CS632 Assignments

Qilin Ye

January 29, 2025

Solution to problem 1. By the constraint $\sum_{p\in\mathcal{P}_i} x_p = 1$ for each $i \in [k]$, we naturally turn to a probabilistic rounding scheme where **for each $i \in [k]$, we randomly choose one path $p \in \mathcal{P}_i$ with probability x_p.**
Fix an edge e and let X_e denote the number of times e appears in a path that is chosen in this rounding; then

$$\mathbb{E}X_e = \sum_{p:e\in p} x_p \leqslant C^*.$$

We use the following formulation of Chernoff bound:

> If $X_i \sim \text{Bernoulli}(p_i)$, and $X = \sum_{i=1}^n x_i, \mu = \mathbb{E}X = \sum_{i=1}^n p_i$, then
>
> $$\mathbb{P}(X > (1+\delta)\mu) < \left(e^\delta/(1+\delta)^{1+\delta}\right)^\mu.$$

It follows that

$$\mathbb{P}(X_e \geqslant \alpha \cdot \mathbb{E}X_e) \leqslant \mathbb{P}(X_e \geqslant \alpha C^*) \leqslant \mathbb{P}(X_e \geqslant \alpha) \leqslant \exp(C^*(\delta - (1+\delta)\log(1+\delta))).$$

The first two inequalities follow from $\mathbb{E}X_e \leqslant C^* \leqslant 1$. Our goal is to make the above $\leqslant n^{-3}$ so that a union bound taken over $\leqslant n^2$ edges gives the desired lower bound $1 - 1/\text{Poly}(n)$ on the probability. Let $\delta' = 1 + \delta$. Then the RHS becomes $\exp(C^*((\delta' - 1) - \delta\log\delta))$.
Following the hint we try δ' of form $\mathcal{O}(\log n/\log\log n)$. It can be numerically verified[1]that when $\delta' = 4\log n/\log\log n$ then $\delta' - 1 - \delta\log\delta < -3\log n$. Therefore, with such δ',

$$\mathbb{P}(X_e \geqslant \alpha) \leqslant \exp(-3\log n) = n^{-3},$$

and by union bound,

$$\mathbb{P}(\text{every edge used at most } \mathcal{O}(\log n/\log\log n) \text{ times}) \geqslant 1 - n^2 \cdot n^{-3} = 1 - 1/n.$$

This completes the proof.

Solution to problem 2. First, we impose the additional assumption that $b_{i,j} \leqslant B_i$ for all i, j, as we did in lecture, for no bidder can go over their budget anyways. By doing so the LP simplifies to

$$\max \sum_{\substack{i\in\mathcal{I}\\ j\in\mathcal{J}}} b_{i,j}x_{i,j} \qquad \text{subject to} \qquad \begin{cases} \sum_{i\in\mathcal{I}} x_{i,j} \leqslant 1 & \text{for all items } j \in \mathcal{I} \\ \sum_{j\in\mathcal{J}} b_{i,j}x_{i,j} \leqslant B_i & \text{for all agents } i \in \mathcal{I} \\ x_{i,j} \geqslant 0 & \text{for all } i, j. \end{cases}$$

(a) We first rigorously prove the claims made in lectures: that (i) there exists an extreme point solution whose induced graph is acyclic, and (ii) there exists an "item" node with the desired property.

(i) Suppose for contradiction that $G(x^*)$ has a cycle C. By the bipartiteness of the variables $x^*_{i,j}$, we may label the cycle as $i_1, j_1, i_2, j_2, \cdots, i_k, j_k, i_1$. In particular, $0 < x^* < 1$ on this cycle because every vertex has degree 2. Call edges of form $i \to j$ *forward* edges and the other type *backward*. Two cases:

[1] Ideally, I would like to derive a bound myself, but since we have encountered a few numerical inequalities toward the second half of the semester (e.g. the 0.878-approximation from SDP), I thought leaving the bound as-is would be okay.

- $\prod_{\text{forward}} b_{i,j} = \prod_{\text{backward}} b_{i,j}$. In this case, we want to perturb the values of $x^*_{i,j}$ by increasing forward edges and decreasing backward edges. To ensure the constraints on i, we ensure the two $x^*_{i,j}$ incident on the same i are perturbed by the same magnitude. Likewise, for the two $x^*_{i,j}$'s incident on the same j, the change should be inversely proportional to their respective $b_{i,j}$'s to ensure the constraints on i are preserved.

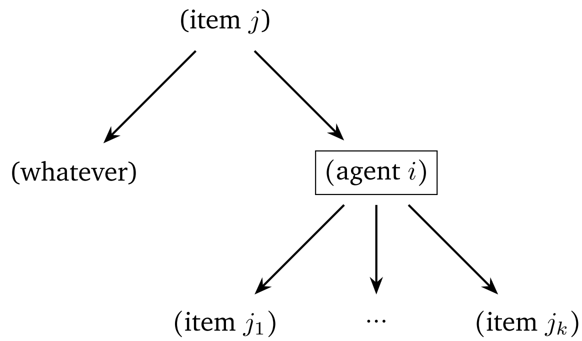
  In other words, if we perturb edges incident on i_1 (these are (j_k, i_1) and (i_1, j_1)) by ϵ_1, then we should perturb the two edges incident on i_2 (these are (j_1, i_2) and (i_2, j_2)) by $\epsilon_1 b(i_1, j_1)/b(j_1, i_2)$. Repeat to compute all ϵ_ℓ for $\ell \in [k]$. Once this is done, incense and decrease forward and backward edges incident on i by $\pm\epsilon_i$, respectively. The constraints are preserved, and so is the objective. We can pick a suitable ϵ_1 to start with to break this cycle, either by making some x^* equal to 1 or 0.

- $\prod_{\text{forward}} b_{i,j} \neq \prod_{\text{backward}} b_{i,j}$ and WLOG let us assume $\neq$ is replaced by $<$. The same argument above would break down when we try to close the loop from (i_k, j_k) to (j_k, i_1). Instead, we replace each x^*_e with some $x^*_e + \epsilon_e$, $\epsilon_e \in \mathbb{R}$, where we want to set ϵ positive for forward edges and negative for backward edges, like before. We want to retain optimality, so we demand that $\epsilon_{e_1} b_{e_1} + \epsilon_{e_2} b_{e_2} = 0$ for any two edges $e_1, e_2 \in C$ incident on the same agent i. But we are forced to relax the conditions on items j: instead of demanding $\epsilon_{e_1} + \epsilon_{e_2} = 0$ for all $e_1, e_2 \in C$ incident on the same item j, we simply require $\epsilon_{e_1} + \epsilon_{e_2} \leqslant 0$. Because the products are related by a strict $<$, there exists some item j such that its two adjacent edges e_1, e_2 satisfy $\epsilon_{e_1} + \epsilon_{e_2} < 0$, strict as well. This means we are decreasing the RHS faster than the LHS, and by choosing ϵ appropriately, we will eventually reduce this to the previous case (=).

(ii) Part (a) shows that E^* is a forest. Let us now narrow our attention down to a specific tree T with edges E. If T contains only one agent then there is nothing to show — by the bipartiteness of x^*, all adjacent items must either be the agent' parent node or leaves. Now assume otherwise.

We are only interested in agent nodes that are connected to item nodes with degree 1, so we discard all item nodes with degree 1, the reason being deleting these nodes will expose the agents as leaves (or even roots). Let the pruned tree be (T', E') where all leaf nodes in T' are now agents. Note all agents are still present. The rank lemma on extreme point solutions imply that T' has $|E'|$ tight constraints and at most 1 non-tight. Because we have $\geqslant 2$ agent nodes, there will be at least one degree 1, tight agent i which is adjacent to some item j. Because item j survives in T', its degree is $\geqslant 2$, so $x^*_{i,j} < 1$. Since i is tight, this implies that we must have pruned some other edge incident on i in the previous step. This gives i the leaf-adjacency property we seek.

(b) The rounding scheme is identical to the one mentioned in lecture: **identify an agent satisfying the descriptions in (a)(ii), assign *all* children item to them, prune the induced graph and adjust the agent's remaining balance, and repeat.**



The proof that this algorithm yields a 2-approximation is identical to the one in lecture. Fix agent i. Say its parent is j and children are $j_1 \cdots, j_k$. The original revenue generated by agent i would be

$$R_i = b_{i,j} x^*_{i,j} + \sum_{\ell=1}^{k} b_{i,j_\ell} x^*_{i,j_\ell}.$$

By LP constraint on i, we immediately know $B_i \geqslant b_{i,j} + \sum_{j_\ell} b_{i,j_\ell} \geqslant R_i$. After assigning all items j_ℓ to agent i, the remaining bid available for parent item j becomes $b'_{i,j} = \min(b_{i,j}, B_i - \sum_{\ell=1}^{k} b_{i,j_\ell})$. The same constraint also implies $b'_{i,j}$ is well-defined (i.e. nonnegative). The new renevue is

$$R'_i = b'_{i,j} x^*_{i,j} + \sum_{\ell=1}^{k} b_{i,j_\ell}.$$

We claim that $R'_i \geqslant R_i/2$. If $\sum_{\ell=1}^{k} b_{i,j_\ell} \geqslant B_i/2 \geqslant R_i/2$ then there is nothing to show. Otherwise, if $\sum_{\ell=1}^{k} b_{i,j_\ell} < B/2$, then

$$b'_{i,j} \geqslant \min(b_{i,j}, B_i/2) \Rightarrow b'_{i,j} x^*_{i,j} \geqslant b_{i,j} x^*_{i,j}/2,$$

and the claim also follows. By applying this assignment argument to every leaf item in ever iteration, we eventually get a 2-approximation, for the revenue gained from each agent is at least $1/2$ of the optimal.

Solution to problem 3. (a) Let V be given and assume it is large. Define item 1 to have value $v_1 = V - 1$ and weight $w_1 = 1/V$. Define item 2 to have value $v_2 = V$ and weight $w_2 = V$. In an integral solution, to accumulate a total value of $\geqslant V$ one must choose item 2, so the cheapest way is to do so, where the total weight is V. On the other hand, in a fractional solution, we can fully pick item 1 and pick $1/V$ portion of item 2. This yields a total value of $(V - 1) + V \cdot 1/V = V$, meeting the threshold, while the total weight is $1/V + V \cdot 1/V = 1 + 1/V$. The ratio between V (integral optimal weight) against $1 + 1/V$ (fractional optimal) can be made arbitrarily large.

(b) We replace the constraints with

$$\sum_{i \notin S} \min(v_i, V - v(S)) x_i \geqslant \max(V - v(S), 0) \qquad \text{for all } S \subset \{1, \cdots, k\}.$$

The advantage of doing this, in particular using $\min(v_i, V - v(S))$ on the LHS is to specifically prevent the unfavorable event in (a) from happening.

(c) With the updated constraints, setting $S = \varnothing$ gives $(V-1)x_1 + Vx_2 \geqslant V$, and setting $S = \{1\}$ gives $x_2 \geqslant 1$. The other two subsets, $\{2\}$ and $\{1,2\}$ do not yield any additional constraints. Based on these, the fractional and integer optimal solutions are identical as they both pick item 2 and nothing else. Here, the integrality gap reduces to 1.

(d) Let x be the fractional optimal of the enhanced LP and let $\tilde{x}_i = \min\{2x_i, 1\}$ as described. Clearly, this yields a feasible solution, since $\tilde{x}_i \geqslant x_i$, so

$$\min_{i \notin S}(v_i, V - v(S))\tilde{x}_i \geqslant \min_{i \notin S} \min(v_i, V - v(S))x_i \geqslant \max(V - v(S), 0).$$

Our goal is to construct integer-valued solutions $y^{(j)}$ that recovers $\tilde{x}$ via a convex combination. We let $\mathcal{L}$ denote the set of **large** items: those whose $x_i \geqslant 1/2$ so $\tilde{x}_i = 1$. It follows that all $y_i^{(j)} = 1$ for the integer-valued solutions can only take values 0 or 1.

More interesting is $\mathcal{S}$, the set of **small** items where $x_i < 1/2$ so $\tilde{x}_i < 1$. We let M be sufficiently large so $m\tilde{x}_i \in \mathbb{Z}$ for all small items i. This is doable since the $\tilde{x}_i$'s are assumed to be rational.

In short, we construct M integer-valued solutions, where given small item i, we choose item i in $M\tilde{x}_i$ integer-valued solutions and discard it in the rest, achieving an average of $\tilde{x}_i$. The round-robin approach goes as follows.

- Sort small items in decreasing values of v_i.
- Let the first $M\tilde{x}_1$ solutions include item 1. Then let the next $M\tilde{x}_2$ solutions include item 2. Whenever we reach the last solution, take everything modulo M and repeat, so it goes on forever, until we are done assigning all small items.

The power of round-robin on these sorted small items is that the distribution of weights is fairly uniform. Let $S \subset \{1, \cdots, k\}$ be given.

Observe that if a solution z satisfies the constraint with respect to $\mathcal{S}$, then $\tilde{z}$ that is set to equal z on $\mathcal{S}$ and takes 1 on all items in $\mathcal{L}$ satisfies any constraint S. To see this, write S^c as $(S^c \cap \mathcal{S}) \cup (S^c \cap \mathcal{L})$. Then, by the feasibility of x (the fractional optimal for LP),

$$\begin{aligned}\sum_{i \notin S} \min(v_i, V - v(S))\tilde{z}_i &= \sum_{i \in S^c \cap \mathcal{S}} \min(v_i, V - v(S)) \cdot z_i + \sum_{i \in S^c \cap \mathcal{L}} \min(v_i, V - v(S)) \\ &\geqslant \max(V - v(S \cup \mathcal{L}), 0) + \max(V - v(S \cup \mathcal{S}), 0) \\ &\geqslant \max(V - v(S), 0). \qquad (*)\end{aligned}$$

Therefore, to ensure a solution is overall feasible, it suffices to check (i) that the solution picks everything from $\mathcal{L}$, and (ii) it satisfies the constraint specifically for $\mathcal{S}$. Our goal is to show that all solutions obtained via round robin satisfies this constraint. **While this is not an intended separation oracle for (b), it posts a set of sufficient conditions to make a solution feasible.**

Observe that by the nature of round robin, the number of items picked by each $y^{(j)}$ must be identical or differ by at most 1:

$$\max_{j \geqslant 1} |\{i \in \mathcal{S} : y_i^{(j)} = 1\}| - \min_{j \geqslant 1} |\{i \in \mathcal{S} : y_i^{(j)} = 1\}| \leqslant 1.$$

Define

$$v(y^{(j)}) = \sum_{i\in\mathcal{S}} \min(v_i, V - v(\mathcal{S}))\mathbf{1}[y_i^{(j)} = 1]$$

the "strengthened" total value of items chosen by $y^{(j)}$ among $\mathcal{S}$. Because items are sorted in decreasing order of value, we know the earlier solutions accumulate more values, so $v(y^{(j)}) \geqslant v(y^{(j+1)})$. Thus it suffices to consider the disparity between $y^{(1)}$ and $y^{(M)}$ (the last one).

By round robin, $y^{(1)}$ can contain at most 1 extra element than $y^{(M)}$, and we assume so. We pair the 2nd element chosen by $y^{(1)}$ with the *first* of $y^{(M)}$, and likewise, the i^{th} element of $y^{(1)}$ with the $(i-1)^{\text{th}}$ element of $y^{(M)}$. This way, we see that in each pair, the value of $y^{(1)}$ is in fact smaller than that of $y^{(M)}$, and so $v(y^{(1)}) - v(y^{(M)}) \leqslant$ the value of the first element chosen by $y^{(1)}$, namely, $\min(v_{\max}, V - v(\mathcal{S}))$ where $v_{\max} = \max\{v_i : i \in \mathcal{S}\}$ corresponds to the most valuable item in $\mathcal{S}$, which by construction is included in $y^{(1)}$. Certainly, $\min(v_{\max}, V - v(\mathcal{S})) \leqslant V - v(\mathcal{S}) \leqslant \max(V - v(\mathcal{S}), 0)$, so

$$\max_{j_1,j_2} |v(y^{(j_1)}) - v(y^{(j_2)})| \leqslant \max(V - v(S), 0). \tag{**}$$

Now suppose one of these solutions $y^{(j)}$ is infeasible. In particular, by the cahracterization in (*), $y^{(j)}$ must have violated the constraint for $\mathcal{S}$, so $\sum_{i\notin\mathcal{S}} \min(v_i, V - v(S))y_i^{(j)} < \max(V - v(S), 0)$. By (**) this implies that all other solutions' $v()$ values must also be bounded by 2 times the RHS. Summing them over,

$$\sum_{j=1}^{M} v(y^j) = \sum_{j=1}^{M}\sum_{i\in\mathcal{S}} \min(v_i, V - v(\mathcal{S}))y_i^{(j)} < 2M \max(V - v(\mathcal{S}), 0).$$

On the other hand, $\sum_{j=1}^{M} y_i^{(j)}$ is also $M\tilde{x}_i$ by definition. Using $2x_i = \tilde{x}_i$ on $\mathcal{S}$, we further have

$$2M \sum_{i\in\mathcal{S}} \min(v_i, V - v(S))x_i = M \sum_{i\in} \min(v_i, V - v(\mathcal{S}))\tilde{x}_i < 2M \max(V - v(\mathcal{S}), 0).$$

We see that x fails to meet the constraint w.r.t. $\mathcal{S}$, contradiction!

Therefore, all $y^{(j)}$ are feasible on $\mathcal{S}$, and when combined with all items in $\mathcal{L}$, they become feasible solutions for the strengthened min-weight knapsack LP. We have shown $\tilde{x} = M^{-1}\sum_{j=1}^{M} y^{(j)} \leqslant 2x$, so $\tilde{x}$ is indeed a 2-approixmation, as claimed.

Solution to problem 4. (a) This follows directly from the embedding theorem and the fact that a composition of $\operatorname{polylog}(n)$ with a $\mathcal{O}(\log(n))$ transformation is also $\operatorname{polylog}(n)$.

(b) Let the root of the tree be r. We are itnerested in finding a subtree such that for all $i \in [k]$, there exists a vertex $v \in S_i$ that connects r to v. In the language of LP, this can be formulated as

$$\min_{e\in E} \sum_{e\in E} w_e x_e \qquad \text{subject to} \qquad \begin{cases} \sum_{e\in\delta(S)} x_e \geqslant 1 & \text{for all } S \subset V \text{ where } r \in S \text{ and } S \cap S_i = \varnothing \text{ for some } i \\ 0 \leqslant x_e \leqslant 1 & \text{for all } e \in E. \end{cases}$$

In other words, any cut that completely separates r from some S_i need to have value of at least 1.

There are exponential number of constraints, so we use a separation oracle. For each $i \in [k]$, an imaginary sink t_i adjacent to all $v \in S_i$, with edge weights all set to 1. The min-cut from r to t_i is < 1 if and only if the original constraint corresponding to (r, S_i) is violated.

(c) Consider a deep tree with $2n+1$ nodes and height h: one root r, two leaves u, v, and two disjoint paths of length n from $r \to u$ and $r \to v$. Let $S_1 = \{u, v\}$. An optimal solution would assign $x_e = 1/2$ for each edge, but a randomized algorithm rounding $\mathbb{P}(x_e \text{ chosen}) = x_e$ will only yield a feasible solution with probability $\leqslant (1/2)^{n-1}$, for a full path from r to u or v needs to be entirely selected.

(d) (i) Let the output of this rounding scheme be ALG. We first show that $\mathbb{P}(e \in \mathrm{ALG}) = x_e$ in this matter still; the only difference is that the events of edges being selected are no longer independent: if e has ancestors $v_k, \cdots, v_1, r$, then

$$\mathbb{P}(x_e \in \mathrm{ALG}) = \frac{x_e}{x_{v_k}} \cdot \frac{x_{v_k}}{x_{v_{k-1}}} \cdot \cdots \cdot \frac{x_{v_1}}{x_r} \cdot x_r = x_e.$$

Therefore, the expected cost of $\mathrm{ALG} = \sum_{e \in E} c_e \mathbb{P}(e \in \mathrm{ALG}) = \sum_{e \in E} c_e x_e =$ the optimal LP solution.

(ii) I am aware that the GKR paper on the Group Steiner Tree problem gives a detailed proof of this claim via Janson's inequality, but even after digesting their proof, I still believe this is far beyond the scope of our class. While I'd *love* to have extra credit, I'm not worthy of it. :)

(iii) We repeat the GKR rounding for a total of $\Theta(\mathrm{poly}(h) \cdot \mathrm{polylog}(n) \cdot \log k)$ times, and take their union and call it SOL. Then, with appropriate coefficients we may remove the big Θ, resulting in

$$\begin{aligned} \mathbb{P}(\text{SOL is not feasible}) &\leqslant \left(1 - \frac{1}{\mathrm{poly}(h)\mathrm{polylog}(n)}\right)^{\Theta(\mathrm{poly}(h) \cdot \mathrm{polylog}(n) \cdot \log k)} \\ &\leqslant \exp\left(-\frac{\Theta(\mathrm{poly}(h) \cdot \mathrm{polylog}(n) \cdot \log k)}{\mathrm{poly}(h) \cdot \mathrm{polylog}(n)}\right) = e^{-\log k} = \frac{1}{k}. \end{aligned}$$

With probability $1 - 1/k$, our output is a feasible solution with approximation factor $\Theta(\mathrm{poly}(h) \cdot \mathrm{polylog}(n) \cdot \log k) = \mathrm{polylog}(n)$, since $h, k \leqslant n$. Otherwise, with low probability, we can brute force compute the shortest path from r to each S_i and append them paths to the solution. Clearly, any such shortest path has cost bounded by OPT (of the GST problem), so the total weight added is $\leqslant k \cdot \mathrm{OPT}$. With probability $1/k$ of this happening, we add 1 to our approximation factor, and we see that $\mathrm{polylog}(n)$ is still preserved.