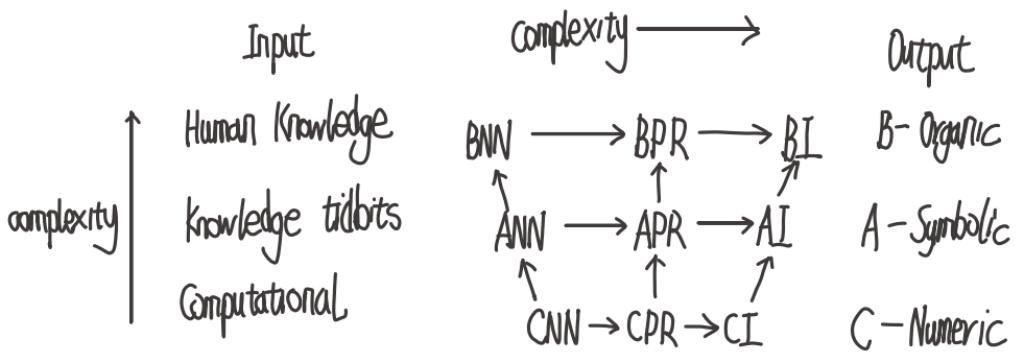


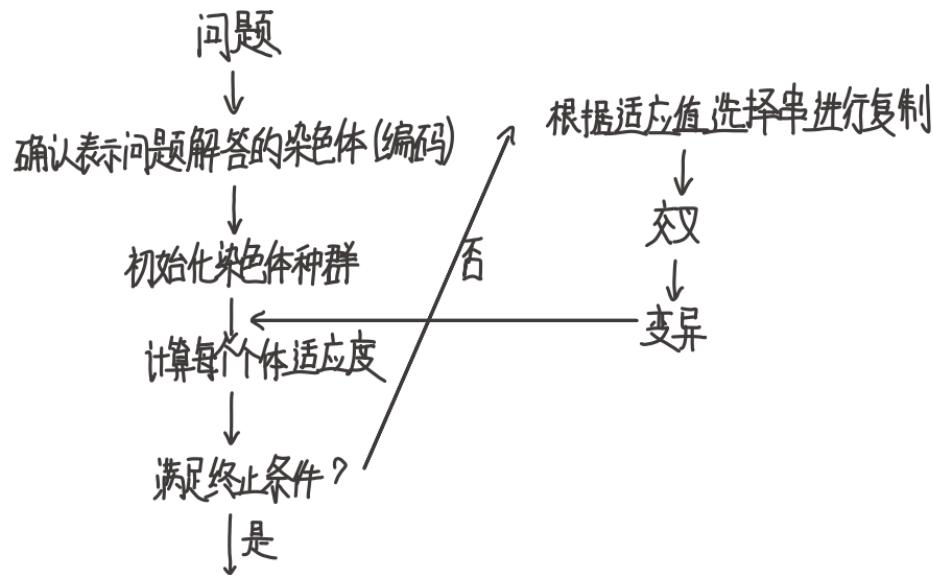
ABC框架和NN、PR、AI之间关系



本图由贝兹德克于1994年提出，中间具有9个节点，表示9个研究领域
ABC三者对应三个系统不同的复杂度级别，复杂度自左向右自下至上
不断提高，节点间的距离衡量领域间的差距，CNN和CPR的差距远小
于BNN和BPR之间的差距，CI和AI的差异远小于AI和BI的差距，“→”意味着适当的
子集，如ANN ⊂ APR ⊂ AI, CI ⊂ AI ⊂ BI

从机器人应用为案例：见Notion总结

遗传算法流程



输出最优解

PPT里是“=”，考试最好写“=”

编码计算核心公式 $\delta \geq \frac{U_{max} - U_{min}}{2^l - 1}$

$$l = 18 \rightarrow \begin{array}{cccccc} 65536 & 65528 & 6552 & 70352 \\ \uparrow & 4096 & 512 & 4800 \\ 010001001011010000 & \frac{70352}{362143} \times 151-3 \end{array}$$

$$x = U_{min} + \sum_{i=1}^l b_i \cdot 2^{l-i} \quad \frac{U_{max} - U_{min}}{2^l - 1}$$

$$= -30 +$$

$$\begin{array}{r} 65536 \rightarrow 16 \\ 131072 \\ 262144 \end{array}$$

- 遗传算法特点
1. 对求解问题没有太多数学要求
 2. 能有效地进行概率意义的全局搜索
 3. 算法灵活性高，对于特殊问题也有效

适应度函数尺度变换

(1) 函数值必须非负

选择算子作用：从当前群体选出优良个体

交叉算子作用：模拟物种繁衍，促进个体多样性

变异算子作用：产生有“个性”的新个体

遗传算法步骤

1. 令进化代数 $g=0$ ，定义初始化种群 $P(g)$
2. 对 $P(g)$ 中每一个体计算适应度和选择得到新种群
3. 按一定方法依适应度选择两个个体进入交叉变异操作，重复多次得到新一代群体 $P(g+1)$
4. 满足终止条件算法结束，否则回到步骤 2

遗传算法 5 基本要素

1. 个体的参数编码形式
2. 初始群体的设定
3. 适应度函数设计
4. 控制参数的设定（群体规模 · · ·）
5. 遗传操作设定

粒子群算法

基本思想、原理

一群粒子 速度

迭代更新寻找最优 个体极值 全局极值

公式：

i 表示粒子编号， k 表示迭代次数， g 表示 global 連續

$x^i(k)$ $p^i(k)$ $v^i(k)$ $p_j^g(k) \leftarrow$ 向量
 ↓
 当前位置 个体最优位置 个体方向 群体最

优位置

均匀分布随机数

$$v_j^i(k+1) = w(k) v_j^i(k) + \varphi_1 \text{rand}(0, 1) (p_j^i(k) - x_j^i(k)) + \varphi_2 \text{rand}(0, 1) (p_j^g(k) - x_j^i(k))$$

惯性因子 加速度常数

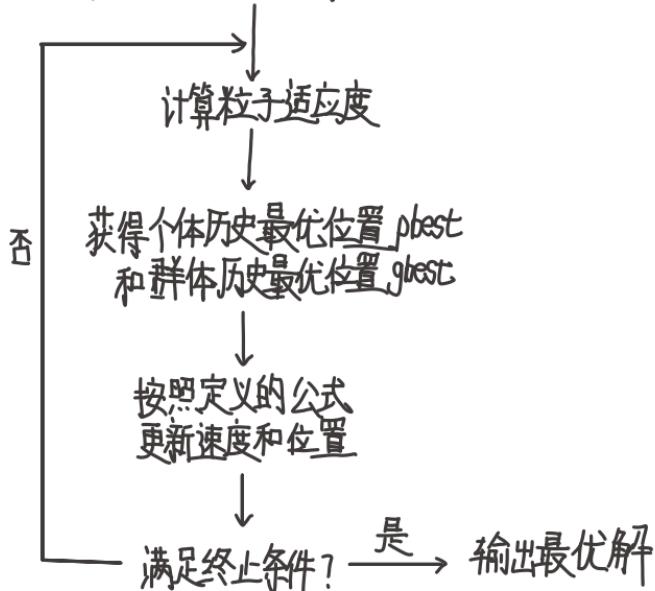
$$x_j^i(k+1) = x_j^i(k) + v_j^i(k+1)$$

向量中的分量值

跟踪全局最优 gbest

流程图

初始化粒子群，设置算法参数



速度更新模型 各部分作用

- 1 只有第一部分
- 2. 没有第一部分
- 3 没有第二部分
- 4 没有第三部分

基本量子粒子群优化算法

1. 粒子不用位置和速度而是波函数来表示

$$\text{波函数 } \Psi(x) = \frac{1}{\sqrt{L}} \exp\left(-\frac{\|p-x\|}{L}\right) \quad \begin{matrix} \text{为粒子个体} \\ \text{最好位置} \end{matrix}$$

$$\text{概率密度函数 } Q(x) = \Psi^2(x)$$

2 引入 m_{best} 作为所有微粒的中心

$$m_{best} = \sum_{i=1}^M \frac{p_i}{M} \quad M \text{ 为种群数, } p_i \text{ 为粒子个体最好位置}$$

基本步骤

1 确定种群规模和粒子维数，初始化粒子群体

2. 计算粒子适应度

3 计算个体历史最优值 p_{best} 和全局最优值 g_{best} (比较、替换)

4 计算所有粒子重心 m_{best}

5 更新所有粒子位置，产生新种群

6 判断算法是否满足终止条件

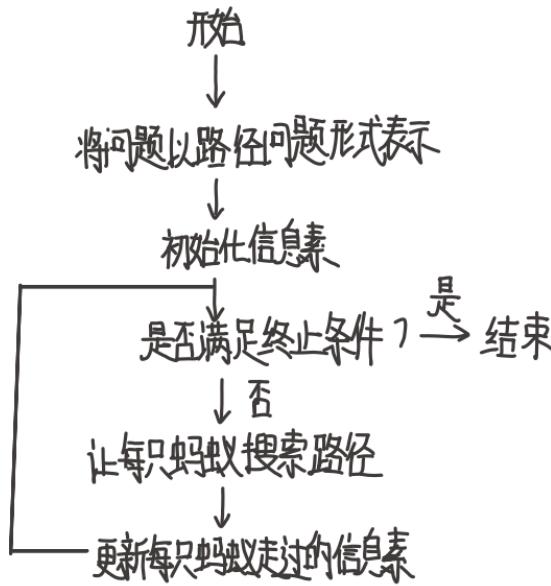
优点：具有更好收敛性和全局搜索能力

缺点：求解约束问题会产生大量不可行解，会破坏种群多样性

蚁群算法：离散

信息素思想：蚂蚁行走留下信息素，其它蚂蚁按一定概率沿信息素较强的路径觅食

流程图



基本蚁群公式

$$I_{xy}(t) \rightarrow t \text{ 时刻留在 } xy \text{ 连线上信息素} \quad I_{xy}(0) = C$$

$$\eta_{xy}(t) \rightarrow \text{启发信息函数}, \text{TSP 中为距离的倒数} \frac{1}{d_{xy}}$$

蚂蚁在运动过程中，根据路径上信息素决定转移方向

$$1 \text{ 路径信息素消散规则: } I_{xy}(t) = \rho I_{xy}(t) + \Delta I_{xy}(t)$$

$$\text{蚁群信息素浓度更新规则: } \Delta I_{xy}(t) = \sum_{k=1}^m \Delta I_{xy}^k(t)$$

$$2 \text{ } t \text{ 时刻蚂蚁 } k \text{ 从城市 } x \text{ 转到城市 } y \text{ 概率 } p_{xy}^k(t) = \frac{|I_{xy}(t)|^\alpha |\eta_{xy}(t)|^\beta}{\sum |I_{xy}(t)|^\alpha |\eta_{xy}(t)|^\beta}$$

α 是信息素启发式因子，反映了随机性因素作用， α 过大陷入局部最优
过早

β 是期望值启发式因子，反映了先验性、确定性因素作用， β 过大陷入局部最优，加快收敛性

3 计算 $\Delta T_{xy}(t)$ 的三种方法

(1) 蚂蚁圈系统: $\Delta T_{xy}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{第 } k \text{ 只蚂蚁在本次循环中从 } x \text{ 到 } y \\ 0 & \text{该蚂蚁本次环所走路径西侧} \end{cases}$

\downarrow 一个循环更新一次

(2) 蚂蚁数量系统: $\Delta T_{xy}^k(t) = \begin{cases} \frac{Q}{d_k} & \text{一根路径长度} \\ 0 & \text{ } \end{cases}$

(3) 蚂蚁密度系统: $\Delta T_{xy}^k(t) = \begin{cases} Q & \text{ } \\ 0 & \text{ } \end{cases}$

1-P 信息素挥发度 过大降低算法随机性和全局搜索能力
过小降低收敛速度

蚁群算法 A 和 遗传算法 G 的区别联系

联系 1 都是不确定性算法，求解过程有随机性要素

2 都使用了“种群”概念，多个体搜索，用统计规律战胜个体局限性

3 使用了“适应度”对个体进行评价

4 去中心化，算法鲁棒性好

区别 1 GA 处理完整的解，ACO 是一步步构建解

更新环境

2 GA 中信息共享是隐式的（优势个体存活），ACO 中信息共享是显式的（信息素）

进化·种群中遗传性状在世代之间的变化