An optimal algorithm for the minimum cost perfect matching in a bipartite graphs

Linear independence

A collection of row vectors $\{v_i^T\}$ are *independent* if there are no constants $\{c_i\}$ so that $\sum_i c_i v_i^T = 0$.

For an $n \times n$ matrix the rows are independent if and only if the determinant is not 0.

The rank of a matrix the maximum subset of rows that are independent. The rank of the rows and the rank of the columns is the same.

This can be shown by Gaus eliminations.

The rank of a matrix

If we have Maximize $c \cdot v$ such that $Ax \geq b$, and A is an $m \times n$ type matrix. Since for every variable we have and $x \geq 0$, these rows induce the identity matrix of dimension $n \times n$. Thus the rank of the columns is n so this is also the rank of the rows.

Recall the a BFS is obtained by taking n independent *rows* and put equality and solve this $n \times n$, system of equalities. It looks as $A' \cdot x = b'$. (Note that x does not change since x had size n to begin with). Since A' has independent rows the inverse matrix ${A'}^{-1}$ exists since this is equivalent to the determinant is not 0..

Thus $x = {A'}^{-1} \cdot b$. A unique solution exists. Which is a corner (a basic feasible solution).

Summary

For minimize $c \cdot x$ under $Ax \geq b$, $x \geq 0$,

The main property we use is:

Theorem 1 *The number of independent rows is the number of variables.*

All corners or basic feasible solution are derived by taking n independent *rows and putting* $A' \cdot x = b'$*. The BFS is* $(A')^{-1} \cdot b'$*.*

Minimum cost perfect matching in a bipartite graph

Definition 2 *A* balanced *bipartite graph is an independent set V_1 with another independent set V_2 both of size n, nd a collection of edges each with one vertex of V_1 and another in V_2 so that there exists a perfect matching (a matching that contains $V_1 \cup V_2$).*

A property of the vertex versus edges bipartite graph

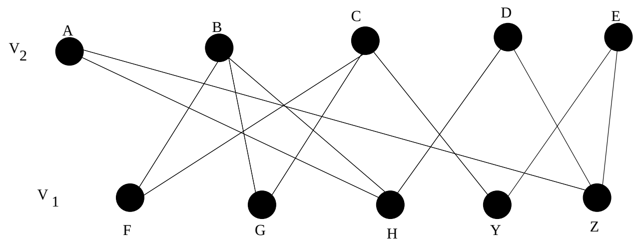
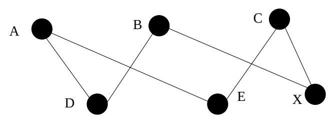


Figure 1: An example of a bipartite graph

The variables versus edges representation



	AD	AE	BD	BX	CE	CX
A	1	1	0	0	0	0
B	0	0	1	1	0	0
C	0	0	0	0	1	1
D	1	0	1	0	0	0
E	0	1	0	0	1	0
X	0	0	0	1	0	1

Figure 2: We later use this matrix. It is the matrix of vertices versus edges

The basic property of this matrix

Theorem 3 *These rows are* dependent.

Proof. In the example, note that if we add the rows of A, B, C this gives the same as adding the rows of D, EX. The sum in both cases is $(1, 1, \ldots, 1)$

More generally if we add the rows of the variables of V_1 and we add the rows of the variables in V_2 both will be the all 1 vector.

Thus in a bipartite graph with n vertices on each side the rank at most $2n - 1$. Because we can give all the rows of V_1 multiplied by 1 and all the rows of V_2 are given -1 we will get 0 □

Minimum cost perfect matching

Say that we are given a balanced bipartite graph with both sides having n vertices. Say that every edge has a cost $c(e)$.

The minimum cost perfect matching is a perfect matching of minimum cost.

Note that there could be exponentially many matchings and we want to minimum cost one.

Example

There is an example of a minimum cost perfect matching.

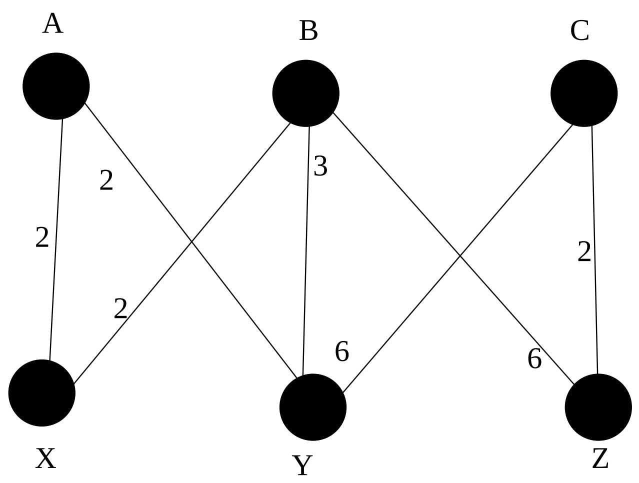


Figure 3: The minimum cost perfect matching is AY, XB, CZ of cost 6

.

An optimal iterative rounding algorithm

Minimize $\sum_{e=uv\in E} c_e \cdot x_{uv}$

Subject to: $\sum_{v|vu\in E} x_{uv} = 1$

$\sum_{u|vu\in E} x_{uv} = 1$

$x_{uv} \geq 0,$ for all $uv \in E$

x_{uv} is the fraction by which uv is taken.

Both lines say that the sum of fraction of the edges of a given vertex must be 1. If the solution is integral this is so by the definition of an *a minimum cost perfect matching*.

Note that the above is the vertices versus edges matrix, of a bipartite graph.

How to get BFS

Theorem 4 *For any BFS solution, there is at least one edge $e = uv$ so that either $x_{uv} = 0$ or $x_{uv} = 1$.*

Say that its true, how do we find an optimal solution? If $x_e = 0$ we remove the edge and the LP value does not change.

If there is $x_e = 1$ we take the edge and pay its cost $c(e)$. We remove the edge and its two vertices. Thus the LP value goes down by $c(e)$

Thus if this can be done in every iteration we take exactly a cost that is subtracted from the LP.

That we have a matching of value $opt_f = opt$.

Proof by contradiction

Say that there is no x_e so that $x_e = 1$ or $x_e = 0$ Thus every edge touching a vertex is fractional.

We note that $\sum_u x_{vu} = 1$. This means that each vertex in a given side has at least two edges that are fractional. This follows since one edge can not get a 1 by assumption.

The number of edges is the number of variables and so the rank of the matrix. It is at least $2n$.

On the other hand if all $0 < x_e < 1$ tight inequalities tight equalities can only happen in the first $2n$ rows.

But we saw that the vertices versus edges matrix has row rank at most $2n - 1$. Thus there are at most $2n - 1$ edges.

This is a contradiction

We found out that the number of edges is at least $2n$ and at most $2n-1$.

It must be the case that at every iteration for one x_e, $x_e = 0$ or $x_e = 1$.

This implies an optimum algorithm as seen before.