HW2 - Algorithm Design

Federico Mascoma

May 20, 2018

Exercise 1

Algorithm

```
def Exercise1(pointList):
    Xmin = Xmax = Ymin = Ymax = 0
    for c in pointList:
        x = c[0]
        y = c[1]
        if x < Xmin:
            Xmin = x
        if x > Xmax:
            Xmax = x
        if y < Ymin:
            Ymin = y
        if y > Ymax:
            Ymax = y
    w = (float)(Xmax-Xmin)
    h = (float)(Ymax-Ymin)
    diagonalApprox = math.sqrt(w*w + h*h)
    return diagonalApprox
```

Running Time O(n) where n are the cardinality of the pointlist.

Claim: The outcome of the algorithm is an $\sqrt{2}$-approximation of the diagonal.

Proof: $diagonal = \max_{a,b \in A} dist(a, b)$
Now just draw a polygon that contains all the set of points.
$diagonalApprox = \sqrt{w^2 + h^2}$ is the max distance between 2 points inside the polygon, hence in the set.
Besides, the minimum distance between two points inside the polygon is at least as long as the longer side of the polygon ($\max(w, h)$).
This is an $\sqrt{2}$-approximation because $diagonalApprox \geq diagonal \geq \max(w, h) \geq \frac{diagonalApprox}{\sqrt{2}}$.
indeed, in the extreme case when $w = h = l \Rightarrow \frac{diagonalApprox}{\sqrt{2}} = \frac{\sqrt{l^2+l^2}}{\sqrt{2}} = l$

Exercise 2

Algorithm

$V =$ vertexes of G

S set of $\frac{24n}{\epsilon^2}$ edges of E

$MaxCut(A, B) = computeMaxCut(V, S)$:

return $|(A \times B) \cap S| \cdot \frac{|E|}{|S|}$

Claim: $|(A\times B)\cap S|\frac{|E|}{|S|}$ is a $(1\pm\epsilon)-$approximation of $|(A\times B)\cap E| = w(A, B)$ where (A, B) is a $maxcut$

Proof: Just take into account the probabilistic inequalities given.
$P[\sum_i^{|S|} X_i > (1+\epsilon)\cdot|S|\cdot\mu] = P[|(A\times B)\cap S| > (1+\epsilon)\cdot|S|\cdot\frac{|(A\times B)\cap E|}{|E|}] =$
$P[|(A\times B)\cap S|\cdot\frac{|E|}{|S|} > (1+\epsilon)\cdot|(A\times B)\cap E|] < exp(-\frac{\epsilon^2|S|\cdot\mu}{3})$

Just prove that $exp(-\frac{\epsilon^2|S|\cdot\mu}{3}) \to 0$

$exp(-\frac{\epsilon^2|S|\cdot\mu}{3}) = exp(-\frac{1}{3}\frac{24n\cdot|(A\times B)\cap E|}{|E|})$

Since $(A, B) = maxcut \to |(A\times B)\cap E| \geq \frac{1}{2}|E|$ =minimum number of edges of a $maxcat$ having E

Follows that $exp(-\frac{1}{3}\frac{24n\cdot|(A\times B)\cap E|}{|E|}) \leq exp(-\frac{1}{3}\frac{24n}{2}) = exp(-4n)$

Since $|S| = \frac{24n}{\epsilon^2}$ and $S \subseteq E \Rightarrow n \gg 48$

Therefore $exp(-4n) \leq exp(-4\cdot 49) \to 0$

In the same way the second inequality is true with probability $exp(-\frac{\epsilon^2|S|\cdot\mu}{2})$
Just to do the same calculations to conclude $exp(-\frac{\epsilon^2|S|\cdot\mu}{2}) \ll exp(-6\cdot 49) \approx 0$

Claim: $|(A'\times B')\cap S|\frac{|E|}{|S|} > |(A'\times B')\cap E| + \epsilon\cdot|(A\times B)\cap E|$ with probability $\leq \frac{1}{2}$ considering any cut (A', B') and $maxcut$ (A, B)
Let $\dot{\mu} = \frac{|(A'\times B')\cap E|}{|E|}$
Since $\epsilon\cdot|(A\times B)\cap E| \geq \epsilon\cdot|(A'\times B')\cap E|$ I'll prove $P[|(A'\times B')\cap S|\frac{|E|}{|S|} > (1+\epsilon)\cdot|(A'\times B')\cap E|] \leq \frac{1}{2} \Rightarrow |(A'\times B')\cap S|\frac{|E|}{|S|} > |(A'\times B')\cap E| + \epsilon\cdot|(A\times B)\cap E| \leq \frac{1}{2}$

Proof: We have to prove $exp(-\frac{\epsilon^2|S|\cdot\dot{\mu}}{3}) \leq \frac{1}{2}$ for any cut (A', B')
We can simply say that, since our goal is to estimate a $maxCut$ with edges $\in S$, we can follow that the algorithm will produce a cut (A', B') s.t: $|(A'\times B')\cap E| \geq \frac{|S|}{2}$ (also if (A', B') is not a $maxcut$ in $G(V, E)$).

Under this assumption trivially follows that :
$exp(-\frac{\epsilon^2|S|\cdot\dot{\mu}}{3}) \leq exp(-\frac{8n\cdot|(A'\times B')\cap E|}{|E|}) \leq exp(-\frac{8n\cdot\frac{|S|}{2}}{|E|}) = exp(-\frac{4n\cdot|S|}{|E|})$
since $|E| \leq \frac{n(n-1)}{2}$ follows that :$exp(-\frac{4n\cdot|S|}{|E|}) \leq exp(-\frac{4n\cdot\frac{24n}{\epsilon^2}}{\frac{n(n-1)}{2}})$
Since $\frac{4n\cdot 24n}{\epsilon^2}\cdot\frac{2}{n(n-1)} > 192 \Rightarrow exp(-\frac{\epsilon^2|S|\cdot\dot{\mu}}{3}) < exp(-192) < \frac{1}{2}$

Proof if is the algorithm want to estimate any cut

Aniway Just prove that $exp(-\frac{\epsilon^2|S|\cdot\dot{\mu}}{3}) \leq \frac{1}{2}$ for any *cut* (A', B').
$exp(-\frac{1}{3}\frac{24n\cdot|(A'\times B')\cap E|}{|E|}) = exp(-\frac{8n\cdot|(A'\times B')\cap E|}{|E|})$
Since we din't know $G(V, E)$ we can see it as a random graph $G(V, p)$.
Every pair of nodes are connected by an edge with probability p
The occurrence of each edge in the graph is independent from other edges in the graph.
Besides we can follow that:

$p\cdot\frac{n(n-1)}{2} = |E|$

$p\cdot n =$ average of number of edges that each node gets.

Now just execute the calculations considering a possible *mincat* where: $|A'| = 1$ and $|B'| = |V| - 1$, $v \in A\ \forall\ v \in V$:
Hence $exp(-\frac{8n\cdot|(A'\times B')\cap E|}{|E|}) \approx exp(-\frac{8n\cdot p\cdot n}{\frac{n(n-1)p}{2}}) = exp(-\frac{16n}{n-1}) < \frac{1}{2}$

Since $(A, B) = maxcut \Rightarrow \epsilon\cdot|(A\times B)\cap E| > \epsilon\cdot|(A'\times B')\cap E|$.
Since $P[|(A'\times B')\cap S|\frac{|E|}{|S|} > (1+\epsilon)\cdot|(A'\times B')\cap E|] \leq \frac{1}{2}$, is true that
$P[|(A'\times B')\cap S|\frac{|E|}{|S|} > |(A'\times B')\cap E| + \epsilon\cdot|(A\times B)\cap E|] \leq \frac{1}{2}$.

We can observe that we have approximated $\dot{\mu} = \frac{(A'\times B')\cap E|}{|E|} \approx \frac{p\cdot n}{\frac{n(n-1)p}{2}}$.
We should prove that the minimum value of $\dot{\mu}$ ensure that the probability $exp(-8n\cdot\dot{\mu}) \leq \frac{1}{2} \Rightarrow \dot{\mu} \geq \frac{1}{4n}$

Using chebyshev's inequality, Take into account:

$\mu = \frac{|(A\times B)\cap E|}{|E|}$

$\dot{\mu} = \frac{|(A'\times B')\cap E|}{|E|}$

$\tilde{\mu} = \frac{|A'\times B'\cap S|}{|S|}$

$P[\sum_i^{|S|} X_i > (1+\epsilon)\cdot|S|\cdot\dot{\mu}] = P[|(A'\times B')\cap S| > (1+\epsilon)\cdot|S|\cdot\frac{|(A'\times B')\cap E|}{|E|}] =$
$P[|(A'\times B')\cap S|\cdot\frac{|E|}{|S|} > (1+\epsilon)\cdot|(A'\times B')\cap E|] \geq$
$P[|(A'\times B')\cap S|\cdot\frac{|E|}{|S|} > |(A'\times B')\cap E| + \epsilon\cdot|(A\times B)\cap E|] =$
$P[\frac{|(A'\times B')\cap S|}{|S|} > \frac{|(A'\times B')\cap E|}{|E|} + \epsilon\cdot\frac{|(A\times B)\cap E|}{|E|}] =$

$P[\tilde{\mu} > \dot{\mu} + \epsilon\mu] = P[|\tilde{\mu} - \dot{\mu}| > \epsilon\mu] < exp(-\frac{\epsilon^2|S|\cdot\dot{\mu}}{3})$

By chebyshev's inequality : $P[|\tilde{\mu} - \dot{\mu}| > \epsilon\mu] \leq \frac{Var[\tilde{\mu}]}{\epsilon^2\mu^2} \leq \frac{1}{4|S|\cdot\epsilon^2\cdot 0.5^2} = \frac{1}{24n} \ll \frac{1}{2}$

Exercise 3

3.1 Just read the slide about primal-dual 3-approximation algorithm presented in the course and prove that the structure of the algorithm hold also in the case we are considering demand d for each city.
Is important to note that the demand d_j is strictly connected to the city and it is the same for every path and every facility that the city choose.
The new dual relaxed problem is represented below:

- $maximize \sum_{j\in C} \alpha_j = \sum_{j_\in C} d_j \cdot c_{ij} + \beta_{ij}$
- S.T :

  $\alpha_j - \beta_{ij} \leq c_{ij} \cdot d_j$
  $\sum_{j\in C} \beta_{ij} \leq f_i$
  $\forall \alpha_j, \beta_{ij}, d_j \geq 0$

3-Approximation Algorithm

- At time t_0, set all $\alpha_j = 0$ and $\beta_{ij} = 0$ and declare all cities unconnected.
- While there is an unconnected city:
  - Raise uniformly all α's of unconnected cities:
  - If $\frac{\alpha_j}{d_j} = c_{ij}$, declare city j tight with facility i
  - For a tight constraint ij, raise both α_j and β_{ij}
  - If $\sum_j \beta_{ij} = f_i$ at time t_i declare:
    - Facility i temporarily opened at time t_i;
    - All unconnected cities j that are tight with i connected.

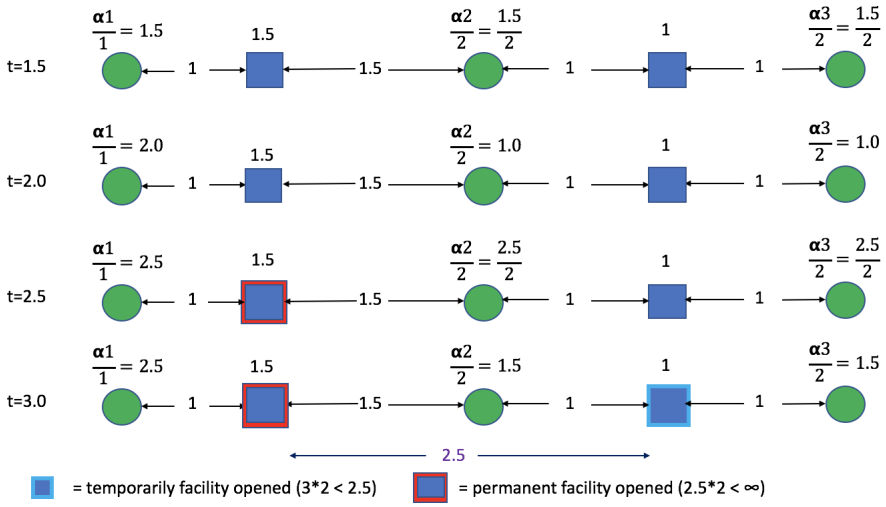
Opening facilities:
The Cities contribute to more permanently opened facilities. Not enough money for all of them.

Facility i temporarily opened at time t_i;

Declare facility i permanently opened if there is no permanently opened facility within distance $2t_i$.

Open all permanently opened facilities.
Connect each City to the nearest opened facility.

Example of the Algorithm



Running Time: Running time of the algorithm is equal to the running time of the algorithm in the slide. For each cycle of while it has to do:

- Rising each α
- Check if cities have connected to any facility with cost $c_{ij} = \alpha_j$
- Declare Facility which are opening checking if is temporarily or permanently opened.

The running time is $O((Nc \cdot Nf + Nf) \cdot T)$ where:

Nc =number of city

Nf =number of facility

$Nc \cdot Nf$ = number of edges that link all city to all facility,

T = number of times where the last α_j has been raised = $\min_{i \in N_f}(c_{ij}) \cdot d_j$ in the worst case where last unconnected city j have connected to the facilities with the most expensive edges $\in M$

Claim: Each $\alpha_j \leq \frac{1}{3}\alpha_j^*$ where each α_j^* is the optimal cost of α_j to maximizes the dual problem.

Proof: Taking any solution proposed by the algorithm above, we know that each city is connected to a facility. Just analyze 2 cases for each city j.

1. City j is connected to opened facilities i:
   $\alpha_j = c_{ij} \cdot d_j + \beta_{ij}$.
   α_j pays for connection cost $c_{ij} \cdot d_{ij}$ and contribute β_{ij} to open f_i.
   Since other opened facilities are at distance $> t_i$, α_j does not pay for opening any other facility.

2. City j is connected to temporarily opened facilities i, hence there exists an opened facility i' with $c_{ii'} \leq 2t_i$.
   Since $c_{ji} \cdot d_j \leq \alpha_j$ and $t_i \leq \alpha_j$ trivially follows that:
   $c_{ji'} \leq c_{ji} + c_{ii'} \leq \frac{\alpha_j}{d_j} + 2\alpha_j \leq 3\alpha_j$.

3.2 (Bonus Problem)

- $maximize \sum_{j \in C} \alpha_j$
- S.T :
  - $\alpha_j - \beta_{ij} \leq c_{ij}$
  - $\sum_{j \in C} \beta_{ij} \leq y_i f_i$
  - $\sum_{i \in F} u_i y_i \geq |C|$
  - $\forall \alpha_j, \beta_{ij}, x_{ij}, y_i \geq 0$
    where $x_{ij} = 1$ if city j is tight with facility i, 0 otherwise
    and y_i is the number of time that facility i is opened
- At time t_0, set all $\alpha_j = 0$, $\beta_{ij} = 0$, $x_{ij} = 0$, $y_i = 0$ and declare all city unconnected.
- While there is an unconnected city:
  - Raise uniformly all α's of unconnected city:
  - If $\alpha_j = c_{ij}$ declare city j tight with facility i, setting $x_{ij} = 1$,
  - For a tight constraint ij, raise both α_j and β_{ij}
  - If $\sum_j \beta_{ij}$ mod $f_i = 0$ and $\sum_j \beta_{ij} > 0$ at time t_i declare:
    - Facility i temporarily opened at time t_i;
    - u_i unconnected cities j,randomly chosen, that are tight with i connected.

Opening/closing facilities:
Facility i temporarily opened at time t_i;
Declare facility i permanently opened if there have been no permanently opened facility within distance $2t_i$.
Open all permanently opened facilities.
Connect each City to the nearest opened facility.
Declare facility i closed at time t_i when $\sum_j x_{ij}$ mod $u_i = 0$ and $\sum_j x_{ij} > 0$

Claim: Every $4\alpha_j \leq \alpha_j^*$ where α_j^* is the value that city j pay to maximize total cost.

Proof:

- if City j is connected to permanent opened facilities i:
  $\alpha_j = c_{ij} + \beta_{ij}$.
  α_j pays for connection cost c_{ij} and contribute with β_{ij} to f_i.
  Since all other facilities opened at least once are at distance $> t_i$, α_j does not pay for opening any other facility.

- if City j is connected to temporarily opened facilities i, hence there exists at least an opened facility i' (or have been opened at least once) with $c_{ii'} \leq 2t_i$.
  Therefore city j could contributed several time to open some other facility paying $\beta_{i'j} = \alpha_j - c_{ji} - \beta_{ij} \leq \alpha_j$
  Since $c_{ji} \leq \alpha_j$ and $t_i \leq \alpha_j$ trivially, from the triangle inequality, follows that:
  $c_{ji'} + \beta_{i'j} \leq c_{ji} + c_{ii'} + \beta_{i'j} \leq \alpha_j + 2t_i + \alpha_j \leq 4\alpha_j$.

Exercise 4

4.1 A pure Nash Equilibrium is an assignment of agents to machines so that no agent can unilaterally deviate to a different machine and decrease her load. The potential function is $\Phi = \sum_j \frac{1}{2} L_j^2$.

Claim: Starting from any solution it is possible to converge to a pure Nash equilibrium.

Proof: Just start from any possible solution. By definition if $\nexists$ an agent s.t deviating to a different machine decreases her machine load, means that this solution is a pure Nash Equilibrium.

Suppose by contradiction that $\exists$ always at least an agent s.t deviating to a different machine decreases the load machine when the agent works.

If the agent a decreases her load deviating from machine m_i to a machine m_j means that $L_j + w_a < L_i$ when $w_a \in \Re > 0$

Trivially follows that also Φ is decreased because : $L_i^2 + L_j^2 > (L_i - w_a)^2 + (L_j + w_a)^2 \; \forall L_i > L_j + w_a$, In fact Φ reach its lower bound when all the machines have the same L, that is when $\Phi = \frac{1}{2} M \cdot L^2$ (considering the same set of job).

We can go on with the deviations if these decrease the load of the agent, but is surely no possible go on after reached the lower bound of Φ.
Therefore no agent can unilaterally deviate to a different machine and decrease her load because this means decrease the lower bound of Φ too, hence every solution converge to a pure Nash Equilibrium.

4.2 Be the $makespan = \max_j L_j$

Claim: The makespan of the pure Nash equilibrium is at most twice the minimum makespan.

Proof: Let $L_j = makespan$ of the pure Nash Equilibrium, $OPT = minimum\ makespan$ and $AVG = \frac{\sum_j L_j}{M}$ the average (hence the LowerBound)
Thus $L_j \geq OPT \geq AVG$
Assume By Contradiction $L_j > 2 \cdot OPT$.
We know that $L_j \neq OPT \Rightarrow$ machine j contains at least 2 job
$L_j = \max L$ in PNE $\Rightarrow$ all other machines have $L \geq \frac{L_j}{2}$.
Assuming the extreme case where all $L = \frac{L_j}{2}$ and L_j gets 2 jobs, we can write that $AVG = \frac{(M+1)L}{M} = L + \frac{L}{M}$
Since $L_j = 2L \leq 2(L + \frac{L}{M}) = 2AVG \Rightarrow L_j \leq 2 \cdot OPT$, and this is a contradiction.

Exercise 5 (Bonus Problem)

If the coin is unfair $\Rightarrow p(head) \neq p(tail)$.
Let assume that $p(head) = p$, hence $p(tail) = 1 - p$. Therefore toss once the coin doesn't simulate a fair coin toss.
Just see what happens tossing the coin twice:

- $p(head, head) = p \cdot p$
- $p(head, tail) = p \cdot (1 - p)$
- $p(tail, head) = (1 - p) \cdot p$
- $p(tail, tail) = (1 - p) \cdot (1 - p)$

Trivially follows that only 2 among 4 possible combinations have the same probability, indeed $p(head, tail) = p(tail, head) = p \cdot (1 - p)$.
Therefore for simulating a fair coin toss each elve has to choose: $(head, tail)$ or $(tail, head)$ before the first coin toss.

If happens (head,head) or (tail,tail) they'll must repeat the two tosses.

Algorithm

```
choice_of_the_elves()
repeat = true
while repeat:
   toss1 = toss_coin()
   toss2 = toss_coin()
   if toss1 != toss2:
      repeat = false
win = (toss1,toss2)
```

Running Time: Consider Random Variable X = number of consecutive tosses before we have an outcome.

$$E[X] = \sum_{i=0}^{\infty} i \cdot P(X = i) = \sum_{i=0}^{\infty} i(1 - 2p(1 - p))^{i-1}(2p(1 - p)) = \frac{2p(1-p)}{1-2p(1-p)} \cdot \sum_{i=0}^{\infty} i(1 - 2p(1 - p))^{i} = \frac{2p(1-p)}{1-2p(1-p)} \cdot \frac{1-2p(1-p)}{(2p(1-p))^2} = \frac{1}{2p(1-p)}$$

The expected number of iterations s.t the algorithm produce an outcome is $\frac{1}{2p(1-p)}$ where $2p(1 - p) = p(head, tail) + p(tail, head) = p(outcome)$.
To get iterations as less as possible $p - (1 - p)$ should be as more as possible equal to 0, $(p(head) = p(tail))$.
The elves have seen that one side appears more frequently than the other, (having no idea as to how big the bias is), so we assume that difference between p and $1 - p$ is not too large, therefore the algorithm is not too long.