

CS416 Spring 2007

Prof. Wright

Assignment #10

Due Friday May 4, 2007

The major part of this assignment is to write a Matlab program to solve the assignment problem described below, for arbitrary problem size N , and for a random cost vector.

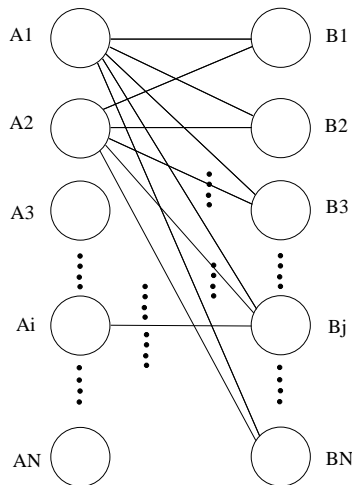
The assignment problem can be stated mathematically as follows. We have two sets each containing N nodes. Label the nodes in the first set A_1, A_2, \dots, A_N and the nodes in the second set B_1, B_2, \dots, B_N . Associated with each pair (A_i, B_j) there is a nonnegative cost c_{ij} . Our aim is to pair off the nodes in an optimal way, that is to find the N pairs (where the first element of each pair is a unique A node, and the second element is a unique B node), such that the sum of the N costs associated with these pairs is minimized.

Suppose for example that $N = 3$, and that the costs of the pairings are defined as follows:

$$\begin{aligned} c_{11} &= 1, & c_{12} &= 1, & c_{13} &= 2, \\ c_{21} &= 0.5, & c_{22} &= 1, & c_{23} &= 4, \\ c_{31} &= 0.5, & c_{32} &= 2, & c_{33} &= 1. \end{aligned}$$

Then the optimal pairing is (A_1, B_2) , (A_2, B_1) , and (A_3, B_3) with an optimal total cost of 2.5.

For the general case, the problem is illustrated in the figure below (note that not all arcs appear in the figure.)



The problem can be formulated as a minimum-cost network flow problem (as discussed in class and in the handout) in which there are a total of N^2 arcs, each one connecting an A node to a B node. The A nodes can be thought

of as generating one unit of flow each, while the B nodes can be thought of as consuming one unit each. The constraint matrix in this problem will have $2N$ rows and N^2 columns, and will be sparse, containing mostly zeros, with a relatively few $+1$ and -1 elements.

1. Write out in full the constraint matrix and constraint right-hand sides for the assignment problem, for the case of $N = 3$. Rows 1, 2, 3 of this matrix should contain the balance constraints for the nodes A_1, A_2, A_3 , respectively, while nodes 4, 5, 6 should contain balance constraints for the nodes B_1, B_2, B_3 . *Does the constraint matrix have full rank?* (Justify your answer.)
2. Write a matlab code to set up and solve the assignment problem for any given value of N (assigned at the start of the code) and for values of the cost vector drawn randomly from the interval $[0, 1]$. The value of N can be assigned in your code, and the output should be a list of the optimal collection of N pairs.

- You should use the code `pathfollow.m`, available from the web site, to solve the linear program.
- Order the N^2 columns of your matrix so that the arc connecting A_i to B_j is represented by column $(i - 1)N + j$.
- In assigning the random cost vector, use the statement

```
rand('twister',5621);
```

to initialize the random number generator, and assign the costs by simply calling `rand(n,1)`, where n (the number of variables) is N^2 .

- Use the `spconvert` function in MATLAB to set up the matrix; do `help spconvert` to see details. *Use a version of this command that does not involve setting up a large dense matrix first*, so that your code can be scaled efficiently to high values of N . I recommend using the form

```
A = spconvert(D);
```

where D is a $K \times 3$ matrix in which each row is a triple of the form (i, j, v) , where i represents the row number, j represents the column number, and v represents the (nonzero) value stored in the (i, j) location of the constraint matrix. K represents the number of nonzero entries in the matrix.`matchings`

- If your code works correctly, the solution vector returned from the linear programming solver will contain N components with the value 1 and the remaining components zero.
- Include code that inspects the final value of x and prints the N optimal pairs.
- Run your code (and gather its output) for the following values of N : 4, 10, 20, 100, 200.

- Make a table (by hand if you wish) of
 - (a) the times required to run `pathfollow` (which is printed out on exit from the routine) for each N (make sure you gather all the times on the same computer);
 - (b) the number of iterations required by `pathfollow` for each value of N . Roughly how does the solution time scale with N ?
- `pathfollow` will generate many warning messages but these can be ignored for this assignment.

Hand in a hard copy of your code `randomAssignment.m` and your output containing the optimal pairings for $N = 4, 10, 20, 100, 200$, along with your written answers. Put your code `randomAssignment.m` into a directory called `homework10`. From the parent directory of `homework10`, run the following command:

```
handin -c cs416-1 -a hwk10 -d homework10
```