Fall 2013 Math 566 Implementation Assignment 3

Problem: Maximum or Minimum Perfect Matchings in Bipartite Graphs

In this report, you will implement the augmenting path algorithm for maximum matching in unweighted bipartite graphs and the Hungarian algorithm for maximum weighted perfect matchings in bipartite graphs. As part of your implementation, you will also output the min-size vertex cover or the minimum-weight vertex cover for these problems. You will also build a method for solving the minimum weight perfect matching problem for bipartite graphs, but you may simply program a reduction to the maximum weighted perfect matching problem. (However, if you perform the reduction, you must still present a maximum-weight vertex under-cover for the proof by duality.) While you may use a graph library such as the Sage Graph Library, you cannot use existing flow algorithms or use a linear programming solver.

Some Extra Information.

Something that may have become clear during the runs of the problem instances is that implementation details can matter a lot in the performance of an algorithm. There are a few things that I did in my implementation to speed it up:

1. Before running the augmenting paths algorithm to find a maximum matching in the equality subgraph, I selected a maximal matching using a greedy algorithm. This led to fewer iterations and more quickly finding a maximum matching.

2. Instead of building an edge list for the equality subgraph, I performed the check for "Is the edge $x_i y_j$ in the equality subgraph?" (i.e. "Is $u[i] + v[j] = W[i][j]$?") whenever I wanted to know if that edge was present. This greatly sped up the search.

3. There are two natural ways to compute a minimum-weight perfect matching and max-weight vertex under-cover from a max-weight matching and min-weight vertex cover algorithm:

   (a) Find a max-matching M and vertex cover u', v' for the matrix $W' = -W$. Then, M is minimum for W and $u = -u'$, $v = -v'$ is a max-weight vertex under-cover.

   (b) Let $b = \max\{w_{i,j}\}$ and find a max-matching M and vertex cover u', v' for the matrix W' where $w'_{i,j} = b - w_{i,j}$. Then, M is minimum for W and u, v is a max-weight vertex under-cover when $u_i = b - u'_i$ and $v_j = -v'_j$.

   These will both work, but to avoid negative-weight edges in the reduced matrix the second option is necessary. My implementation did not work over negative weights, so I used the second formulation. Recall that in the dual formulation of vertex cover and vertex under-cover the variables are free, so negative values are allowed.

Problem Instances.

I1.

(Maximum) The matching and vertex cover each have value 82 A matching M:

$$(0,4)(1,5)(2,1)(3,0)(4,2)(5,9)(6,7)(7,3)(8,6)(9,8)$$

A vertex cover:

$$u : [7,5,7,5,6,6,6,6,6,5]$$

$$v : [4,0,3,2,2,2,2,2,3,3]$$

(Minimum) The matching and vertex under-cover each have value 9 A matching M:

$$(0,0)(1,3)(2,8)(3,5)(4,1)(5,2)(6,4)(7,6)(8,7)(9,9)$$

The vertex under-cover:

$$u : [5,5,5,4,6,3,6,5,5,5]$$

$$v : [-4,-5,0,-4,-6,-4,-3,-4,-5,-5]$$

I2.

(Maximum) The matching and vertex cover each have value 265. The matching M:

$$(0,2)(1,1)(2,5)(3,14)(4,8)(5,13)(6,11)(7,10)(8,0)(9,4)(10,9)(11,7)(12,3)(13,12)(14,6)$$

The vertex cover:

$$u : [16,12,17,17,16,16,16,15,17,14,19,17,17,16,16]$$

$$v : [2,2,2,0,3,1,3,2,0,0,2,3,2,0,2]$$

(Minimum) The matching and vertex under-cover each have value 31 The matching M:

$$(0,5)(1,11)(2,6)(3,4)(4,1)(5,7)(6,13)(7,8)(8,12)(9,9)(10,3)(11,2)(12,0)(13,14)(14,10)$$

The vertex under-cover:

$$u : [5,3,6,4,6,4,5,2,7,4,5,4,4,7,3]$$

$$v : [-4,0,-1,0,-3,-2,-6,-3,-1,-3,-2,-2,-4,-4,-3]$$

I3.

(Maximum) The matching and vertex cover each have value 4864,
The matching M:

$$(0,35)(1,26)(2,54)(3,51)(4,11)(5,63)(6,9)(7,46)(8,99)(9,19)(10,1)(11,87)(12,94)(13,92)$$
$$(14,15)(15,67)(16,45)(17,20)(18,28)(19,98)(20,95)(21,65)(22,33)(23,29)(24,97)$$
$$(25,47)(26,3)(27,43)(28,13)(29,93)(30,59)(31,86)(32,82)(33,39)(34,79)(35,48)(36,61)$$
$$(37,66)(38,12)(39,23)(40,14)(41,81)(42,5)(43,70)(44,36)(45,8)(46,58)(47,21)(48,56)$$
$$(49,68)(50,32)(51,73)(52,38)(53,75)(54,60)(55,74)(56,24)(57,42)(58,89)(59,41)(60,2)(61,18)$$
$$(62,69)(63,7)(64,85)(65,34)(66,90)(67,17)(68,49)(69,4)(70,76)(71,50)(72,91)(73,22)(74,55)$$
$$(75,96)(76,25)(77,16)(78,37)(79,57)(80,78)(81,80)(82,83)(83,84)(84,31)(85,71)(86,10)(87,40)$$
$$(88,27)(89,62)(90,44)(91,0)(92,53)(93,30)(94,77)(95,88)(96,64)(97,6)(98,52)(99,72)$$

The vertex cover:

$$\begin{aligned}u :&[48,47,47,46,46,47,48,47,48,48,47,47,47,47,48,48,48,46,47,47,47,48,48,47,47,45,\\&48,47,46,46,48,47,47,47,46,48,47,47,47,48,47,46,49,45,47,47,48,46,47,47,48,47,\\&48,47,46,47,48,48,47,47,47,48,47,47,47,47,46,47,47,48,48,48,48,47,46,48,47,47,\\&46,47,47,46,46,47,48,47,48,48,49,47,46,49,46,47,47,47,47,47,49,47]\\v :&[0,2,2,0,1,0,2,1,2,1,1,3,2,2,2,1,2,2,1,1,2,3,2,1,0,2,2,0,2,2,1,1,1,1,2,\\&1,1,2,1,0,0,2,1,2,2,0,2,2,0,1,0,2,0,3,2,3,2,2,1,1,3,2,2,2,2,1,0,1,2,2,\\&1,2,2,2,2,2,1,1,2,3,2,2,1,2,2,2,2,2,2,2,2,1,2,1,2,0,1,1,1,1]\end{aligned}$$

(Minimum) The matching and vertex under-cover each have value 39 The matching M:

$$(0,28)(1,4)(2,37)(3,75)(4,59)(5,62)(6,95)(7,51)(8,7)(9,11)(10,2)(11,18)(12,55)(13,12)(14,52)(15,93)$$
$$(16,97)(17,3)(18,10)(19,77)(20,66)(21,99)(22,17)(23,90)(24,89)(25,49)(26,85)(27,24)(28,86)(29,78)$$
$$(30,32)(31,81)(32,53)(33,33)(34,70)(35,0)(36,82)(37,98)(38,8)(39,22)(40,71)(41,35)(42,80)(43,76)$$
$$(44,54)(45,40)(46,30)(47,39)(48,61)(49,34)(50,68)(51,60)(52,44)(53,56)(54,96)(55,14)(56,25)(57,19)$$
$$(58,41)(59,38)(60,47)(61,16)(62,64)(63,45)(64,42)(65,15)(66,29)(67,83)(68,67)(69,9)(70,13)(71,26)$$
$$(72,92)(73,23)(74,84)(75,57)(76,94)(77,91)(78,50)(79,6)(80,1)(81,46)(82,21)(83,58)(84,48)(85,87)$$
$$(86,73)(87,79)(88,74)(89,31)(90,63)(91,72)(92,43)(93,27)(94,65)(95,69)(96,88)(97,20)(98,5)(99,36)$$

The vertex under-cover:

$$\begin{aligned}u :&[2,2,1,2,2,3,2,1,3,2,2,0,0,2,2,4,0,2,1,2,3,3,\\&2,2,2,2,1,2,2,2,2,2,1,2,2,1,2,2,2,2,2,2,1,1,1,2,\\&2,2,1,1,2,2,2,1,2,1,1,2,2,2,2,2,2,2,2,3,1,2,4,2,\\&2,4,2,1,2,0,2,3,2,2,2,2,2,1,2,3,1,1,2,2,1,1,2,2,\\&0,2,2,1,1,2]\\v :&[-1,-2,-1,-2,-1,-1,0,-2,-2,-1,-1,-2,-1,-2,-1,-3,-2,\\&-1,0,-2,0,-1,-2,-1,-2,-1,-1,-2,-2,-1,-2,-2,-1,-2,-1,\\&-1,-2,-1,-2,-1,-2,-2,-2,-2,-2,-2,-1,-2,-2,-2,-2,-1,-2,\\&-1,-1,0,-1,0,-1,-2,-2,0,-2,-1,0,0,-1,-2,-2,-2,-2,-2,\\&-1,-1,0,0,-1,-2,-2,-1,0,-2,-2,-2,-1,-1,-2,-2,-2,-1,-2,\\&-2,-2,-2,-1,-2,-1,0,-2,-2]\end{aligned}$$

I4. This instance is a bit special. The maximum matching has weight $99,000$, and the vertex cover is given by $u_i = 99$ and $v_j = 0$ for all $i, j \in [1000]$. Similarly, the minimum matching has weight 0 and the vertex under-cover is the all zeros vector.

This is by "accident" but should have been expected. Here's why:

The matrix W is a 1000×1000 matrix whose entries were randomly selected from the integers between 0 and 99. By this random procedure, we should *expect* about ten 0's and ten 99's in every row and every column. This means, that when we consider only the edges of the bipartite graph that have maximum or minimum weight, that graph has almost all vertices of degree about 10.

Such a bipartite graph is actually built using a well-studied random graph process. This could be seen as an instance of $G_{1000,1000,p}$ where $p = \frac{1}{100}$. It turns out that the probability that a random balanced bipartite graph of order 2000 where every edge is included with probability $\frac{1}{100}$ has a high probability of having a perfect matching. That is certainly the case with the matrix W as described above.

This instance is slow for several people, especially if you built the equality subgraph as an edge list.