ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Bachelor semester 4-DO

DISCRETE OPTIMIZATION WEEK 3

EXERCISE 1

Consider the following *greedy algorithm* for finding a vertex cover S in a given graph $G = (V, E)$: Set $S = \emptyset$. As long as E is nonempty, add to S a vertex v of largest degree (number of incident edges) in G. Then delete the edges incident to v from G and repeat the process with the resulting graph $G' = (V, E')$.

Show that there is no constant factor k with the property that this algorithm always finds a vertex cover whose size is at most k times as large as the size of a smallest vertex cover.

SOLUTION

We will prove something even stronger, namely the following statement.

Statement.

For every $n \in \mathbb{Z}_{>1}$, there exist a bipartite graph $G_n = (A_n, B_n, E_n)$ such that $|A_n|/|B_n| = n/2$, and such that the optimal vertex cover is B_n while the one found by the greedy algorithm could be A_n. Furthermore, we ask that each vertex of B_n has degree $|B_n|$ (here, the vertices are the elements in $A_n \cup B_n$ and the edges, denoted by E_n, have one point in A_n and one in B_n).

Clearly the proof of this statement is enough to solve the exercise. We proceed by induction, for $k = 2$ a graph with 2 vertices and one edge satisfies the requirements. Hence, let us assume that the claim is true for all $k \leq n$ and we shall prove that the statement holds for $k = n + 1$.

We built the graph $G_{n+1} = (A_{n+1}, B_{n+1}, E_{n+1})$ as follows : make two copies of G_n, say $G_n = (A_n, B_n, E_n)$ and $G'_n = (A'_n, B'_n, E'_n)$ and create $|B_n|$ new vertices, which we denote $a_1, ...a_{|B_n|}$. Add an edge between every a_i and each element of B_n and similarly between every a_i and each element of B'_n. We will let $A_{n+1} = A_n \cup A'_n \cup \{a_1, ...a_{|B_n|}\}$, $B_{n+1} = B_n \cup B'_n$, and the edge set E_{n+1} will be the one given by the union of E_n, E'_n and the new edges we created. Note now that the greedy algorithm could choose first all the new vertices $a_1, ..., a_{|B_n|}$. Following, by the inductions hypothesis, the greedy algorithm could choose all vertices in A_n and in A'_n, hence choosing all vertices in A_{n+1}. The optimal set is B_{n+1}, and it is clear that such vertices have degree $|B_{n+1}|$, now note that by the inductions hypothesis $|A_{n+1}|/|B_{n+1}| = (2|A_n| + |B_n|)/(2|B_n|) = (n + 1)/2$.

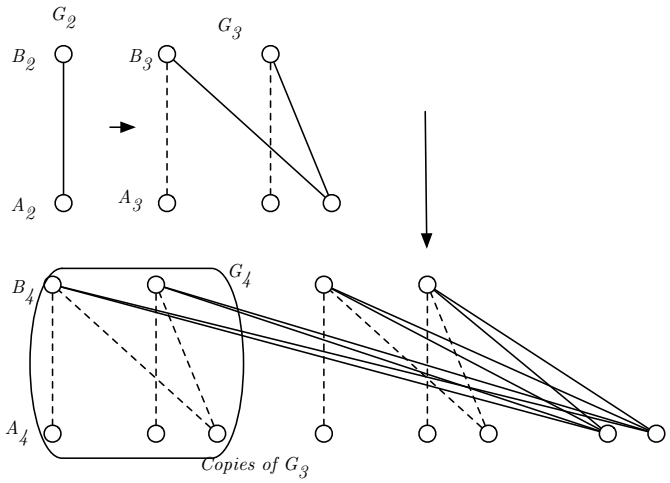

FIG. 1 – The construction of G_2, G_3, G_4. The full lines represent the new edges created while the dotted-lines come from the previous graph.

Note. In the above solution, the greedy algorithm is not forced to choose A_n but it is, nevertheless, a possible outcome. This is because ties arise where the greedy algorithm must choose between vertices with the same degree. It is

possible to construct examples where the greedy algorithm is completely forced to choose a bad solution (regardless of how it manages ties). If instead of adding $|B_n|$ vertices at each step, in the above construction, we would add $|B_n|/2$ vertices at each step, the construction would yield such an example, though with k growing at half of the speed, that is $1/4$ at every iteration (instead of $1/2$). In order for $|B_n|/2$ to be an integer, we would also need a different base for the induction, G_3, from above, would work.

EXERCISE 2

We are given a set R of m red points and a set B of n blue points in the plane. We want to find a non-vertical line in the plane that passes below all red points and above all blue points (the line is also allowed to pass through points). Write a linear system of inequalities (meaning, a system of the form $A\boldsymbol{x} \leq \boldsymbol{b}$) that expresses whether such a line exists or not.

SOLUTION 2

Let $(x_1, y_1), ..., (x_m, y_m)$ be the red points and $(a_1, b_1), ..., (a_n, b_n)$ the blue points. We create two variables q and b, these will denote the line $y = qx + b$. In order for this line to be below (possibly passing through) all red points, the following inequality must be satisfied

$$y_i \geq qx_i + b \Longleftrightarrow qx_i + b \leq y_i$$

Symmetrically, in order for the line to be above all blue points, the following inequality must be satisfied :

$$b_i \leq qa_i + b \Longleftrightarrow -qa_i - b \leq -b_i.$$

There are no other restrictions on the variables. To write it in the regular form $A\boldsymbol{v} \leq \boldsymbol{d}$ (change from usual notation because we have already used the variables x and b), let $\boldsymbol{v} = [q, b]^T$, and $\boldsymbol{d} = [y_1, y_2, ...y_m, -b_1, -b_2, ..., -b_n]^T$ and we have

$$A = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ . & . \\ . & . \\ x_m & 1 \\ -a_1 & -1 \\ -a_2 & -1 \\ . & . \\ . & . \\ -a_m & -1 \end{pmatrix}$$

EXERCISE 3

Consider the following bipartite graph :

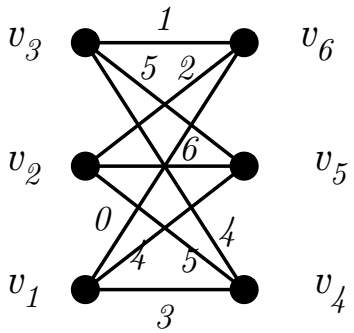

FIG. 2 –

We are interested in finding a maximum-weight perfect matching in this graph.

a) Find a perfect matching of weight 10.

b) Prove that there is no perfect matching of weight larger than 10, as follows : Write the linear program corresponding to the problem, as we learned in class. Multiply the six constraints corresponding to the vertices by 0, 2, 1, 3, 4, and 0, respectively ; add up these constraints ; and interpret the result.

(This is a prelude to the duality theorem that we will learn later in the course.)

SOLUTION 3

a) Matching $(v_1, v_4), (v_2, v_5), (v_3, v_6)$ is a perfect matching of weight 10.

b)Create one variable for each edge, for ease we will let $a_{i,j}$ denote the variable assigned to edge $v_i v_j$. We must have $0 \leq a_{i,j} \leq 1$ for every edge. We will have one equality per vertex.

- For vertex v_1 we have $a_{1,4} + a_{1,5} + a_{1,6} = 1$
- for v_2, $a_{2,4} + a_{2,5} + a_{2,6} = 1$
- for v_3, $a_{3,4} + a_{3,5} + a_{3,6} = 1$
- for v_4, $a_{1,4} + a_{2,4} + a_{3,4} = 1$
- for v_5, $a_{1,5} + a_{2,5} + a_{3,5} = 1$
- for v_6, $a_{1,6} + a_{2,6} + a_{3,6} = 1$

And we will wish to maximize the following function : $3a_{1,4} + 5a_{2,4} + 4a_{3,4} + 4a_{1,5} + 6a_{2,5} + 5a_{3,5} + 0a_{1,6} + 2a_{2,6} + 1a_{3,6}$.

On the other hand, if one sums up as specified in the exercise, we get the following equation

$$3a_{1,4} + 5a_{2,4} + 4a_{3,4} + 4a_{1,5} + 6a_{2,5} + 5a_{3,5} + 0a_{1,6} + 2a_{2,6} + 1a_{3,6} = 10.$$

This shows that the objective function is always 10 , proving that any matching will have weight 10.