产生初始种群?
- 如何定义适应函数?
- 如何进行遗传操作(复制、交叉、变异)?
- 如何产生下一代种群?
- 如何定义停止准则?

编码(Coding)


表现型空间
基因型空间 = {0,1}L
编码(Coding)
10010001
10010010
010001001
011101001
解码(Decoding)

选择(Selection)

- 选择(复制)操作把当前种群的染色体按与适应值成正比例的概率复制到新的种群中
- 主要思想: 适应值较高的染色体体有较大的选择(复制)机会
- 实现1：”轮盘赌”选择(Roulette wheel selection)
  - ✓ 将种群中所有染色体的适应值相加求总和，染色体适应值按其比例转化为选择概率**Ps**
  - ✓ 产生一个在**0**与总和之间的的随机数**m**
  - ✓ 从种群中编号为**1**的染色体开始，将其适应值与后续染色体的适应值相加，直到累加和等于或大于**m**

选择(Selection)

设种群的规模为N

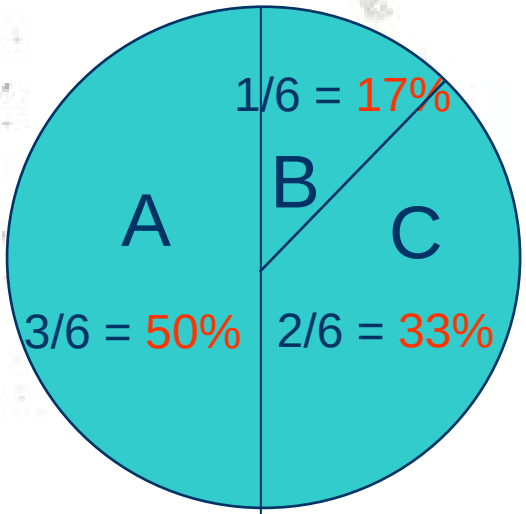
x_i是i为种群中第i个染色体

$f(x_i)$ 第i个染色体的适应度值

$\sum f(x_i)$ 种群中所有染色体适应度值之和。

染色体x_i被选概率

$$p_s(x_i) = \frac{F(x_i)}{\sum_{j=1}^{N} F(x_j)}$$



fitness(A) = 3

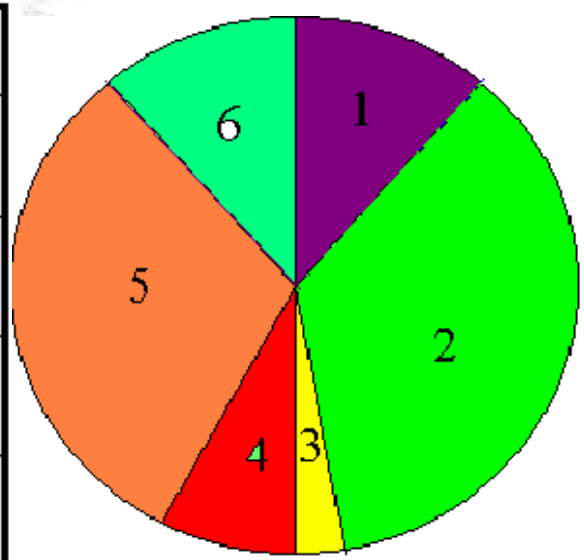
fitness(B) = 1

fitness(C) = 2

选择(Selection)

染色体的适应值和所占的比例

序号	染色体	适应度值	所占比例	累计
1	01110	8	16	8
2	11000	15	30	23
3	00100	2	4	25
4	10010	5	10	30
5	01100	12	24	42
6	00011	8	16	50



轮盘赌选择

选择(Selection)

染色体被选的概率

染色体编号	1	2	3	4	5	6
染色体	01110	11000	00100	10010	01100	00011
适应度	8	15	2	5	12	8
被选概率	0.16	0.3	0.04	0.1	0. 24	0.16
适应度累计	8	23	25	30	42	50

被选的染色体

随机数	23	49	13	38	6	27
所选号码	2	6	2	5	1	4
所选染色体	11000	00011	11000	01100	01110	10010

选择(Selection)

- ✓ 轮盘上的片分配给群体的染色体，使得每一个片的大小与对于染色体的适应值成比例
- ✓ 从群体中选择一个染色体可视为旋转一个轮盘，当轮盘停止时，指针所指的片对于的染色体就时要选的染色体。
- ✓ 模拟“轮盘赌”算法：
  - (1) **r=random(0, 1)，s=0，i=0；**
  - (2) 如果**s≥r**，则转**(4)；**
  - (3) **$s=s+p(x_i)$，i=i+1,** 转**(2)**
  - (4) **xi**即为被选中的染色体，输出**I**
  - (5) 结束

选择(Selection)

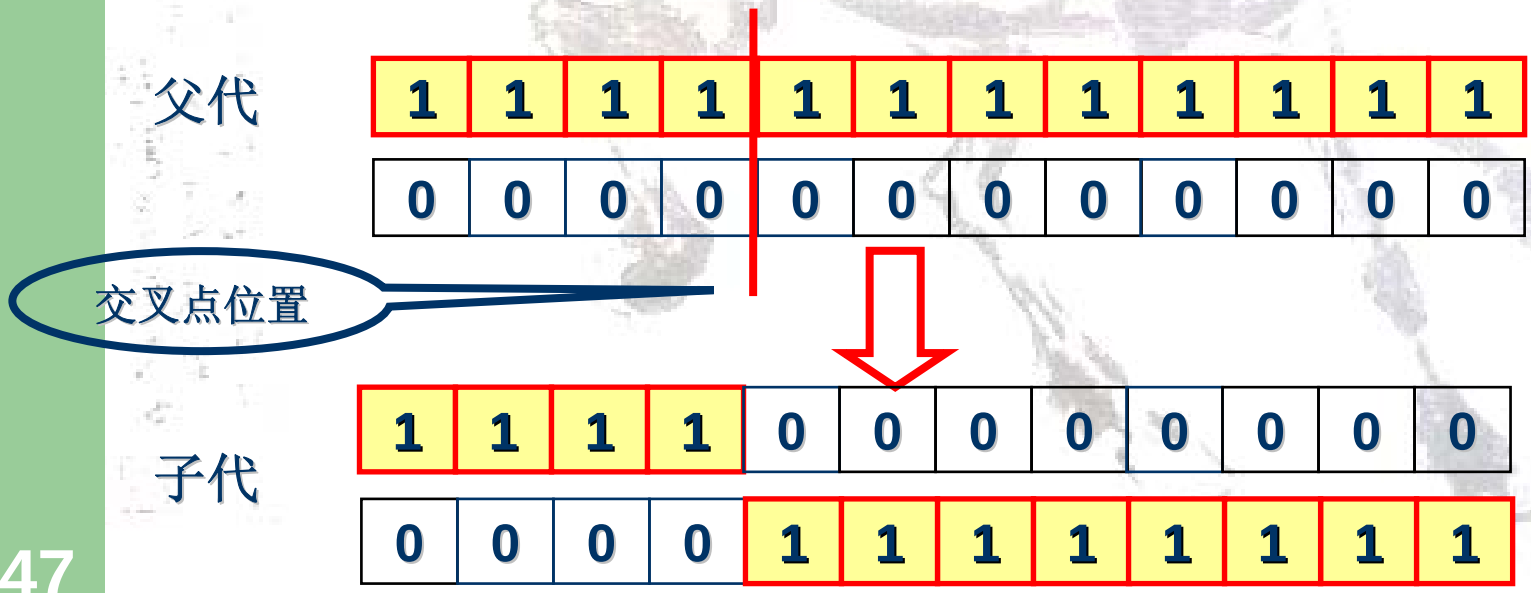
- 其他选择法：
  - ✓ 随机遍历抽样**(Stochastic universal sampling)**
  - ✓ 局部选择**(Local selection)**
  - ✓ 截断选择**(Truncation selection)**
  - ✓ 竞标赛选择**(Tournament selection)**
- 特点：选择操作得到的新的群体称为交配池，交配池是当前代和下一代之间的中间群体，其规模为初始群体规模。选择操作的作用效果是提高了群体的平均适应值（低适应值个体趋于淘汰，高适应值个体趋于选择），但这也损失了群体的**多样性**。选择操作没有产生新的个体，群体中最好个体的适应值不会改变。

交叉(crossover, Recombination)

- 遗传交叉(杂交、交配、有性重组)操作发生在两个染色体之间，由两个被称之为双亲的父代染色体，经杂交以后，产生两个具有双亲的部分基因的新的染色体，从而检测搜索空间中新的点。
- 选择(复制)操作每次作用在一个染色体上，而交叉操作每次作用在从交配池中随机选取的两个个体上(交叉概率 P_c)。
- 交叉产生两个子染色体，他们与其父代不同，且彼此不同， 每个子染色体都带有双亲染色体的遗传基因。

单点交叉(1-point crossover)

- 在双亲的父代染色体中随机产生一个交叉点位置
- 在交叉点位置分离双亲染色体
- 互换交叉点位置右边的基因码产生两个子代染色体
- 交叉概率P_c一般范围为(60%, 90%)，平均约80%



交叉(crossover, Recombination)

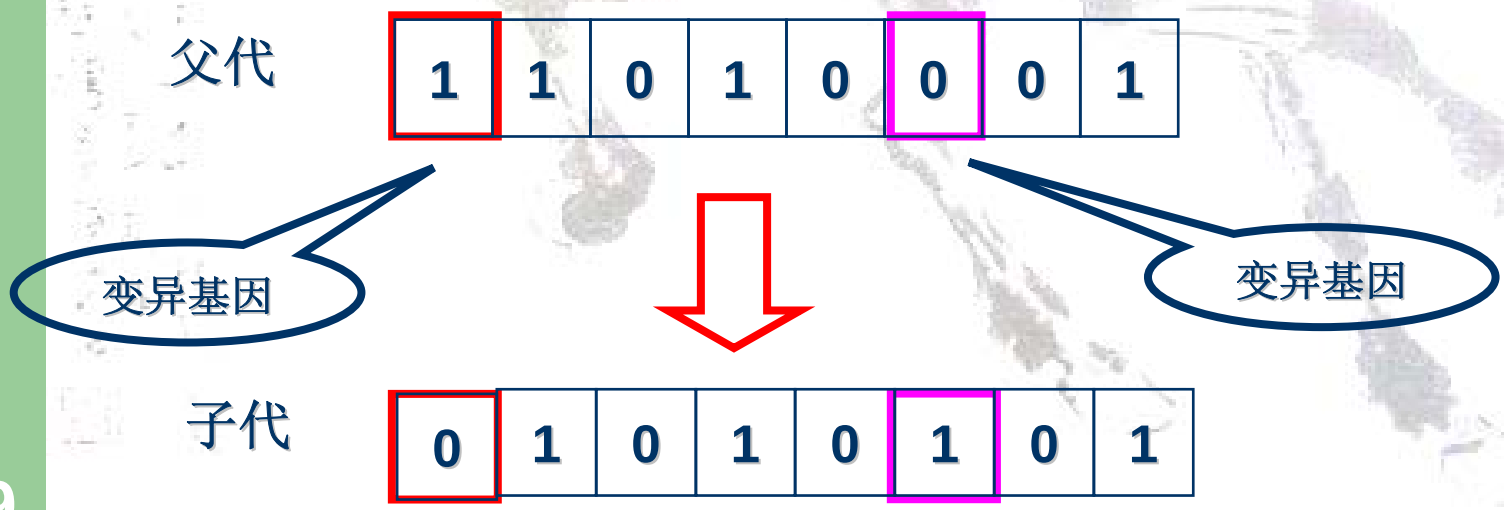
- 单点交叉操作可以产生与父代染色体完全不同的子代染色体；它不会改变父代染色体中相同的基因。但当双亲染色体相同时，交叉操作是不起作用的。

假如交叉概率P_c＝50%,则交配池中50%的染色体(一半染色体)将进行交叉操作，余下的50%的染色体进行选择(复制)操作。

- GA利用选择和交叉操作可以产生具有更高平均适应值和更好染色体的群体

变异(Mutation)

- 以变异概率P_m改变染色体的某一个基因，当以二进制编码时，变异的基因由0变成1，或者由1变成0。
- 变异概率P_m 一般介于1/种群规模与1/染色体长度之间，平均约1-2%



变异(Mutation)

- 比起选择和交叉操作，变异操作是GA中的次要操作，但它在恢复群体中失去的**多样性**方面具有潜在的作用。

在GA执行的开始阶段，染色体中一个特定位上的值1可能与好的性能紧密联系，，即搜索空间中某些初始染色体在那个位上的值1可能一致产生高的适应值。因为越高的适应值与染色体中那个位上的值1相联系，选择操作就越会使群体的遗传多样性损失。等到达一定程度时，值0会从整个群体中那个位上消失，然而全局最优解可能在染色体中那个位上位0。如果搜索范围缩小到实际包含全局最优解的那部分搜索空间，在那个位上的值0就可能正好是到达全局最优解所需要的。

适应函数(Fitness Function)

- GA在搜索中不依靠外部信息，仅以适应函数为依据，利用群体中每个染色体(个体)的适应值来进行搜索。以染色体适应值的大小来确定该染色体被遗传到下一代群体中的概率。染色体适应值越大，该染色体被遗传到下一代的概率也越大；反之，染色体的适应值越小，该染色体被遗传到下一代的概率也越小。因此适应函数的选取至关重要，直接影响到GA的收敛速度以及能否找到最优解。
- 群体中的每个染色体都需要计算适应值
- 适应函数一般由目标函数变换而成

适应函数(Fitness Function)

- 适应函数常见形式：
  - ✓ 直接将目标函数转化为适应函数
    - 若目标函数为最大化问题：

      $$Fitness(f(x)) = f(x)$$

    - 若目标函数为最小化问题：

      $$Fitness(f(x)) = -f(x)$$

    - 缺点：

      (1) 可能不满足轮盘赌选择中概率非负的要求

      (2) 某些代求解的函数值分布上相差很大，由此得到的评价适应值可能不利于体现群体的评价性能，影响算法的性能。

适应函数(Fitness Function)

✓ 界限构造法

- 目标函数为最大化问题

$$\text{Fitness}(f(x)) = \begin{cases} f(x) - C_{\min}, & f(x) > C_{\min} \\ 0, & others \end{cases}$$

其中C_{min}为f(x)的最小估计值

- 目标函数为最小化问题

$$\text{Fitness}(f(x)) = \begin{cases} C_{\max} - f(x), & f(x) < C_{\max} \\ 0, & others \end{cases}$$

其中C_{maxn}为f(x)的最大估计值

停止准则(Termination Criteria)

- 种群中个体的最大适应值超过预设定值
- 种群中个体的平均适应值超过预设定值
- 种群中个体的进化代数超过预设定值

基本步骤(Step by Step)

(1) 随机产生初始种群;

(2) 计算种群体中每个个体的适应度值,判断是否满足停止条件，若不满足，则转第(3)步,否则转第(7)步;

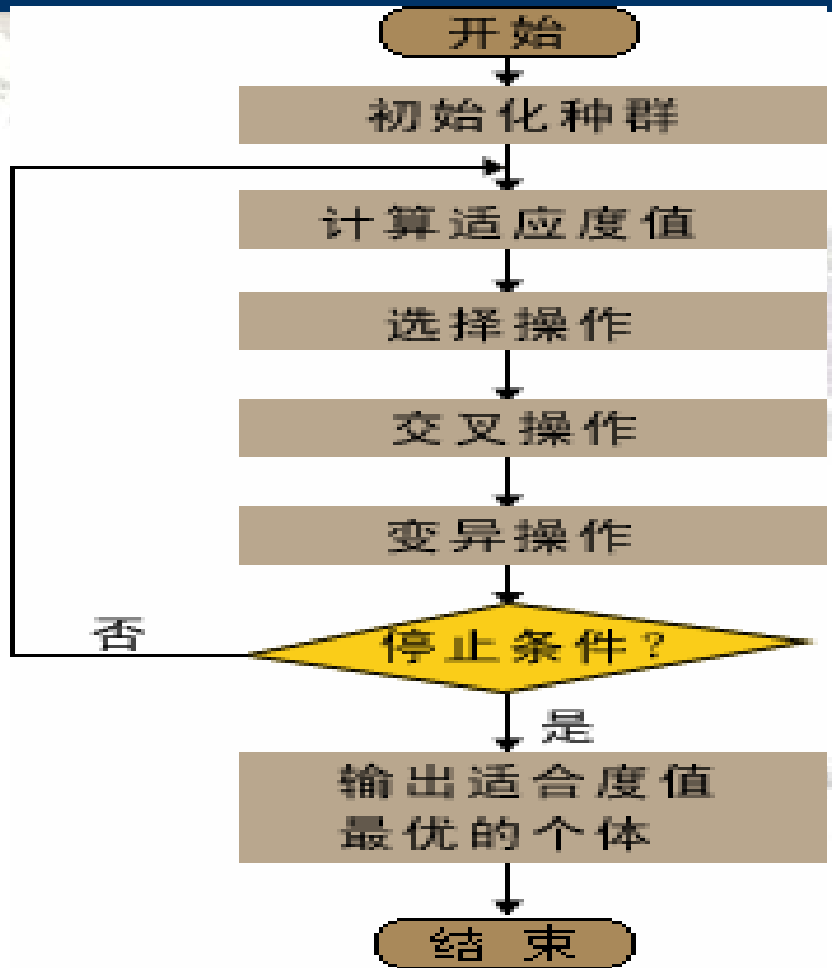
(3) 按由个体适应值所决定的某个规则选择将进入下一代的个体;

(4) 按交叉概率P_c进行交叉操作,生产新的个体;

(5) 按变异概率P_m进行变异操作,生产新的个体;

(6) 输出种群中适应度值最优的染色体作为问题的满意解或最优解。

流程图(Flow Chart)



基本遗传算法

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。

SGA伪码描述

Procedure Genetic Algorithm

```
begin
  t = 0 ;
  初始化 P(t) ;
  计算 P(t) 的适应值 ;
  while (不满足停止准则) do
      begin
           t = t+1 ;
           从P(t-1)中选择 P(t) ;     % selection
           重组 P(t) ;                % crossover and mutation
            计算 P(t) 的适应值;
      end
end
```

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。

遗传算法的应用

- 组合优化　遗传算法是寻求组合优化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。例如，遗传算法已经在求解旅行商问题(Traveling Salesman Problem，TSP)、背包问题(Knapsack Problem)、装箱问题(Bin Packing Problem) 等方面得到成功的应用。
- 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。现在遗传算法已经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，例如基于遗传算法的模糊控制器的优化设计、基于遗传算法的参数辨识、基于遗传算法的模糊控制规则的学习、利用遗传算法进行人工神经网络的结构优化设计和权值学习等。
- 机器人智能控制　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，所以机器人智能控制自然成为遗传算法的一个重要应用领域。

遗传算法的应用

- 图象处理和模式识别　图像处理和模式识别是计算机视觉中的一个重要研究领域。在图像处理过程中，如扫描、特征提取、图像分割等不可避免地存在一些误差，这些误差会影响图像处理的效果。如何使这些误差最小是使计算机视觉达到实用化的重要要求，遗传算法在这些图像处理中的优化计算方面得到了很好的应用。
- 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。

遗传算法的应用

- 遗传程序设计　Koza发展了遗传程序设计的概念，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，在很多领域中都得到了应用。例如基于遗传算法的机器学习可用来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。分类器系统在多机器人路径规划系统中得到了成功的应用。

SGA实例1：函数最值

求函数$f(x)=x^2$的最大值，x为自然数且$0 \leqslant x \leqslant 31$.

SGA参数:

- ✓ 编码方式：二进制码

  e. g. 00000↔0； 01101 ↔ 13； 11111↔31
- ✓ 种群规模：4
- ✓ 随机初始群体
- ✓ “转盘赌”选择
- ✓ 一点杂交，二进制变异

- 手工方式完成演示SGA过程

SGA实例1 max x^2 ：选择操作

编号	初始群体	变量 x	适应值 $f(x)=x^2$	选择比	期望	实际数目
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
和			1170	1.00	4.00	4
平均值			293	0.25	1.00	1
最大值			576	0.49	1.97	2

SGA实例1 max x^2 ：交叉操作

编号	交配池	交叉位置	交叉后 新的种群	变量x	适应值 $f(x)=x^2$
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
和 平均值 最大值					1754 439 729

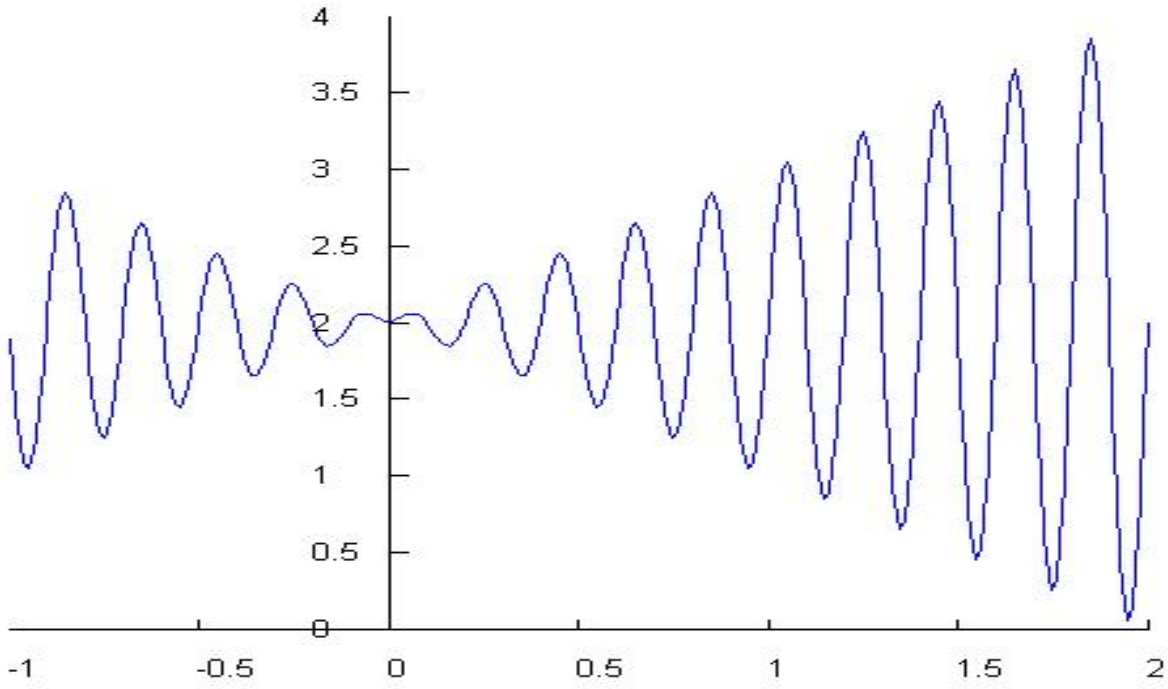
SGA实例1 max x^2 ：变异操作

编号	交叉后 新的种群	变异后 新的种群	变量x	适应值 $f(x)=x^2$
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	1 1 0 1 1	1 1 0 1 1	27	729
4	1 0 0 0 0	1 0 1 0 0	18	324
和 平均值 最大值				2354 588.5 729

SGA实例2：连续函数最值

求下列函数的最大值：

$$f(x) = x\sin(10\pi x) + 2.0 \qquad x \in [-1, 2]$$



SGA实例2：编码

- 编码
  - ✓ 实数问题：变量x为实数，如何把z∈[x,y]⟶$\{a_1,\ldots,a_L\} \in \{0,1\}^L$
  - ✓ [x,y] → {0,1}L 必须可逆(一个表现型对应一个基因型)
  - ✓ 解码算子：Γ: $\{0,1\}^L \rightarrow [x,y]$

$$\Gamma(b_L,\ldots,a_0) = x + \frac{y-x}{2^L-1}\cdot\left(\sum_{i=0}^{L-1} b_i \cdot 2^i\right) \in [x,y]$$

  - ✓ 染色体长度L决定可行解的最大精度
  - ✓ 高精度 ⟷ 长染色体(慢进化)

SGA实例2：编码

设定求解精确到6位小数，因区间长度位2-(-1)=3,则需将区间分为3X10^{6}等份。因 2097152＝2^{21}< 3X10^6≤2^{22}＝4194304。故编码的二进制串长L=22。

将一个二进制串(b21b20…b0)转化为10进制数：

$$(b_{21},...,b_0)_2 = -1+\frac{2-(-1)}{2^L-1}\cdot(\sum_{i=0}^{21} b_i\cdot 2^i)_{10} \in [-1,2]$$

e.g. <0000000000000000000000> ↔ -1;
<1111111111111111111111> ↔ 2
<1110000000111111000101> ↔ 1.627 888
1.627888 = -1+3x(1110000000111111000101) 2 /(2^{22}-1)
= -1+3x3674053/(222-1)

SGA实例2：初始化种群、适应函数

- 随机初始化种群
- 适应函数

本实例目标函数在定义域内均大于0，且是求函数最大值，故直接引用目标函数作为适应函数：

$$f(s) = f(x)$$

其中二进制串s对于变量x的值。

e.g. s_1 =<000000111000000010000> $\leftrightarrow$ x_1= -0.958 973

适应值: $f(s_1) = f(x_1)$ =1.078 878

s_2=<111000000111111000101> $\leftrightarrow$ x_2= 1.627 888

适应值: $f(s_2) = f(x_2)$ = 3.250 650

SGA实例2 :遗传操作

- 选择操作(“轮盘赌”选择)
- 交叉操作(单点交叉)

交叉前(父): s_1=<00000 | 0111000000010000>

s_2=<11100 | 0000011111000101>

交叉后(子): s'_1=<00000 | 0000011111000101>

s'_2=<11100 | 0111000000010000>

适应值: $f(s'_1) = f(-0.998\ 113) = 1.940\ 865$

$f(s'_2) = f(1.666\ 028) = 3.459\ 245$

s'_2的适应值比其双亲个体的适应值高。

SGA实例2 :遗传操作

- 变异操作

  变异前(父): s_2=<11100000001111111000101>

  变异后(子): s'_2=<11101000001111111000101>

  适应值 $f(s'_2) = f(1.721\ 638) = 0.917\ 743$ 比 $f(s_2)$小

  变异前(父): s_2=<11100000001111111000101>

  变异后(子): s''_2=<11100000011111111000101>

  适应值 $f(s''_2) = f(1.630\ 818) = 3.343\ 555$ 比 $f(s_2)$大

变异操作有"扰动"作用，同时具有增加种群多样性的效果。

SGA实例2 :模拟结果

遗传算法的参数:

- 种群规模: **50**
- 染色体长度: **L=22**
- 最大进化代数: **150**
- 交叉概率: **P_c=0.25**
- 变异概率: **P_m=0.01**

SGA实例2 :模拟结果(最佳个体进化情况)

世代数	染色体编码	变量x	适应值
1	100011100001011000111 1	1.831 624	3.534 806
4	000001101100010100111 1	1.842 416	3.790 362
11	011010101110011100111 1	1.854 860	3.833 286
17	111010101111111010011 11	1.847 536	3.842 004
34	110000110111101100111 1	1.853 290	3.843 402
40	110100100010001100111 1	1.848 443	3.846 232
54	100011011010001100111 1	1.848 699	3.847 155
71	010011011000101100111 1	1.850 897	3.850 162
89	110100111111001100111 1	1.850 549	3.850 274
150	110100111111001100111 1	1.850 549	3.850 274

最优化问题(Optimization Problem)

最优化问题:

$$Minimize \quad f(x) = f(x_1, x_2, \cdots, x_n)$$

$$subject\ to \quad x = (x_1, x_2, \cdots, x_n) \in S \subset X$$

组合优化问题(Combinatorial Optimization Problem) : 最优化问题中的解空间X或S由离散集合构成。其中很多问题是NP完全(Nondeterministic Polynomial Completeness)问题.

最优化问题算法

传统的优化方法(局部优化方法)

共轭梯度法、牛顿法、单纯形方法等

特点:

1)依赖于初始条件。

2)与求解空间有紧密关系，促使较快地收敛到局部 解，但同时对　解域有约束，如连续性或可微性。利用这些约束，收敛快。

3)有些方法，如Davison-Fletcher-Powell直接依赖于至少一阶导数；共轭梯度法隐含地依赖于梯度。

最优化问题算法

全局优化方法

下降轨线法、Monte-Carlo随机试验法、模拟退火法、GA等

特点：

1)不依赖于初始条件；

2)不与求解空间有紧密关系,对解域无可微或连续的要求;容易实现，求解稳健。

3)但收敛速度慢,能获得全局最优;适合于求解空间不知的情况。

4)GA可应用于大规模、多峰多态函数、含离散变量等全局优化问题;求解速度和质量远超过常规方法。

无约束最优化问题

无约束最优化问题:

$$Minimize \quad f(x_1, x_2, \cdots x_n)$$

$$subject\ to \quad l_i \le x_i \le u_i \quad i = 1, 2, \cdots, n$$

GA编码:

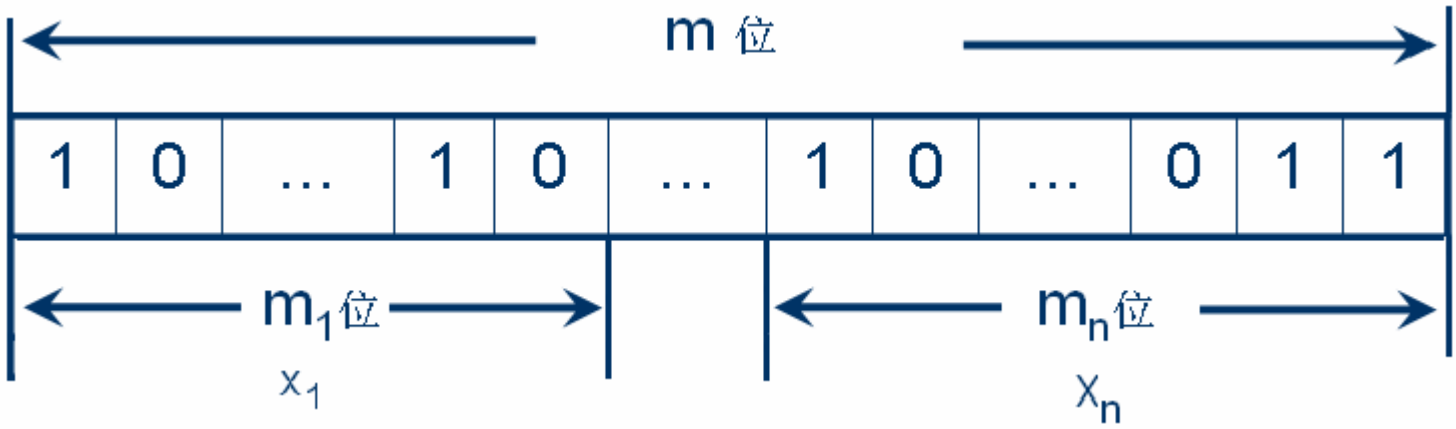
$X=(x_1,x_2,\ldots,x_n)$的各个变量可以按二进制编码方法分别编码。对于变量x_i的上、下限约束$l_i \leqslant x_i \leqslant u_i (i=1,2,\ldots,n)$，依据解的精度要求(有效位数)求得各个变量$X=(x_1,x_2,\ldots,x_n)$的二进制码位数$(m_1,m_2,\ldots,m_n)$(确定方法类似于SGA实例2)，因此将n个二进制位串顺序连接起来，构成一个个体的染色体编码，编码的总位数$m=m_1+m_2+\ldots+m_n$。

无约束最优化问题

GA解码：

解码时仍按各个变量的编码顺序分别实现常规的二进制编码解码方法。

二进制遗传编码示意图如下：



约束最优化问题

约束最优化问题:

$$Minimize \quad f(x_1, x_2, \cdots x_n)$$

$$g_i(x) = 0, \quad i = 1, 2, \cdots, m$$

$$h_i(x) \le 0, \quad i = 1, 2, \cdots, l$$

$$l_i \le x_i \le u_i, i = 1, 2, \cdots, n$$

常规解法:

(1)把约束问题转化为无约束问题，在用无约束问题方法求解，如罚函数法

(2)改进无约束问题的方法，再用于约束问题，如梯度投影法、广义简约梯度法

约束最优化问题

- **遗传算法求解关键：约束条件的处理**

等式约束可以包含到适应函数，仅考虑不等式约束。

假设按无约束问题那样求解，在搜索过程中计算目标函数值，并检查是否有约束违反。如果没有违反，则表明是可行解，就根据目标函数指定一适应值；否则，就是不可行解，因而没有适应值(适应值为0)。这样的处理实际不可行，因为找到一个可行解几乎与找到最优解一样困难。

一般解法：通过引入罚函数，从不可行解中得到一些信息。将罚函数包含到适应函数中。

- **关键是如何设计罚函数；**
- **不同问题需要设计不同的罚函数；**
- **对一般的约束处理，通常很困难。**

组合最优化问题

典型问题：

巡回旅行商问题(Traveling Salesman Problem)

作业调度问题(Job Shop Scheduling Problem)

背包问题(Knapsack Problem)

图着色问题

… … …

很多组合最优化问题是NP难问题或NP完全问题

巡回旅行商问题(TSP)

TSP，也称货郎担问题，是一个NP完全问题。

TSP描述：

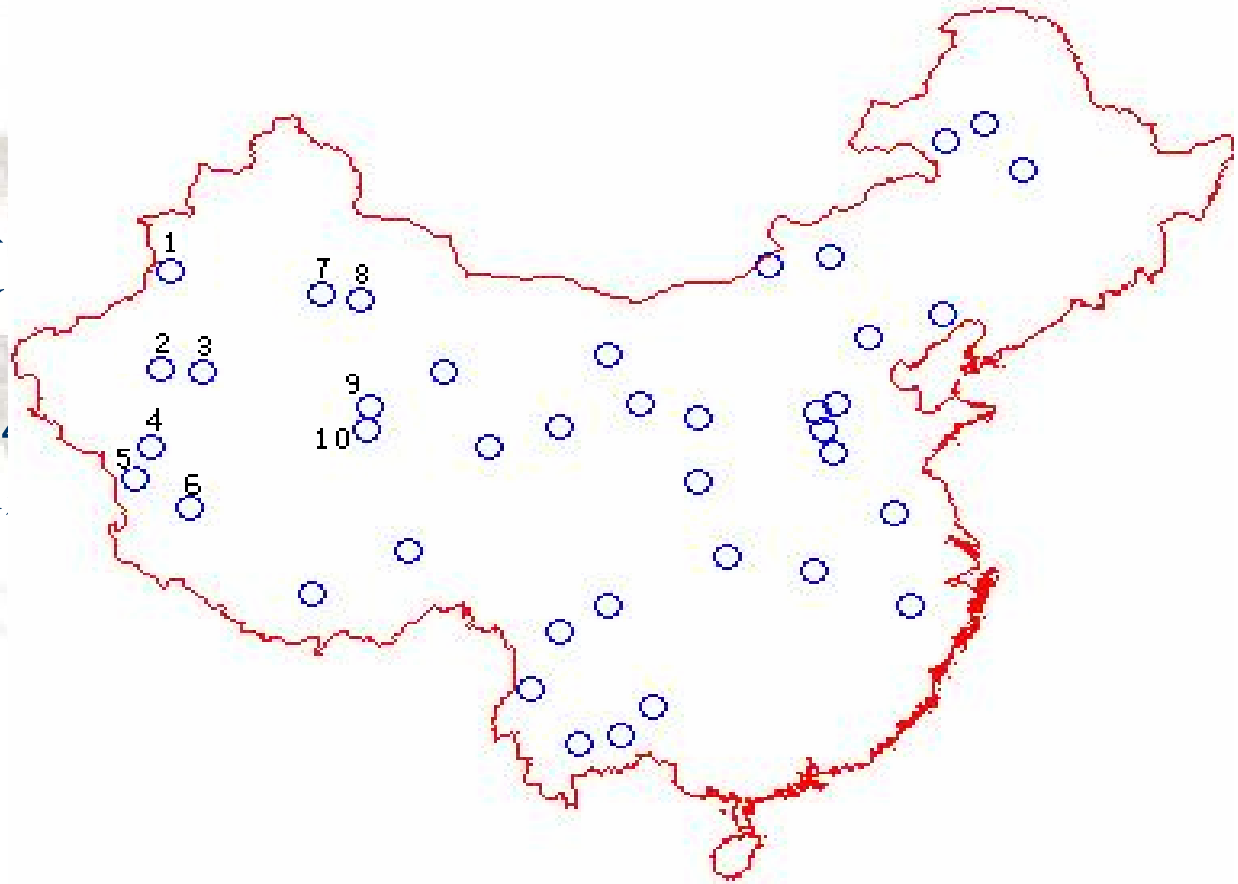
- 图论:设图**G=(V,E)**,其中**V**是顶点集，**E**是边集。设**C=(C_{ij})**是与**E**相联系的距离矩阵。寻找一条通过所有顶点且每个顶点只通过一次的最短距离回路**(Hamilton**回路**)**。实际应用中，**C**也可解释为费用或旅行时间矩阵。

- 实际:一位推销员从自己所在城市出发，必须遍访所有城市之后又回到原来的城市，求使其旅行费用最少的路径。

巡回旅行商问题(TSP)

中国货郎担问题

- 城市数: 40
- 城市编号1,2,…,
- 寻找一条最短路



TSP复杂性

搜索空间庞大

TSP涉及求多个变量的函数的最小值，求解很困难。其可能的路径条数随着城市数目n成指数增长，如，5个城市对应12条路径；10个城市对应181 440条路径；100个城市对应4.6663×10^{155}条路径。如此庞大的搜索空间，常规解法和计算工具都遇到计算上的困难。只能寻找近似解法，如神经网络方法、模拟退火法、遗传算法等。

TSP编码:路径表示

染色体表示成所有城市的一个排列，若有n个城市，一条可能路径编码为长度为n的整数向量($i_1,i_2,\ldots,i_n$),其中i_k表示第i_k个城市。例如: 路径编码向量(5 1 7 8 9 4 6 2 3)直接表示一条旅行路程(5->1->7->8->9->4->6->2->3)。

此向量是1到n的一个排列，即从1到n的每个整数在这个向量中正好出现一次,不能有重复。这样，基本遗传算法的基因操作生成的个体不能满足这一约束条件，需寻求其他遗传操作。

TSP交叉

一般的交叉操作会产生不合适的解，如

需其他方式的交叉(重组)操作,如部分匹配交叉(**Partially Matched Crossover,PMX**)、顺序交叉(**Ordered Crossover,OX**)、循环交叉(**Cycle Crossover,CX**)、边重组(**Edge Recombination**)。

TSP交叉1:部分匹配交叉(PMX)

- 双亲**P1,P2**随机选取两个交叉点，得到一个匹配段,根据交叉点中间段给出映射关系。

P1	1	2	3	4	5	6	7	8	9
P2	9	3	7	8	2	6	5	1	4

映射关系：

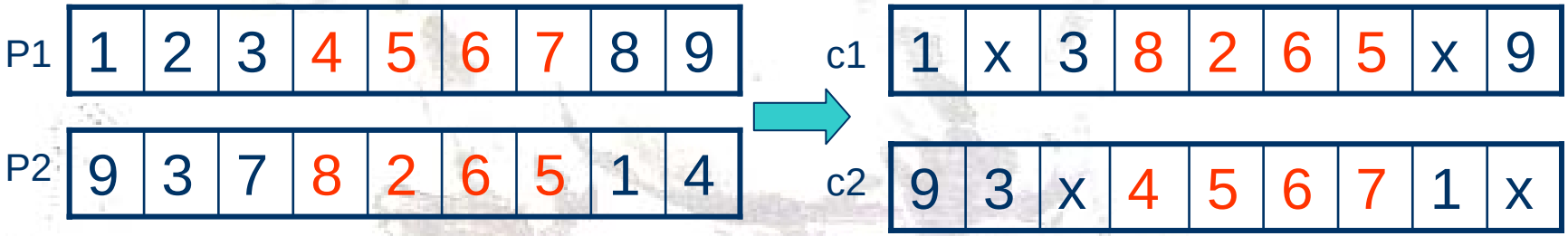
4 ↔8、5 ↔2、7 ↔5

- 交换两个交叉点之间的编码,(X表示未定码)

c1	x	x	x	8	2	6	5	x	x
c2	x	x	x	4	5	6	7	x	x

TSP交叉1:部分匹配交叉(PMX)

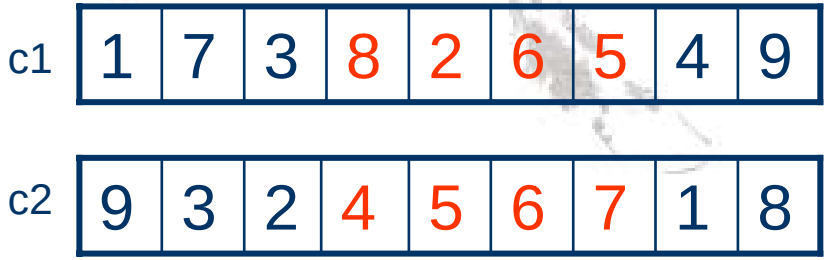
- 子个体C_1，C_2分别从其父个体中继承未映射城市码



- 再根据映射关系，对以上未定码，使用最初双亲码，得到子个体的对应码。映射关系存在传递关系，则选择未定码交换。

映射关系：

4 ↔8、5 ↔2、7 ↔5



TSP交叉2:顺序交叉(OX)

- 双亲**P1,P2**随机选取两个交叉点

P1	1	2	3	4	5	6	7	8	9
P2	9	3	7	8	2	6	5	1	4

- 两个交叉点间的中间段保存不变

c1	x	x	x	4	5	6	7	x	x
c2	x	x	x	8	2	6	5	x	x

- 子个体**C1**的未定码的确定需要父个体**P2**的未选定城市码，子个体**C2**的未定码的确定需要父个体**P1**的未选定城市码

TSP交叉2:顺序交叉(OX)

- 记取父个体P_2从第二个交叉点开始城市码的排列顺序，当到达表尾时，返回表头继续记录，直到第二个交叉点。

P2	9	3	7	8	2	6	5	1	4

c1	x	x	x	4	5	6	7	x	x

- 得到父个体**P2**的排列顺序**1-4-9-3-7-8-2-6-5,**并将**C1**已有城市码**4,5,6,7**从中去掉，得到排列顺序**1-9-3-8-2**，再将此顺序复制到**C1**，复制点也是从第二个交叉点开始，得到**C1**。

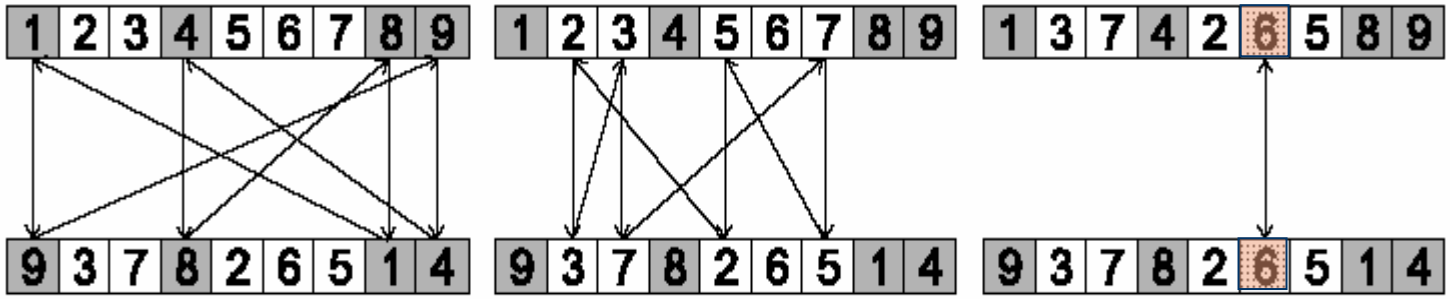
c1	3	8	2	4	5	6	7	1	9

同理的**C2**，

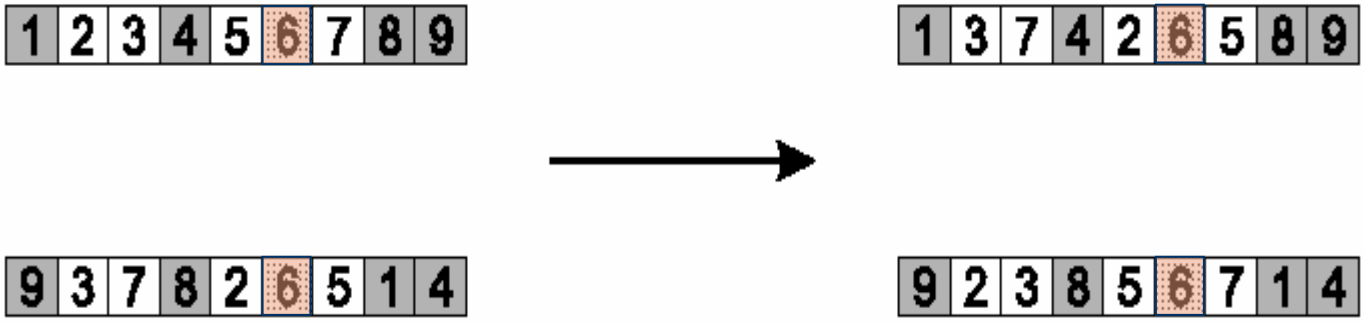
c2	3	4	7	8	2	6	5	9	1

TSP交叉3:循环交叉(CX)

- CX操作中子个体中的城市码顺序根据任一父个体产生
- 确定循环编码



- 复制循环编码到子个体



TSP变异

- **Insert Mutation**

随机选取个体中两个编码，然后把第二个编码放在第一个编码之后，其他编码顺次调节位置。

- **Swap mutation**

随机选取个体中两个编码，然后交换它们的位置。

TSP变异

- **Inversion mutation**

随机选取个体中一段编码，然后颠倒这段编码的顺序。

- **Scramble mutation**

随机选取个体上一段编码,然后打乱这段编码的顺序。

选取的编码不一定是邻接编码

TSP的GA过程

- 从N个随机起点开始产生N条路径，N为种群的规模;
- 利用最优方法搜索每条路径的局部最优解;
- 选择交叉对使在平均性能之上的个体得到更多的子代;
- 交叉和变异;
- 搜索每条路径得到其极小解，如果不收敛，则回到第3步；否则，停止。

GA的MATLAB实现

软件平台(Software Platforms):

- MATLAB 7.x
  - ✓ Genetic Algorithm and Direct Search Toolbox 2.0.1
- MATLAB 6.x(or 7.x)+GAOT
  - ✓ GAOT: Genetic Algorithm Optimization Toolbox
  - ✓ 美国North Carolina State University开发
- MATLAB 6.x(or 7.x)+GEATbx
  - ✓ GEATbx: Genetic and Evolutionary Algorithm Toolbox
  - ✓ 英国The University of Sheffield开发
  - ✓ 《**MATLAB**遗传算法工具箱及应用》**(**雷英杰等,西安电子科技大学出版社,**2005)**基于此工具箱

GAOT工具箱

核心函数：

(1) [pop]=**initializega**(num,bounds,eevalFN,eevalOps,options)-----初始种群的生成函数

【输出参数】

pop-----生成的初始种群

【输入参数】

num-----种群中的个体数目

bounds-----代表变量的上下界的矩阵

eevalFN-----适应度函数

eevalOps-----传递给适应度函数的参数

options-----选择编码形式(浮点编码或是二进制编码)与精度，如 [type prec]，

type-----为1时选择浮点编码，否则为二进制编码

prec-----变量进行二进制编码时指定的精度，默认[1e-6 1]

GAOT工具箱

(2) [x,endPop,bPop,traceInfo] =
ga(bounds,evalFN,evalOps,startPop,opts,termFN,termOps,selectFN,…
selectOps,xOverFNs,xOverOps,mutFNs,mutOps)
-------遗传算法函数

【输出参数】

x------求得的最优解

endPop------最终得到的种群

bPop------最优种群的一个搜索轨迹

traceInfo------每代种群中最优及平均个体构成的矩阵

【输入参数】

bounds------代表变量上下界的矩阵

evalFN------适应度函数

evalOps------传递给适应度函数的参数

startPop------初始种群

GAOT工具箱

【输入参数】

opts------- [epsilon prob_ops display]，opts(1:2)等同于initializega的options参数，第三个参数控制是否输出，一般为0。如[1e-6 1 0]

termFN-------终止函数的名称,如['maxGenTerm']

termOps-------传递个终止函数的参数,如[100]

selectFN-------选择函数的名称,如['normGeomSelect']

selectOps-------传递个选择函数的参数,如[0.08]

xOverFNs-------交叉函数名称表，以空格分开，如['arithXover heuristicXover simpleXover']

xOverOps-------传递给交叉函数的参数表，如[2 0;2 3;2 0]

mutFNs-------变异函数表，如['boundaryMutation multiNonUnifMutation nonUnifMutation unifMutation']

mutOps-------传递给交叉函数的参数表,如[4 0 0;6 100 3;4 100 3;4 0 0]

GAOT:函数最值(实例2)

求下列函数的最大值:

$$f(x) = x\sin(10\pi x) + 2.0 \qquad x \in [-1, 2]$$

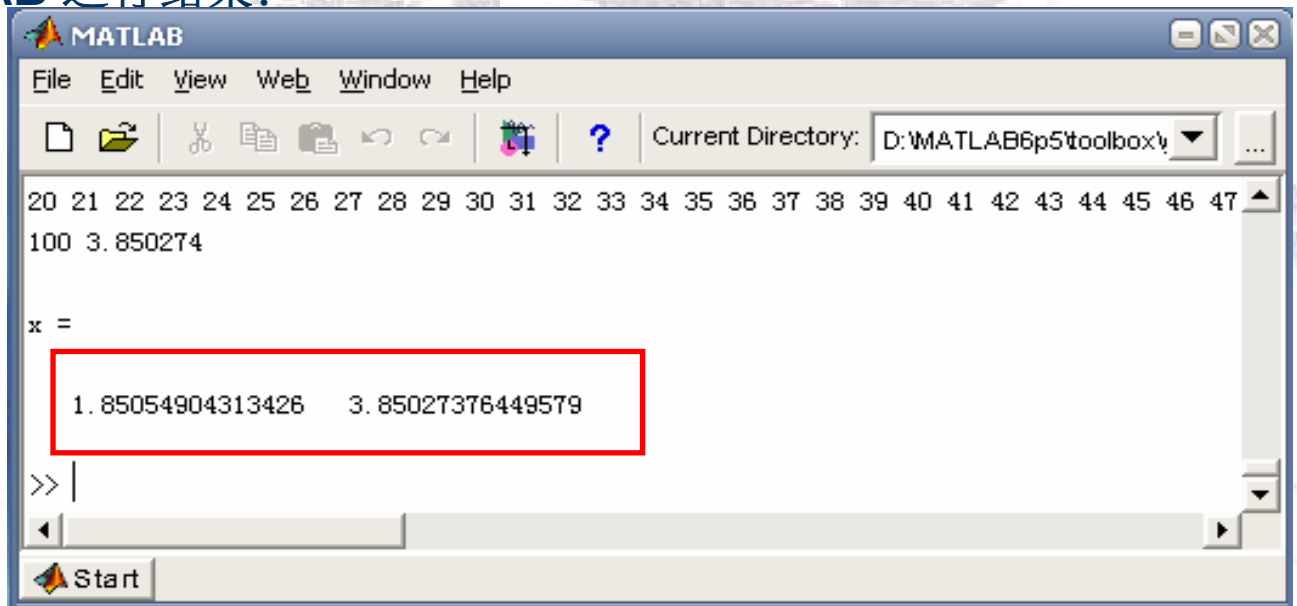
40	[illegible]1001111	1.848 443	3.846 232
54	10001101101000 11001111	1.848 699	3.847 155
71	01001101100010 11001111	1.850 897	3.850 162
89	11010011111110 011001111	1.850 549	3.850 274
150	11010011111110 011001111	1.850 549	3.850 274

GAOT:函数最值(实例2)

Fitness.m文件

```
function [sol,eval]=fitness(sol,options)
      x=sol(1);
      eval = x*sin(10*pi*x)+2.0;
```

MATLAB 运行结果：



Genetic Algorithm and Direct Search Toolbox (GA&DS)

包含工具:

遗传算法/直接搜索

运行方式: **command line; gatool**

GA解决的问题类:

- 最小化问题 $\min_x imize \quad f(x)$
- 最大化问题要转化为最小化问题

$$\max_x imize \quad f(x) \Rightarrow \min_x imize \quad -f(x)$$

Genetic Algorithm and Direct Search Toolbox (GA&DS)

- 最优化问题

$$\underset{x}{Minimize} \quad f(x)$$

$$such \quad that$$

$$Ax \le b$$

$$A_{eq}x = beq$$

$$C_i(x) \le 0, i = 1,\ldots,m$$

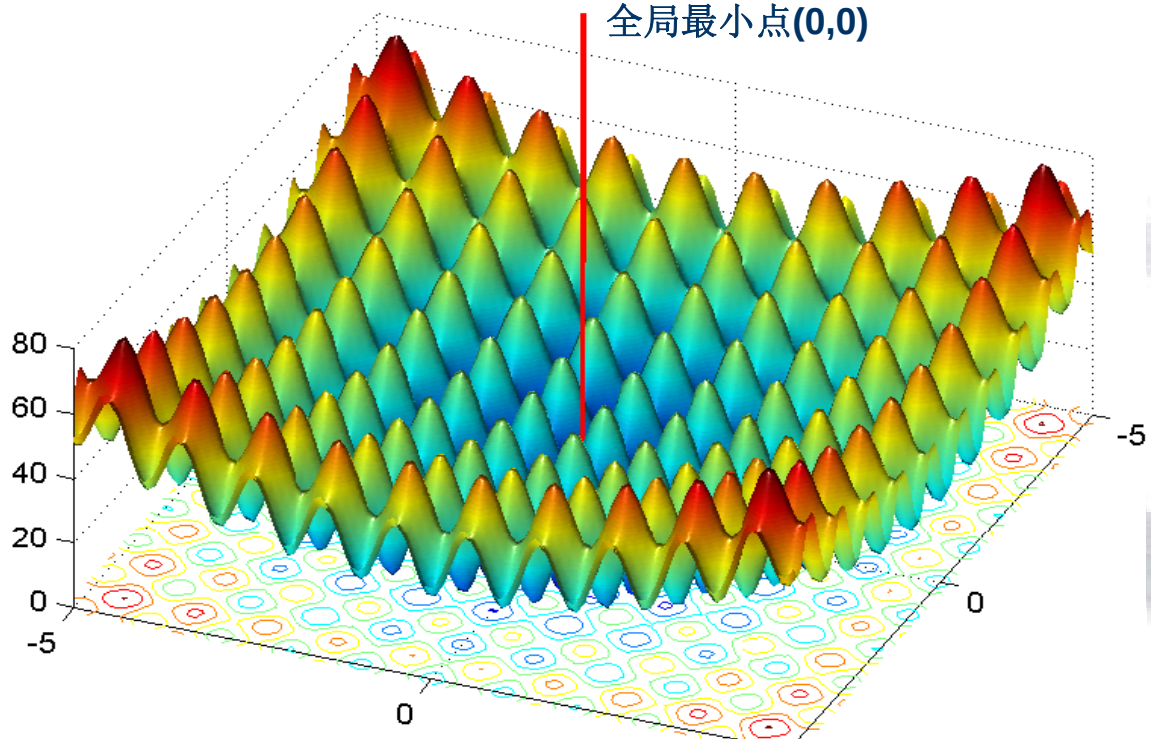
$$C_i(x) = 0, i = m+1,\ldots,mt$$

$$Lb \le x \le UB$$

算例1:函数最值

Rastrigin's Function

$$Ras(x) = 20 + x_1^2 + x_1^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$$



算例2:最优化

$$\max \quad x + y$$

$$such \quad that$$

$$x + y - 14 \le 0$$

$$y - 8 \le 0$$

$$x^2 + y^2 - 100 \le 0$$

$$x \ge 0, y \ge 0$$

最优解:(6,8),(8,6),(7,7)

$$\min \quad w - x - y - w y + w z + x y - x z$$

$$such \quad that$$

$$8 - w - 2x \ge 0$$

$$12 - 4w - x \ge 0$$

$$12 - 3w - 4x \ge 0$$

$$8 - 2y - z \ge 0$$

$$8 - y - 2z \ge 0$$

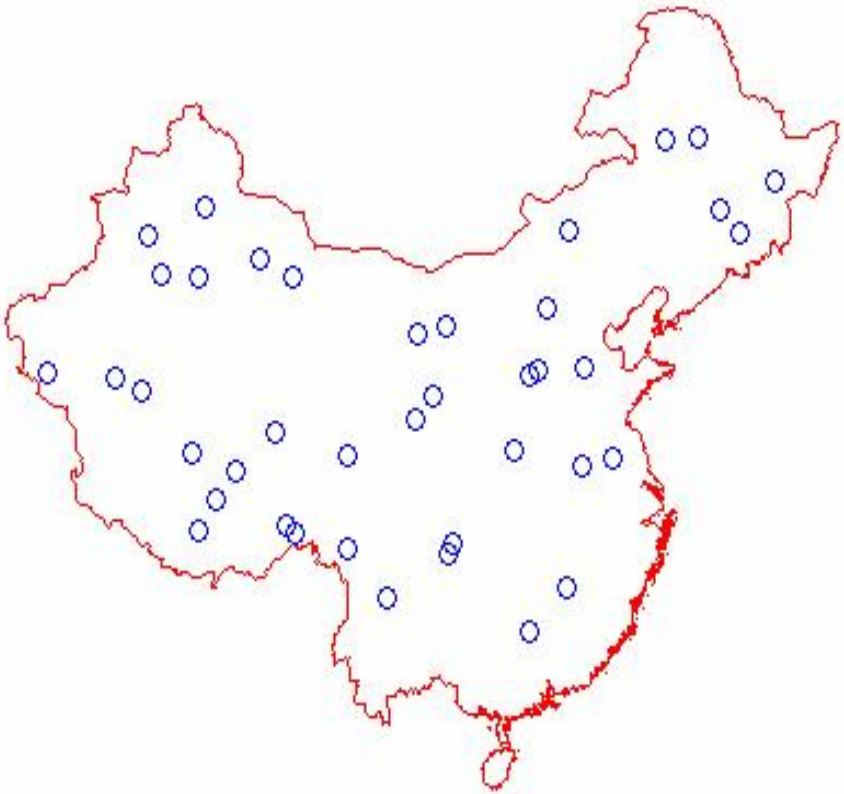
$$5 - y - z \ge 0$$

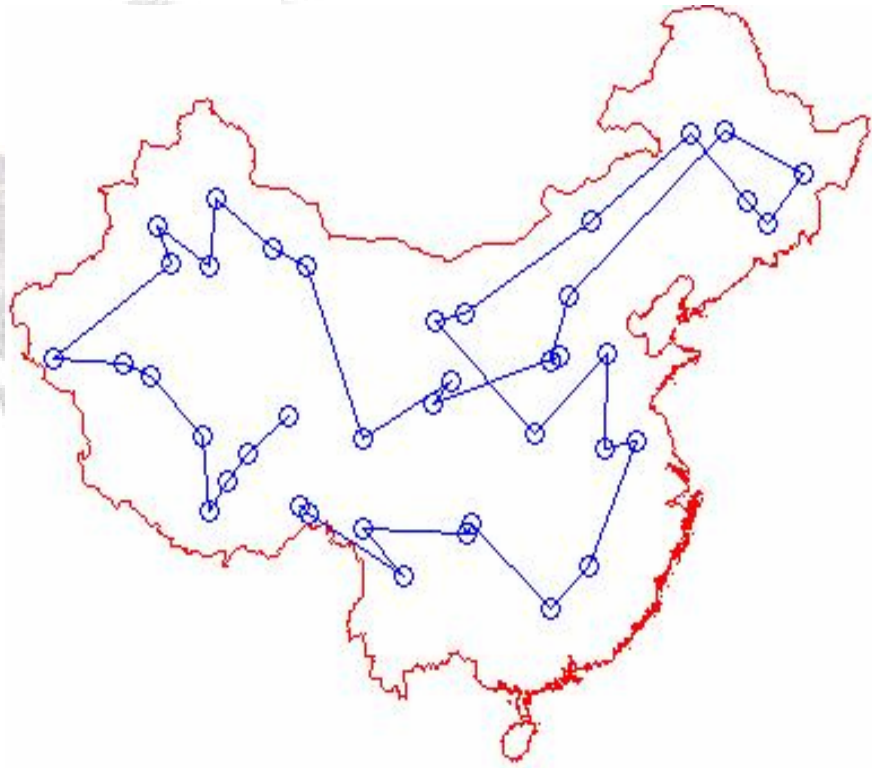
$$w \ge 0, x \ge 0, y \ge 0, z \ge 0$$

最优解:(0,3,0,4)

算例2:TSP

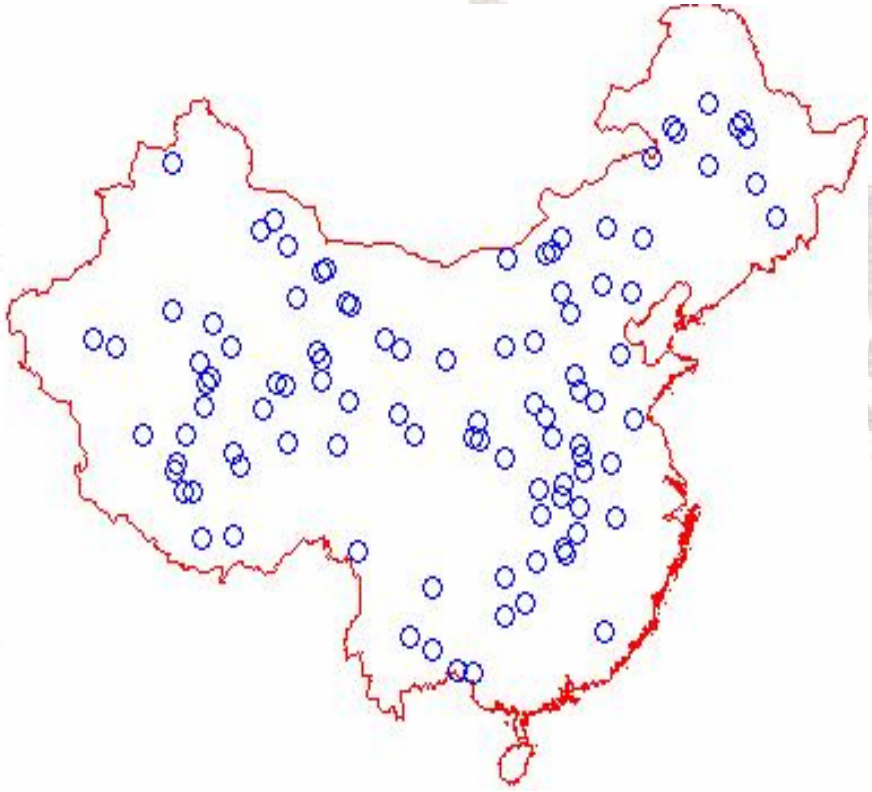
城市数：**40**

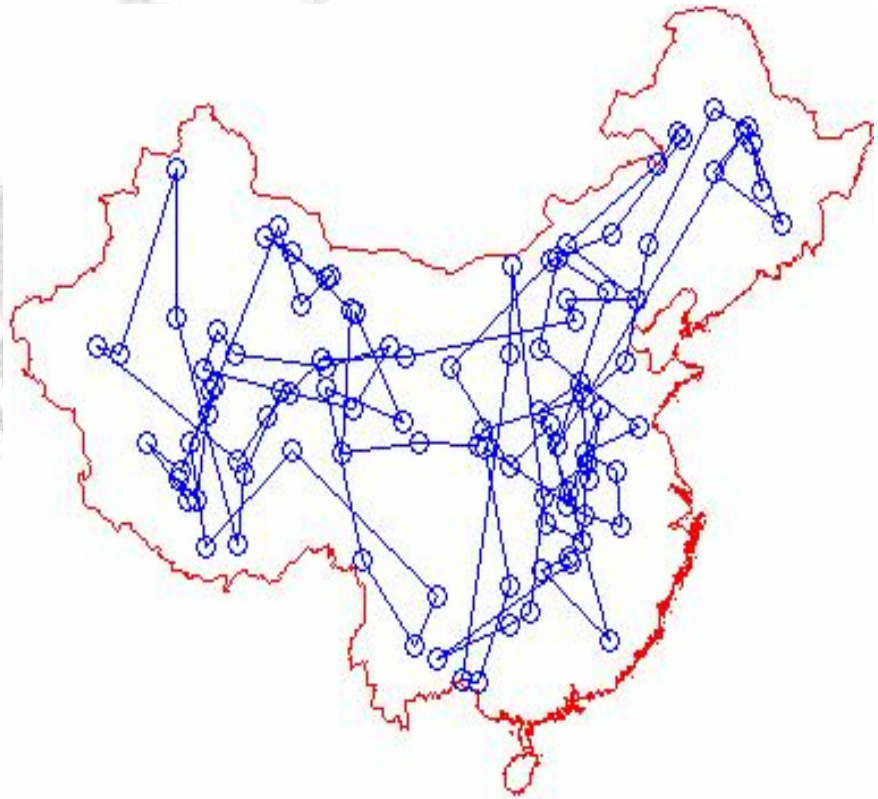




算例2:TSP

城市数：**100**





References

- *J. H. Holland. Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, Michigan, 1975.*
- *Z. Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin, third edition, 1996.*
- 康立山等，非数值并行算法(第二册)—遗传算法，科学出版社，2003.
- 王小平等，遗传算法—理论、应用与软件实现，西安交通大学出版社，2002.
- *www.mathworks.com*

Thank you!

Any question?

may you succeed in CUMCM