The Hungarian Method – Egerváry's algorithm

Combinatorial Optimization – Group K

Class 24

Spring 2023

Now we give an efficient, combinatorial algorithm for the Optimum Assignment Problem that extends (and relies on) the previously seen Augmenting Path Algorithm for the Maximum Bipartite Matching Problem. As we have seen before, the Maximum Weight Bipartite Matching Problem can be reduced to the Optimu Assignment Problem, so the algorithm to be presented below can also be used to efficiently solve this problem.

The main tool of the algorithm is the following lemma that was also proved before. Recall that given a bipartite graph $G = (A, B; E)$ and a weight function $w\colon E \to \mathbb{R}$ on its edges, a *labeling* is an assignment $c\colon (A \cup B) \to \mathbb{R}$ of real values to the vertices such that $c(a) + c(b) \geq w(e)$ holds for every edge $e = \{a, b\}$. Furthermore, the edge $e = \{a, b\}$ is *binding* with respect to a labeling c if $c(a) + c(b) = w(e)$.

Lemma. Assume that a perfect matching M and a labeling c are given in the bipartite graph G such that every edge of M is binding with respect to c. Then M is a maximum weight perfect matching.

The basic idea of the algorithm is the following. It maintains a (not necessarily perfect) matching M and a labeling c such that, throughout the whole procedure, every edge of M is binding. In each step, it either modifies c or increases the size of M, but it takes care never to violate the condition that all edges of M are binding (with respect to the current labeling). When it succeeds in making M a perfect matching, it terminates.

In what follows, we use the terminology introduced in the description of the Augmenting Path Algorithm for finding a maximum size bipartite matching. (That is, we refer to elements of A and B as "girls" and "boys" and the terms "Kőnig ritual", "The Grass", etc. is used without further explanation.)

Egerváry's algorithm

Step 0. Let $M = \emptyset$ and

$$c(v) = \begin{cases} \max\left\{w(e)\colon e \text{ is incident to } v\right\} & \text{if } v \in A, \\ 0 & \text{if } v \in B. \end{cases}$$

Step 1. Find all binding edges (with respect to the current labeling c). Then starting from the current matching M, run the Augmenting Path Algorithm for finding a matching M' of maximum size in the subgraph formed by the binding edges. If M' is a perfect matching, then STOP and return M' (and c).

Step 2. Let A_G and B_G be the set of girls and boys standing on the grass, respectively, when the Kőnig ritual froze in Step 1. Let

$$\delta = \min\left\{c(a) + c(b) - w(e)\colon e = \{a, b\} \in E, a \in A_G, b \in (B \setminus B_G)\right\}$$

and

$$c'(v) = \begin{cases} c(v) - \delta & \text{if } v \in A_G, \\ c(v) + \delta & \text{if } v \in B_G, \\ c(v) & \text{otherwise.} \end{cases}$$

Continue at Step 1 (with M' and c' instead of M and c). □

The correctness of the algorithm is justified by the following claims.

Claim. The assignment c defined in Step 0 is a labeling.

Indeed, since the maximum of the edge weights across all edges incident to a was defined to be $c(a)$ for every girl a, $c(a) + 0 \geq w(e)$ is obviously true for every edge $e = \{a, b\}$ with $a \in A$ and $b \in B$.

Claim. If the algorithm terminates in Step 1 then the returned matching M' is a maximum weight perfect matching.

This is obvious (as also mentioned before) from the above lemma.

Claim. The definition of δ in Step 2 gives a positive number.

To show this, recall that the Kőnig ritual freezes with a few couples and a few single girls standing on the grass and the fact that the ritual froze implies that "girls standing on the Grass only like boys also standing on the Grass". In the present context, this means that no vertex in A_G is adjacent to any vertex of $B \setminus B_G$ along a binding edge (since the Augmenting Path Algorithm was run on the subgraph of binding edges only).

This has two important implications. Firstly, since no edge $e = \{a, b\} \in E$ with $a \in A_G$, $b \in (B \setminus B_G)$ is binding, $c(a) + c(b) > w(e)$ must hold for all such edges. In other words, the minimum that defines δ is taken over a finite set of positive numbers. Secondly, this set of numbers is not empty: there must exist at least one edge $e = \{a, b\} \in E$ with $a \in A_G$ and $b \in (B \setminus B_G)$. Indeed, if this were not true, then the set A_G would violate the Hall-condition by $|A_G| > |B_G|$ (which is true since M' is not perfect, so the number of single girls – all of whom stand on the grass from the beginning of the ritual – is positive). So there would be no perfect matching in G, contradicting the definition of the Optimum Assignment Problem (where the existence of a perfect matching was assumed).

Claim. The assignment c' defined in Step 2 is a labeling.

To prove this, consider the following table that shows how the sum of the labels on the endpoints of an edge changes with the modification in the definition of c'.

$e = \{a, b\}$	$b \in B_G$	$b \in (B \setminus B_G)$
$a \in A_G$	0	$-\delta$
$a \in (A \setminus A_G)$	$+\delta$	0

Apparently, the only edges threatened by the danger of violating the definition of a labeling are the ones between A_G and $B \setminus B_G$. However, as seen above, each of these edges have a positive "surplus" (that is, $c(a) + c(b) - w(e) > 0$) and δ was defined to be the minimum of these surpluses. Therefore, none of these edges loses more from the sum of the labels on its endpoints than it is allowed to lose (that is, $c'(a) + c'(b) \geq w(e)$ remains to be true).

Claim. All edges of M' remain to be binding with respect to the new labeling c'.

As it can be seen from the above table, the only edges that can cease to be binding when c is modified to c' are the ones between $A \setminus A_G$ and B_G. So to show the above claim, we only have to ensure that no edge of M' is of this type. However, this is obvious from the Kőnig ritual: couples always "move together" (that is, either they both stand on the grass or they both stand off the grass). Therefore, every edge of M' either connects vertices of A_G and B_G or vertices of $A \setminus A_G$ and $B \setminus B_G$.

Claim. The algorithm terminates after at most n^2 cycles, where $n = |A| = |B|$.

The fact that only edges between $A \setminus A_G$ and B_G can cease to be binding has another important consequence: when the Kőnig ritual is started over again in Step 1 (after finding the new subgraph of binding edges), it can reproduce its previous running until the point it froze. Indeed, all the binding edges that were used in the previous running of the ritual to call a new couple to the grass remained to be binding (since, obviously, all such edges are between A_G and B_G).

So assume that the Kőnig ritual was started again and reached the point where it previously froze. The essential observation is that it will surely not freeze at this point in the present running. This is guaranteed by the simple fact that at least one new edge between A_G and $B \setminus B_G$ became binding with respect to c' (that was not binding with respect to c): the one (or the ones) where the minimum in the definition of δ in Step 2 was attained. (Indeed, such an edge $e = \{a, b\}$ loses all its surplus $c(a) + c(b) - w(e) = \delta$, so $c'(a) + c'(b) = w(e)$ is true.) In other words, a new binding edge between a girl a on the grass and a boy b off the grass is born which obviously results in either a calling b to the grass (with b's pair) or, if he is single, in b shouting "YAHVEETOE OOT".

Consequently, after each performance of Step 2 either the size of the matching, or the number of couples on the grass increases. This implies that after at most n cycles the size of M increases (after reaching the point where all couples are on the grass). Since the size of M can only increase at most n times and each increase takes at most n cycles, the total number of cycles is indeed at most n^2.

We can conclude that not only does Egerváry's algorithm work correctly, it is also very efficient. It is not hard to check that each cycle of the algorithm can be performed in $c \cdot m$ time, where m is the number of edges (and c is a suitable constant). Therefore the total running time is at most $c \cdot n^2 \cdot m$, which makes the algorithm polynomial.