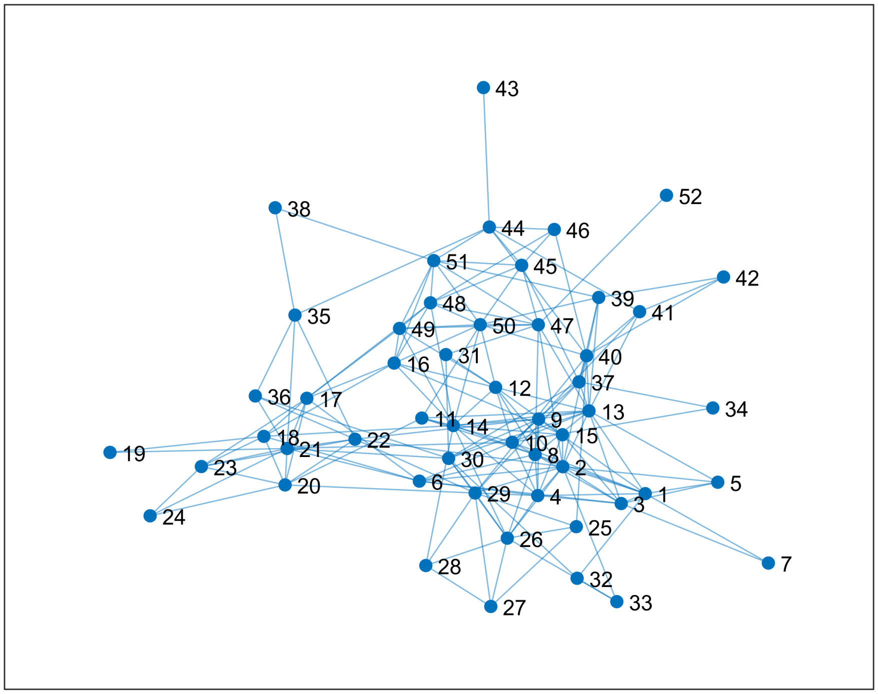
Optimization Division

```
clear all; close all; clc
% % Load matrix for imputation of subregion
M = readmatrix('cp_exp_unique_1.csv');
l = length(M(:,1));

%%
% Travel time matrix
PT = readmatrix('PT.csv',"NumHeaderLines",1);
% Distance matrix
PD = readmatrix('PD.csv',"NumHeaderLines",1);

ind = zeros(l);
m = max(M(:,4));
ad_sub = zeros(m,m);
for i=1:l
    ind = find(PD(i,:) == min(PD(i,M(:,4)~=M(i,4))));
    % Neighbours adyacent to subregion M(i,4)
    ad_sub(M(i,4),M(ind,4)) = 1;
end

ad_sub = (ad_sub + ad_sub')/2>0;
G = graph(ad_sub);
gr = plot(G);
saveas(gr,'graph_sub','svg')
```



Load dataset for the problem

```
Ms = readmatrix('sub_exp_unique_.csv');
```

Additive statistic for balance

```
X = 0.8*Ms(:,2)+0.1*Ms(:,3)+0.1*Ms(:,4); % Diligencias + Address + Km2 (all scaled)
```

We declare the optimization variables

```
D = optimvar('D',length(X),25,'Type','integer','LowerBound',0,'UpperBound',1);
T = optimvar('T',25,25,'LowerBound',0);
```

We declare the constraints

Define the aggregated statistic for every region

```
y = X'*D;
```

The next constraint require every subregion k to be assigned to exactly one region r

```
assignment = sum(D,2) == ones(length(X),1);
```

Aditional constraints (LP formulation)

```
for i=1:25
    for j=1:25
        additional1(i,j) = T(i,j) >= y(j)-y(i);
        additional2(i,j) = T(i,j) >= y(i)-y(j);
    end
end
```

Adyacency constraints & Grouping constraints

```
for i=1:length(X)
    for j=i+1:length(X)
        if ad_sub(i,j)==0 && i~=j
            ad_const(i,j,1:25) = reshape(D(i,:)+D(j,:)<=ones(1,25),1,1,25);
        end
        if Ms(i,7)~=Ms(j,7) && i~=j
            gp_const(i,j,1:25) = reshape(D(i,:)+D(j,:)<=ones(1,25),1,1,25);
        end
    end
end
```

Declare the optimization problem

```
optDiv = optimproblem;
```

The objective function to minimize is the sum of the absolute value of differences.

```
balance = sum(sum(T));
optDiv.Objective = balance;
```

Include the constraints in the problem.

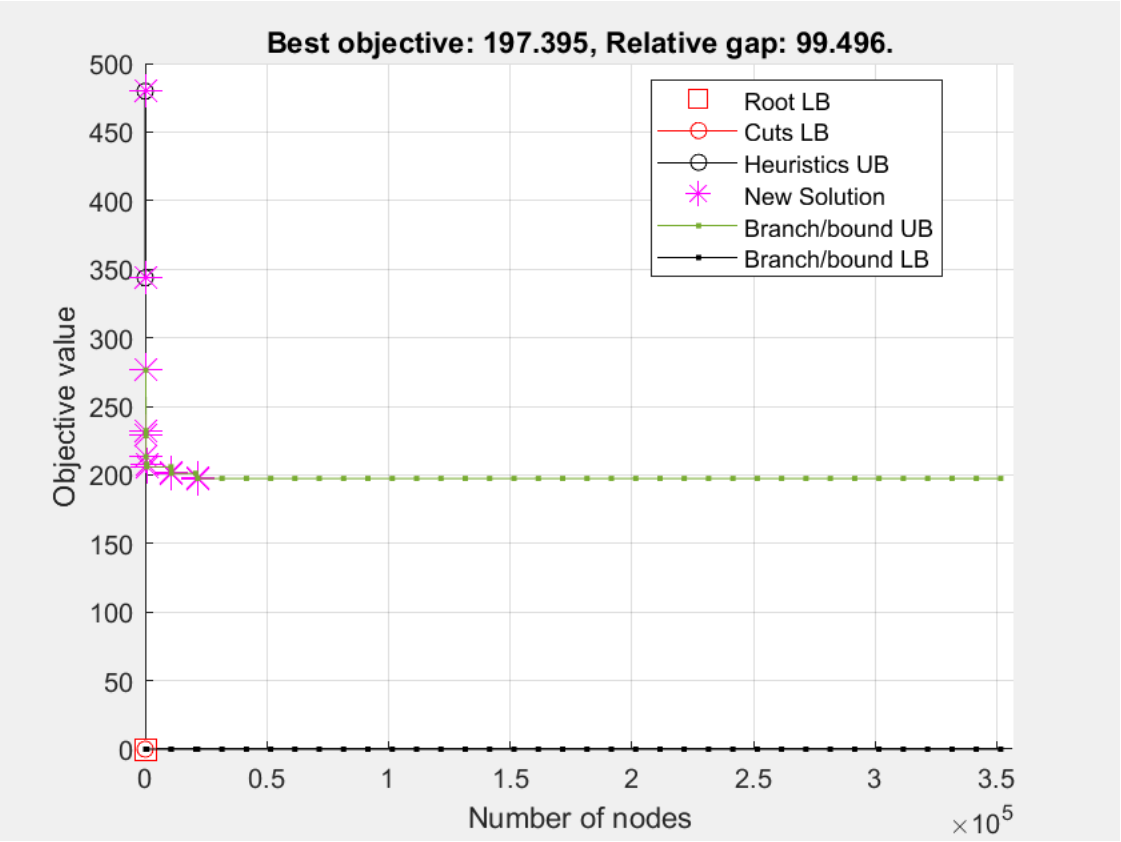
```
optDiv.Constraints.as = assignment;
optDiv.Constraints.ad1 = additional1;
optDiv.Constraints.ad2 = additional2;
optDiv.Constraints.adc = ad_const;
optDiv.Constraints.adc = gp_const;
```

Solution to the problem

```
opts = optimoptions('intlinprog','MaxTime',18000, 'Display','iter','PlotFcn',@optimplotmilp);
[sol,fval,exitflag,output] = solve(optDiv,'options',opts);
```

```
Solving problem using intlinprog.
LP:                Optimal objective value is 0.000000.

Cut Generation:    Applied 2 Gomory cuts, and 5 mir cuts.
                   Lower bound is 0.000000.

Heuristics:        Found 2 solutions using diving.
                   Upper bound is 343.695678.
                   Relative gap is 99.71%.

Cut Generation:    Applied 8 Gomory cuts,
                   and 116 flow cover cuts.
                   Lower bound is 0.000000.
                   Relative gap is 99.71%.

Branch and Bound:

   nodes     total   num int        integer       relative
explored  time (s)  solution           fval        gap (%)
     294    308.57         3   2.765048e+02   9.963965e+01
     328    315.42         4   2.321947e+02   9.957117e+01
     331    316.22         5   2.286407e+02   9.956454e+01
     332    316.80         6   2.133358e+02   9.953344e+01
     644    371.84         7   2.078453e+02   9.952118e+01
     651    373.12         8   2.058895e+02   9.951665e+01
   10651    888.44         8   2.058895e+02   9.951665e+01
   10757    892.76         9   2.020824e+02   9.950759e+01
   10757    892.76        10   2.011705e+02   9.950537e+01
   20757   1399.60        10   2.011705e+02   9.950537e+01
   21560   1439.51        11   1.976733e+02   9.949666e+01
   21562   1439.68        12   1.973947e+02   9.949595e+01
   31562   1976.59        12   1.973947e+02   9.949595e+01
   41562   2510.83        12   1.973947e+02   9.949595e+01
   51562   3059.03        12   1.973947e+02   9.949595e+01
   61562   3617.47        12   1.973947e+02   9.949595e+01
   71562   4132.55        12   1.973947e+02   9.949595e+01
   81562   4677.61        12   1.973947e+02   9.949595e+01
   91562   5189.48        12   1.973947e+02   9.949595e+01
  101562   5742.14        12   1.973947e+02   9.949595e+01
  111562   6300.06        12   1.973947e+02   9.949595e+01
  121562   6807.17        12   1.973947e+02   9.949595e+01
```

```
   131562   7334.71         12   1.973947e+02   9.949595e+01
   141562   7858.14         12   1.973947e+02   9.949595e+01
   151562   8336.16         12   1.973947e+02   9.949595e+01
   161562   8806.26         12   1.973947e+02   9.949595e+01
   171562   9317.33         12   1.973947e+02   9.949595e+01
   181562   9775.56         12   1.973947e+02   9.949595e+01
   191562  10256.31         12   1.973947e+02   9.949595e+01
   201562  10773.90         12   1.973947e+02   9.949595e+01
   211562  11216.04         12   1.973947e+02   9.949595e+01
   221562  11668.37         12   1.973947e+02   9.949595e+01
   231562  12134.97         12   1.973947e+02   9.949595e+01
   241562  12615.93         12   1.973947e+02   9.949595e+01
   251562  13105.11         12   1.973947e+02   9.949595e+01
   261562  13594.82         12   1.973947e+02   9.949595e+01
   271562  14041.55         12   1.973947e+02   9.949595e+01
   281562  14505.56         12   1.973947e+02   9.949595e+01
   291562  15000.53         12   1.973947e+02   9.949595e+01
   301562  15476.23         12   1.973947e+02   9.949595e+01
   311562  15963.00         12   1.973947e+02   9.949595e+01
   321562  16417.06         12   1.973947e+02   9.949595e+01
   331562  16844.37         12   1.973947e+02   9.949595e+01
   341562  17270.42         12   1.973947e+02   9.949595e+01
   351562  17745.76         12   1.973947e+02   9.949595e+01
```



```
Solver stopped prematurely. Integer feasible point found.

Intlinprog stopped because it exceeded the time limit,
options.MaxTime = 18000 (the selected value). The intcon
variables are integer within tolerance,
options.IntegerTolerance = 1e-05 (the default value).
```

Optimal division

```
sol.D
```

```
ans = 52×25
    1.0000         0         0         0         0         0         0         0⋯
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
    1.0000         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
         0    1.0000         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
      ⋮
```

Optimal value function

```
fval
```

```
fval = 197.3947
```

```
if isempty(sol) % If the problem is infeasible or you stopped early with no solution
    disp('The solver did not return a solution.')
    return % Stop the script because there is nothing to examine
end
```

Examine the exit flag and the infeasibility of the solution.

```
exitflag
```

```
exitflag = 
    IntegerFeasible
```

```
infeas1 = max(max(infeasibility(assignment,sol)))
```

```
infeas1 = 7.9536e-13
```

```
res = [Ms, sol.D]
```

```
res = 52×33
    1.0000    1.0948    1.5882    2.9762   19.3246  -99.1627    1.0000    1.0000⋯
    2.0000    0.6313    0.9758    3.0088   19.3101  -99.1290    1.0000    2.0000
    3.0000    0.4352    0.4360    3.0696   19.2757  -99.1960    1.0000    3.0000
    4.0000    0.3001    0.3910    1.6239   19.2857  -99.1680    1.0000    4.0000
    5.0000    0.0064    0.0069    1.0542   19.2485  -99.1663    1.0000    5.0000
    6.0000    0.2604    0.3287    2.9106   19.2506  -99.0909    1.0000    6.0000
    7.0000    0.0042    0.0138    0.0396   19.2937  -99.1837    1.0000    7.0000
    8.0000    0.3815    0.4913    1.4084   19.3567  -99.0635    2.0000    8.0000
    9.0000    0.4296    0.6194    0.9395   19.3665  -99.1065    2.0000    9.0000
   10.0000    0.9851    0.8512    1.4711   19.3777  -99.0973    2.0000   10.0000
      ⋮
```

```
colNames = {'reg_sub','num_diligencias_sub', 'num_addr_sub', 'area_sqkm_sub', ...
```

```
    'lat_c', 'lon_c', 'region', 'id', ...
    'r1','r2','r3','r4','r5','r6','r7','r8','r9','r10',...
    'r11','r12','r13','r14','r15','r16', 'r17','r18','r19','r20','r21','r22','r23','r24','r25']
sTable = array2table(res,'VariableNames',colNames);
writetable(sTable,'sub_exp_res.csv')
```