

## 题型

### 一、搜索

1.1 无信息搜索（仅仅使用问题提供的状态信息）

1.2 启发式搜索（优先拓展代价最低节点）

1.3 博弈树搜索

### 二、遗传算法

### 三、约束满足问题(CSP)

### 四、规划基础

### 五、命题逻辑和一阶逻辑

### 区别和联系

## 题型

1. 名词解释 30' 5\*6?

2. 简述题 48'（算法论述？） 12\*4?

3. 论述题 22'（区别和联系？） 11\*2?

### 强化学习：贪心动作->非贪心动作的后面（必考）

**主要部分：卡尔曼滤波、神经网络（会有超纲部分）、k-means、强化学习，有公式计算（不多），A\*最佳优先整合**

算法横向、纵向区别和联系（确定和随机问题），对算法理解，算法总结，以图的方式画出算法优化？

有变动部分：

chapter6：神经网络

chapter8：强化学习 和原文档有出入

## 一、搜索

### 1.1 无信息搜索（仅仅使用问题提供的状态信息）

- BFS: queue实现、open-closed实现
  - UCS(一致代价)：引入代价函数，优先访问当前代价最低的节点
- DFS: 回溯实现、stack实现、open-closed实现
  - DLS(深度受限)：搜索到一定深度时回溯或停止搜索
  - IDDFS(迭代加深)：深度从0开始迭代加深
  - ILDFS(迭代加长DFS)：引入代价函数的DFS

### 1.2 启发式搜索（优先拓展代价最低节点）

代价函数： $f(n) = g(n) + h(n)$ ，其中  $g(n)$  为初始  $\rightarrow n$  节点实际代价、 $h(n)$  为  $n \rightarrow$  目标估计代价

- 最佳优先(BestFS)：在UCS基础上引入 $h(n)$
- 贪心最佳优先(GBFS)：只启用启发函数 $h(n)$  即 $g(n) = 0$
- A\*：
  - $h(n)$ 可接受性： $h(n) \leq h^*(n)$  即估计不超过实际
  - $h(n)$ 一致性： $h(n) \leq cost(n, n') + h(n')$

## 1.3 博弈树搜索

- Minimax算法：己方利益最大、对方利益最小
- $\alpha - \beta$ ：将原本只计算最大最小值改为计算范围，从而达到剪枝目的
- MCTS：从未访问节点选择UCB最佳的，随机下棋，得分回传
  - MC方法：类比pi 的计算
  - Naive方法：尝试n次，统计每个拉杆的平均回报，始终选择平均回报最大的拉杆
  - $\epsilon - \text{greedy}$ 算法：以  $1 - \epsilon$  的概率选择当前最佳拉杆，而以  $\epsilon$  的概率选择其它拉杆
  - UCB(置信度上界)：平均回报越大，选中次数越小，UCB1越大

	完备性	最优性	时间复杂度	空间复杂度
BFS	$\checkmark$	$\checkmark$	$O(b^d)$	$O(b^d)$
UCS	代价>0时完备	$\checkmark$	$O(b^{1+C^*/\epsilon})$	$O(b^{1+C^*/\epsilon})$
DFS	状态空间无限时不完备	$\times$	$O(b^m)$	$O(bm)$
DLS	$\times$	$\times$	$O(b^l)$	$O(bl)$
IDDFS	$\checkmark$	$\checkmark$	$O(b^d)$	$O(bd)$
A*	$\checkmark$	$\checkmark$		
Minimax			$O(b^m)$	$O(bm)$

## 二、遗传算法

运用生物遗传与进化概念，通过繁殖、变异、竞争等方法，实现优胜劣汰，以逐步得到问题的最优解或次优解

- 编码方式：二进、十进制
- 选择算子：轮盘赌策略(转为相对适应度值)
- 交叉方式：点式和均匀
- 理论基础：模式定理与积木块假设

模式定理与积木块假设：如果模式的定义矩较短、阶次较低、适应度大于群体的平均适应度，那么随着群体的进化，该模式在群体中出现的次数将按指数规律增长

## 三、约束满足问题(CSP)

针对给定的一组变量及需要满足的一些限制或约束条件，找出满足这些约束关系的一个解、全部解或最优解

- 一致的：一个不违反任何约束条件的赋值
- CSP问题的解：相容且完整的赋值
- 避免无效搜索的2种措施P10
- 回溯法存在的2个问题P12
  - BJ：多个层级之间回跳，以避免抖动现象的发生
  - CBJ：改进BJ，允许执行多个回跳
  - BM：阻止相同的约束检测反复发生

- 节点一致性:  $D_i \subseteq R_i$  即**节点可取值域满足约束**。不满足节点一致性意味着至少存在一个节点  $X_i$  其某个实例化值总会导致回溯失败
- 弧一致性: 弧  $(x_i, x_j)$  具有一致性当且仅当**对于  $D_i$  中的每个元素, 在  $D_j$  中存在相应的元素满足  $R_{ij}$**   
弧一致性具有方向性
  - AC-1、AC-2 (在AC-1的基础上将变量值域的修改通过弧传播出去)、AC-3 (在AC-2基础上值重新检测与之相关的弧)
- 路径一致性
- k-一致性

## 四、规划基础

将人工智能领域中的搜索与逻辑结合, 使得智能体除了具有应激行为或反应式行为之外, 还具有某种程度的深思熟虑的能力

- 应激行为/反应式行为: 智能体为了实现其自身目标, 在与环境的交互过程中, 会依据环境的状态生成一系列动作
- 经典的规划环境: 智能体在完全可观察的、确定的、离散的、静态的环境中的规划
- 特征表示/因式化表示: 用一组变量或特征表 示环境中的一个状态
- 基于状态空间的规划
  - 前向搜索 (从初始状态开始)、后向搜索 (从目标状态开始)

## 五、命题逻辑和一阶逻辑

允许智能体对知识进行表示, 并基于知识库执行某种推理, 以指导智能体自身的决策与行动

- 形式逻辑: 使用一些列规则和推理方法, 帮助人们正确思考问题和表达思想
- 逻辑智能体: 智能体能够将接收到的新信息与知识库中的原有知识进行综合, 执行推理, 并获得有效的决策, 以适应环境的新变化
- 知识库: 知识库是由描述环境及智能体目标的逻辑语句组成的, 即知识库包含了智能体在环境中进行决策与行动所需的一切知识。
- 逻辑: 基于形式化知识的有效推理
- 原子命题: 其真值不依赖于任何其他命题的命题
- $p \rightarrow q$ : 强调  $p$  成立时,  $q$  一定要成立这一关系 等价于  $\neg p \vee q$
- SAT问题: 为逻辑语句中所有的命题变量找到一组或全部的真值赋值使得所给定的逻辑语句成立, **SAT是一个特殊的CSP问题**

## 区别和联系

区别和联系	
DFS、BFS区别	1. 搜索方式不同 2. BFS时间复杂度更低、DFS空间复杂度更低
一致代价(UCS)和最佳优先BestFS区别	只有一行代码有区别 <code>priority = new_cost + h(n)</code>
$\alpha - \beta$ 、A*	相似剪枝思路: $f > f^*$ 时剪枝
A*、约束满足问题	A*中是针对具体的问题对启发函数的设计, CSP是使用一种通用策略对问题求解
CSP、基于状态空间搜索	1. 基于状态空间的搜索将状态看作没有内部结构的黑盒子, 而CSP将状态进一步细分, 利用一组变量和值来表示状态 2.CSP可以使用通用策略, 而基于状态空间的只能基于特定问题的启发式 3.常规的只能执行搜索, 而CSP还可以执行推断, 即约束传播
规划、搜索	规划结合了搜索与逻辑两个方面有明确的目标, 在搜索的基础上还包括逻辑推理、概率等各种组合问题。而搜索只是找到满足条件的解
CSP、命题逻辑和一阶逻辑、规划	都使用变量及其赋值来表示对象的状态
命题逻辑、一阶逻辑	一阶逻辑的表达粒度要比命题逻辑细些, 能够表达语句中的精细结构与内容。引入了谓词变量以及全称、存在量词

回溯:

- DFS
- Minimax、 $\alpha - \beta$  博弈树
- CSP
- SAT (约束满足问题)

利用与探索:

- 博弈树搜索中MCTS节点的选取
- SAT中的Walksat算法 (类似 $\epsilon - greedy$ )

绝对转相对:

- GA轮盘赌策略

结合方式: 在开始/过程中结合

- 约束满足问题中**回溯法避免无效搜索的方式**
- 约束满足问题中**回溯法与约束传播的结合**

一致:

- A\*的一致性:  $h(n)$

- CSP中的一致性等
- 命题逻辑中命题集合一致性：集合中所有命题可能同时为真