First Assignment:Action Problem

Given a bipartite graph. Two sets, one contains people and another contains objects.
Find an one-to-one relationship to make the benefit MAX.

aij : benefit
pj : price of object j

$$a_{ij_i} - p_{j_i} = \max_{j=1,\ldots,n} \{a_{ij} - p_j\}.$$

But there is a problem: this algorithm may never end.
So, we add a increment for every bidding.
After one iteration, the price of the object will increase by

$$\gamma_i = v_i - w_i + \epsilon,$$

And make some people *almost happy.*
IF

$$a_{ij_i} - p_{j_i} \geq \max_{j=1,\ldots,n} \{a_{ij} - p_j\} - \epsilon.$$

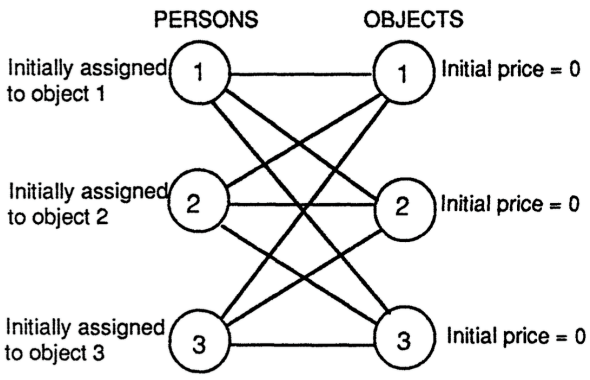
問題在於e,該怎麼選適合的大小

increment越大，演算法越快結束
increment越小，演算法出來的結果越接近optimal
Only persons without an object submit a bid, and objects are awarded to their highest bidder.

在這樣的情況下，我們可以證明最後的total benefit 與最佳的結果相差在ne之內
而又因為aij都是整數的情況下，total benefit 也是整數，因此假如ne<1的話就可以說最後的結果就是最佳的

通常會這樣做

$$C = \max_{i,j} |a_{ij}|.$$



Here $a_{ij} = C > 0$ for all (i,j) with i = 1,2,3 and j = 1,2
and $a_{ij} = 0$ for all (i,j) with i = 1,2,3 and j = 3

Algorithm Pseudo Code

Bidding Phase

set of unassigned person = I, i belong to I, i determines an object ji (BEST VALUE)
Submit a bid : pj+yi (yi=vi - wi + e, vi : best object ; wi : second best object ; e < 1/n)
Every unassigned person send bid to its best object.

Assignment Phase

Every object find the largest bid and assigned itself to that person and increase its price to highest bid. The person who was originally assigned to this object would leave without an object.

Terminate:
a.the auction terminates in a finite number of iterations, with all persons almost happy, before every object receives a bid
b.the auction continues until, after a finite number of rounds, all objects receive at least one bid, at which time the auction terminates.

Pregel Implementation

0.Bidding Phase1:每個Object處理空訊息 & 傳自己的價格給所有還沒擁有物品的Person，每個有得標的Person處理得標msg或是喪失原物品
1.Bidding Phase2:每個Person決定自己的的prefer並且傳送標價給該Object，同時每個Object送一個空訊息給對應的Person
2.Assignment Phase:每個有人標的Object處理所有標價並選出得標者 & 重設本身價格以及擁有者載送訊息給得標者及原主，每個Person送一個空訊息給對應的Object

Person VD: ObjectID(VertexID)=-1 , bid(double)=0, status(Byte)=0
Object VD : OwnerID(VertexID)=-1 , price(double)=0, status(Byte)=0
A : VertexID, double1, double2
Edge ED : aij(benefit)

initialMsg : A(0L ,0 ,0)
iteration : C/e e=constant
vprog : `(VertexId, VD, A) ⇒ VD`
case ObjectVD : if(status == 0){

OwnerId = OwnerID, price = price , status = 1
}

if(status == 1){
if(A._3 == 1)
if(A._1 == 0)// no bid
OwnerId = OwnerID, price = price , status = 0
else{//have bid
OwnerID = A._1, price = A._2, status=0
}
case PersonVD : if(status == 0){
if(A == empty msg)
status = 1
else{
if(A != (0,0,2)){//Get an new Object
ObjectID = A._1, bid=0, status = 1
}
else if(A == (0,0,2)){//Lost the Object OR get Nothing
ObjectID = -1, bid=0, status = 1
}
}
}
if(status == 1){
ObjectID = A._1, bid = first prefer object's value - second prefer object's value + e, status = 0
}

sendmsg :`(EdgeTriplet[VD, ED]) ⇒ Iterator[(VertexId, A)]`

if(Person.status == 1){
Object send A(ObjectVD.VertexID, edge.attr - price, 0) to PersonVD.object == -1
}
if(Object.status == 1&& Person.status == 0){
Person send A(PersonVD.VertexId , bid, 1) to Object<best> : send A(0,0, 1) to other Object
}
if(Object.status == 0 && Person.status == 0){
if(owner != -1) Object send A(objectID, 0, 2) to Person (Owner) or ; send A(0,0,2) to other Person
else empty
Person send A(0,0,2) to all Object01
}

mergemsg :`(A, A) ⇒ A`

case ObjectVD : Do nothing <Bidding Phase1>
if(A._3 == 1) {//bid msg
find Max A._2 return A(Max A._2's A._1, Max A._2, 0) <Bidding Phase2>
}
if(A._3 == 2) return A <Assignment Phase>
case PersonVD : if(A._3 == 0)//value msg
{Find A(prefer ObjectVD.VertexID, first prefer object's value, second

prefer object's value) <Bidding Phase1>
}
Do nothing <Bidding Phase2>
if(A._3 == 2) find max A._1 <Assignment Phase>