CS 539
Take Home Exam

Christopher Marcum
A20385764

① a) Given: $C_u = \alpha n_u + \beta \cdot \Sigma_v (dist(u,v))^2$, $\alpha, \beta \geq 0$

let us assume three cases: $\alpha = \beta$, $\alpha < \beta$, $\alpha > \beta$

for $\alpha = \beta = k$

$C_u = k(n_v + \Sigma_v dist^2(u,v))$

as α & β grow, the cost grows linearly with respect to k.

if the graph is a star, then $dist(u,v)$ is at max 2

but if G is complete then $dist(u,v)$ is max 1 and $n_v = \frac{n(n-1)}{2}$

any other graph would have $dist(u,v) \geq 1$ with $n_v \geq n-1$ nodes

meaning the OPT is a star graph. for $\alpha = \beta$

we can use a ratio of sums for the other 2 cases:

(*) $\frac{C_{star}}{C_{complete}} = \frac{4\beta + a(n-1)}{\beta + a\left(\frac{n(n-1)}{2}\right)}$ [note: $\lim_{n\to\infty} \frac{C_s}{C_c} = \frac{1}{2}$ for $\beta = \alpha$]

if $\beta < a$: then a star is OPT since adding a node decreases cost by $4\beta - \alpha$ percent and removing an edge from a complete graph decreases cost by a ...

so if $\alpha \geq 4\beta$ then adding a node is not incentivized and only removing nodes is...

if $a \geq \beta$ then a complete graph is OPT since reducing an ~~node~~ edge lowers cost by 1 and adding an edge can reduce a cost by $a - 4\beta$ for $n-1$ paths where all nodes are incentivized to add a node.

b) Similarly (*) shows us a * graph is NE since ~~all players~~ paths want to mimize their max cost so from a star graph, adding a node doesn't decrease their max cost unless it creates a complete graph, and from a complete graph, the cost decreases By $4\beta - a\left(\frac{n(n-1)}{2}\right)$ so a star graph is ideal since removing nodes to get cost $4\beta + a(n-1)$ is ideal.

② Given $S_i = k \ \forall m$, social objective $O_G = \Sigma_i L(A_i) = \Sigma_i \Sigma_j w_j$

$PoA = \frac{\max_{NE}(O)}{OPT}$

since all machines have the same speed, average cost is $C_i = \frac{1}{m} \sum_i L(A_i)$

$C_i = \frac{1}{m} \sum_i \sum_j w_j$ which is social optimum since no machine is incentivized to have a higher cost than average since they are all the same speed. i.e. $OPT = \min\{c_i \in C\} = \frac{1}{m} \sum_i L(A_i)$. In case one machine requires a higher load, then you can assign jobs in order of weight

for max NE (O):

Let us assume not all jobs are equal such that

$w_1 \leq w_2 \leq \dots \leq w_n \quad \forall w \in W$

$\frac{W_k}{S_i} \leq \frac{w_k + \binom{W}{\alpha \neq k}}{S_i}$ where 'α' is a subset of any size in W

this implies $\max_{NE}(O) = \max\left\{\frac{W_k}{S_i}\right\} \forall i \in m \ \& \ \forall k \in W$

$\max_{NE}(O) = \frac{\max\{W_k\} \in W}{S_i}$

where the $\max_{NE}$ is determined by the 'heaviest' job.

hence $PoA = \frac{\max\{W_k\}/S_i}{\frac{1}{m} \cdot \sum_i L(A_i)} = \frac{m}{S_i} \cdot \frac{\max\{W_k\}}{\sum L(A_i)} = \frac{m}{S_i} \cdot \frac{\max\{W_k\}}{O_G}$

let $\frac{m}{S_i} = \alpha$ since s_i & m are constants so:

$$PoA = \alpha \cdot \frac{\max\{w_k\}}{O_G}$$

(3) Given: $\text{Cost}_e = C(e) \geq 0$, $\text{delay}_e = d(e) \geq 0$, $N = K_n$ with $\binom{n}{2}$ edges

Find: path $p \in N$ where $\sum_p d(e) < D$ where $\sum_p c_e$ is minimized

Note: $C(e)$ is a bid by ISP & $d(e)$ is not

Answer|

We want to find a VCG mechanism s.t. $f(v_i \ldots) = \arg\max_{a \in A} \sum_i v_i(a)$

in terms of our problem, we find the shortest path p where $d_p = \sum_p d(e) < D$. That is, we find a path p minimizing delay where $d(\hat{e}) = \Sigma_{e \in p'} d(e) - \sum_{e \in p - \{e_0\}} d(e)$.

essentially, we bid for a min cost path from the set of min delay paths with delay less than D.

so we find $\underset{\sum_p C(e)}{\arg\min} \left\{ \underset{\sum_p d(e)}{\arg\min} \{ P_{s \to t} \in N \} \right\}$.

so given a complete graph N, we can assume there is always a path containing e and not containing e.

hence: total delay $d = \sum_{e \in p^*} d(\hat{e})$ where p^* is the winner path

so $d = \sum_{e \in p^*} \sum_{e \in p'} d(e) - \sum_{e \in p^*} \sum_{e \in p - \{e_0\}} d(e)$

where there can be any path $\{p^*\}$ with delay $\{d_{p^*}\} < D$.

since we only care if $d_{p^*} < D$ and we want to minimize cost, Then the problem is incentive compatible where we procure any path with delay $d(p^*) < D$ and minimize the cost such that we get:

$$p^* = \underset{\sum_p C(e) \in N}{\arg\min} \left\{ \sum_{e \in p} \left(\sum_{e \in p'} d(e) - \sum_{e \in p - \{e_0\}} d(e) \right) < D \right\}$$

4) a) Traders maximize utility when all endowment is spent.

so: when goods j & j' are bought at prices P_j & $P_{j'}$, then

$$k \cdot \frac{P_j}{P_{j'}} = \frac{u_{ij}}{u_{ij'}}$$ where the ratio of prices are proportional to the utility gain of the goods.

$u_{ij} = v_{ij} x_{ij}$ where $v_{ij} = r_{ij} \cdot s_i$ so:

$$k \cdot \frac{P_j}{P_{j'}} = \frac{x_{ij}}{x_{ij'}} \cdot \frac{r_{ij}}{r_{ij'}} \cdot \underbrace{\frac{s_i}{s_i}}_{1}$$ so to prove ratios equal

we construct an endowment m_i where $K = \frac{x_{ij'}}{x_{ij}} = 1$

we know for linear fisher: $m_i = \sum_j p_{ij} x_{ij} \rightarrow p_j = \frac{m_{ij}}{x_{ij}} \rightarrow p_i = \frac{m_i}{\sum_i x_{ij}}$

→ and if a product has a $P_j > 0$ then $\sum_i x_{ij} = \sum_i x_{ij'} = 1$

which shows us $x_{ij} = x_{ij}$ since the endowment spent per good equals $x_{ij} = x_{ij'}$

hence: $$\frac{P_j}{P_{j'}} = \frac{r_{ij}}{r_{ij'}}$$

b) Market equilibrium occurs when profits (utility) are maximized which happens when a_j of j is sold and m_i for i is spent.

from part A, we can get $P_{ij} r_{ij'} = P_{ij'} r_{ij}$ and we know $m_i = \sum_i p_{ij} x_{ij}$ where x_{ij} converges to 1 so $m_i \leq p_i$, meaning we want to sell products to break even or make a profit

so we set the price (*) $P_{i'j} \geq P_{ij} \geq m_{ij}$ or the price at least what we bought but no greater than our competitor. This incentivizes i to buy more j while i' does not since they would not gain anything

hence allocation $a_{ij} = \begin{cases} 1 & \text{if (*) satisfied} \\ 0 & \text{if (*) not satisfied} \\ [0, a_j - a_{i'j}] & \text{if competitor sells at our price} \end{cases}$