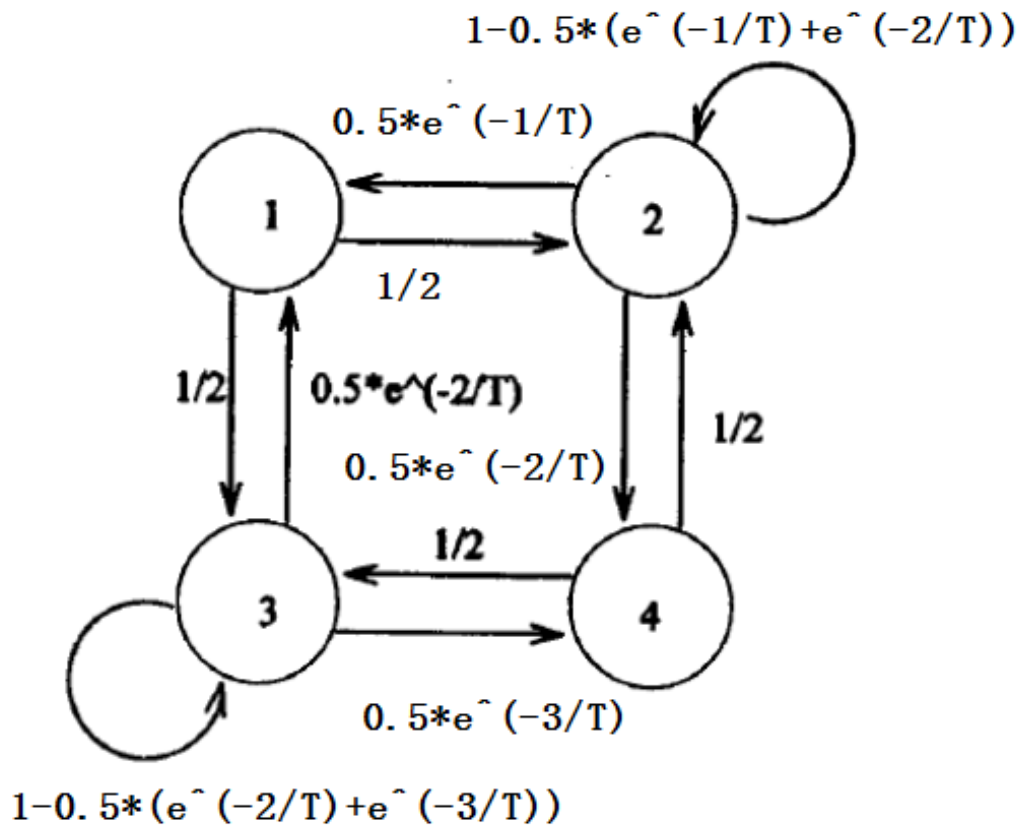


**Problem.1****Part a)**

The cost for each state is:

State	$x_0$	$x_1$	Cost
1	0	1	2
2	1	1	1
3	0	0	0
4	1	0	3

Therefore, we have the transition probability diagram as:



For both 1 and 4, all the neighbors cost less than the current state. Thus, whenever selecting a neighbor, the move will always happen. Thus it's impossible to stay at the current state.

**Part b)**

We have  $\Theta =$

0.000	0.500	0.500	0.000
0.184	0.748	0.000	0.068
0.068	0.000	0.907	0.025
0.000	0.500	0.500	0.000

**Part c)**

We have  $\theta^5 =$

0.098	0.325	0.541	0.036
0.120	0.495	0.341	0.044
0.073	0.126	0.774	0.027
0.098	0.325	0.541	0.036

We have  $\theta^{10} =$

0.092	0.272	0.602	0.034
0.100	0.341	0.522	0.037
0.082	0.192	0.697	0.030
0.092	0.272	0.602	0.034

We have  $\theta^{1000} =$

0.087	0.237	0.644	0.032
0.087	0.237	0.644	0.032
0.087	0.237	0.644	0.032
0.087	0.237	0.644	0.032

Thus, the probability distribution will simply be [0.25 0.25 0.25 0.25] multiplied by each matrix, which gives us:

N = 5: [0.0972 0.3177 0.5493 0.0358]

N = 10: [0.0912 0.2695 0.6057 0.0336]

N = 1000: [0.0871 0.2369 0.6439 0.0321]

**Part d)**

For T = 10, we have  $\Theta =$

0.000	0.500	0.500	0.000
0.452	0.138	0.000	0.409
0.409	0.000	0.220	0.370

0.000	0.500	0.500	0.000
-------	-------	-------	-------

$$\theta^5 =$$

0.128	0.368	0.388	0.116
0.333	0.165	0.200	0.302
0.317	0.181	0.215	0.287
0.128	0.368	0.388	0.116

$$\theta^{10} =$$

0.277	0.221	0.251	0.251
0.200	0.297	0.322	0.181
0.206	0.291	0.316	0.186
0.277	0.221	0.251	0.251

$$\theta^{1000} =$$

0.236	0.261	0.289	0.214
0.236	0.261	0.289	0.214
0.236	0.261	0.289	0.214
0.236	0.261	0.289	0.214

The probability distribution will be [0.25 0.25 0.25 0.25] multiplied by each matrix, which gives us:

$$N = 5: [0.2267 \quad 0.2707 \quad 0.2974 \quad 0.2051]$$

$$N = 10: [0.2399 \quad 0.2576 \quad 0.2853 \quad 0.2171]$$

$$N = 1000: [0.2363 \quad 0.2612 \quad 0.2887 \quad 0.2138]$$

### Part e)

For  $T = 0.2$ , we have  $\Theta =$

0.000	0.500	0.500	0.000
0.003	0.997	0.000	0.000
0.000	0.000	1.000	0.000
0.000	0.500	0.500	0.000

$$\theta^5 =$$

0.002	0.496	0.503	0.000
0.003	0.990	0.007	0.000

$\theta^{10} =$ 

0.000	0.000	1.000	0.000
0.002	0.496	0.503	0.000

 $\theta^{1000} =$ 

0.002	0.492	0.507	0.000
0.003	0.982	0.015	0.000
0.000	0.000	1.000	0.000
0.002	0.492	0.507	0.000

0.000	0.096	0.903	0.000
0.001	0.187	0.813	0.000
0.000	0.005	0.994	0.000
0.000	0.096	0.903	0.000

The probability distribution will be [0.25 0.25 0.25 0.25] multiplied by each matrix, which gives us:

N = 5: [0.0017 0.4954 0.5029 0.0000]

N = 10: [0.0017 0.4912 0.5071 0.0000]

N = 1000: [0.0003 0.0962 0.9034 0.0000]

### Part f)

Yes, as we see that the lower the temperature is, the more likely we will converge to the global optimum as time tends to infinity, but it will converge really slowly, especially when reached a local optimum. On the other hand, the higher the temperature is, the more likely we will converge to a unified distribution for all initial conditions, but at the same time, the more unlikely we will stay at the global optimum.

We need to set the temperature high enough so that the algorithm reaches the optimal solution for any initial conditions within reasonable amount of iterations. But we also need to set the temperature low enough so that we reach the global minimum with high enough probability.

## Problem.2

### Part a)

This is greedy search. Since the probability of moving is always  $\frac{1}{3}$  for all potential moves, it cannot be a simulated annealing, where the probability of uphill move is  $\frac{1}{3}e^{\frac{\text{delta}T}{T}}$ .

**Part b)**

The neighbors of 1 are {2, 4, 5}, since the move probability from 1 to these states are 1/3 each.

**Part c)**

The neighbors of 2 are {1, 3, 4}. We firstly notice that  $M_{23} = M_{24} = 1/3$ , this means that 3 and 4 are neighbors of 2 and are better solution than 2. That leaves up with one unknown neighbor that is worse than 2. We also notice that  $M_{35} > 0$ , meaning 5 is better than 3, and therefore it cannot be the neighbor of 2 (otherwise,  $M_{25}$  will  $> 0$ , as it's a better solution.) Thus 5 cannot be the other neighbor. We notice that 6 cannot be the other neighbor, as we noticed  $M_{66} = 1$ , meaning 6 is a local minimum. And therefore 6 has to be better than at least 3 of its neighbors. If 6 is to be a neighbor of 2, then 2 will need to be better than 4 states (including 6 and its 3 neighbors). This is impossible as we showed both 3, 4, and 5 are better than 2. Thus, the remaining neighbor can only be 1.

**Part d)**

1/3, as the possibilities of moving to all states has to sum to 1.

**Part e)**

0, for the same reason as d)

**Part f)**

We know from matrix that we have the following inequalities:

$J(1) < J(2), J(1) < J(3), J(1) < J(4), J(1) < J(5), J(1) < J(6), J(2) < J(3), J(2) < J(4), J(2) < J(5), J(2) < J(6), J(3) < J(4), J(3) < J(5), J(3) < J(6).$

We cannot conclude any relationships between 4, 5, or 6.

**Part g)**

Yes. Both 5 and 6 are local optimum. 4 Could also be a local optimum. As we have identified the sequence of 1, 2, 3. For the rest, we could assign 4, 5, 6 to have neighbors 1, 2, 3. Then, all of them are local optimum. We could not infer from the table which one is better/equal to each other for 4, 5, 6.

**Problem.3****Part a)**

The lower bound for the probability of a schema surviving crossover in one generation is  $1 - p_c \left( \frac{\delta(H)}{n-1} \right)$ , where  $p_c$  is the crossover probability.

In our case,  $p_c = 1$  and  $n = 11$

For 1\*\*\*\*\*1:  $lowerbound = 1 - \frac{10}{10} = 0$

For 11\*\*\*\*\*:  $lowerbound = 1 - \frac{1}{10} = \frac{9}{10}$

For \*\*\*\*\*111\*\*\*:  $lowerbound = 1 - \frac{2}{10} = \frac{4}{5}$

For \*\*1\*\*\*1\*\*0\*:  $lowerbound = 1 - \frac{7}{10} = \frac{3}{10}$

### Part b)

The probability of surviving mutation in one generation is  $(1 - p_m)^{O(H)}$ , where  $p_m$  is the mutation probability

In our case,  $p_m = 0.9$

For 1\*\*\*\*\*1:  $prob_{survive} = (1 - 0.9)^2 = 0.1^2 = 0.01$

For 11\*\*\*\*\*:  $prob_{survive} = (1 - 0.9)^2 = 0.1^2 = 0.01$

For \*\*\*\*\*111\*\*\*:  $prob_{survive} = (1 - 0.9)^3 = 0.1^3 = 0.001$

For \*\*1\*\*\*1\*\*0\*:  $prob_{survive} = (1 - 0.9)^3 = 0.1^3 = 0.001$

### Part c)

The lower bound of surviving both crossover and mutation in one generation is simply

$$\left(1 - p_c \left(\frac{\delta(H)}{n-1}\right)\right) \left((1 - p_m)^{O(H)}\right)$$

Where  $p_c = 1, p_m = 0.9$ , and  $n = 11$

For 1\*\*\*\*\*1:  $lowerbound = 0 * 0.01 = 0$

For 11\*\*\*\*\*:  $lowerbound = \frac{9}{10} * 0.01 = 0.009$

For \*\*\*\*\*111\*\*\*:  $lowerbound = \frac{4}{5} * 0.001 = 0.0008$

For \*\*1\*\*\*1\*\*0\*:  $lowerbound = \frac{3}{10} * 0.001 = 0.0003$

### Code used for Problem.1

```
function t = getMatrix(T)
t = zeros(4, 4);
t(1, 2) = 1/2;
```

```
t(1, 3) = 1/2;  
t(2, 1) = 0.5*exp(-1/T);  
t(2, 4) = 0.5*exp(-2/T);  
t(2, 2) = 1 - t(2, 1) - t(2, 4);  
t(3, 1) = 0.5*exp(-2/T);  
t(3, 4) = 0.5*exp(-3/T);  
t(3, 3) = 1 - t(3, 1) - t(3, 4);  
t(4, 2) = 1/2;  
t(4, 3) = 1/2;
```

```
t  
t^5  
t^10  
t^1000  
sum(0.25*t^5,1)  
sum(0.25*t^10,1)  
sum(0.25*t^1000,1)
```