Solutions to Assignment# 4

Liana Yepremyan

1 Nov.12: Text p. 651 problem 1

Solution: (a) One example is the following. Consider the instance $K = 2$ and $W = \{1, 2, 1, 2\}$. The greedy algorithm would load 1 onto the first truck. The second weight is too heavy for the first truck and the greedy algorithm would send it away and load 2 onto the next truck. This truck is now fully loaded and must be sent off. This continues with the third and fourth truck as well. The optimal number of trucks is three. By loading the two weights of 1 onto a single truck and loading the two weights of 2 onto two other trucks we achieve the optimal number of trucks, three.

More generally, suppose $n > 4$ and the weights of the containers are given by the set $W = \{1, K, 1, K, \ldots, 1, K\}$ for any $K > 1$. The greedy algorithm will use n trucks. But in fact we only need need $n/2$ trucks for all containers of weight K and $\lceil n/2K \rceil$ trucks for all weights of 1. Hence in total we need at most $n/2 + \lceil n/2K \rceil \leq n/2 + n/2K + 1 \leq 3n/4 + 1 < n$ trucks.

(b) Suppose there are n items to unload and the weights are $w_1, w_2, \ldots, w_n$, then let N^* be the optimal number of trucks needed. Then, since each truck cannot carry more then K units of load, we get that $\sum_{i=1}^{n} w_i \leq KN^*$ and hence

$$N^* \geq \frac{1}{K} \sum_{i=1}^{n} w_i. \tag{1}$$

Now let N be the number of trucks that the greedy algorithm finds. We prove that it is within a factor two of the minimum possible number, for any set of weights and any value of K. More precisely we show that

Claim 1.1. $N \leq 2N^*$.

Proof. Let I_j denote the set of items that truck j loads and let W_j be the total weight of the items in I_j, that is $W_j := \sum_{a \in I_j} w(a)$. By analyzing the greedy algorithm we can conclude that the following holds for any $j > 1$,

$$W_j + W_{j-1} > K.$$

On the other hand, we have that

$$\sum_{j=1}^{N} W_j = \sum_{i=1}^{n} w_i. \tag{2}$$

Suppose $N = 2m$, for some m, then

$$\sum_{j=1}^{N} W_j = \sum_{j=1}^{m} (W_{2j} + W_{2j-1}) > Km.$$

If $N = 2m + 1$ for some m, then

$$\sum_{j=1}^{N} W_j = \sum_{j=1}^{m} (W_{2j} + W_{2j-1}) + W_{2m+1} > Km.$$

Hence in general we get that $\sum_{j=1}^{N} W_j > \frac{1}{2}K(N-1)$. And now by inequalities (1) and (2) we get that

$$\frac{1}{2}K(N-1) < \sum_{j=1}^{N} W_j = \sum_{i=1}^{n} w_i \leq KN^*,$$

from where it follows that $N - 1 < 2N^*$ or equivalently $N \leq 2N^*$. $\square$

2 November 14, problem

Draw a non-bipartite graph on 6 *vertices with* 7 *edges with no perfect matching. Write down the integer linear programs for the maximum matching and minimum vertex cover problems and the LP relaxations. Find the maximum matching, min vertex cover and solve the LP relaxations either by inspection or by using lp-solve. Also compute the rounded solution from the vertex cover LP. Verify that the size of maximum matching $\leq$ max fractional matching = min fractional vertex cover $\leq$ min vertex cover $\leq$ rounded vertex cover $\leq$ 2$*$ min vertex cover.*

Solution: Consider the graph G in Figure 2 and the given assignment of the variables to the vertices and edges. The integer programm for the maximum matching looks like this.

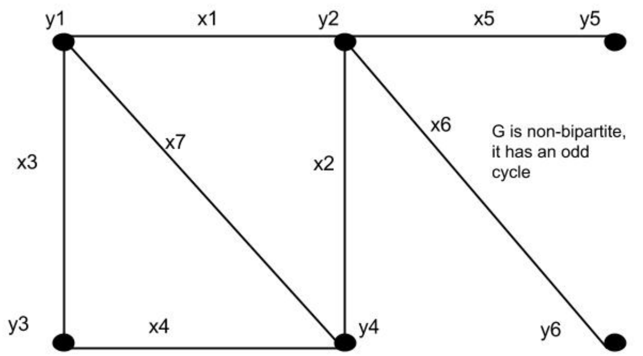
$$\begin{aligned}
\text{maximize } & z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\
\text{subject to } & x_1 + x_3 + x_7 \leq 1 \\
& x_3 + x_4 \leq 1 \\
& x_2 + x_7 + x_4 \leq 1 \\
& x_1 + x_2 + x_5 + x_6 \leq 1 \\
& x_5 \leq 1 \\
& x_6 \leq 1 \\
& x_i \in \{0, 1\}
\end{aligned}$$

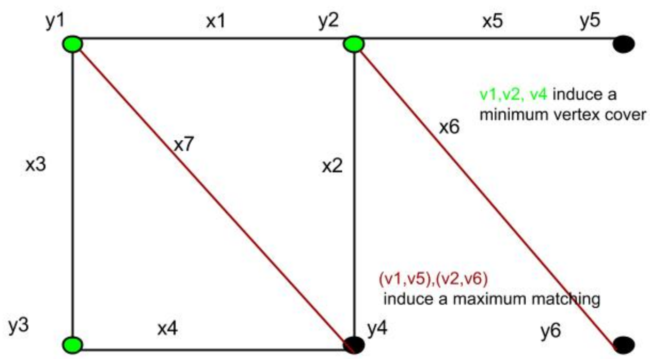
The dual of this problem which corresponds to the minimum vertex cover problem is the following.

$$\begin{aligned}
\text{minimize } & z = y_1 + y_2 + y_3 + y_4 + y_5 + y_6 \\
\text{subject to } & y_1 + y_2 \geq 1 \\
& y_1 + y_3 \geq 1 \\
& y_1 + y_4 \geq 1 \\
& y_2 + y_5 \geq 1 \\
& y_2 + y_6 \geq 1 \\
& y_2 + y_4 \geq 1 \\
& y_4 + y_3 \geq 1 \\
& y_i \in \{0, 1\}
\end{aligned}$$

It is easy to see that the the optimal solution for the first problem is $z^* = 2$(i.e. the maximum matching has size two in G) and for the second one is $z^* = 3$ (i.e. the minimum vertex cover has size three in G).

In the LP relaxations of these two problems, we just let $0 \leq x_i \leq 1$ and $0 \leq y_i \leq 1$. Using lp-solve, one can find out that for LP relaxiation of the first problem the optimal solution has value $z' = 2.5$ (when f.e. $x_3 = x_4 = x_7 = 0.5, x_6 = 1, x_1 = x_2 = x_5 = 0$), and for the second one it is $z' = 2.5$ (when f.e. $y_1 = y_3 = y_4 = 0.5, y_2 = 1, y_5 = y_6 = 0$), while for the rounded problem it is $z'' = 4$ (when f.e. $y_1 = y_3 = y_4 = 1, y_2 = 1, y_5 = y_6 = 0$). Hence one can see that all the inequalities are satisfied (i.e. $2 \leq 2.5 = 2.5 \leq 3 \leq 4 \leq 2 * 3$).





Remark: Thanks to Cleo Kesidis for a nice example, I like it so decided to include in the solution set.

3 November 19, problem

The vertex cover algorithm in section 10.1 relies on the fact that for any edge uv, at least one of u or v must be in a minimum vertex cover. Show that a similar result does not apply to independent set : i.e, find a graph and edge uv such that neither u nor v is in the maximum independent set. Use this or similar example to show that the algorithm in Section 10.1 does not adapt to finding an independent set of size k.

Solution: As an example take a triangle K_3, that is $V(K_3) = \{u, v, w\}$ and $E(K_3) = \{uv, vw, wu\}$. It is easy to see that all maximum independent sets in this graph are singletones. Take any maximum independent set, say $\{u\}$, then the edge vw has no vertices in this independent set.

This is the main reason why the algorithm to find a minimum vertex cover described in the section 10.1 cannnot be adapted to find a maximum independent set. Indeed, we would like to have something similar to the claim (10.3), that says the following.

Claim 3.1. *Let $e = (u, v)$ be any edges of G. The graph G has a vertex cover of size at most k if and only if at least one of the graphs $G - \{u\}$ and $G - \{v\}$ has a vertex cover of size at most $k - 1$.*

We would like to say that if for some edge $e = (u, v)$ we have that $G - e$ has a maximum independent set of size at least $k - 1$, then we can extend it to a maximum independent set of size k in G by adding either u or v to it. In fact, this is wrong, since u and v might be adjacent to some vertex in the maximum independent set in $G - e$ as our previous example shows. Therefore, sometimes we have to look for a maximum independent set of size k in the graph $G - e$, which is the same problem as our initial one. This observation shows that the algorithm in 10.1 cannot be adapted to solve maximum independent set problem.

4 Nov. 21, Problem.

Let $G = (V, E)$ be any connected graph with $n = |V|$. Construct a TSP on n cities as follows. The weight w_{ij} is the the number of edges in a shortest path in G between i and j. Show that the bounds on the two heuristics studied today apply to this TSP.

Solution: In the class we showed that if the distances between cities satisfy the metric properties, (that is $d_{i,j} \leq d_{i,k} + d_{k,j}$ for any distinct i, k, j), then we can find a 2 and 3/2 approximating algorithms for TSP.

To see, why the same heuristics apply to the TSP given above, it is enough to show that in fact the triangle inequality is still satisfied.

Claim 4.1. *For any distinct i, j, k $w_{i,j} \leq w_{i,k} + w_{k,j}$.*

Proof. By definition, $w_{i,j}$ is the number of edges in the shortest path from i to j. Let $P_{i,k}$ be the shortest path from i to k. Note that $w_{i,k} = |P_{i,k}|$. Define $P_{k,j}$ and $P_{i,j}$ similarly. Consider the walk $P^* = P_{i,k} \cup P_{k,j}$. It must contain a subpath from i to j, call it $Q_{i,j}$. Then, we have

$$w_{i,j} = |P_{i,j}| \leq |Q_{i,j}| \leq |P^*| \leq |P_{i,k}| + |P_{k,j}| = w_{i,k} + w_{k,j},$$

and we are done. $\square$