Q1: 10 points The proof that the double coverage algorithm is k-competitive for the k-server problem on the line is by a potential argument. The proof follows from Table 5.1. Verify table 5.1.

Move type	ΔDC	$\Delta\Phi_1$	$\Delta\Phi_2$	$\Delta\widetilde{DC}$	ΔOPT
event (a) OPT moves	0	$\leq k \cdot \Delta OPT$	0	$\leq k \cdot \Delta OPT$	ΔOPT
event (b), DC moves rightmost or leftmost server	ΔDC	$-k \cdot \Delta DC$	$(k-1) \cdot \Delta DC$	0	0
event (b), DC moves two servers for an "in-between" request	ΔDC	≤ 0	$-\Delta DC$	0	0

Table 5.1: Summary of changes to DC, OPT, and potential function during one step.

potential function:

$$\Phi(p, q) = \Phi_1 + \Phi_2 \qquad \Phi_1 = k \sum_{i=1}^{k} |p_i - q_i| \qquad \Phi_2 = \sum_{i<j} |p_i - p_j|.$$

Row 1. OPT moves but DC does not move.

So obviously $\Delta DC = 0$.

For $\Delta\Phi_1$: only the positions of OPT's servers change.

So for each i. let q_i' be the new position of q_i.

then $\Delta|p_i - q_i| = |p_i - q_i'| - |p_i - q_i| \leq |q_i' - q_i|$

Sum up for all i, we get

$$\sum_{i=1}^{k} \Delta|p_i - q_i| \leq \sum_{i=1}^{k} |q_i' - q_i| = \Delta OPT.$$

multiply both sides with k.

$$k \sum_{i=1}^{k} \Delta|p_i - q_i| \leq k \cdot \Delta OPT.$$

So we get $\Delta\Phi_1 = \Delta\left(k \cdot \sum_{i=1}^{k} |p_i - q_i|\right) = k \cdot \sum_{i=1}^{k} \Delta|p_i - q_i| \leq k \cdot \Delta OPT$

For $\Delta\phi_2$: Since DC does not move, and ϕ_2 only depends on the positions of DC's servers. So $\phi_2 = 0$.

Hence, for this row: $\Delta DC = 0$, $\Delta\phi_1 \le k \cdot \Delta OPT$, $\Delta\phi_2 = 0$.

Then $\Delta DC + \Delta\phi \le k \cdot \Delta OPT$. verified.

Row 2. DC moves rightmost or leftmost server, OPT does not move.

Case 1. DC moves the rightmost server to the request $x > p_k$.

ΔDC is exactly the distance moved by p_k.

Then $\Delta DC = p_k' - p_k = x - p_k > 0$, $\Delta OPT = 0$.

For ϕ_1:

Since the request is at x, then from event a, OPT already has some server q_n at the request. So $q_n = x$.

We claim that in a minimum cost matching before event b, p_k is matched to $q_n = x$.

If not, suppose p_k is matched to some $q_m \neq x$, and some other p_j is matched to q_n. Then we have the below ordering: $p_j \le p_k \le x = q_n \le q_m$.

In this case, in ϕ_1 we have the terms:

$$|p_j - q_n| = q_n - p_j \quad \text{and}$$

$$|p_k - q_m| = q_m - p_k$$

And notice that $q_n - p_j + q_m - p_k = q_n - p_k + q_m - p_j$

$$= |q_n - p_k| + |q_m - p_j|$$

Which means that if we instead map p_k to q_n and map p_j to q_m, it is still a minimum cost matching.

So we can always let p_k match to q_n before event b.

Then before DC moves, the contribution of $k \cdot |p_k - q_n|$ to ϕ_1 is $k \cdot |p_k - q_n| = k \cdot (x - p_k) = k \cdot \Delta DC$.

After DC moves, the contribution of $k \cdot |p_k' - q_n|$ to ϕ_1 is $k \cdot |x - x| = 0$.

So in this case, $\Delta\phi_1 = -k \cdot \Delta DC$.

For ϕ_2, only the terms including p_k change.

So for each $j < k$,

$$|p_k' - p_j| = |(p_k + \Delta DC) - p_j| = |p_k - p_j| + \Delta DC.$$

There are $k-1$ such pairs, so

$$\Delta\phi_2 = (k-1)\Delta DC$$

Combine with first row (event a):

$$\Delta DC = 0 + \Delta DC = \Delta DC \; ; \quad \Delta OPT = \Delta OPT + 0 = \Delta OPT \; ;$$

$$\Delta\phi_1 \le k \cdot \Delta OPT - k \cdot \Delta DC \; ; \quad \Delta\phi_2 = 0 + (k-1)\Delta DC$$

We get $\Delta DC + \Delta\phi_1 + \Delta\phi_2 \le k \cdot \Delta OPT$.

Case 2. DC moves the leftmost server to the request $x < p_1$.

ΔDC is exactly the distance moved by p_1.

Then $\Delta DC = p_1 - p_1' = p_1 - x > 0$, $\Delta OPT = 0$.

For ϕ_1,

OPT also already has a server q_n at request x, so $q_n = x$

Using similar arguments, we claim that in a minimum cost matching before event b, p_1 is matched to $q_n = x$.

If not, we can use the same argument to find a minimum cost matching with p_1 matched to q_n. Then before DC moves, the contribution of $k \cdot |p_1 - q_n| = k \cdot (p_1 - x) = k \cdot \Delta DC$.

and after DC moves, $k \cdot |p_1' - q_n| = k \cdot (x - x) = 0$.

So $\Delta \phi_1 = -k \cdot \Delta DC$.

For ϕ_2. Similar to case 1, only the terms involving p_1 change.

for each $j > 1$,

$$|p_1' - p_j| = |(p_1 + \Delta DC) - p_j| = |p_1 - p_j| + \Delta DC.$$

there are $k-1$ such pairs,

So $\Delta \phi_2 = (k-1) \Delta DC$.

Combine with event a:

$\Delta DC = 0 + \Delta DC = \Delta DC$; $\Delta OPT = \Delta OPT + 0 = \Delta OPT$;

$\Delta \phi_1 \le k \cdot \Delta OPT - k \cdot \Delta DC$; $\Delta \phi_2 = 0 + (k-1) \Delta DC$

we get $\Delta DC + \Delta \phi_1 + \Delta \phi_2 \le k \cdot \Delta OPT$.

Row 2 verified.

Row 3.

The request x lies in between two consecutive servers of DC, let them be p_a and p_b, then $p_a \le x \le p_b$.

DC moves the two servers towards x at the same speed until one reaches x. Let the distance each moves be $d > 0$.

Then $P_a' = P_a + d$, $P_b' = P_b - d$,

And $\Delta DC = |P_a' - P_a| + |P_b' - P_b| = 2d$.

OPT does not move, so $\Delta OPT = 0$.

For ϕ_1,

Without loss of generality, let P_a be the server that meets the request x. So $P_a' = x$.

before event b, there is an OPT server at request x, call it $q_n = x$.

Case 1. Suppose that before event b, P_a is matched to q_n and P_b is matched to some OPT server $q_m \geq q_n$.

Then $|P_a - q_n| = d$. and $|P_a' - q_n| = 0$.

$|P_b' - q_m| = |(P_b - d) - q_m| \leq |P_b - q_m| + d$

$$\begin{aligned} \text{So } \Delta\phi_1 &= \left(|P_a' - q_n| - |P_a - q_n|\right) + \left(|P_b' - q_m| - |P_b - q_m|\right) \\ &\leq (0 - d) + \left(|P_b - q_m| + d - |P_b - q_m|\right) \\ &= -d + d \\ &= 0. \end{aligned}$$

Case 2. Suppose that before event b, P_a is matched to some OPT server q_m satisfying $q_m \leq q_n$. Then P_b must be matched to some OPT server q_r satisfying $q_m \leq q_r \leq q_n = x$.

$$\begin{aligned} \text{Then } |P_a' - q_m| &= |(P_a + d) - q_m| \\ &\leq |P_a - q_m| + d \end{aligned}$$

And note that $P_b \geq P_b' \geq q_n = x \geq q_r$ by assumption.

So $|p_b' - q_r| = p_b' - q_r = p_b - d - q_r$

$= |p_b - q_r| - d$

So $\Delta\phi_1 = (|p_a' - q_m| - |p_a - q_m|) + (|p_b' - q_r| - |p_b - q_r|)$

$\leq |p_a - q_m| + d - |p_a - q_m| + |p_b - q_r| - d - |p_b - q_r|$

$= d - d$

$= 0.$

Case 3. Suppose that before event b, p_a is matched to some OPT server q_m satisfying $q_m \geq q_n = x \geq p_a$.

Then p_b must be matched to some OPT server q_r satisfying $q_r \geq q_m \geq q_n = x = p_a' \geq p_a$.

Then $|p_a' - q_m| = q_m - p_a' = q_m - p_a - d$

$= |p_a - q_m| - d$

And $|p_b' - q_r| = |(p_b - d) - q_r|$

$\leq |p_b - q_r| + d$

So $\Delta\phi_1 = (|p_a' - q_m| - |p_a - q_m|) + (|p_b' - q_r| - |p_b - q_r|)$

$\leq -d + d$

$= 0.$

Hence, in all three cases, $\Delta\phi_1 \leq 0$.

For ϕ_2,

First consider servers p_i where $i < a$. so $p_i \leq p_a \leq p_a'$

Then $\sum_{i<a} |p_a' - p_i| = \sum_{i<a} (p_a + d - p_i)$

$= \sum_{i<a} (|p_a - p_i| + d)$

And note that $p_i \le p_a \le p_a' \le p_b' \le p_b$ by assumption.

Then $\sum_{i<a} |p_b' - p_i| = \sum_{i<a} (p_b - d - p_i)$

$$= \sum_{i<a} (|p_b - p_i| - d).$$

So we get $\sum_{i<a} (|p_a' - p_i| + |p_b' - p_i|)$

$$= \sum_{i<a} (|p_a - p_i| + d + |p_b - p_i| - d)$$

$$= \sum_{i<a} (|p_a - p_i| + |p_b - p_i|)$$

So no net change from these terms.

Consider servers p_j where $b<j$. So $p_b' \le p_b \le p_j$.

Then $\sum_{j>b} |p_b' - p_j| = \sum_{j>b} (p_j - p_b + d)$

$$= \sum_{j>b} (|p_b - p_j| + d)$$

and $\sum_{j>b} |p_a' - p_j| = \sum_{j>b} (p_j - p_a - d)$

$$= \sum_{j>b} (|p_a - p_j| - d)$$

So we get $\sum_{j>b} (|p_b' - p_j| + |p_a' - p_j|)$

$$= \sum_{j>b} (|p_b - p_j| + d + |p_a - p_j| - d)$$

$$= \sum_{j>b} (|p_b - p_j| + |p_a - p_j|)$$

So no net change from these terms.

Now by assumption, there is no server between p_a and p_b

So for the pair (p_a, p_b), after event b,

$$|p_a' - p_b'| = p_b' - p_a' = p_b - d - p_a - d$$

$$= |p_a - p_b| - 2d.$$

So the net change from this term is $-2d = -\Delta DC$.

Therefore, $\Delta\phi_2 = -\Delta DC$.

Combine with event a:

$\Delta DC = 0 + \Delta DC$; $\Delta OPT = \Delta OPT + 0$. ;

$\Delta\phi_1 \le k \cdot \Delta OPT + 0$; $\Delta\phi_2 = 0 - \Delta DC$

We get $\Delta DC + \Delta\phi_1 + \Delta\phi_2 \le k \cdot \Delta OPT$

Row 3 verified.

Table 5.1 verified.

Q2: 5 points Show that if there is a ROM algorithm (either deterministic or randomized) achieving competitive ratio ρ for the unweighted bipartite matching problem (BMM), then there is a *greedy* ROM algorithm (respectively, deterministic or randomized) achieving competitive ratio at least ρ. That is, a ROM algorithm for the BMM problem can always match an online vertex if one of its neighbours is not yet matched.

1. Deterministic case:

Claim.

If there is a deterministic ROM algorithm achieving competitive ratio ρ for BMM, then there is a greedy deterministic ROM algorithm achieving competitive ratio at least ρ.

Proof.

Let ALG be a deterministic ROM algorithm with competitive ratio ρ. If ALG is greedy, the claim is true. So we consider when ALG is non-greedy.

Take an arbitrary graph G, and an arbitrary arrival order U of the online nodes. When we run ALG on this instance, we get a sequence of decisions and a final matching M. Since ALG is non-greedy. Then in the sequence of decisions, there is a state where the online vertex u has at least one free neighbor $v \in V$. And ALG chose to leave u unmatched. We find the first appearance of such state.

We define algorithm ALG' such that:

1. ALG' behaves the same as ALG up to this state.
2. At this state, ALG' matches u to v.

3. After this state, ALG' behaves the same as ALG.

except:

if later an online vertex u' comes, and ALG would have matched u' to v, then ALG' simply leave u' unmatched.

After ALG' finishes, it gives a matching M'.

we compare the objective values:

if in M (matching from ALG), v is matched to some online vertex w, then $|M'| = |M|$. this because except for the matching involving v, M has w matched to v, and has u unmatched. Meanwhile M' has u matched to v, and has w unmatched.

if in M, v is not matched to any vertex.

Then $|M'| = |M| + 1$. This is because ALG leaves both u and v unmatched, but ALG' matches u and v. That is, $M' = \{(u,v)\} \cup M$.

In either case we have $|M'| \geq |M|$.

Then ALG' has competitive ratio at least

$$\rho' = \frac{|M'|}{|M_{OPT}|} \geq \frac{|M|}{|M_{OPT}|} = \rho.$$

So for ALG', if it is still not greedy, we can perform the same process as above, and so on.

Because there are only finitely many relevant states for a finite graph with finite input sequence, so after finitely many such modifications, we can obtain a deterministic ROM algorithm ALG_{greedy} that:

- is greedy.
- achieves at least the same objective value as the version before the modification for every modification.

Therefore ALG_{greedy} has competitive ratio at least ρ. $\square$

2. Randomized case:

Claim.

If there is a randomized ROM algorithm achieving competitive ratio ρ for BMM, then there is a greedy randomized ROM algorithm achieving competitive ratio at least ρ.

Proof.

Let RAN be an arbitrary randomized ROM algorithm with competitive ratio ρ. And RAN has a finite support P_{RAN}. For any deterministic ROM algorithm $ALG_p \in P_{RAN}$, by the previous argument for deterministic algorithms, we can build a greedy deterministic ROM algorithm ALG_p' that always perform at least as good as ALG_p.

Then we can build a greedy randomized ROM algorithm RAN'

which draws p with the same distribution with RAN, and then runs ALG_p'.

For every graph G and random arrival order, and for each p, let M be the matching of ALG_p and let M' be the matching of ALG_p'. We have $|M'| \geq |M|$.

So taking the expectation over both the random order and the deterministic algorithms, we have

$$\mathbb{E}[RAN'] \geq \mathbb{E}[RAN]$$

So if RAN competitive ratio ρ, then the greedy randomized algorithm RAN' has competitive ratio at least ρ. $\square$

Q3: 5 points Prove that no deterministic algorithm for the Display Ads problem (as defined in Section 7.6.1) can have an asymptotic competitive ratio better than $\frac{1}{2}$.

Consider the special case:

Every advertiser V_j has capacity $C_j = 1$. All bids are either 0 or 1.

In this case, for advertiser V_j, the set of assigned users S_j has size at most 1, because $C_j = 1$ and we don't need to assign more. Also, free disposal is irrelevant in this case since each advertiser can receive at most one assignment, so no excess items ever need to be discarded.

T_j is either empty or contain that one user. Then W_j is either 0 or 1. So the objective value $\sum_j W_j$ is exactly the size of the matching between users and advertisers.

Hence, this special case is exactly BMM.

For BMM, we know that no deterministic online algorithm can achieve a competitive ratio better than $1/2$.

Now suppose for contradiction that there is a deterministic online algorithm for Display Ads problem with asymptotic competitive algorithm strictly better than $1/2$. Then when running this algorithm on the special case described above, it would be a deterministic online algorithm for BMM that has competitive ratio strictly better than $1/2$.

This contradicts with the lower bound of deterministic competitive ratio for the BAM problem. So the assumption is incorrect.

That is, no deterministic online algorithm for the Display Ads problem can have an asymptotic competitive ratio better than $1/2$. $\square$

Q4: 5 points Prove Lemma 8.2.20. That is, for the RSIC problem, prove that $OPT(\mathbf{x}) \geq \max(span(\mathbf{x}), util(\mathbf{x}))$.

1. Claim. $OPT(x) \geq span(x)$.

Proof.

For an optimal schedule of servers OPT,

at any timestamp when there is an active job, there must be at least one active server, otherwise that job is lost. which shouldn't happen.

So the set of timestamps when some server is active cover entirely the set of timestamps when some job is active.

So $\sum_B d(B) \geq |\bigcup_i [a_i, f_i)| = span(x)$.

We have $OPT(x) = \sum_B d(B) \geq span(x)$.

2. Claim $OPT(x) \geq util(x)$.

Proof.

Let $f(t)$ be the set of jobs active at time t.

Let $\beta(t)$ be the set of servers active at time t.

Since each server has capacity 1 and every job x_i has size s_i, and x_i has to be processed entirely on the same server. So for all t, we have

$$|\beta(t)| \geq \sum_{x_i \in f(t)} s_i$$

Then integrate both sides over the time.

Right-hand side:

$$\int_t \sum_{x_i \in f(t)} s_i \, dt = \sum_{x_i \in \mathcal{X}} \left[s_i \cdot \int_{t \in [a_i, f_i)} dt \right]$$

$$= \sum_{x_i \in \mathcal{X}} \left[s_i d(x_i) \right]$$

$$= \mathrm{util}(\mathcal{X})$$

Left-hand side:

$$\int_t |\beta(t)| \, dt = \sum_B d(B)$$

So $\sum_B d(B) \geq \mathrm{util}(\mathcal{X})$.

We have $\mathrm{OPT}(\mathcal{X}) = \sum_B d(B) \geq \mathrm{util}(\mathcal{X})$.

With $\mathrm{OPT}(\mathcal{X}) \geq \mathrm{span}(\mathcal{X})$ and $\mathrm{OPT}(\mathcal{X}) \geq \mathrm{util}(\mathcal{X})$.

We have proven that

$$\mathrm{OPT}(\mathcal{X}) \geq \max(\mathrm{span}(\mathcal{X}), \mathrm{util}(\mathcal{X})) \qquad \square$$