1．代码对应函数

函数 1： 随机生成状态 S。函数 $S = f_1(I, L, W, M, x, \mathscr{H}, \lambda)$。

参数 I，L，W，M，x 都是常数；向量 $\mathscr{H} = (h_1, h_2, \ldots, h_M)$，其中 $h_m \in \{1, \ldots, I\}$；矩阵 $\lambda = (\lambda_{il})$ 是 $I \times L$ 维矩阵。

函数用于随机生成状态 S, $S = ((n_{il}), (i_1, w_1), \ldots, (i_M, w_M))$, 其中 (n_{il}) 是 $I \times L$ 维矩阵，且矩阵中的任意元素取值为 $0, 1, \ldots, x$，服从 Possion 分布，即对于 (n_{il}) 中的第 i 行第 l 列的元素 n_{il} 取值为 $j = 0, \ldots, x$ 的概率为

$$P_j = \frac{\lambda_{il}^j}{j!} \cdot e^{-\lambda}$$

这样便生成了 (n_{il})。

对于 $m = 1, \cdots, M$，(i_m, w_m) 中 $i_m \in \{1, 2, \ldots, I\}$，均匀分布，即取到任何一个值的概率为 $1/I$；$w_m \in \{0, 1, 2, \ldots, W\}$，取值也是均匀分布，$1/(W+1)$。特别注意：当 $w_m = W$，则 $i_m = h_m$ 是固定的。

这样给定各参数，就可以随机生成一个状态 S。

函数 2： 生成 S 到 $\bar{S}$ 的函数。函数 $\bar{S} = f_2(Z, S, X)$

参数：给定状态 $S = ((n_{il}), (i_1, w_1), \ldots, (i_M, w_M))$，和常数 Z 和 X。

函数用于生成压缩后的状态 $\bar{S} = ((N)_{zl}, (g_1, \ldots, g_Z), w)$，其中 Z 将 L 切分为 Z 份，每一份有 L/Z 元素，如 $L = 15$，$Z = 4$，则 $\mathscr{Z}_1 = \{1, 2, 3, 4\}$，$\mathscr{Z}_2 = \{5, 6, 7, 8\}$，$\mathscr{Z}_3 = \{9, 10, 11, 12\}$，$\mathscr{Z}_4 = \{13, 14, 15\}$ 四份，尽量平均分配，有余数则余数作为一类。

(N_{zl}) 是 $Z \times L$ 维的矩阵，如 $z = 1$ 中第 l 列的元素为 $N_{1l} = \max\{X, n_{1l} + n_{2l} + \ldots + n_{L/Z,l}\}$，表示在 $\mathscr{Z}_1$ 这一份中的元素之和，以此类推。

令 $e_z = \sum_{m=1}^{M} \mathbb{I}(i_m \in \mathscr{Z}_z)$ 是在聚类 z 中的数量。并且令 $\bar{M} = \sum_{m=1}^{M} \mathbb{I}(w_m \neq 0)$，这里 $\mathbb{I}(\cdot)$ 是指示函数，等于括号中的就为 1，否则为 0。

令 g_z 满足：

$$g_z = \begin{cases} 0, & e_z = 0; \\ 1, & e_z < \frac{\bar{M}}{Z}; \\ 2, & e_z \geq \frac{\bar{M}}{Z}. \end{cases} \tag{1}$$

最后，令 $W \leftarrow w_1$。

函数 3： 生成 S 到 A 的函数: $A = f_3(S, \mathscr{L}, \mathscr{H}, r_1, (c_1(ij)), c_2)$

参数为 $S = ((n_{il}), (i_1, w_1), \ldots, (i_M, w_M))$ 和 $\mathscr{L} = (l_1, \ldots, l_M)$，其中 $l_m \in \{1, \ldots, L\}$；$\mathscr{H} = (h_1, h_2, \ldots, h_M)$；$r_1 = (r_1(0), r_1(1), \ldots, r_1(L))$；$(c_1(i, j))$ 是 $I \times I$ 的矩阵，c_2 是常数。

状态生成决策 $A = [(i^1, l^1), \ldots, (i^M, l^M)]$。其中 $i^m \in \{1, 2, \ldots, I\}$，$l^m \in \{0, 1, 2, \ldots, L\}$。

A 中 (i^m, l^m) 需要满足如下约束条件：

$$(i^m, l^m) = (h_m, 0), \qquad if\ w_m = 0, \qquad (2)$$

$$(i^m, l^m) \in \{(i, l) | n_{il} > 0, l \geq l_m\} \cup \{(h_m, 0)\}, \qquad if\ w_m > 0, \qquad (3)$$

$$\phi_A(i, l) := \sum_m \mathbb{I}(i^m = i, l^m = l) \leq n_{il}, \qquad \forall i, l. \qquad (4)$$

目标函数是：

$$R(S, A) = \max \sum_m (r(l^m) - c_1(i_m, i^m)) - c_2 \sum_{i,l} [n_{il} - \phi_A(i, l)], \qquad (5)$$

这是一个线性规划问题，需要找到最优的决策 A。

函数 4： 生成 S 到 $\mathcal{L}$ 的函数: $\mathcal{L} = f_4(S, \mathscr{L})$

参数 $S, \mathscr{L}$ 以上都有介绍。

输出分类之后的 l，如 $\mathcal{L} = [(l(1), \ldots, l(k_1)), (l(k_1{+}1), \ldots, l(k_2)), \ldots, (k_n, \ldots, L)]$ 的形式类似，其中 $k_1, \ldots, k_n$ 是根据 S 的变化而变化的，具体对应如下：

令 $N_1(l = j) = \sum_{m=1}^{M} \mathbb{I}(l_m = j, w_m \neq 0)$，其中 $j = 1, \ldots, L$，这里 $N(l = j)$ 表示 $l = j$ 的 m 的数量。令 $N_2(l = j) = \sum_{k=1}^{I} (n_{il})_{kj}$，表示矩阵中第 j 列的加和，表示 $l = j$ 的 n 的数量。

这样，我们假设 $L = 7$，则能够列出 $N_1(l = 1), \ldots, N_1(l = 7)$ 和 $N_2(l = 1), \ldots, N_2(l = 7)$ 如下表，tag 用于比较每一行的大小，则有 $6 > 5$，$6 + 2 > 2 + 1$，$6 + 2 + 3 \leq 5 + 1 + 6$；遇到小于等于号则终止，将 $l = 1, 2, 3$ 归为一类；接着，$2 > 1$，$2 + 4 \leq 1 + 17$，所以 $l = 4, 5$ 归为一类；最后一类不论最后一个 tag 是 $>$ 或 $\leq$，都归为一类。所以在这个例子中的输出结果为 $\mathcal{L} = [(1, 2, 3), (4, 5), (6, 7)]$。注意，若 tag 全为 $>$，则只有一类 $[(1, \ldots, L)]$。

表 1: example

l	N_1	tag	N_2
1	6	$>$	5
2	2	$>$	1
3	3	$\leq$	6
4	2	$>$	1
5	4	$\leq$	17
6	5	$>$	4
7	7	$>$	3

函数 5： 生成 S 到 $A = [A^1, \ldots, A^n]$ 的函数（这里的 A 和函数 3 中的 A 不一样）：$A = f_5(S, \mathcal{L}, N_1, N_2)$.

参数：这里利用了函数 4 得到的结果 N_1 和 N_2，以及 $\mathcal{L}$。

函数 3 是对所有的 m 做分配，而这里函数 5 是对在一个 $\mathcal{L}$ 中的每个元组（类）做分配，例如在表 1 中的 $l = 1, 2, 3$，$l = 2, 4$ 以及 $l = 6, 7$ 独立进行分配，分配的方式和函数 3 一致。

每个类的目标函数为：

$$R^k(S, A^k) = \max \sum_{m \in \mathcal{M}^k(S)} \big[r(l^m) - c_1(i_m, i^m)\big] - c_2 \sum_{i \in \mathcal{I}, l \in \mathcal{L}^k} \big[n_{il} - \sum_{m \in \mathcal{M}^k(S)} \mathbb{I}(i^m = i, l^m = l)\big]. \tag{6}$$

当每个类中的最后一个 tag 是 $\leq$ 时，则有数学规划满足：

$$(i^m, l^m) = (h_m, 0), \qquad if\ w_m = 0, \tag{7}$$

$$(i^m, l^m) \in \{(i, l) | n_{il} > 0, l \geq l_m\}, \qquad if\ w_m > 0, \tag{8}$$

$$\sum_{m \in \mathcal{M}^k(S)} \mathbb{I}(i^m = i, l^m = l) \leq n_{il}, \qquad \forall i \in \mathcal{I}, l \in \mathcal{L}^k, \tag{9}$$

当最后一个 tag 是 $>$ 时，则有数学归纳满足：

$$(i^m, l^m) = (h_m, 0), \qquad if\ w_m = 0, \tag{10}$$

$$(i^m, l^m) \in \{(i, l) | n_{il} > 0, l \geq l_m\} \cup \{(h_m, 0)\}, \qquad if\ w_m > 0, \tag{11}$$

$$\sum_{m \in \mathcal{M}^k} \mathbb{I}(i^m = i, l^m = l) = n_{il}, \qquad \forall i \in \mathcal{I}, l \in \mathcal{L}^K. \tag{12}$$

以上有三个变量需要解释：A^k，$\mathcal{L}^K$ 和 $\mathcal{M}^k(S)$。其中 $A^k=(i^m,l^m)_{m\in\mathcal{M}^k(S)}$，$\mathcal{M}^k(S)=\{m=1,\ldots,M\mid l_m\in\mathcal{L}^k,w_m\neq 0\}$ 是在 $\mathcal{L}$ 中类 k 对应的 m。$\mathcal{L}^k$ 是类 k 对应的 l，如表 1 中 $\mathcal{L}^1=(1,2,3)$；$\mathcal{L}^2=(4,5)$，$\mathcal{L}^3=(6,7)$。

最终对于任意的类 k，得到 $A^k=(i^m,l^m)_{m\in\mathcal{M}^k(S)}$。$A=[A^1,\ldots,A^n]$，实际上，这个也可以写成和函数 3 的输出一样的形式：$A=[(i^1,l^1),\ldots,(i^M,l^M)]$。

函数 6： 函数 5 是输出最优决策，而另一个输出是决策空间 $\mathcal{A}$。记为 $\mathcal{A}=f_6(S,\mathcal{L},N_1,N_2)$。

$\mathcal{A}=[\mathcal{A}^1,\ldots,\mathcal{A}^L]$，其中 $\mathcal{A}^m=[(i_1^m,l_1^m),\ldots,(i_n^m,l_n^m)]$，表示所有 $m=1$ 可能的决策，这个决策满足约束条件(7)-(12)。

换句话说函数 f_5 就是在 f_6 的基础上确定了最优的决策。

函数 7： 生成 $\Xi=f_7(T,x)$

T 和 x 是常数，$\Xi=((\xi_{il})_1,\ldots,(\xi_{il})_T)$。任意的 $(\xi_{il})_t$ 是一个 $I\times L$ 的矩阵，矩阵中的任意元素取值为 $0,1,\ldots,x$，服从 Possion 分布，即对于 $(\xi_{il})_t$ 中的第 i 行第 l 列的元素 ξ_{il} 取值为 $j=0,\ldots,x$ 的概率为

$$P_j=\frac{\lambda_{il}^j}{j!}\cdot e^{-\lambda}$$

这样便生成了 (n_{il})。

函数 8（带起始探索的强化学习算法）： 生成最优值函数和最优策略：$[S_1,V^1,\pi^1]=f_8(f_1,V^0,\Xi,\pi^0)$。

参数介绍：S_1 是由函数 f_1 随机生成的；$V^0=[V_1^0(S_1),V_2^0,\ldots,V_T^0]$，除了 $V_1^0(S_1)$ 是一个常数值，其他的均为向量：$V_t^0=(V_t^0(\bar{S}))_{\bar{S}\in\bar{\mathcal{S}}}$，其中，$\bar{\mathcal{S}}$ 是函数 2 中 $\bar{S}$ 的状态空间，即 $\bar{S}$ 所有可能的情况，这里设 V^0 中所有的元素都为 0。令第 0 次迭代的策略 $\pi^0=[A_1^0,\ldots,A_T^0]$，对于 $t=1,2,\ldots,T$，满足

$$A_t^0=\underset{A_t\in f_6(S_t)}{\arg\max}[R(S_t,A_t)] \tag{13}$$

这里的 $R(S_t,A_t)$ 是(5)中 max 内的部分（简化记为 R_t）。而在 $t+1$ 阶段，$S_{t+1}=S_t+(\xi_{il})_t$ 指的是 S_t 中的 (n_{il}) 加上由函数 7 产生的 $(\xi_{il})_t$ 部分。

以下介绍如何得到 V^1 和 π^1。令 $G_t=G_{t+1}+R_t$，从 $T,\ldots,1$ 逆向进行计算，其中 $G_{t+1}=0$，这样可以计算出所有的 $G_t,t=1,\ldots,T$。更新 V^1

如下：

$$V_t^1(\bar{S}_t) = V_t^0(\bar{S}_t) + \frac{1}{\mathbb{N}(\bar{S}_t)}[G_t - V_t^0(\bar{S}_t)], \tag{14}$$

这里 $V_t^0(\bar{S}_t)$ 表示 V_t^0 这个向量中的第 $\bar{S}_t$ 这个状态的值需要进行更新，更新为 $V_t^1(\bar{S}_t)$，其余向量中的元素不变；$\frac{1}{\mathbb{N}(\bar{S}_t)}$ 表示 $\bar{S}_t$ 这个状态在 t 阶段之前迭代出现的次数，比如这是第一次迭代，肯定都是第一次出现，因此 $\frac{1}{\mathbb{N}(\bar{S}_t)} = 1$；随着出现次数的增加，$\frac{1}{\mathbb{N}(\bar{S}_t)}$ 的值越来越小。

更新 $\pi^1 = (A_1^1, \ldots, A_T^1)$ 如下，对于任意的阶段 t，有：

$$A_t^1 = \underset{A_t \in f_6(S_t)}{\arg\max}[R(S_t, A_t) + V_{t+1}^1(\overline{S_t + (\xi_{il})_t})] \tag{15}$$

这里 $\overline{S_t + (\xi_{il})_t}$ 表示状态 S_t 加上 $(\xi_{il})_t)$ 之后形成下阶段状态后再进行状态压缩（函数 2）。从而我们获得更新后的状态。

函数 9 按照函数 8 进行的规律学习 $[S_1, V^j, \pi^j] = f_8(f_1, V^{j-1}, \Xi, \pi^{j-1})$，只是将 V^0 变为 V^{j-1}，V^1 变为 V^{j-1}。$j = 1, \ldots, J$。特别注意：在每次迭代中，第一个状态 S_1 都要重新根据 f_1 获取。

最终在第 J 次迭代后，我们获得 $V^J = [V_1^J(S_1), V_2^J, \ldots, V_T^J]$。

函数 10 $[V_t, A_t] = f_{10}(S_t)$

给定任意的状态 S_t，能够得到该状态下的最优值 V_t（一个常数）和最优策略 A_t（一个分配策略），满足以下公式：

$$\begin{aligned} A_t &= \underset{A_t \in f_6(S_t)}{\arg\max}[R(S_t, A_t) + V_{t+1}^J(\overline{S_t + (\xi_{il})_t})] \\ V_t &= \underset{A_t \in f_6(S_t)}{\max}[R(S_t, A_t) + V_{t+1}^J(\overline{S_t + (\xi_{il})_t})] \end{aligned} \tag{16}$$

基准策略：

函数 11（随机分配）： $v_1 = f_{11}(\Xi, S_t)$

表示给定 $\Xi = ((\xi_{il})_t, \ldots, (\xi_{il})_T)$ 和状态 S_t 就能够得到在状态 S_t 下的值 v_1。

生成的轨迹如下：$S_t, A_t, R_t, (\xi_{il})_t, S_{t+1}, \ldots, S_T, A_T, R_t$。在这个轨迹中，$A_t$ 的确定是随机的，只需要 $A_t = (i^m, l^m)$ 满足(2)-(4)中，随机选择一个决策即可。

函数 11（就近分配）： $v_2 = f_{11}(\Xi, S_t)$

生成的轨迹如下：$S_t, A_t, R_t, (\xi_{il})_t, S_{t+1}, \ldots, S_T, A_T, R_t$。在这个轨迹中，$A_t$ 的确定是就近的，需要 $A_t = (i^m, l^m)$ 满足(2)-(4)的约束，同时有目标函数：

$$R^2 = \min \sum_{m=1}^{M} c_1(i_m, i^m) \tag{17}$$

函数 12（单阶段最优分配）： $v_2 = f_{12}(\Xi, S_t)$

生成的轨迹如下：$S_t, A_t, R_t, (\xi_{il})_t, S_{t+1}, \ldots, S_T, A_T, R_t$。在这个轨迹中，$A_t$ 的确定是就近的，需要 $A_t = (i^m, l^m)$ 满足(2)-(4)的约束，同时有目标函数(5)。这个是保证每个阶段内是最优的决策。