

# Evolutionary Computation

## 大纲

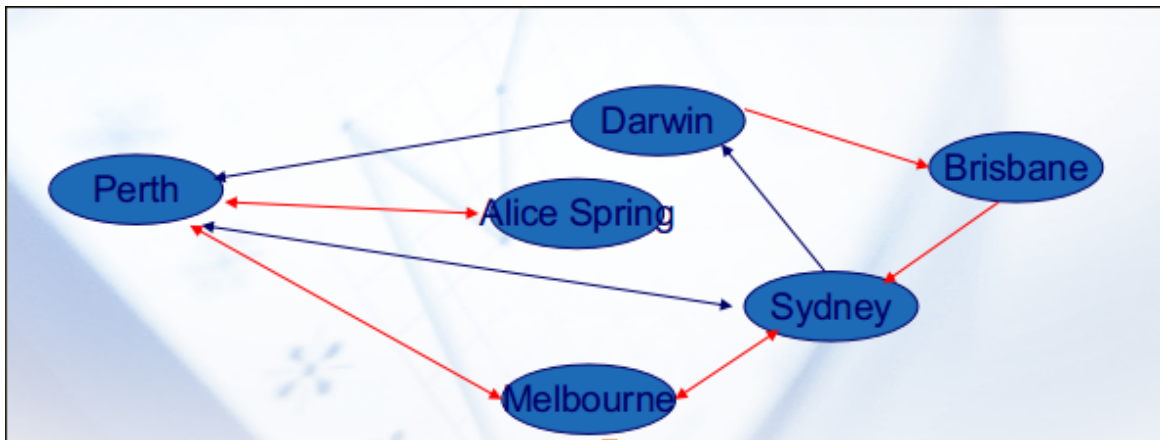
- 生物启发式搜索
- DNA计算
- 进化计算
- 进化算法
  - 遗传算法
  - 进化算法
  - 进化策略

## 生物启发式搜索

- 专注群体而不是个体
  - 个体可能并不会生存的很好
  - 但是作为一个群体，始终是在想好的方向发展和进步
- 专注适应度,适者生存
  - 好基因个体得到更好的繁殖
  - 差基因个体得到更少的资源
- 专注群体的多样性
- 专注变异
  - 基因重组
    - 现有解的重组
  - 基因突变
    - 低概率
    - 危险性大(效果不是很好，但是可以扩大解的搜索范围，发现更好的解)
- 进化式搜索

## DNA计算

- 基因可以理解成是基于4(A, C, G, T)的存储单元
- 基因本身不牵扯任何的计算，但是我们可以从编码的角度让DNA参与计算的过程
- DNA计算的优点
  - 海量信息存储
  - 巨大的并行性(天然并行性)
  - 能量利用率高
- DNA计算的经典实验(TSP)
  1. 问题



## 2. 编码城市

1	Sydney	- TTAAGG
2	Perth	- AAAGGG
3	Melbourne	- GATACT
4	Brisbane	- CGGTGC
5	Alice Spring	- CGTCCA
6	Darwin	- CCGATG

## 3. 编码边

- 编码所有可能的边
- 将所有的可能的边都加入实验计算
- 在试管中并行计算，最终筛选出合适的解

## 进化计算

- 进化计算和搜索算法的关系
  - 个体：候选解
  - 适应度：解的质量
  - 环境：问题
- 进化的机制
  1. 增加多样性
    - 变异
    - 重组
  2. 减少多样性
    - 选择子代
    - 选择父代
  3. 机制
    - 群体选择出父代
      - 选择
    - 父代产生子代
      - 变异
      - 重组
    - 子代替换群体
      - 选择
  4. 算法思路

```

BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END

```

- 编码解码
  - 针对具体的问题而言
  - 编码：表型 → 基因型
  - 解码：基因型 → 表型
- 适应度
  - 目标函数评估适应度
- 遗传操作
  - 重组：结合父母的信息
  - 变异：引入新的改变，保证考虑解的充分性
- 选择
  - 选择父代
    1. 根据适应度情况概率的选择父代
    2. 上述概率的方式有利于我们逃出局部最优解
  - 选择子代
    1. 适应度为标准：父代和子代共同排名选出
    2. 年龄为标准：父代删除，只从子代中排名选择
  - 平衡性考虑
    - Exploitation:
      1. 深度开发
      2. 容易陷入局部最优
      3. 但是对解的诠释更精确
    - Exploration:
      1. 广度开发
      2. 但是过于随机的广度开发，解不容易收敛
      3. 但是对解的看的更多，接的效果会更好(结果更全面)
    - 两者权衡
- 初始解 / 终止解
  - 初始解：随机初始化，挥着精心的设计初始解
  - 终止解：
    - 检查终止条件
      - 适应度到达需求
      - 迭代次数

- 多样性限制
  - 多次迭代解没有良好的改善
- 进化算法步骤
  1. 设计问题的编码方式
    - 找到表现型：问题的显著表示
    - 找到基因型：找到编码表现性的方式
  2. 设计适应度函数
  3. 设计遗传算子
    - 突变
    - 重组
  4. 选择父代和子代
  5. 决定初始解和终止解的思路
- 八皇后问题实例分析
  1. 编码方式
    - 棋盘 → 数组
  2. 适应度
 

设定非法情况的惩罚
  3. 遗传算子
  4. 选择替换的方式
  5. 初始解随机，终止解可以考虑最大的迭代次数
- 进化计算(EA / EC)的好处
  1. 广泛的适用于离散，连续，混合优化问题
  2. 不需要先验知识
  3. 噪声不敏感(群体的力量)
  4. 易于并行
- 进化计算(EA / EC)的缺点
  1. 局部最优
  2. 没有理论基础
  3. 参数调整

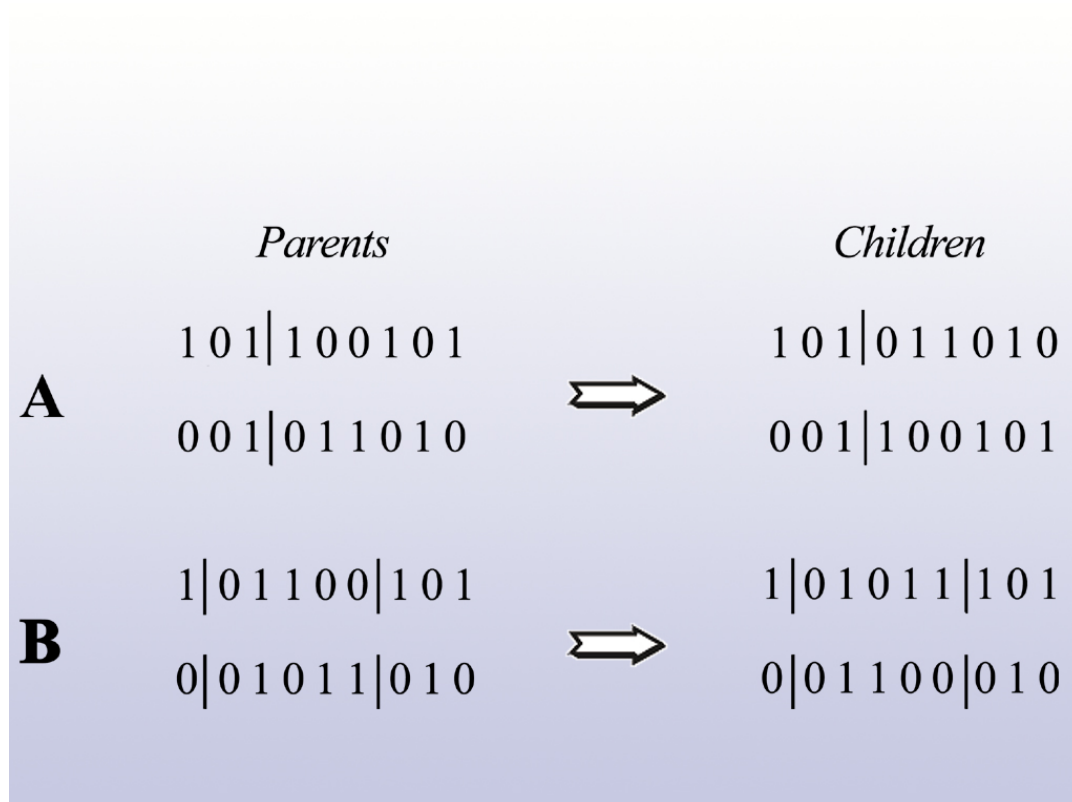
## 主流进化算法

- 分类
  - GA
  - EP
  - ES
  - GP
- GA
  - 编码方式
    - 固定长度的二进制串编码方式已经证明可收敛
    - 其他的编码方式难以证明
  - 选择机制不同
    - 贪心策略：不考虑概率和随机性
    - 概率策略：和适应度成比例的概率是合适的

$$Pr(h_i) = \frac{Fitness(h_i)}{\sum_j^n Fitness(h_j)}$$

- 遗传算子不同

- 重组的概率性：0.6 ~ 0.9
- 多级重组：



- 突变的概率性：相对低

- EP

- 算法

1. 传统EP：有限状态机的机器学习任务
2. 当代EP：（数值）优化
3. 非常开放的框架：任何表示和变异都可以。
4. 没有重组
5. 参数标准自适应（当代EP）
6. 持续的参数优化
7. 重点

- 解
- 参数
- 表征

$$\begin{cases} x'_i = x_i + \sqrt{\sigma_i} \times N_i(0,1) \\ \sigma'_i = \sigma_i + \sqrt{\eta \times \sigma_i} \times N_i(0,1) \end{cases}$$

- 决定参数的分布方式
  - 正态分布
  - 标准差等等参数

8. 在搜索空间中只有种群，没有个体的概念，不存在重组等遗传算子
9. 每个个体都会产生子代

## 10. 锦标赛的机制去竞争

- ES
  - 数值优化
  - 快速
  - 实值优化好
  - 比较多的理论
  - 参数标准的自适应