MATLAB实例：Munkres指派算法

作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

1．指派问题陈述

指派问题涉及将机器分配给任务，将工人分配给工作，将足球运动员分配给职位等。目标是确定最佳分配，例如，使总成本最小化或使团队效率最大化。指派问题是组合优化领域中的一个基本问题。

例如，假设我们有四个工作需要由四个工作人员执行。因为每个工人都有不同的技能，所以执行工作所需的时间取决于分配给该工人的工人。

下面的矩阵显示了工人和工作的每种组合所需的时间（以分钟为单位）。作业用J1，J2，J3和J4表示，工人用W1，W2，W3和W4表示。

	J1	*J2*	*J3*	*J4*
W1	82	83	69	92
W2	77	37	49	92
W3	11	69	5	86
W4	8	9	98	23

每个工人应仅执行一项工作，目标是最大程度地减少执行所有工作所需的总时间。

事实证明，将工人1分配给作业3，将工人2分配给作业2，将工人3分配给作业1，将工人4分配给作业4是最佳选择。那么，总时间为69 + 37 + 11 + 23 = 140分钟。所有其他任务导致需要更多时间。

2．Munkres指派算法MATLAB程序

munkres.m

```
function [assignment,cost] = munkres(costMat)
% MUNKRES    Munkres Assign Algorithm
%
```

```
% [ASSIGN,COST] = munkres(COSTMAT) returns the optimal assignment in ASSIGN
% with the minimum COST based on the assignment problem represented by the
% COSTMAT, where the (i,j)th element represents the cost to assign the jth
% job to the ith worker.
%

% This is vectorized implementation of the algorithm. It is the fastest
% among all Matlab implementations of the algorithm.

% Examples
% Example 1: a 5 x 5 example
%{
[assignment,cost] = munkres(magic(5));
[assignedrows,dum]=find(assignment);
disp(assignedrows'); % 3 2 1 5 4
disp(cost); %15
%}
% Example 2: 400 x 400 random data
%{
n=5;
A=rand(n);
tic
[a,b]=munkres(A);
toc
%}

% Reference:
% "Munkres' Assignment Algorithm, Modified for Rectangular Matrices",
% http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html

% version 1.0 by Yi Cao at Cranfield University on 17th June 2008

assignment = false(size(costMat));
cost = 0;

costMat(costMat~=costMat)=Inf;
validMat = costMat<Inf;
validCol = any(validMat);
validRow = any(validMat,2);

nRows = sum(validRow);
nCols = sum(validCol);
n = max(nRows,nCols);
if ~n
    return
end
```

```
dMat = zeros(n);
dMat(1:nRows,1:nCols) = costMat(validRow,validCol);

%*************************************************
% Munkres' Assignment Algorithm starts here
%*************************************************

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   STEP 1: Subtract the row minimum from each row.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 dMat = bsxfun(@minus, dMat, min(dMat,[],2));

%************************************************************************
%   STEP 2: Find a zero of dMat. If there are no starred zeros in its
%           column or row start the zero. Repeat for each zero
%************************************************************************
zP = ~dMat;
starZ = false(n);
while any(zP(:))
    [r,c]=find(zP,1);
    starZ(r,c)=true;
    zP(r,:)=false;
    zP(:,c)=false;
end

while 1
%************************************************************************
%   STEP 3: Cover each column with a starred zero. If all the columns are
%           covered then the matching is maximum
%************************************************************************
    primeZ = false(n);
    coverColumn = any(starZ);
    if ~any(~coverColumn)
        break
    end
    coverRow = false(n,1);
    while 1
        %************************************************************************
        %   STEP 4: Find a noncovered zero and prime it.  If there is no starred
        %           zero in the row containing this primed zero, Go to Step 5.
        %           Otherwise, cover this row and uncover the column containing
        %           the starred zero. Continue in this manner until there are no
        %           uncovered zeros left. Save the smallest uncovered value and
        %           Go to Step 6.
        %************************************************************************
        zP(:) = false;
        zP(~coverRow,~coverColumn) = ~dMat(~coverRow,~coverColumn);
```

```
    Step = 6;
    while any(any(zP(~coverRow,~coverColumn)))
        [uZr,uZc] = find(zP,1);
        primeZ(uZr,uZc) = true;
        stz = starZ(uZr,:);
        if ~any(stz)
            Step = 5;
            break;
        end
        coverRow(uZr) = true;
        coverColumn(stz) = false;
        zP(uZr,:) = false;
        zP(~coverRow,stz) = ~dMat(~coverRow,stz);
    end
    if Step == 6
        % ********************************************************************
        % STEP 6: Add the minimum uncovered value to every element of each covered
        %         row, and subtract it from every element of each uncovered column.
        %         Return to Step 4 without altering any stars, primes, or covered lines.
        %********************************************************************************
        M=dMat(~coverRow,~coverColumn);
        minval=min(min(M));
        if minval==inf
            return
        end
        dMat(coverRow,coverColumn)=dMat(coverRow,coverColumn)+minval;
        dMat(~coverRow,~coverColumn)=M-minval;
    else
        break
    end
end
%**************************************************************************
% STEP 5:
%  Construct a series of alternating primed and starred zeros as
%  follows:
%  Let Z0 represent the uncovered primed zero found in Step 4.
%  Let Z1 denote the starred zero in the column of Z0 (if any).
%  Let Z2 denote the primed zero in the row of Z1 (there will always
%  be one).  Continue until the series terminates at a primed zero
%  that has no starred zero in its column.  Unstar each starred
%  zero of the series, star each primed zero of the series, erase
%  all primes and uncover every line in the matrix.  Return to Step 3.
%**************************************************************************
rowZ1 = starZ(:,uZc);
starZ(uZr,uZc)=true;
while any(rowZ1)
    starZ(rowZ1,uZc)=false;
```

```
        uZc = primeZ(rowZ1,:);
        uZr = rowZ1;
        rowZ1 = starZ(:,uZc);
        starZ(uZr,uZc)=true;
    end
end

% Cost of assignment
assignment(validRow,validCol) = starZ(1:nRows,1:nCols);
cost = sum(costMat(assignment));
```

demo_munkres.m

```
A=[82,83,69,92;77,37,49,92;11,69,5,86;8,9,98,23];
[assignment,cost]=munkres(A)
[assignedrows,dum]=find(assignment);
order=assignedrows'
```

3. 指派结果

```
>> demo_munkres

assignment =

  4×4 logical 数组

   0   0   1   0
   0   1   0   0
   1   0   0   0
   0   0   0   1


cost =

   140


order =

     3     2     1     4
```

4. 参考文献

[1] Munkres' Assignment Algorithm Modified for Rectangular Matrices

[2] Munkres Assignment Algorithm

[3] Hungarian algorithm: The assignment problem