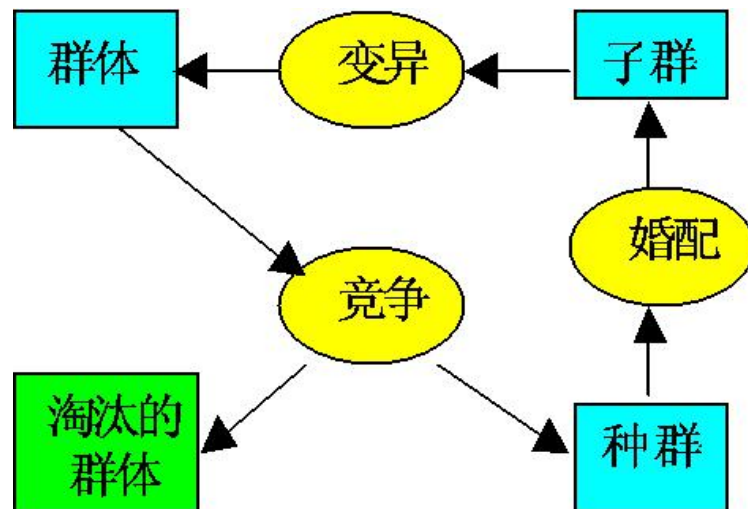
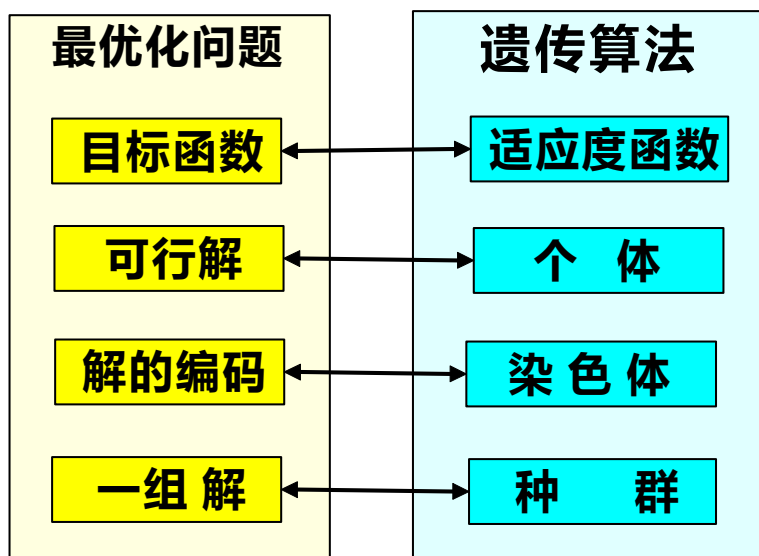


遗传算法实验

- ◆ 解的编码
- ◆ 初始种群的设定
- ◆ 适应度函数的设计
- ◆ 遗传操作（选择、交叉、变异）
- ◆ 算法控制参数的设定
- ◆ 约束条件的处理

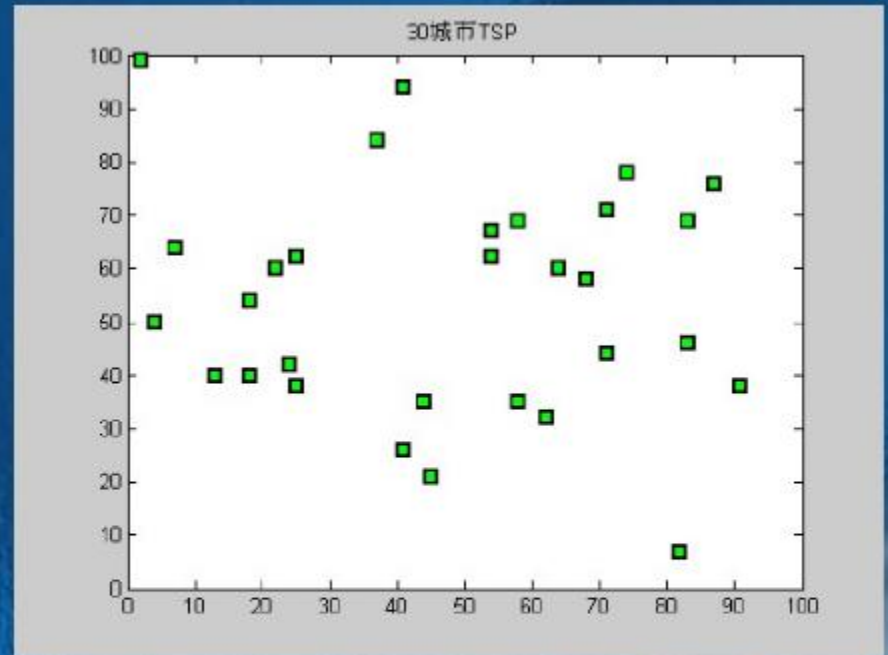
TSP问题：全国34城市为例
路径：31564.651km



遗传算法求解30个城市的TSP问题

◆ TSP Benchmark 问题

41 94;37 84;54 67;25 62;
7 64;2 99;68 58;71 44;54
62;83 69;64 60;18 54;22
60;83 46;91 38;25 38;24
42;58 69;71 71;74 78;87
76;18 40;13 40;82 7;62 32;
58 35;45 21;41 26;44 35;4 50



30个城市

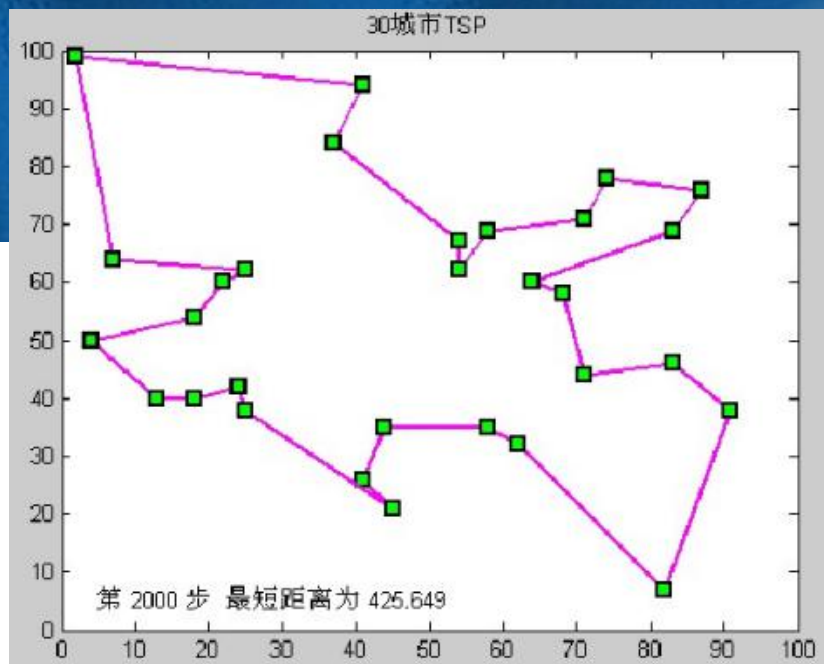
遗传算法求解30个城市的TSP问题

◆ TSP Benchmark 问题

编码: 直接采用解的表示形式, 30位 (30个城市) 长, 每位代表所经过的城市序号 (无重复);

适应度函数: 个体所代表的路径距离的倒数;

选择：轮盘赌方法



遗传算法求解30个城市的TSP问题

◆ TSP Benchmark 问题

交叉：有序交叉法

- 1) 随机选取两个交叉点;
- 2) 两个父个体交换中间部分;
- 3) 替换交换后重复的城市序号。

X1: 9 8 | 4 5 6 7 1 | 3 2 0 X1': 9 8 | 1 4 0 3 2 | 3 2 0

X2: 8 7 | 1 4 0 3 2 | 9 6 5 X2': 8 7 | 4 5 6 7 1 | 9 6 5



X1': 9 8 | 1 4 0 3 2 | 7 5 6

X2': 8 3 | 4 5 6 7 1 | 9 0 2



遗传算法求解30个城市的TSP问题

◆ TSP Benchmark 问题

变异：随机选择同一个个体的两个点进行交换；

初始参数：

种群规模 100

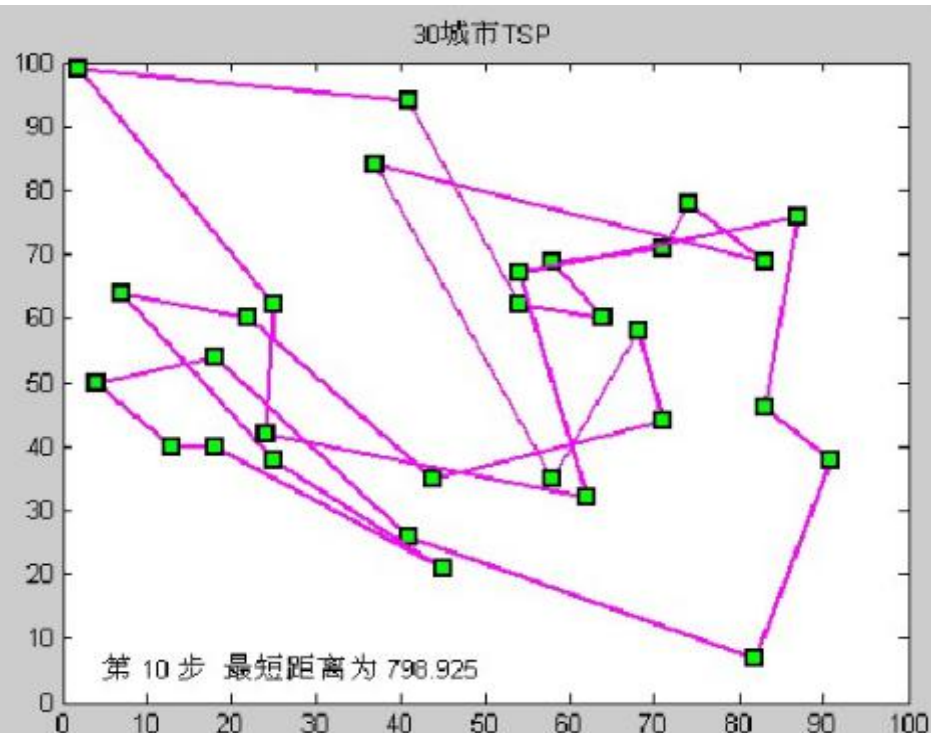
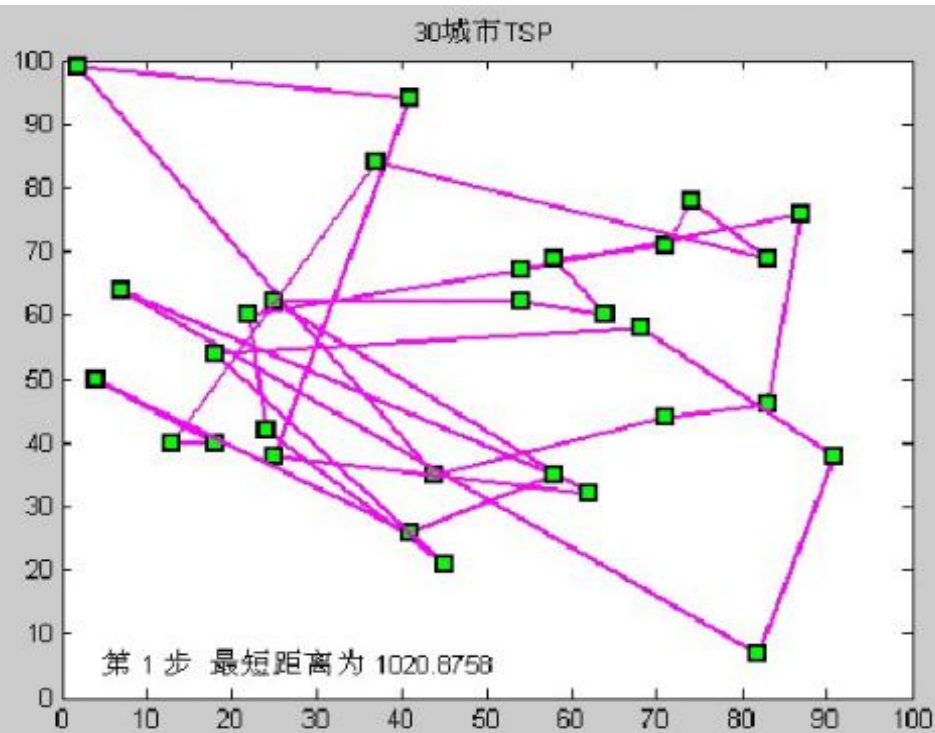
交叉概率 0.8

变异概率 0.8

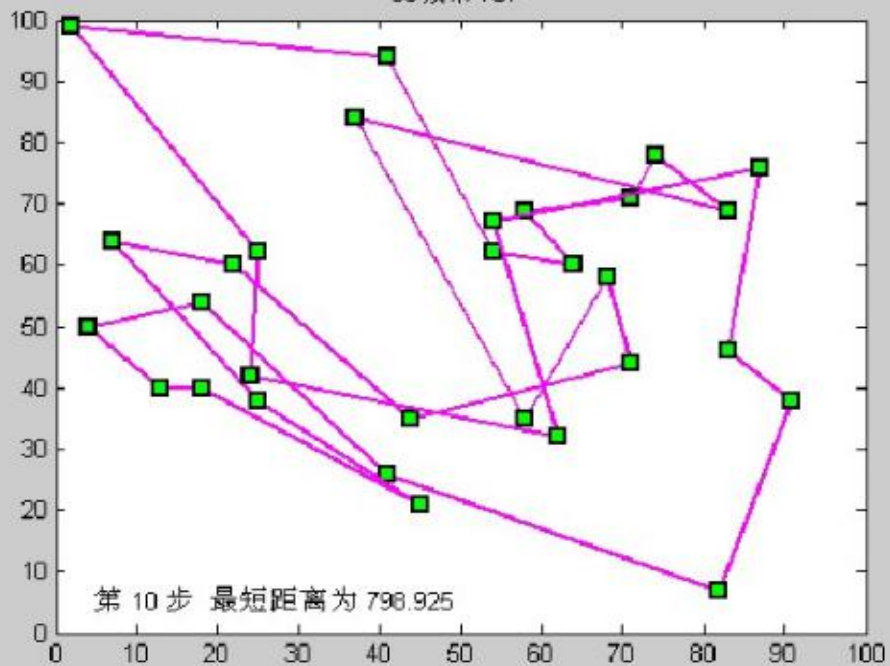
终止代数 2000

遗传算法求解30个城市的TSP问题

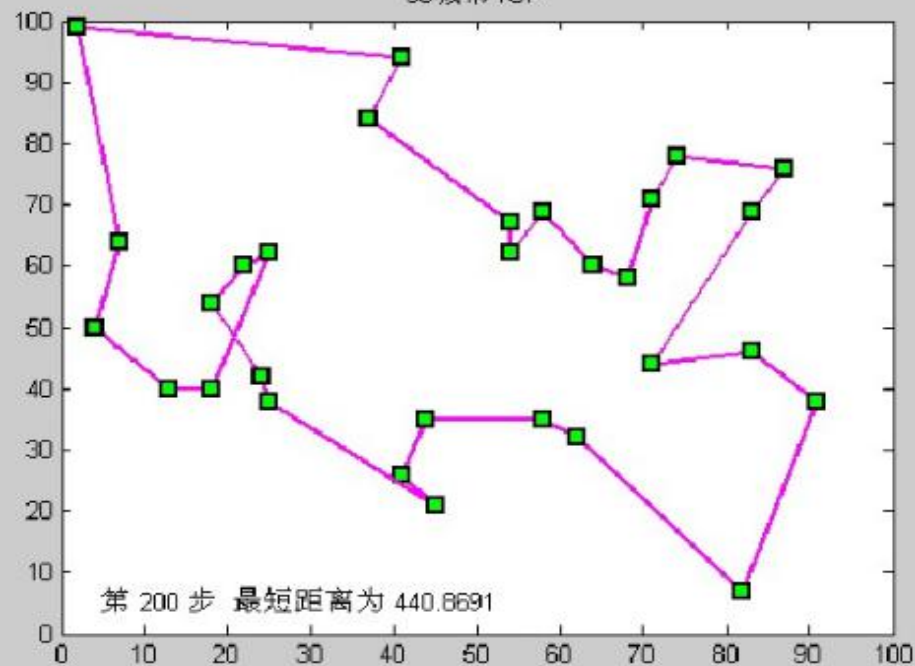
□ 运行结果:



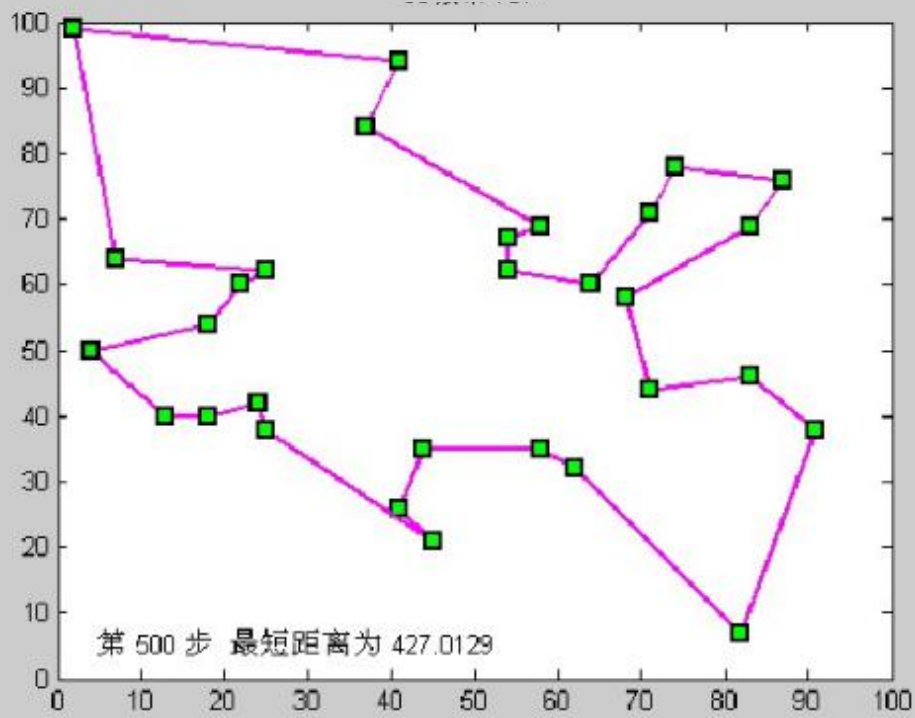
30城市TSP



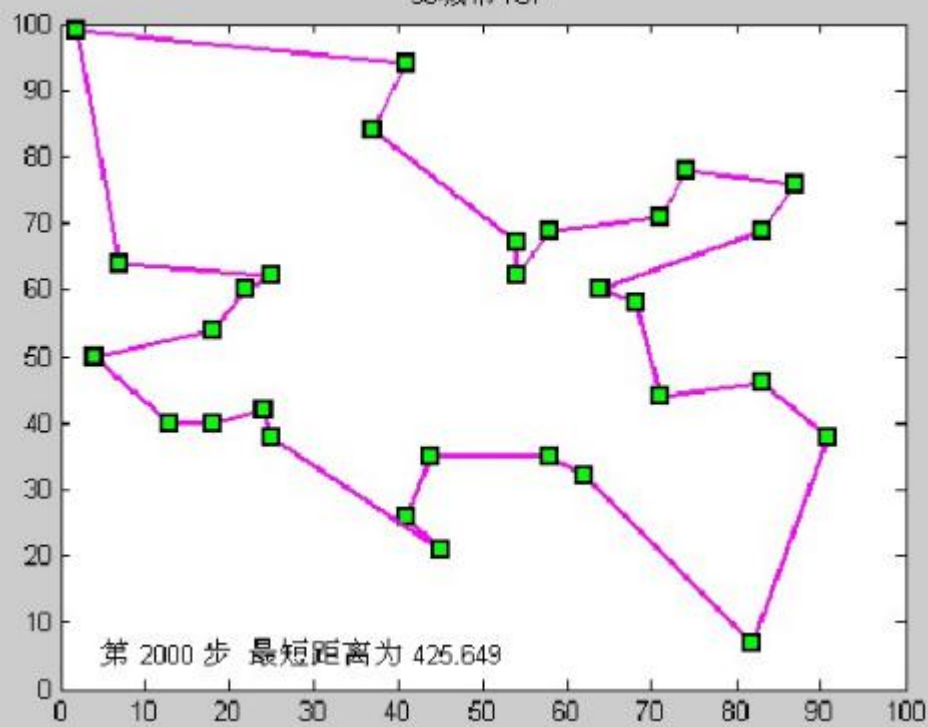
30城市TSP



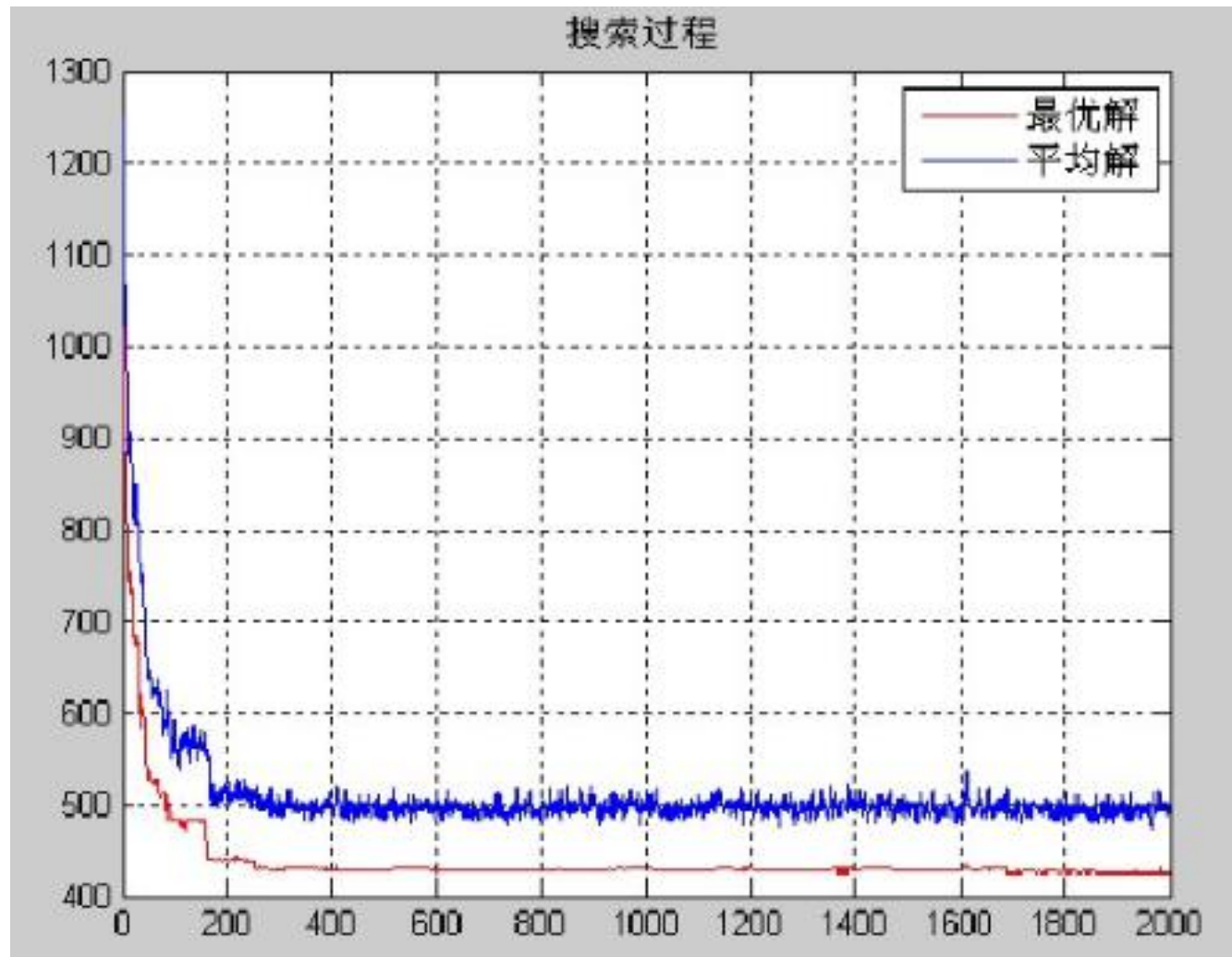
30城市TSP



30城市TSP



遗传算法求解30个城市的TSP问题



第6章 智能计算及其应用

□ 6.1 进化算法的产生与发展

□ 6.2 基本遗传算法

□ 6.3 遗传算法的改进算法

□ 6.4 遗传算法的应用

□ 6.5 群智能算法产生的背景

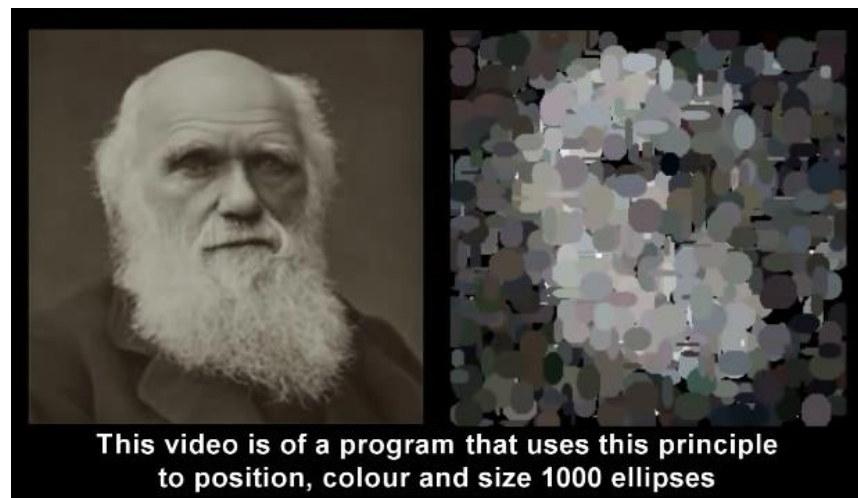
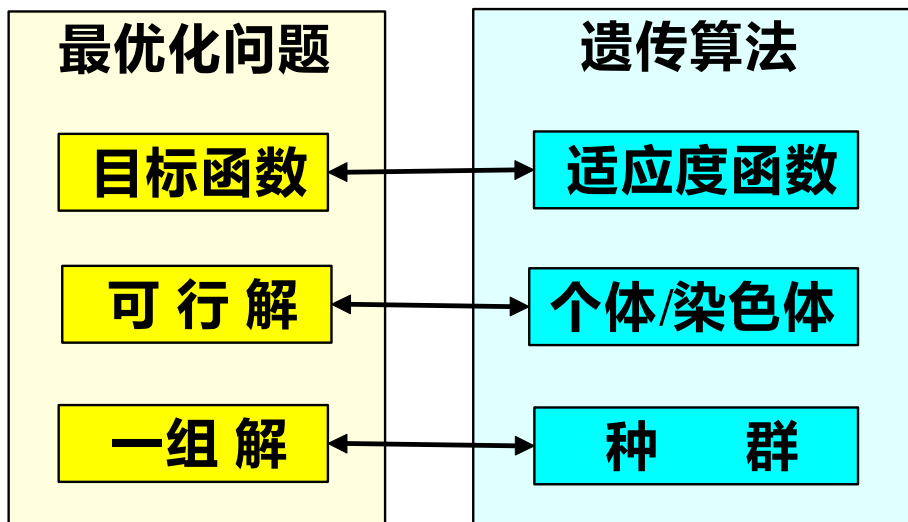
□ 6.6 粒子群优化算法及其应用

□ 6.7 蚁群算法及其应用

- MOOC上：第8讲的课堂讨论、单元测试、[在线作业](#)
- 网络教学平台作业W6-2
- 准备[下周四的遗传算法实验](#)

❑ 基本遗传算法的不足:

- 早熟: 遗传算法的探索能力是有限的, 易收敛到局部最优解。
- 大量计算: 当问题复杂时, 计算时间是个问题。
- 处理规模小。目前对于维数较高的问题, 还是很难处理和优化的。
- 稳定性差: 遗传算法属于随机类算法, 不能稳定的得到解, 需要多次运算。



6.3 遗传算法的改进算法

- 6.3.1 双倍体遗传算法
- 6.3.2 双种群遗传算法
- 6.3.3 自适应遗传算法

■ 基本遗传算法的不足:

- 早熟: 已陷入局部极值
- 大量计算
- 稳定性差

● 改进策略大致可以概括为以下几个方面:

(1) 改进遗传算法的组成成分或使用技术, 如选用优化控制参数、适合问题特性的编码技术等;

(2) 采用混合遗传算法, 例如禁忌搜索、模拟退火、混沌等;

(3) 采用动态自适应技术, 在进化过程中调整控制参数;

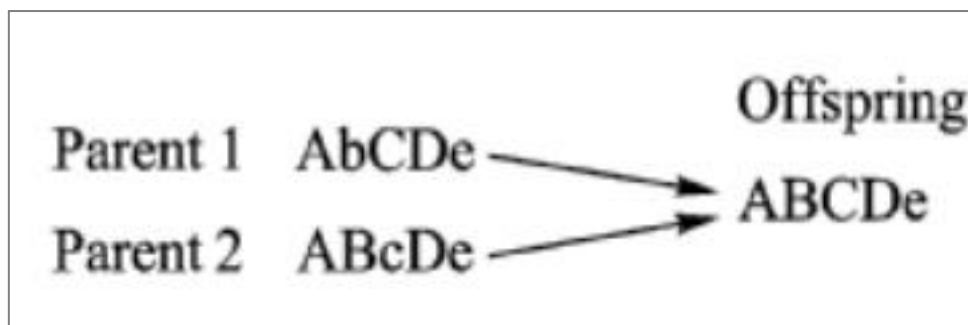
(4) 采用非标准的遗传操作算子, 例如灾变;

(5) 采用并行遗传算法。

6.3.1 双倍体遗传算法

1. 基本思想

- 双倍体遗传算法采用显性和隐性两个染色体同时进行进化，提供了一种记忆以前有用的基因块的功能。
- 双倍体遗传算法采用显性遗传。



- 双倍体遗传延长了有用基因块的寿命，提高了算法的收敛能力，在变异概率低的情况下能保持一定水平的多样性。

6.3.1 双倍体遗传算法

2. 双倍体遗传算法的设计

- (1) 编码：两个染色体（显性、隐性）
- (2) 复制(选择)算子：计算显性染色体的适应度，按照显性染色体的选择概率将个体复制到下一代群体中。
- (3) 交叉算子：两个个体的显性染色体交叉、隐性染色体也同时交叉。
- (4) 变异算子：个体的显性染色体按正常的变异概率变异；隐性染色体按较大的变异概率变异。
- (5) 显隐性重排算子：个体中适应值较大的染色体设为显性染色体，适应值较小的染色体设为隐性染色体。

6.3 遗传算法的改进算法

- 6.3.1 双倍体遗传算法
- 6.3.2 双种群遗传算法
- 6.3.3 自适应遗传算法

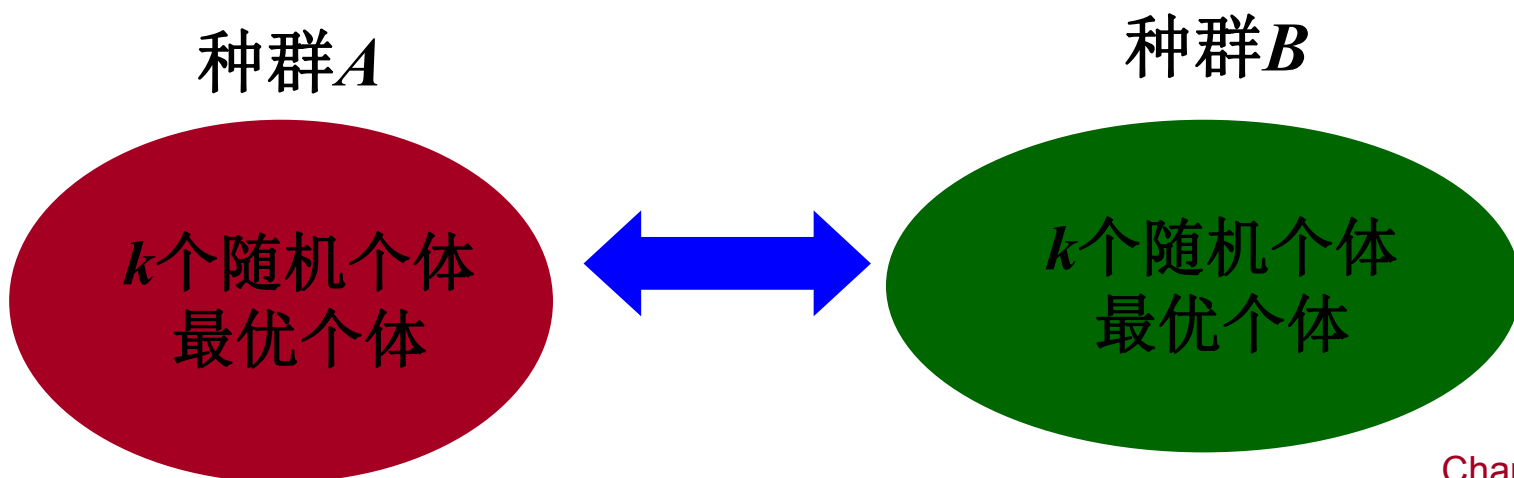
6.3.2 双种群遗传算法

1. 双种群遗传算法：建立两个遗传算法群体，分别独立地运行复制、交叉、变异操作，同时当每一代运行结束以后，选择两个种群中的随机个体及最优个体分别交换。

2. 双种群遗传算法的设计

- (1) 编码设计
- (2) 交叉算子、变异算子
- (3) 杂交算子（迁移操作）

打破原种群内的平衡态
达到更高的平衡态，有
利于算法跳出局部最优。



6.3 遗传算法的改进算法

6.3.1 双倍体遗传算法

6.3.2 双种群遗传算法

6.3.3 自适应遗传算法

■ Srinivas M., Patnaik L. M.等在1994年提出一种自适应遗传算法(adaptive genetic algorithms, AGA): 交叉概率 P_c 和变异概率 P_m 能随个体的适应度自动改变。

6.3.3 自适应遗传算法

1. 基本思想

■ AGA: 当种群各个体适应度趋于一致或者趋于局部最优时, 使 P_c 和 P_m 增加, 以跳出局部最优; 而当群体适应度比较分散时, 使 P_c 和 P_m 减少, 以利于优良个体的生存。

■ 对于适应度高于群体平均适应度 (f_{avg}) 的个体, 选择较低的 P_c 和 P_m , 使得该解得以保护进入下一代; 对低于平均适应度 (f_{avg}) 的个体, 选择较高的 P_c 和 P_m , 使该解被淘汰。

$$P_c = \begin{cases} \frac{k_1(f_{\max} - f')}{f_{\max} - f_{avg}}, & f' > f_{avg} \\ k_2, & f' \leq f_{avg} \end{cases}$$

交叉个体中较大的适应度

$$k_2 > k_1, \quad k_4 > k_3$$

$$P_m = \begin{cases} \frac{k_3(f_{\max} - f)}{f_{\max} - f_{avg}}, & f > f_{avg} \\ k_4, & f \leq f_{avg} \end{cases}$$

变异个体的适应度

6.3.3 自适应遗传算法

2. 自适应遗传算法的步骤

- (1) 编码设计。
- (2) 初始种群产生：产生 N (N 是偶数) 个候选解，组成初始种群。
- (3) 计算适应度 f_i 。
- (4) 按轮盘赌规则选择 N 个个体，计算 f_{avg} 和 f_{max} 。
- (5) 将群体中的各个个体随机搭配成对，共组成 $N/2$ 对。
对每一对个体，按照自适应公式计算自适应交叉概率 P_c ，随机产生 $R(0,1)$ ，如果 $R < P_c$ ，则对该对染色体进行交叉操作。

6.3.3 自适应遗传算法

2. 自适应遗传算法的步骤（续）

- (6) 对于群体中的所有个体，共 N 个，按照自适应变异公式计算自适应变异概率 P_m ，随机产生 $R(0,1)$ ，如果 $R < P_m$ ，则对该染色体进行变异操作。
- (7) 计算由交叉和变异生成新个体的适应度，新个体与父代一起构成新群体。
- (8) 判断是否达到预定的迭代次数，是则结束；否则转第（4）步。

6.3.3 自适应遗传算法

交叉中个体较大的适应度

3. 自适应的交叉概率与变异概率

变异个体的适应度

$$P_c = \begin{cases} \frac{k_1(f_{\max} - f')}{f_{\max} - f_{\text{avg}}}, & f' > f_{\text{avg}} \\ k_2, & f' \leq f_{\text{avg}} \end{cases}$$

$$P_m = \begin{cases} \frac{k_3(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f > f_{\text{avg}} \\ k_4, & f \leq f_{\text{avg}} \end{cases}$$

■ 普通自适应算法中，当个体适应度值越接近最大适应度值时，交叉概率与变异概率就越小；当等于最大适应度值时，交叉概率和变异概率为零。这种方法对于进化后期比较合适，但进化初期不利。

■ 改进的思想：当前代的最优个体的交叉概率与变异概率最小，较优个体要对应于较大的交叉概率与变异概率，较差个体对应于最大的交叉概率与变异概率。

6.3.3 自适应遗传算法

3. 自适应的交叉概率与变异概率（进一步改进）

■ F—自适应方法:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{\text{avg}})}{f_{\text{max}} - f_{\text{avg}}}, & f' > f_{\text{avg}} \\ P_{c1}, & f' \leq f_{\text{avg}} \end{cases}$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_{\text{max}} - f)}{f_{\text{max}} - f_{\text{avg}}}, & f > f_{\text{avg}} \\ P_{m1}, & f \leq f_{\text{avg}} \end{cases}$$

$$P_{c1} = 0.9, P_{c2} = 0.6, P_{m1} = 0.1, P_{m2} = 0.001$$

当 $f' = f_{\text{max}}$, $P_c = P_{c2} > 0$;
当 $f = f_{\text{max}}$, $P_m = P_{m2} > 0$ 。

- 最优个体的交叉概率与变异概率最小
- 较优个体要对应于较大的交叉概率与变异概率
- 较差个体对应于最大的交叉概率与变异概率

第6章 智能计算及其应用

□ 6.1 进化算法的产生与发展

□ 6.2 基本遗传算法

□ 6.3 遗传算法的改进算法

□ 6.4 遗传算法的应用

□ 6.5 群智能算法产生的背景

□ 6.6 粒子群优化算法及其

□ 6.7 蚁群算法及其应用

• 基于小生境技术的遗传算法：包括预选机制、排挤机制、共享机制，具有更好的多样性和全局搜索性能。

• 多目标遗传算法