智能优化算法课程历年考试题目整理

一、 优化算法

1 什么是 P 问题，什么是 NP 问题？

问题类型	描述解释
P	能在**多项式时间内**解决的问题
NP	**不能在**或**不确定能不能在多项式时间内解决**，但能在**多项式时间验证**的问题
NP-C	NP 完全问题，所有 NP 问题在**多项式时间内**都能**约化**(Reducibility)到它的 NP 问题，即解决了此 NPC 问题，所有 NP 问题也都得到解决
NP-Hard	NP 难问题，所有 NP 问题在**多项式时间内**都能**约化**(Reducibility)到它的问题 (不一定是 NP 问题)

2 智能优化算法主要是针对什么问题而提出的？

智能优化算法主要是针对**组合优化问题**而提出的。

当最优化问题中的可行域D是一个由有限个元素组成的集合时，该最优化问题称为组合优化问题。

通常组合优化问题可表示为：$\begin{cases} min\, f(x) \\ s.t. \quad g(x) \geq 0 \\ \quad\quad x \in \{x_1, x_2, \dots, x_n\} = D \end{cases}$

3 描述组合优化问题中的一个典型例子，并建立其数学模型。

3.1 描述约束机器排序问题，建立其数学模型，并指出一种可行解的编码方式。(X 3)

3.2 描述背包问题、建立其数学模型，并指出一种可行解的编码方式。(X 2)

(1) **旅行商问题** (Traveling Salesman Problem，TSP)

设有n个城市$1,2,\dots,n$，城市i与城市j间的距离为d_{ij}，一旅行商要去这些城市推销货物，他希望从一城市出发后走遍所有的城市且旅途中每个城市只经过一次，最后回到起点。选择一条路径是的收货商所走路线总长度最短。

引进决策变量x_{ij}，若商人从城市i出来后紧接着到城市j，则$x_{ij}=1$,否则$x_{ij}=0$ $(i,j=1,2,\dots,n)$。那么 TSP 的数学模型可表示为：

$$\begin{cases} min \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \\ s.t. \quad \begin{aligned} &\sum_{j=1}^{n} x_{ij} = 1, i = 1,2,\dots,n \\ &\sum_{i=1}^{n} x_{ij} = 1, j = 1,2,\dots,n \\ &\sum_{i,j\in S} x_{ij} \leq |S| - 1,\ S\text{为}\{1,2,\dots,n\}\text{的非空真子集}, \\ &x_{ij} \in \{0,1\}, i,j = 1,2,\dots,n \quad i \neq j \end{aligned} \end{cases}$$

(其中$|S|$表示集合 S 中元素的个数。)

可行解 编码方法：实数解编码

(2) **背包问题** (knapsack problem)

设有一个容积为b的背包，n 个尺寸分别为w_i，价值分别为$c_i(i=1,2\dots,n)$的物品，如何装包以使装入的物品产总价值最大？

引入决策变量x_i，若第 i 个物品被放入包中，则 $x_i=1$，否则 $x_i=0(i=1,2\dots,n)$

背包问题的数学模型为：$\begin{cases} max \sum_{i=1}^{n} c_i x_i \\ s.t. \quad \begin{aligned} &\sum_{i=1}^{n} w_i x_i \leq b \\ &x_i \epsilon \{0,1\}, i = 1,2,\dots,n \end{aligned} \end{cases}$

可行解 编码方法：二进制序列

（3）**并行机排序问题**

设有m台同型机器$M_1, M_2, \dots, M_m$，n 个相互独立的工件$J_1, J_2, \dots, J_n$。现在要安排这些工件到机器上进行加工，设每个工件只需在任一台机器上加工，工件J_i的加工时间为$t_i(i = 1,2, \dots, n)$。如何安排这些工件的加工方案，以使机器完成所有工作的时间最少。

引入决策变量x_{ij}，若工件J_i在机器M_j上加工，则$x_{ij} = 1$，否则$x_{ij} = 0$。

并行机排序问题的数学模型为:$\begin{cases} min\ T \\ s.t. \end{cases} \begin{matrix} \sum_{j=1}^{m} x_{ij} = 1, i = 1,2, \dots, n \\ T \geq \sum_{i=1}^{n} t_i x_{ij}, j = 1,2, \dots, n \\ x_{ij} \in \{0,1\}, i = 1,2, \dots, n \ \ j = 1,2, \dots, m \end{matrix}$

可行解 编码方法：矩阵编码

（4）**约束机问题**

n个加工量为$\{d_i | i = 1,2, \dots, n\}$的产品在一台机器上加工，机器在第$t$个时段的工作能力为$c_t$，求出所有产品得以加工的最少时段数。

数学模型：$\begin{cases} min\ T \\ s.t. \end{cases} \begin{matrix} \sum_{t=1}^{T} x_{it} = 1\ , i = 1,2, \dots, n \\ \sum_{i=1}^{n} d_i x_{it} \leq c_t\ , t = 1,2, \dots, T \\ x_{it} \in \{0,1\}, i = 1,2, \dots, n \ \ t = 1,2, \dots, T \end{matrix}$，其中$x_{it}$，$T$为决策变量，$x_{it}$表示$t$时段加工产品$i$。

对于约束机排序问题，用 $0-1$ 编码是不太有效的。一个有效的编码是给产品$(1,2, \dots, n)$一个加工序$(i_1, i_2, \dots, i_n)$，由于工序按c_t能力约束，因此以c_t最小的剩余能力依次安排加工时段，在加工的工程中不允许改变产品的加工序。

4　对 TSP 问题，可行解邻域构造方法？

对 TSP 问题，解 S 可表示为城市的一个排列，解的邻域可用不同的操作算子定义：

1. 互换操作：即随机交换解码中两不同的字符位置
2. 逆序操作：即将解码中两不同的随机位置间的字符串逆序
3. 插入操作：即随机选择某个点插入到串中的不同随机位置

5　实例、实例的规模、数的规模定义？

实例：问题中参数赋与了具体值的例子称为问题的实例。

实例的规模：实例中所有参数数值的规模之和。

数的规模：数在计算机中存储中所占据的位数。

任何一个正整数的输入规模是$l(x) = \lceil log_2(|x| + 1) \rceil + 2$ (2 表示包含符号位与分隔符，⌈ ⌉表示取最小整数)

6　数学规划包括：

线性规划、非线性规划、多目标规划、目标规划、动态规划、多层规划。

线性规划（LP）是指目标函数是线性函数，约束条件由线性函数确定的优化问题，

标准的LP可表示为：$\begin{cases} min\ C \cdot x \\ s.t. \end{cases} \begin{matrix} Ax = B, \text{ 其中 } C = (c_1, \dots c_n), x = (x_1, \dots x_n)^T, A = (a_{ij})_{m \times n}, B = (b_1, \dots b_m)^T \\ x \geq 0 \end{matrix}$

非线性规划（NLP）中目标函数是非线性函数或约束条件由非线性函数确定，

NLP的一般形式为：$\begin{cases} max\ f(x) \\ s.t. \end{cases} \begin{matrix} g_j(x) \leq 0\ , j = 1,2, \dots, p \end{matrix}$

- 线性目标规划可用**单纯形法求解**
- 非线性目标规划的求解方法大致有以下几种：

  a.基于单纯形的方法　b.直接搜索法　c.基于梯度的方法　d.交互法　e.遗传算法

7　启发式算法定义，启发式算法分类

定义：启发式算法是一种技术，这种技术使得在可接受的计算费用内去寻找最好的解，但不一定能保证所得解的可行性和最优性，甚至在多数情况下，无法阐述所得解同最最优解的近似程度。

分类：

1. 简单直观的算法（1）一步算法：如背包问题的贪婪算法（2）改进算法：如邻域搜索算法
2. 数学规划算法. 如分支定界法，割平面法
3. 智能算法

二、 模拟退火

1 模拟退火算法步骤

Step 1. 初始化可行解和温度 (任选一个初始解i_0，$i=i_0$；$k=0$；$t_0=t_{max}$[初始温度])

Step 2. 根据 Boltzmann 概率退火

（1）从邻域 $N(i)$ 中随机选取 j，计算 $\Delta f_{ij}=f(s_j)-f(s_i)$；

（2）若 $\Delta f_{ij}\le 0$，则 $i=j$，否则若 $e^{-\frac{\Delta f_{ij}}{t_k}}>random(0,1)$ 则 $i=j$；

Step 3. 重复 Step2 直到稳定状态

Step 4. 降温

（1）计算$t_{k+1}=d(t_k)$，$k=k+1$；

Step 5. 重复第二步至第四步知道满足终止条件或直到给定的步数

Step 6. 输出最好的解作为最优解

2 模拟退火算法中，描述内循环、外循环的步骤和停止规则。

内循环：

步骤：(退火)

（1）若在该温度达到内循环停止条件，则结束

（2）从邻域$N(i)$中随机选取j，计算$\Delta f_{ij}=f(s_j)-f(s_i)$；

（3）若$\Delta f_{ij}\le 0$，则$i=j$，否则若$e^{-\frac{\Delta f_{ij}}{t_k}}>random(0,1)$则$i=j$；

（4）回到步骤（1）

终止准则：

（1）循环总数控制法：总的温度下降次数为一定值K，当温度迭代次数达到K时，停止运算，整个算法的总迭代步数，它表示各温度时马氏链迭代步数的总和为一给定的数。

（2）邻域法：连续若干步目标值变化较小

（3）由接受和拒绝的比率控制迭代步数。

外循环：

步骤：(降温)

（1）计算$t_{k+1}=d(t_k)$，$k=k+1$；

（2）若达到外循环停止条件，则停止；否则运行内循环。

终止准则：

（1）零度法：模拟退火的最终温度为零，因而最为简单的原则是给定一个比较小的正数 t，当温度$T_k<t$时，算法停止。

（2）循环总数控制法：总的温度下降次数为一定值K，当温度迭代次数达到K时，停止运算，温度下降次数是一个定值。

（3）基于不改进规则控制法：在给定的迭代步数内没有改进当前局部最优解，则停止运算。

（4）接受概率控制法：给定一个指标$X>0$是一个比较小的数，在给定温度，除局部最优解，其他状态的接受概率都小于X时，停止运算。

简单总结：

内循环终止准则：

1.固定步数

2.连接若干步的目标值变化较小

3.由接受和拒绝的比率控制迭代步数

外循环终止准则：

1.设置终止温度的阈值(比较小的正数)

2.设置循环总数

3.连续若干步搜索到的最优解不再改进

4.设置接受概率

3　描述模拟退火算法中的接收准则(Metropolis 接收准则)。(X 2)

退火接收准则：在一给定温度下，由一个状态变到另一个状态，每一个状态到达的次数服从一个概率分布，即基于Metropolis接受准则的过程，该过程达到平衡时停止。在状态s_i时，产生的状态s_j被接受的概率为：

$$A_{ij}(t)=\begin{cases}1 & f(s_i)\geq f(s_j)\ \ or\ \ \Delta f_{ij}\leq 0\\ e^{(-\frac{\Delta f_{ij}}{t})} & f(s_i)<f(s_j)\ \ or\ \ \Delta f_{ij}>0\end{cases}，其中\ \Delta f_{ij}=f(s_j)-f(s_i)$$

4　证明：在温度趋于 0 时，分子停留在最低能量状态概率为 1。

在温度T下，金属物体分子停留在状态r满足Boltzmann概率分布 $P\{\bar{E}=E(r)\}=\frac{1}{Z(T)}\times e^{-\frac{E(r)}{k_BT}}$

其中，$E(r)$表示状态r的能量，$k_B>0$为Boltzmann常数，$\bar{E}$表示分子能量的一个随机变量。$Z(T)$为概率分布的标准化因子$Z(T)=\sum_{s\in D}e^{-\frac{E(s)}{k_BT}}$。温度$T$很高时，概率$P\{\bar{E}=E(r)\}$接近常数$\frac{1}{|D|}$，其中$|D|$是状态空间$D$中状态的个数。

假设$\mathrm{E}_1<\mathrm{E}_2$，因为$P\{\bar{E}=E_1\}-P\{\bar{E}=E_2\}=\frac{1}{Z(T)}e^{-\frac{E_1}{k_BT}}(1-e^{-\frac{E_2-E_1}{k_BT}})$ 可得$P\{\bar{E}=E_1\}-P\{\bar{E}=E_2\}>0,\forall T>0$

因为$\frac{\partial P\{\bar{E}=E(r_{min})\}}{\partial T}<0$，所以$P\{\bar{E}=E(r_{min})\}$是$T$的**衰减函数**，当$\mathrm{T}\to 0$，$P\{\bar{E}=E(r_{min})\}\to\frac{1}{|D_0|}$，其中$\mathrm{D}_0$是具有最低能量的状态集合。故分子停留在最低能量状态的概率趋于 1。

5　在温度T，状态r的能量为$E(r)$，用E表示分子能量的一个随机变量，写出其 Boltzmann 概率分布。并证明分子停留在最小的状态的概率比停留在能量大的状态的概率要大。

Boltzmann概率分布为 $P\{\bar{E}=E(r)\}=\frac{1}{Z(T)}\times e^{-\frac{E(r)}{k_BT}}$，标准化因子$Z(T)=\sum_{s\in D}e^{-\frac{E(s)}{k_BT}}$

假设$E_1<E_2$，因为$P\{\bar{E}=E_1\}-P\{\bar{E}=E_2\}=\frac{1}{Z(T)}e^{-\frac{E_1}{k_BT}}(1-e^{-\frac{E_2-E_1}{k_BT}})$ 可得$P\{\bar{E}=E_1\}-P\{\bar{E}=E_2\}>0,\forall T>0$

因此分子停留在最小的状态的概率比停留在能量大的状态的概率要大。

6　模拟退火算法中，初始温度如何选取？(X 2)

Step 1.　随机产生一个可行解s_0；

Step 2.　初始化t_0；理论上初始温度应保证平稳分布中每一状态的概率相等，即$e^{(-\frac{\Delta f_{ij}}{t_0})}\approx 1$；

其中$\Delta f_{ij}=f(s_j)-f(s_i)$，$t_0=K\Delta_0$，$K$为充分大的数，$\Delta_0=max\{f(s_i)|s_i\in D\}-min\{f(s_i)|s_i\in D\}$

Step 3.　给定一个常数T；温度t_0；常数χ_0(例如 0.9)；$R_0=0$；$k=1$；ε 比率常数。

Step 4.　在这个温度退火L步，记接受状态的个数为L'，计算$R_k=\frac{L'}{L}$

Step 5.　如果 $|R_k-\chi_0|<\varepsilon$ 则停止，输出初始温度 t_0；

否则：

如果 $R_{k-1},R_k<\chi_0$，　则 $t_0=t_0+T$；

如果 $R_{k-1},R_k\geq\chi_0$，　则 $t_0=t_0-T$；

如果 $R_{k-1}\geq\chi_0,R_k\leq\chi_0$，　则 $t_0=t_0+\frac{T}{2}$；

如果 $R_{k-1}\leq\chi_0,R_k\geq\chi_0$，　则 $t_0=t_0-\frac{T}{2}$；

之后 k=k+1，返回 Step 4

三、 遗传算法

1 写出遗传算法中变异算子的三种确定方法。

变异运算：

1. 单点、多点变异法(取反)；
2. 2-opt 法(对调)；
3. (实数码) 设V为变异的染色体，取$M>0$ 适当大，随机选个变异方向d，如$V+M\times d$不可行，则置M为$0\sim M$之间的随机数，直到可行为止；若该过程在规定的代数还不能找到可行解，则置$M=0$。
   由V变异产生的后代为：$X=V+M\times d$。

2 用遗传算法解决实数编码求连续函数优化问题，写出一种变异的运算方法。(X 3)

实数编码变异运算方法：

1. 转二进制编码法
   连续的实数变量在一定精度下也可以采用二进制编码。对给定区间$[a,b]$，设二进制编码的长度为n，
   则变量$x=a+a_1\frac{b-a}{2}+a_2\frac{b-a}{2^2}+a_3\frac{b-a}{2^3}+\cdots+a_n\frac{b-a}{2^n}$，与二进制码$a_1a_2\ldots a_n$相对应。二进制码与实际变量的误差为$\frac{b-a}{2^n}$。再用单点变异法或多点变异法即可完成实数码的变异方法。（随机选一个或几个变异位取反）
2. 互换操作：即随机交换解码中两不同的字符位置
3. 逆序操作：即将解码中两不同的随机位置间的字符串逆序
4. 插入操作：即随机选择某个点插入到串中的不同随机位置

3 用遗传算法解决实数编码求连续函数优化问题，写出一种交叉的运算方法。
(如果用遗传算法解 TSP，其中交叉算法如何实现？)

部分映射交叉(PMX)法：

Step 1.随机产生两个交叉点； Step 2.交换父代两个交叉点之间的基因片断；

Step 3.将互换的基因段以外的部分中与互换后基因段中元素冲突的用另一父代的相应位置代替，直到没有冲突。

举例：父代 $\begin{matrix}264|7358|91\\452|1876|93\end{matrix}$ → 交换基因并解冲突 $\begin{matrix}264|1876|91\\452|7358|93\end{matrix}$ → 子代 $\begin{matrix}234|1876|95\\412|7358|96\end{matrix}$

4 写出遗传算法中的两种交叉运算方法，并分别举例说明。

二进制编码：

1. 常用方法-双亲双子：对换随机交叉位后的基因
2. 变化交叉位法：从不同基因位后开始选择交叉位，并对换交叉位后的基因
3. 多交叉位法：随机选择多个交叉位，交替进行基因互换
4. 双亲单子法：选择父代交叉位前的基因与另一个父代交叉位后的基因组成子代
5. 显性遗传法：将两个父代进行按位或操作

实数编码：

随机产生一个数$c\in(0,1)$，由父代V_1',V_2'产生子代：

双亲双子：$\begin{cases}X=c\cdot V_1'+(1-c)\cdot V_2'\\Y=(1-c)\cdot V_1'+c\cdot V_2'\end{cases}$；双亲单子：$X=c\cdot V_1'+(1-c)\cdot V_2'$。

5 详述遗传算法中，轮盘赌选择种群的方法。

Step 1. 计算所有染色体V_i的累积概率q_i，$q_0=0,q_i=\sum_{j=1}^{i}Eval(V_j)$，$i=1,2,\ldots,pop\_size$；

Step 2. 在$(0,q_{pop_size}]$中产生一个随机数r；

Step 3. 若$q_{i-1}<r\le q_i$，则选择染色体 V_i ；

Step 4. 重复 Step 2 至 Step 3 pop_size 次以获得 pop_size 个染色体 (种群的规模可以与群体的规模不相同)

6 详述遗传算法中，评价函数的计算方法。

评价函数$Eval(V)$是根据每个染色体V的适应函数$fitness(V)$而得到与其他染色体的比例关系，可用来决定该染色体被选为种群的概率。公式为： $Eval(V)=\frac{fitness(V)}{\sum_V fitness(V)}$

7　写出遗传算法中，适应函数的两种确定方法。

1. 简单适应函数：设$f(x)$为目标函数，则 $\begin{cases} fitness(x)=f(x) & max\text{优化问题} \\ fitness(x)=M-f(x) & min\text{优化问题},M>\max\limits_{x} f(x) \end{cases}$；

2. 非线性适应函数：$fitness(x)=\begin{cases} \frac{1}{f_{max}-f(x)} & ,x<f_{max} \\ M & f_{max}\le x \end{cases}$，其中$M>0$是一个充分大的数，$f_{max}$是当前最优目标值；

3. 线性加速适应函数：$fitness(x)=\alpha\cdot f(x)+\beta$，$\alpha,\beta$满足$\begin{cases} \alpha\frac{\sum_{i=1}^{pop_size}f(x_i)}{pop_size}+\beta=\frac{\sum_{i=1}^{pop_size}f(x_i)}{pop_size} \\ \alpha\cdot\max\limits_{1\le i\le pop_size}\{f(x_i)\}+\beta=M\cdot\frac{\sum_{i=1}^{pop_size}f(x_i)}{pop_size} \end{cases}$

4. 排列适应函数：设$a\in(0,1),b>0$ 取 $fitness(V_i)=b(1-a)^{i-1},i=1,2,\dots,pop\_size,$
其中i代表最好个体，$i=pop\_size$ 代表最坏的个体。

8　在模板理论中，记t时刻的群体为$POP(t)$所有具有模板H的$POP(t)$中染色体集合为$T(H,POP(t))$。若一个个体Y被选入种群$NewPOP(t+1)$的概率为$P_Y=\frac{fitness(Y)}{\sum_{X\in POP(t)}fitness(X)}$，若种群的规模同群体的规模，求模板$H$所包含的染色体在$t+1$适合的期望值。

模板H所包含的染色体在$t+1$时刻的期望数为：$E_1(H,t+1)=f(H,t)\cdot N(H,t)$；

其中 $N(H,t)$ 为 t 时刻 $T(H,POP(t))$ 所包含的染色体数；$f(H,t)=\frac{\sum_{Y\in T(H,POP(t))}\frac{fitness(Y)}{|T(H,POP(t))|}}{\sum_{X\in POP(t)}\frac{fitness(X)}{|POP(t)|}}$

证明：

个体 $pop_i(t)$ 被选中的概率为：$p_i=\frac{f_i}{\sum_{j=1}^{N}f_j},i=1,2,\dots,N$；

模板H的每一个染色体被选中的平均概率为：$\frac{1}{|T(H,POP(t))|}\cdot\sum_{Y\in T(H,POP(t))}\frac{fitness(Y)}{\sum_{X\in POP(t)}fitness(X)}$；

而种群$NewPOP(t+1)$的个体数还是$|POP(t)|$,

故含H模板的染色体数为：$\frac{|POP(t)|}{|T(H,POP(t))|}\cdot\sum_{Y\in T(H,POP(t))}\frac{fitness(Y)}{\sum_{X\in POP(t)}fitness(X)}$

9　在模板理论中，假设模板H在t时刻存在的概率为$P(H,t)$，证明经过简单变异后，有$P(H,t+1)\ge 1-P_m o(H)$，其中P_m为每个基因变异概率，$o(H)$为H阶。(X 2)

证明：变异后模板 H 没有改变的概率为：$(1-p_m)^{o(H)}\ge 1-p_m\cdot o(H)$，即 $P(H,t+1)\ge 1-P_m o(H)$。

10　* 在模板理论中，若在t时刻，模板H的长度为$\delta(H)$。采取简单交叉方式，即随机选一个位置，交叉该位置之后的全部基因，交叉概率为p_c，则 $t+1$ 时刻模板H仍保留下来的概率为：$p_1(H,t+1)\ge 1-\frac{p_c\delta(H)}{n-1}(1-p(H,t))$，其中$p(H,t)$为$t$时刻模板$H$出现的概率。

证明：如果交叉双方有相同的模板，则交叉后模板不变。如果交叉双方中一方有H模板，而另一方不具有H模板，并且交叉位介于第一个模板位至最后一个模板位之间，则模板H可能改变，其改变的最大概率为：$\frac{p_c\delta(H)}{n-1}(1-p(H,t))$。

不改变(保留)概率即 $p_1(H,t+1)\ge 1-\frac{p_c\delta(H)}{n-1}(1-p(H,t))$。

11　* 遗产算法计算过程

Step 1. 随机初始化 pop_size 个染色体
Step 2. 用交叉算法更新染色体
Step 3. 用变异算法更新染色体
Step 4. 计算所有染色体的目标值
Step 5. 根据目标值计算每个染色体的适应度
Step 6. 通过轮盘赌的方法选择染色体
Step 7. 重复 Step 2 到 Step 6 直到终止条件满足
Step 8. 输出最好的染色体作为最优解

四、 蚁群算法

1 在蚁群优化算法中，假设信息素为τ_{ij}，预见值为η_{ij}，试写出人工蚂蚁从i到j的选择概率。

一般规则：$p_{ij}(t)=\begin{cases}\frac{a_{ij}(t-1)}{\sum_{l\in T}a_{il}(t-1)} & j\in T\\ 0 & j\notin T\end{cases}$，$\mathrm{A}_i(t-1)=\{a_{ij}(t-1)|(i,j)\in A\}$ 取决于三部分因素：

1. 信息素$\tau_{ij}(t-1)$和预见度$\eta_{ij}(t-1)$；2. 每个蚂蚁自身的历史信息；3. 问题的约束条件

蚂蚁转移概率为：$p_{ij}(t)=\begin{cases}\frac{\tau_{ij}^{\alpha}(t-1)\eta_{ij}^{\beta}(t-1)}{\sum_{l\in T}\tau_{il}^{\alpha}(t-1)\eta_{il}^{\beta}(t-1)} & j\in T\\ 0 & j\notin T\end{cases}$，

其中α为残留信息的相对重要程度，β为预见值的相对重要程度，T是从i可以到达j的节点集合。

2 在蚁群优化算法中，写出三种信息素的更新方法。(X 2)

2.1 解释蚁群智能优化算法中信息素的一种更新方法。(X 2)

方法一： $\tau_{ij}(t)=\begin{cases}(1-\rho_{t-1})\cdot\tau_{ij}(t-1)+\frac{\rho_{t-1}}{|\hat{s}|} & if\ (i,j)\in\hat{s}\\ (1-\rho_{t-1})\cdot\tau_{ij}(t-1) & otherwise\end{cases}$

其中 $0<\rho_t<1$ 是挥发因子且满足 $\rho_t\le 1-\frac{\ln t}{\ln(t+1)},(t\ge K)$，$\sum_{t=1}^{\infty}\rho_t=\infty$

方法二：(MAX-MIN 方法) $\tau_{\mathrm{ij}}(t)=\begin{cases}\max\{(1-\rho)\cdot\tau_{ij}(t-1)+\frac{\rho}{|\hat{s}|},\tau_{min}(t-1)\} & if\ (i,j)\in\hat{s}\\ \max\{(1-\rho)\cdot\tau_{ij}(t-1),\tau_{min}(t-1)\} & otherwise\end{cases}$

其中 $0<\rho<1$ 是挥发因子，而$\tau_{min}(t-1)$为一个实数。

方法三： $\tau_{ij}(t)=\begin{cases}max\{(1-\rho)\cdot\tau_{ij}(t-1)+\boldsymbol{\rho}\cdot\boldsymbol{g}(\hat{\boldsymbol{s}}),\tau_{min}\} & if\ (i,j)\in\hat{s}\\ max\{(1-\rho)\cdot\tau_{ij}(t-1),\tau_{min}\} & otherwise\end{cases}$

其中 $0<\rho<1$ 是挥发因子，τ_{min}是一个参数，

$0<g(s)<+\infty$是一个满足 $f(s)<f(s')\rightarrow g(s)\ge g(s')$ 的函数，可取 $g(s)=\frac{1}{|f(s)|+1}$

3 写出并解释蚁群优化算法中人工蚂蚁获取可行解的方法。

Step 1. 初始化所有的信息素具有同样的量

Step 2. 根据信息素构造人工蚂蚁行动路线(解)

根据 $p_{ij}(t)=\begin{cases}\frac{\tau_{ij}^{\alpha}(t-1)\eta_{ij}^{\beta}(t-1)}{\sum_{l\in T}\tau_{il}^{\alpha}(t-1)\eta_{il}^{\beta}(t-1)} & j\in T\\ 0 & j\notin T\end{cases}$选择下一个城市，直到完成路线

Step 3. 重复 Step 2 直到所有蚂蚁完成一次行动

Step 4. 根据当前最好解更新路径上的信息素 (信息素更新的三种办法)

Step 5. 重复 Step 2 到 Step 4 直到满足终止条件

Step 6. 输出最好解作为最优解

五、 神经网络

1 简述 Hopfield 神经网络求解组合优化问题的计算步骤。

Step 1. 针对实际的组合优化问题**构造能量函数E**，能量函数有好的稳定性，如满足(稳定|Lasalle不变原理)定理条件;

Step 2. 由能量函数建立**动力系统方程** (该动力系统的平衡点恰好是E的极值点)。

Step 3. 用数值的方法求解动力系统方程，其数值解即为**E的极值点的近似解**

2 描述 Hopfield 人工神经网络的函数逼近一连续函数的方法。

假设$f(x)$是一个连续函数。我们希望训练一个NN去逼近函数$f(x)$。对于一个固定神经元和网络结构的NN，网络权可作成一个向量w。设$F(x,w)$是由NN所得出的输出。训练过程是寻找权向量w以最好地逼近函数$f(x)$。设$\{(x_i^*,y_i^*)|i=1,2,\dots,N\}$是训练数据集。我们希望选择权向量$w$使得$F(x^*,w)$对输入$x_i^*$来说最接近要求的输出$y_i^*$。即训练过程是找权向量$w$以极小化误差函数：$Err(w)=\frac{1}{2}\sum_{i=1}^{N}\left||F(x_i^*,w)-y_i^*\right||^2$

Step 1. 构造函数逼近的能量函数，使得能量函数有好的稳定性，如$Err(\boldsymbol{w})$；

Step 2. 由能量函数$Err(w)$，根据$-\frac{dz_i}{dt}=\frac{\partial Err(w)}{\partial y_i}$求解出动力系统方程 $\begin{cases}\frac{dz_i}{dt}=-A_iz_i+\sum_{j=1}^{n}w_{ij}y_j+I_i\\ y_i=f_i(z_i),\ \ i=1,2,\dots,n\end{cases}$；

Step 3. 用数值计算的方法求解动力系统方程的**平衡点**，用定理判断平衡点是否为稳定点或渐近稳定点，网络达到稳定状态即达到**极小值**。

3 描述求解 TSP 的 Hopfield 方法中可行解的编码方式，并举例说明。

使用 $n\times n$ 矩阵表示 TSP 的一个解，第 i 行表示商人到达城市的顺序，第i行由0,1数字组成一个 n 维向量，其中只能有一个分量为 1，1 所在的序数表示商人到达的序数。

城市\顺序	1	2	3	……
城市 1	0	0	1	
城市 2	1	0	0	
城市 3	0	1	0	
……				

表示商人访问城市的顺序为： 2->3->1 ……

4 单隐层前向型人工神经网络输出方式？(X 2)

单隐层NN其中输入层有n个神经元，输出层有m个神经元，而隐含层有p个神经元。

那么在隐层神经元的输出为：$\mathrm{x}_i^1=\sigma\left(\sum_{j=1}^{n}w_{ij}^0x_j+w_{i0}^0\right),i=1,2,\dots,p$。

因此输出层神经元的输出为：$\mathrm{y}_i=\sum_{j=1}^{p}w_{ij}^1x_j^1+w_{i0}^1,i=1,2,\dots,m$。

系数 $\mathrm{w}_{ij}^0,,i=1,2,\dots,p\ \ ,j=1,2,\dots,n$ 和 $\mathrm{w}_{ij}^1,,i=1,2,\dots,m\ \ ,j=1,2,\dots,p$ 称为网络权。

5 叙述双层前向型神经网络的输出方式。

双隐层NN其中输入层有n个神经元，输出层有m个神经元，而隐含层有$\mathrm{p}^1,\mathrm{p}^2$个神经元。

那么在第一隐层神经元的输出为：$x_i^1=\sigma\left(\sum_{j=1}^{n}w_{ij}^0x_j+w_{i0}^0\right),i=1,2,\dots,p^1$。

那么在第二隐层神经元的输出为：$\mathrm{x}_i^2=\sigma\left(\sum_{j=1}^{p^1}w_{ij}^1x_j^1+w_{i0}^1\right),i=1,2,\dots,p^2$。

因此输出层神经元的输出为：$\mathrm{y}_i=\sum_{j=1}^{p^2}w_{ij}^2x_j^2+w_{i0}^2,i=1,2,\dots,m$。

系数 $w_{ij}^0,i=1,2,\dots,p^1\ \ ,j=1,2,\dots,n$，$\mathrm{w}_{ij}^1,i=1,2,\dots,p^2\ \ ,j=1,2,\dots,p^1$ 和 $\mathrm{w}_{ij}^2,i=1,2,\dots,m\ \ ,j=1,2,\dots,p^2$ 称为网络权。

6 叙述前向型人工神经网络的 BP 算法。

Step 1. 初始化权向量w，选置 $\beta,\alpha,\eta,E_0,\lambda=1,k=0$

Step 2. $k=k+1$ Step 3. 调整权重 w Step 4. 计算误差 E_k

Step 5. 如果$k<N$ 返回 Step 2

Step 6. 置 $E=E_1+E_2+\cdots+E_N$

Step 7. 如果$E>E_0$，那么$k=0,\lambda=e^{-\frac{1}{E^2}}$，返回 Step 2 ；否则结束

六、 其他类型问题

1 为什么学“智能优化算法”？

1.1 为什么要研究智能优化算法？

1.2 简述学习智能优化算法的意义?

智能优化算法可以解决人们在工程技术，科学研究和经济管理等诸多领域中经常遇到的问题。例如：

A. 结构设计：要在满足强度要求等条件下使所用材料的总重量最轻；

B. 资源分配：要使各用户利用有限资源产生的总效益最大；

C. 安排运输方案：要在满足物质需求和装载条件下使运输费用很低；

D. 编制生产计划：按照产品工艺流程和顾客需求，尽量降低人力，设备，原材料等成本使总利润最高；

此外，基于客观数据求解问题更简便经济，而且规模不受限制，人们可以根据从经验得到的规则用实验不断校正数学模型，随着数学方法和计算机技术的进步，用建模与数值模拟解决优化问题这一手段将会越来越显示出它的效能和威力。

2 学习之后有什么感想？对本课程考核方法有什么建议。

3 谈谈学习智能优化算法的目的和收获。