

10/6 CS 386

Today:

- FPTAS of Knapsack (Fully Polynomial Time Approximation Scheme)
- Approximation protocol for MMS

We've showed that things like an envy-free protocol is hard, along with others. With all these NP-hard problems, we have to give up some thing to achieve faster runtime. We give up on finding the right answer, but to get close enough.

$$\text{Knapsack} = \left\{ \langle (v_1, w_1), \dots, (v_n, w_n), W, V \rangle \mid \exists S \subseteq [n] \text{ s.t. } \sum_{i \in S} v_i \geq V \text{ and } \sum_{i \in S} w_i \leq W \right\}$$

This is NP-hard. It has a linear lower bound.

If we try picking the most valuable item, there might be the following case.

V	$W = 100$	
100	100	If we picked the item with 100 value, we could've picked 1000 items with value 1.
1	0.1	
1	0.1	
1	0.1	
$\vdots$		

If we sort by value per weight, there might be the following case.

V	$W = 10$	
1	0.00001	If we picked (1, 0.00001), we couldn't pick up (100, 10) and get more value.
100	10	

Algorithm: Sort by value per weight.

Let $\{I_1, \dots, I_k\}$ be the list of items that fit into the bag in the order given, that is $\{I_{k+1}\}$ won't fit.

Take the max of $\{I_0, \dots, I_k\}$ and $\{I_{k+1}\}$.

Gets at least $1/2$ the optimal value.

We want the algorithm to:

$\forall \varepsilon > 0, \varepsilon \neq 1$: I will guarantee at least $(1 - \varepsilon)$ the optimal value. But runtime be polynomial in n & $1/\varepsilon$.

1. Let $K = \lceil \frac{n}{\varepsilon} \rceil$
2. $\hat{v}_i = \left\lfloor \frac{v_i}{V_{max}} \cdot K \right\rfloor$, so all values will be integers.
3. Run the DP algorithm with a runtime of $O(n^2V)$ on the new input.
4. Whatever items that algorithm outputs, those are the ones I output.

Let S be the set of items we select.
consider

$$\begin{aligned} \sum_{i \in S} v_i &\geq \sum_{i \in S} \frac{V_{max}}{K} \left\lfloor \frac{v_i}{V_{max}} K \right\rfloor = \frac{V_{max}}{K} \sum_{i \in S} \hat{v}_i \geq \frac{V_{max}}{K} \sum_{i \in OPT} \hat{v}_i \\ &\geq \frac{V_{max}}{K} \sum_{i \in OPT} \left(\frac{v_i}{V_{max}} K - 1 \right) = \frac{V_{max}}{K} \left(\sum_{i \in OPT} \frac{v_i}{V_{max}} K - \sum_{i \in OPT} 1 \right) \\ &= \sum_{i \in OPT} v_i - |OPT| \frac{V_{max}}{K} \geq opt - n \frac{V_{max}}{K} \geq opt - \frac{n}{n/\varepsilon} V_{max} \varepsilon \\ &\geq opt - opt \cdot \varepsilon = (1 - \varepsilon) opt \end{aligned}$$

APX-MMS (n, N, S, M)
 n : # of people N : Every one S : remaining items unassigned M : all items

if $n == 1$:

 give them all of S .

else:

1. have P_i make their best MMS allocations.
2. Find all the people remaining who would take the allocations.
3. Give them their bundles, but sure to leave enough for everyone else.
4. Recurse on what's left.