Homework 4 Solution

1. *Consider the following problem for n jobs, each one of which takes exactly one minute to complete. At any time $\mathrm{T} = 1,2,3,\cdots$, we can execute exactly one job. Each job i earns a profit of $\mathrm{p_i}$ dollars if and only if it is executed no later than time $\mathrm{d_i}$, where $\mathrm{d_i}$ is given as an input. Assume that $\mathrm{d_i}$ is an integer value. The Problem is to schedule the jobs to maximize the profit.*
   Consider the following greedy strategy: For all the jobs with deadline at 1 minute, schedule the job with the maximum profit. Next, for all the jobs with deadline at 2 minutes or less, pick the job with maximum profit from the remaining unscheduled jobs. And so on. For example, consider $\mathrm{n} = 4$, profits $\mathrm{P} = (50,10,15,30)$ and deadlines $\mathrm{D} = (2,1,2,1)$. The greedy strategy will yield the following solution: job 4 and job 1 to be scheduled for a total of profit of 80 dollars.
   Give a counter example to establish that this greedy strategy does not always work.
   Counter example:
   $P = (50,10,40,30)$, $D = (2,1,2,1)$
   The greedy algorithm will give job 4 and job 1. While the optimal solution should be job 1 and job 3.

2. *Consider the "coin change problem" we discussed in class: Consider a currency system with coins' worth $\mathrm{a_1, a_2, \cdots, a_k}$ cents where $\mathrm{a_1} = 1$. Assume that you are given an unlimited numbers of coins of each type. The input to the problem is an integer M and the objective is to determine the number of coins of each type to make up M cents using the minimum number of coins. Consider a greedy algorithm that takes as many coins as possible from the highest denomination, and repeat this with the next highest one, etc.*
   Prove that this greedy algorithm correctly solves the coin change problem for the case when $\mathrm{a_1} = 1$, $\mathrm{a_2} = 5$, $\mathrm{a_3} = 10$, $\mathrm{a_4} = 25$, and $\mathrm{a_5} = 50$.
   Let x_1, x_2, x_3, x_4, x_5 be the solution generated by greedy algorithm. And y_1, y_2, y_3, y_4, y_5 be an optimal solution.

   Step 1: $x_1 = y_1$
   From the definition of the greedy algorithm, we can obtain that $x_1 = M \mod 5$.
   For any optimal solution, $M = y_1 + 5y_2 + 10y_3 + 25y_4 + 50y_5$. $y_1 = M \mod 5$.
   Thus $x_1 = y_1$

   Step 2: $x_2 = y_2$
   Let $M_1 = M - x_1$.
   From the definition of the greedy algorithm, we can obtain that
   $x_2 = ((M_1 \mod 25) \bmod 10)/5$.
   For any optimal solution, $M_1 = 5y_2 + 10y_3 + 25y_4 + 50y_5$
   $5y_2 + 10y_3 = M_1 \mod 25$
   If $5y_2 + 10y_3 < M_1 \bmod 25$, then we cannot hold the equation $M_1 = 5y_2 + 10y_3 +$

$25y_4 + 50y_5$. If $5y_2 + 10y_3 > M_1 \mod 25$. Then we can always subtract 25 value from a_2 and a_3, and adding to a_4. And this is not an optimal solution as we assumed.
Thus $5y_2 = (M_1 \mod 25) \mod 10$
$y_2 = x_2$

Step 3: $x_3 = y_3$
Let $M_2 = M - x_1 - 5x_2$.
From the definition of the greedy algorithm, we can obtain that:
$10x_3 = M_2 \mod 25$
For any optimal solution, $M_2 = 10y_3 + 25y_4 + 50y_5$
Thus $10y_3 = M_2 \mod 25$
$y_3 = x_3$

Step 4: $x_4 = y_4$
Let $M_3 = M - x_1 - 5x_2 - 10x_3$
From the definition of the greedy algorithm, we can obtain that:
$25x_4 = M_3 \mod 50$
For any optimal solution, $M_3 = 25y_4 + 50y_5$
Thus $25y_4 = M_3 \mod 50$
$y_4 = x_4$

Step 5: $x_5 = y_5$
$M = x_1 + 5x_2 + 10x_3 + 25x_4 + 50x_5 = y_1 + 5y_2 + 10y_3 + 25y_4 + 50y_5$
$x_1 = y_1, x_2 = y_2, x_3 = y_3, x_4 = y_4$
Thus, $x_5 = y_5$

Thus the solution generated by greedy algorithm must be an optimal solution.

3. *A k-coloring og a graph $G = (V, E)$ is a mapping $f: V \to \{1,2,\cdots,k\}$ such that adjacent vertices are mapped out different colors, i.e., no two neighbors in G receive the same color (i.e., same integer).*

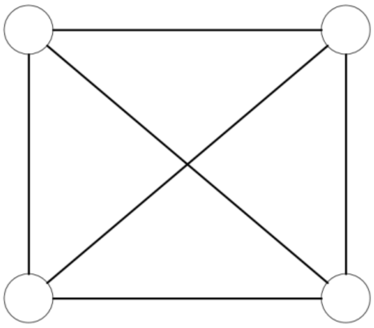
   For $i = 1, \cdots, n$ do

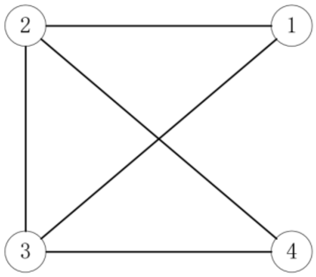
   Color vertex v_i using the smallest available color in $\{1,2,\cdots,\Delta(\mathrm{G})+1\}$.

   (a) *Prove that the following GREEDY algorithm colors the given graph G with at most $\Delta(\mathrm{G})+1$ colors where $\Delta(G)$ denote the maximum degree (number of adjacent vertices) of any node $v \in V$.*

   (b) *Show an example of a graph G that requires $\Delta(G)+1$ colors.*

Let d_i denote the degree of vertex v_i. For any vertex, by applying greedy algorithm we need at most $d_i + 1$ color (treated as all the neighboring vertex are colored). Thus the we at most $\Delta(G) + 1$ color to color the whole graph.





Left figure is an example using at most $\Delta(G) + 1$ color.
Right figure is an example using at most $\Delta(G)$ color.

4. *Consider the knapsack problem discussed in class. Now we have one additional constraint that $x \in \{0,1\}$, i.e., you are not allowed to put a fraction of any object in the knapsack. This problem is called the 0-1 knapsack problem.*
   (a) *Show that the greedy algorithm that consider ratio of $\frac{p_i}{w_i}$ is not an optimal algorithm for this case.*
   (b) *Suppose that the order of the items when sorted by increasing weight is the same as their order when sorted by decreasing profit. The greedy algorithm then finds an optimal solution. Prove the optimality.*

$M = 8, P = (7,8), W = (6,8)$
The greedy algorithm will give x_1 while the optimal solution is x_2

Proof:
Without loss of generality We have additional statement that are:

$$p_1 \geq p_2 \geq \cdots \geq p_n$$
$$w_1 \leq w_2 \leq \cdots \leq w_n$$

Thus we have:

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \cdots \geq \frac{p_n}{w_n}$$

Let $y_1, y_2, \cdots, y_n$ be an arbitrary optimal solution besides greedy and $x_1, x_2, \cdots, x_n$ be greedy solution. Let k denote the least index such that $x_k = 1$. Let j denote the first index such that $y_j \neq x_j$

1) $j > k$
   $x_j = 0, y_j = 1$, Thus total weight of optimal solution exceeds the capacity.
2) $j = k$
   $x_j = 1, y_j = 0$, Thus total profit of optimal solution is less than the total profit of greedy solution. Contradiction.
3) $j < k$
   $x_j = 1, y_j = 0$
   (i) For $i > k, y_i = 0$, total profit of optimal solution is less than the total profit of greedy solution. Contradiction.

(ii) For $i > k, \exists y_i = 1$, total weight of optimal solution exceeds the capacity according to the additional statement.

Thus there exists no such optimal solution besides greedy. Thus the greedy algorithm is optimal.

5. *A source node of a data communication network has n communication lines connected to its destination node. Each line i has a transmission rate r_i representing the number of bits that can be transmitted per second. A data needs to be transmitted with transmission rate at least M bits per second from the source node to its destination node. If a fraction x_i $(0 \le x_i \le 1)$ of line i is used (for example, a fraction x_i of the full band widths of line i is used), the transmission rate through line i becomes $x_i \cdot r_i$ and a cost $c_i \cdot x_i$ is incurred. Assume that the cost function $c_i (1 \le i \le n)$ is given. The objective of the problem is to compute x_i, for $1 \le i \le \mathrm{n}$, such that $\Sigma_{1 \le i \le n} r_i x_i \ge M$ and $\Sigma_{1 \le i \le n} c_i x_i$ is minimized.*
   (a) Describe an outline of a greedy algorithm to solve the problem.
   (b) Prove that your algorithm in part (a) always produces an optimum solution. You should give all the details of your proof.

Sort the lines according to $\frac{r_i}{c_i}$ in decreasing order. Let $l_1, l_2, \cdots, l_n$ denotes the lines after sorted.

$i = 1;$

$x_1, x_2, \cdots, x_n = 0;$

For $i = 1$ to n

    If $l_i < M$

        $M = M - l_i;$

        $x_i = 1;$

        // always trying to add a full line if through put is not filled to lower bound.

    Else

        $x_i = \frac{M}{l_i};$

        // when adding a full line exceed the lower bound then try adding a portion

        // of it to just reach the lower bound.

    End

End

Proof:

Without the loss of generality, we assume $\frac{r_1}{c_1} \ge \frac{r_2}{c_2} \ge \cdots \ge \frac{r_n}{c_n}$. Let $X = (x_1, x_2, \cdots, x_n)$ be the greedy solution. If $x_i = 1, \forall i$, then X is optimal. So, let j be the least index such that $x_j < 1$.

Let $Y = (y_1, y_2, \cdots, y_n)$ be an optimal solution besides greedy. Without the loss of generality, we can assume that $\Sigma_{i=1}^{n} r_i y_i = M$. Let k be the first index such that $x_k \ne y_k$. We next prove that $y_k < x_k$.

1. $k < j$: $x_k = 1$, thus $y_k < x_k$

2. $k = j$:

   If $y_k > x_k$ by reducing y_k to x_k can lower the cost of optimal solution while the transmission rate is still acceptable. Contradiction.

   $y_k \neq x_k$

   Thus $y_k < x_k$

3. $k > j$:

   By cutting y_k down to 0 can lower the cost of optimal solution while the transmission rate is still acceptable. Contradiction.

Now define a new solution $Z = (z_1, z_2, \cdots, z_n)$ such that

a) $z_i = y_i, \forall 1 \leq i < k$;

b) $z_k = x_k$

c) $z_i \leq y_i$,for $k < i \leq n$ such that $r_k(x_k - y_k) = \Sigma_{i=k+1}^{n} r_i(y_i - z_i)$

We then have

$$\sum_{i=1}^{n} c_i z_i = \sum_{i=1}^{n} c_i y_i + (z_k - y_k)c_k - \sum_{i=k+1}^{n} (y_i - z_i)c_i$$

$$= \sum_{i=1}^{n} c_i y_i + (z_k - y_k)r_k\left(\frac{c_k}{r_k}\right) - \sum_{i=k+1}^{n} (y_i - z_i)r_i\left(\frac{c_i}{r_i}\right)$$

$$\leq \sum_{i=1}^{n} c_i y_i + (z_k - y_k)r_k\left(\frac{c_k}{r_k}\right) - \sum_{i=k+1}^{n} (y_i - z_i)r_i\left(\frac{c_k}{r_k}\right)$$

$$\text{since} \frac{c_i}{r_i} \leq \frac{c_k}{r_k} \quad \text{for } i = k+1$$

$$= \sum_{i=1}^{n} c_i y_i + \frac{c_k}{r_k}\left(r_k(z_k - y_k) - \sum_{i=k+1}^{n} (y_i - z_i)\, r_i\right)$$

$$= \sum_{i=1}^{n} c_i y_i$$

Since Y is assumed to be optimal, $\Sigma_{i=1}^{n} c_i z_i = \Sigma_{i=1}^{n} c_i y_i$. By repeating this process, we can transform Y into X. Hence, X must be optimal.