两道半不会：-25

宽搜分析，部分map crossover，锦标selection违反Elimla例子？

A*算法tree和graph分别对应的最优性条件。consistent证admissable

演化算法结构。两个binary的crossover方法举例。

锦标赛selection步骤+Elima

两个漂移分析步骤。加性证乘性

1+1EA，one bit变异，Onemax问题时间下界

NSGA-II的N+N selection步骤

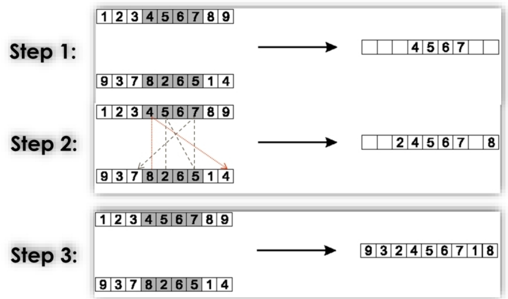
COCZ问题时间上界（作业）

Permutation representation: Partially mapped crossover

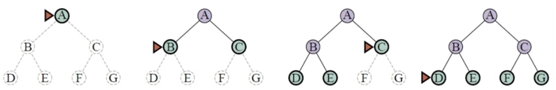
- Partially mapped crossover:
  1. Choose two crossover points randomly, and copy the segment between them from parent P1 into the first offspring
  2. Starting from the first crossover point, look for elements in that segment of P2 that have not been copied
  3. For each of these *i*, look in the offspring to see what element *j* has been copied in its place from P1
  4. Place *i* into the position occupied by *j* in P2, since we know that we will not be putting *j* there (as is already in offspring)
  5. If the place occupied by *j* in P2 has already been filled in the offspring by *k*, put *i* in the position occupied by *k* in P2
  6. Having dealt with the elements from the crossover segment, the rest of the first offspring can be filled from P2
  7. Create the second offspring analogously with parental roles reversed

http://www.lamda.nju.edu.cn/qianc/

Permutation representation: Partially mapped crossover



Breadth-first search - performance



- Complete

  if the depth d of the shallowest goal node is finite
- Optimal

  if all actions have the same cost
- Time complexity

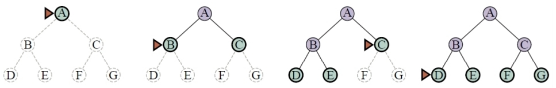
  $b + b^2 + b^3 + \cdots + b^d = O(b^d)$, *where b is the branching factor*

  If goal test is applied when a node is expanded

  $$b + b^2 + b^3 + \cdots + b^d + b^{d+1} = O(b^{d+1})$$

http://www.lamda.nju.edu.cn/qianc/

Breadth-first search - performance



- Time complexity

  $b + b^2 + b^3 + \cdots + b^d = O(b^d)$, *where b is the branching factor*

  If goal test is applied when a node is expanded

  $$b + b^2 + b^3 + \cdots + b^d + b^{d+1} = O(b^{d+1})$$
- Space complexity

  $$\left.\begin{array}{l} b + b^2 + b^3 + \cdots + b^{d-1} = O(b^{d-1}) \text{ in the explored set} \\ b^d \text{ in the frontier} \end{array}\right\} O(b^d)$$

  If using tree-search, $O(b^d)$

NPC(Non-deterministic Polynomial Complete) Problem: 满足两个条件:

- 是一个 NP 问题
- 所有的 NP 问题都可以约化到它

NP-hard Problem: 满足NPC问题的第 2 条，但**不一定要满足第 1 条。**（NP-Hard问题要比 NPC问题的范围广）

tree search frontier&graph search frontier

- tree：初次访问到时不进行判断，探索时才判断
- graph：记录已访问过的集合。不是探索时才判断

uninformed cost search：没有利用除“判断目标状态”外的其他信息（非启发式的）

Uniform-cost search：记录每步的累积cost

- 多个孩子，先展开累积cost小的
- 同一个节点多次出现，保留累积cost最小的那个
- 非搜到目标即停止，其他分支也要继续搜，为了最优

consistent<admissable

- admissable：估计函数h（n）始终小于实际花费
- consistent：任意两相邻状态的估计函数h之差，小于这两状态的动作实际cost

- consistent，则g+h不递减

greedy best-first：只看h（tree死循环）
A*：看g+h，当前最小的进行探索（tree最优性更容易）

> tree要admissable才有最优性
> graph要consistent才有最优性（更严格）

优化空间复杂度：记录次优路径，最优路径的下一步代价大于记录时，回退，回退时最优变次优

二进制01串编码的改进：缺点是相邻整数编码差距大，改成格雷码（不唯一），相邻的永远差一位

binary-crossover

- Cut-and-crossfill crossover：定一个crossover点后，左面不变，右面按照另一个父亲的右面顺序添加不重复的
- Uniform crossover（均匀）：每一位都有pm概率翻转，另一个offspring镜像

parent selection

FPS：把所有评分都减去最小的，fi占总fi比例
LRS：赋分制，最好的i是u-1，最差是0
TS：每次在k个中选最好的，共λ次
US：纯随机

Survive selection

Aged-based：每个个体存活相同轮数
replace worst：父代里最差的λ个被子代替代
（μ,λ）selection：子代里选最优的μ个
（μ+λ）selection：
round-robin tournament：每个都和q个随机抽出来的比，选μ个胜场最多的

种群多样性

fitness sharing

有很多邻居时，降低fitness
个体只有邻近才会在分母贡献

- 特点：最高峰上明显会保存更多

crowding

子代和相近的父代竞争，留一个

- 特点：在不同区域会有相对均匀划分

适应层分析法

步骤

- 将解空间划分为m+1个子空间
- 计算从Si到所有更高层USj的概率上下界
- 概率取倒数就是时间，算出算法期望运行时间的上下界

求上界必背公式：

$$\sum_{i=0}^{n-1} \pi_0(S_i) \sum_{j=i}^{n-1} \frac{1}{v_j} \leq \sum_{j=0}^{n-1} \frac{1}{v_j}$$

- 且1/e约等于（1-1/n）^i
- 无穷级数1+1/2+1/3+...=logn

加性漂移分析

第一步：设计距离函数

$V(x) = n - f(x)$，其中$f(x)$表示x中总共有多少位是1

cl：其实就是单步变稍优的概率

$$\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq c_l$$

Upper bound: $\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{V(x)}{c_l}$

$$\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{V(x)}{c_l} \leq \frac{n}{c_l}$$