Homework 5
Colorado CSCI 5454

Sai Siddhi Akhilesh Appala

November 30, 2022

People I studied with for this homework: None

Problem 1

Given Greedy Algorithm
Greedy, that tries to add items in order of the most "bang per buck" (value per weight):

- For each item i, let $a_i = \frac{v_i}{w_i}$.
- Sort the items from largest a_i to smallest.
- Add items in this order until the next one does not fit; then stop.

Part a

$v_1 = 1, w_1 = 1; v_2 = 5, w_2 = 10; v_3 = 9, w_3 = 10; W = 10.1;$
So $a_1 = 1$, $a_2 = 0.5$, $a_3 = 0.9$
Solution:

What is the value of the optimal solution and which items are in it ?
When we run the original algorithm with the specified weights and values, we get the maximum value of 9, which is determined by the weight. (w_3) of weight 10 and value 9.

What is the value of Greedy's solution and which items does it choose?

- By applying the Given Greedy algorithm we need to check the highest ratio of v and w, the first weight as weight(w_1) as it has a high value/weight ratio of $1/1 = 1$.
- now we are left with a weight of 9.1 in the knapsack. As both (w_2) and (w_3) have weights of 10, it's not possible to include them in the list.

- we stop the algorithm as there are no further steps. the max value is 1 and it is obtained by the weight w1 which has the value of 1.

What is the ratio of Greedy to optimal in this example?
the ratio of greedy to optimal as $\frac{1}{9} \approx 0.11$

Part b - using Cheating greedy

First, what is the performance of Cheating Greedy on the example in part (a)?

Cheating Greedy Algorithm: t gets to include the last item in the loop, the one that does not fit, in its solution. So Cheating Greedy actually will violate the weight constraint W, but analyzing it will help us find a good non-cheating algorithm.

- If we add to the above greedy algorithm which already has the weight w_1 which had a total weight of 1, now we need to add a next weight having a higher value/weight ratio, i.e weight w_3.
- now we have w_1 and w_3, the total value we get using this algorithm is $v_1 + v_3 = 10$.
- Therefore the performance which we measure by the ratio of this algorithm's value to the optimal solution which is $\frac{10}{9} \approx 1.11$.

Part C

From the given information, we are going to prove the below points validates to argue that Cheating Greedy's performance is at least as large as Opt

The cheating algorithm obtains the optimal value for the cheating algorithm's weight.

Let the ratio of sorted value/weight list in sorted order be $A = \{a_1, a_2, ...a_n\}$ and for our convenience the corresponding weights list and values list of those be $Weights = \{w_1, w_2...w_n\}$ and $Values = \{v_1, v_2...v_n\}$.

The difference is that now the last item can be added even if it is any sorted value/weight of the weight left, which can be put in the knapsack. The goal, therefore, is to select values $A = \{a_1, a_2, ...a_n\}$ to maximize a1v1 + a2v2 + · · · + anvn, subjected to the constraint of $a1v1 + a2v2 + \ + anvn \leq W$

proof of correctness:

- Assume towards contradiction that there is an instance of cheating greedy knapsack such that the solution of this algorithm CGA(Cheating greedy algorithm) is not optimal.

- Let OPT denote the optimal solution. Without loss of generality, assume that items are sorted in decreasing order by vi/wi, and that no two items have the same values.

- Let CGA = p1, p2, . . . , pn denote the sequence of decision (last value taken) made by our cheating greedy algorithm and OPT = q1, q2, . . . , qn denote that in Optimal one .

- , Therefore, by assumption we have $\Sigma_{i=1}^{n} pivi \leq \Sigma_{i=1}^{n} qivi$ Let i be there first index at which $pi \neq qi$. By the design of our algorithm, it must be that $pi > qi$. By the optimality of OPT, there must exist an item j ¿ I such that pj ¡ qj.

- Consider a new solution $q' = q'1, q'2, . . . , q'n$ where $q'k = qk$ for all $k \neq i$, j. q' will take a little more of item i and a little less of item j compared to OPT. let that be q'i = qi + e$q'nj = qje(wi/wj)$.

- The total weight required for the last item will be added, but the total value strictly increases. i.e by using the Cheating greedy algorithm value is greater than the optimized one. i.e cheating greedy value $\geq optimizedvalue$.

- That q' is a valid and better solution than OP T is a contradiction. Hence, the solution from our algorithm is optimal.

Pard d

We will now analyze the careful greedy stated earlier and apply it to part a. First, we have the maximum weight w1 ratio. So we have the current value V as 1 and then the next max ratio is for v3. By adding them it will be more than the knapsack. But now we will check element 9 and value 1. As the new one has more value, we can say that the maximum value obtained through careful greedy is 9. and previously we have 9 as the optimal solution. The performance of careful greedy is 9/9 = 1

part E

Now we are gonna compare the performance of the careful greedy with optimal. Now we have 3 algorithm types, cheating algorithm, optimal, and careful greedy.

Careful Greedy algorithm follows an approach of greedy and stops at the weight that exceeds the weight and makes a comparison step as follows.

- Let V be the total value of the items taken by optimal Greedy and after fitting items into the set, let ith be the first item that greedy cannot fit.

- The Careful Greedy solution is just item ith, if $v_i \geq V$, otherwise, the optimal Greedy solution
- $V_{Carefulgreedy} = \max(v_i, V)$ which is the V_{max} here

The difference between the cheating greedy and careful greedy is that we add the i^{th} element. So we know that the cheating algorithm will give a result as $V + V_i$. Let the value from the cheating algorithm be $V_{cheating}$. So,

$$V_{cheating} = V_{total} + V_i.$$

From above we can say that $V_{cheating} = V_{Total} + V_i \leq 2 * V_{max}$. So

$$V_{cheating} \leq 2 * V_{max}.$$

By combining the above both conclusions

$$V_{cheating} \leq 2 * V_{max} = 2 * V_{Carefulgreedy}.$$

Now we can say that

$$V_{cheating} \leq 2 * V_{Carefulgreedy}$$

we have proved that $V_{cheating} \geq V_{optimal}$
From this, we can conclude that

$$V_{optimal} \leq V_{cheating}$$

we have

$$V_{optimal} \leq V_{cheating} \leq 2 * V_{Carefulgreedy}$$

From the above we can conclude that

$$V_{optimal} \leq 2 * V_{Carefulgreedy}$$

$$V_{Carefulgreedy} \geq 0.5 * V_{optimal}$$

From the above equations and proof, we can conclude that the careful greedy algorithm has an approximation factor of 0.5. Thus proved.

Problem 2

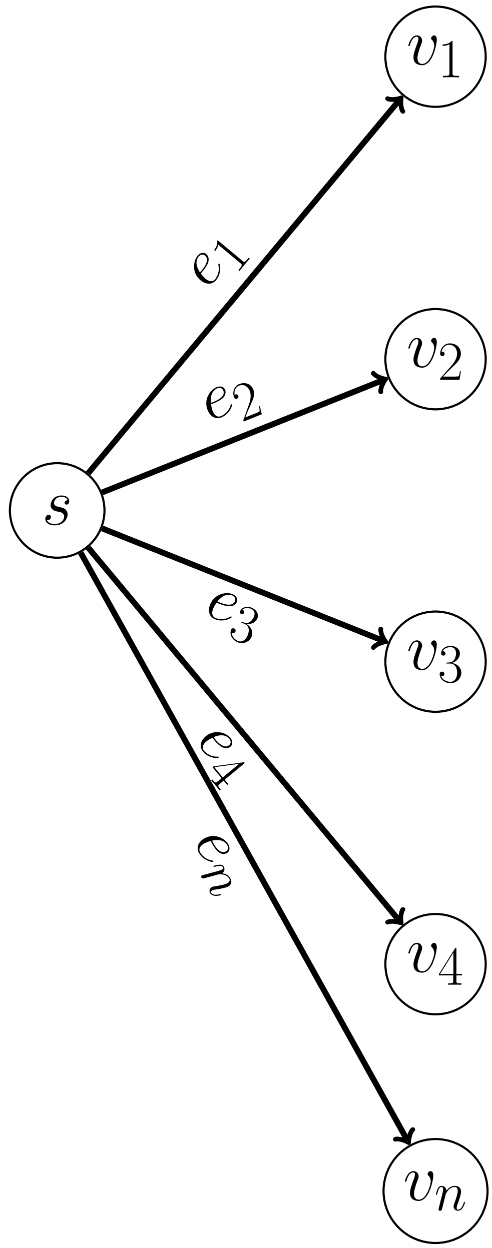
Part a

Consider the following algorithm for vertex cover:

- Initially, the vertex cover W is empty.
- When an edge arrives if neither endpoint is in W, then pick one endpoint and add it to W.

To prove that this algorithm does not guarantee a competitive ratio of C for any C.

Solution: Let's consider the graph network as shown below.



We have the edges of order $e_1, e_2, e_3, e_4...e_n$ for online setup. Based on a given algorithm we keep on choosing the edges. s.

Initially, W_{alg} which is the vertex cover set, is empty. So $W_{alg} = \{\}$. Start iterating the algorithm using the above steps.

- Step 1: e_1 arrives, now add the vertex v_1, since nothing is there in the W_{alg} list. Now $W_{alg} = \{v_1\}$.

- Step 2: e_2 arrives, now add the vertex v_2, since both s, a_2 are not there in the W_{alg} list. Now $W_{alg} = \{v_1, v_2\}$.

- Step n: e_n arrives, we add v_n, both s , v_n are not there in the W_{alg} list.Now $W_{alg} = \{v_1, v_2, v_3, ..., v_n\}$.

So finally we have the vertex cover $W_{alg} = \{v_1, v_2, v_3, ..., v_n\}$, lets say n for 5 vertices then then the$|W_{alg}| = 5$.

But the optimal solution here is $W_{opt} = \{s\}$, $|W_{opt}| = 1$. so we get $\frac{W_{alg}}{W_{opt}} = 5/1 = 5$

we go by the contradiction approach and assume that there exist a C(the optimal ratio) this algorithm guarantees. Which means $W_{alg} \leq C * W_{opt}$. But we show a case where W_{alg} is greater than this value.

Assure that we have a similar graph as above and have $\lceil C \rceil + 1$ vertices instead of just 5. now in this case $W_{alg} > C * W_{opt}$. So our assumption that there exists some C is false and we cant say this is correct for C.

The algorithm is as follows.

- Initially, the vertex covers W_{alg} is empty.
- When an edge arrives if neither endpoint is in W_{alg}, then add both vertices to W_{alg}, else ignore the edge and do nothing.

We prove the competitive ratio for this algorithm as 2.

Part b

Solution:

- we first find the online maximal matching problem from the given
- in the next step we will generate a solution for the vertex cover.

The algorithm is as follows.

- In step 1, the vertex cover W_{alg} is empty.
- When an edge arrives if its endpoints are not in W_{alg}, then add those to W_{alg}, else ignore the edge.

Firstly we note that the above algorithm finds the solution for the matching problem. When we add an edge where its vertices are not seen in the list. so each time we will be adding 2 vertices for every matching edge. So we can conclude that $W_{alg} = 2 * M$ - M matching

$$W_{alg} = 2 * M \tag{1}$$

One more we need to prove is the number of edges in matching is less than or equal to the vertex cover. We say that matching edges are not linked at all. So we can find for every edge, either of its vertexes can be found in the vertex cover. so we conclude that $M \leq W_{opt}$.

$$M \leq W_{opt} \tag{2}$$

By combining the above 2 equations, we can conclude from the above explanation and proofs

$$W_{alg} = 2 * M \leq 2 * W_{opt}$$

Now we have $W_{alg} \leq 2 * W_{opt}$ if we see the upper bound of them. C which is 2 where $\frac{W_{alg}}{W_{opt}} \leq 2$ Therefore the competitive ratio(C) for this algorithm is 2.

$$\frac{W_{alg}}{W_{opt}} \leq 2$$

So the competitive ratio(C) for this algorithm is 2.