CIS675 Take Home Exam

Kailiang Ying

# Answer 1:

Notation**:**

1. Let P(C,M) be the optimum probability of the system performance function properly.

2. M={m1, m2, m3, …. ,mn } is the optimum number of each device when cost constraint is C

3. preM(C\*) is a two dimension array store all optimum M when cost constraint is C\* ( C\* < C )

preM[C-k\*ci]{mi+k} means change mi ∈preM[C-k\*ci] by plus k

## Decomposition equation:

P(C,M) =Max[ P(C-k\*ci, preM(C-k\*ci){mi +k } )]

(1<=preM[C-k\*ci][mi]<=ui ,1<=mi<=ui , ∑ci\*preM[C-k\*ci][mi]<=C-k\*ci , ∑ci\*mi<C ,i∈[1,n],k∈[1,lowerBound(C/ci)])

The simplest case : initial all mi =1 (i∈[1,n]),C =0 ,P(0,M) =0)

Use bottom up approach : every time increase C by 1. P(C+1,M) =Max[P(C+1 –k\*ci ,preM[C-k\*ci]{mi +k})]

### Pseudo-code:

Optimum()

For (I= 0🡪 C)

If(mi>ui or ∑ci\*mi>C)

P(C,M) =0

postM[C]=M

Else

P(C,M) =Max(P(C-k\*ci,preM[C-k\*ci])) // i∈[1,n]

M=postM[C-k\*ci]

M[i] =M[i] +k

postM[C] =M

return P(C,M)

## Explain work correctly:

The notation P(C,M) represent two state :

1. C: the cost constraint
2. M: the optimum number of each device when cost constraint is C

So every P(C\*,M\*) represent the optimum probability when cost constraint is C\* <C. Then every iteration change from the optimum P(C\*,M\*) and choose the maximum from these change results will get the optimum probability at the next state C’

P(C-k\*ci,preM[C-k\*ci]) is already the best probability when cost constraint is C-k\*ci. So if Max[P(C-k\*ci,preM[C-k\*ci]{mi+k})>P(C-1.M) ,it will be the optimum probability ,other wise P(C,M) = P(C-1,M)

So ,finally ,return P(C,M) will get the optimum probability

## Explain bottom up approach:

The simplest case is when C =0; all M[i] =1 for i∈[1,n] , P(0,M)=0

Every time ,increase state C by plus one

EX:P(30,M) =Max[P(30-k\*ci),preM[0]{mi +k}]

So at each state C , traverse all possible sub optimum probability and get the maximum increase from the sub optimum probability and finally get the be the best probability P(C,M).

## Analysis the algorithms :

Running time include 1. Initial the parameter :O(n) 2.dynamic programming : T\*(n)

T\*(n) include 1. C+1 times to find the optimum probability 2. N times ci to traverse 3. O(C) times compare to find the maximum probability when C =C-k\*ci

T\*(n)= (C+1)\*n\*O(C) =O(C\*C\*n)

So T(n) =T\*(n) + O(n) =O(n)

# Answer 2:

P(30,M) =Max[P(30-k\*ci, preM[30-k\*ci]{mi+ k})] for i∈[1,4]

The analysis data is here: <file://localhost/Users/kailiangying/Desktop/Learning/Algorithms/DataAnlysis.xlsx>

M1: 2 M2: 2 M3: 3 M4:2

∑ci\*mi =2\*3 + 2\*5 + 3\*2 +2\*4 <=30

u1= 6 > m1 u2= 4 > m2 u3= 9 > m3 u4 = 5 > m4

So M ={2,2,3,2} is the optimum number of each device

And P(30,M) = 0.869559 is the optimum performance probability when cost constraint is 30