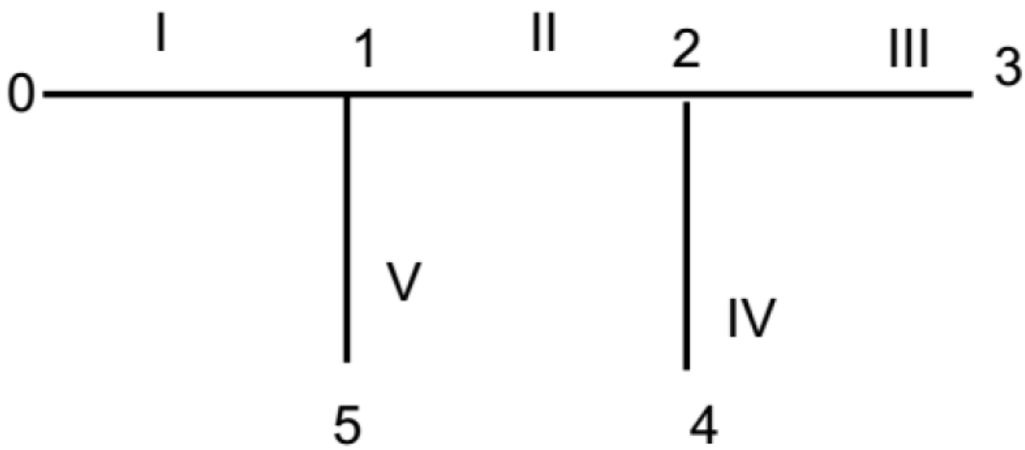
ASSIGNMENT- WATER NETWORKS

-Meghana P.V

CH12B083

PART 1:

Optimal pipe selection for a branched network (From a set of discrete diameters)



Introduction:

In the following network, node 0 is the source node and the rest are demand nodes. The pressure drop (Head in metres) is given by:

$$H_A - H_B = \Delta H = 4.457 \times 10^8 \frac{LQ^{1.85}}{D^{4.87}}$$

Where, Q is the flow rate in m3/min, HA ; HB ; ΔH are the heads in meters, L is the length of pipe in meter , D is the inner diameter of the pipe in mm.

Link specifications:

The flow in links are as follows:
Link I : 8.5 m^3/min
II : 5.8
III : 1.2
IV : 1.6
V : 1.3

The lengths of the pipes are as follows:
Pipe I :1000m
II :600
III :400

#	Pipe Dia (mm)	Unit cost (Rs/m)
1	80	424
2	100	570
3	125	767
4	150	977
5	200	1431
6	250	1924
7	300	2451
8	350	3008
9	400	3591
10	450	4198
11	500	4828
12	600	6149
13	700	7545
14	750	8269

IV :300
V :300

Constraints on the heads at nodes:
The head available at node 0 is 115 m and the minimum required head at the other nodes are as follows:
Node 1: 90m
2: 85
3: 80
4: 80
5: 80

Given the optimal diameters (if any positive real value diameter was available) of each pipe, 3 discrete commercially available diameter sizes closest to these optimal diameters for each pipe are chosen and the discrete optimal diameters are found through the '**Discrete Merge Method**' detailed in the paper by Richard S Mah, (AIChE, 1976)

Paths and constraints:

There are 3 paths from the source to end demand nodes in this network:

Path 1: Links 1, 2, 3; Net Pressure drop= $\Delta H1+\Delta H2+\Delta H3=H0-H3<=35$ (Since H0=115 and H3>=80)

Path 2: Links 1, 2, 4; Net Pressure drop= $\Delta H1+\Delta H2+\Delta H4=H0-H4<=35$ (Since H0=115 and H4>=80)

Path 2: Links 1, 5; Net Pressure drop= $\Delta H1+\Delta H1= H0-H5<=35$ (Since H0=115 and H5>=80)

Since minimum heads at the end demand nodes 3, 4 and 5 are the same, the net pressure drop has the same constraint for all the paths, i.e **ΔHnet<=35**. So we can directly compare all the paths to decide the maximum pressure drop in the parallel merge approaches in the problem.

Link pressure and cost calculations for discrete diameters:

Link No.	Optimal Dia	k=1			k=2			k=3		
		Dia	Pressure Drop(m)	Cost	Dia	Pressure Drop(m)	Cost	Dia	Pressure Drop(m)	Cost
1	304.9	250	49.0342802	1924000	300	20.17841902	2451000	350	9.524808893	3008000
2	267.5	200	43.00506354	858600	250	14.50667329	1154400	300	5.9697365	1470600
3	152	125	15.33331887	306800	150	6.309914855	390800	200	1.554432442	572400
4	159.8	125	19.5809651	230100	150	8.057891682	293100	200	1.985042355	429300
5	119.8	100	39.53293793	171000	125	13.33543931	230100	150	5.487754303	293100

Discrete Merge Method:

Branches are merged at nodes to form equivalent networks, with the process starting from the end demand nodes (terminal nodes) of the tree all the way till the source node. Merging is done by two ways:

Parallel Merging: For links at nodes of degree greater than 2. Pressure drop of equivalent branch is the maximum of that of the constituent branches and cost is the sum that of constituent branches.

Series Merging: For links at nodes of degree equal to 2. Pressure drop of equivalent branch is the sum of that of the constituent branches and cost is the sum of that of the costs of constituent branches.

The combinations are ordered in order of decreasing pressure drops. The combinations with decreasing costs in this list are deleted (as the costs are supposed to be in ascending order).

Procedure:

1. Node 2: Parallel Merging of Links 3 and 4 to give equivalent link for 3&4
2. Node 2: Series Merging of Links 2 and equivalent link for 3&4 to give equivalent link for 2,3&4
3. Node 1: Parallel Merging of Links 5 and equivalent link for 2,3&4 to give equivalent link for 2,3,4&5
4. Node 1: Series Merging of Links 1 and equivalent link for 2,3,4&5 to give final equivalent link for 1, 2,3,4&5

Once this is done, the networks are checked for satisfaction of head constraints. The design at the top of the branch list for the final pipe section (i.e having min. cost, max. pressure drop) satisfying these constraints is the optimal solution.

1. Parallel Merging of Links 3 and 4:

Link specs:

k3	**Link 3**		
	Dia	**Pressure Drop(m)**	**Cost**
1	125	15.33331887	306800
2	150	6.309914855	390800
3	200	1.554432442	572400

k4	**Link 4**		
	Dia	**Pressure Drop(m)**	**Cost**
1	125	19.58097	230100
2	150	8.057892	293100
3	200	1.985042	429300

Merge list:

keq1	**k3 (k for link 3)**	**k4 (k for link 4)**	**Pressure Drop(m)**	**Cost**	**Dia3**	**Dia4**
1	1	1	19.5809651	536900	125	125
2	1	2	15.33331887	599900	125	150
3	2	2	8.057891682	683900	150	150
4	2	3	6.309914855	820100	150	200
5	3	3	1.985042355	1001700	200	200

2. Series Merging of Links 2 and equivalent link for 3&4

All combinations:

keq1\k2	1		2		3	
	Pressure Drop(m)	**Cost**	**Pressure Drop(m)**	**Cost**	**Pressure Drop(m)**	**Cost**
1	62.58602864	1395500	34.08763839	1691300	25.5507	2007500
2	58.33838241	1458500	29.83999216	1754300	21.30306	2070500
3	51.06295522	1542500	22.56456497	1838300	14.02763	2154500
4	49.3149784	1678700	20.81658815	1974500	12.27965	2290700
5	44.9901059	1860300	16.49171565	2156100	7.954779	2472300

Merge list:

keq2	keq1	k2	Pressure Drop(m)	Cost	Dia3	Dia4	Dia2
1	1	1	62.5860286	1395500	125	125	200
2	2	1	58.3383824	1458500	125	150	200
3	3	1	51.0629552	1542500	150	150	200
4	4	1	49.3149784	1678700	150	200	200
5	1	2	34.0876384	1691300	125	125	250
6	2	2	29.8399922	1754300	125	150	250
7	3	2	22.564565	1838300	150	150	250
8	4	2	20.8165881	1974500	150	200	250
9	3	3	14.0276282	2154500	150	150	300
10	4	3	12.2796514	2290700	150	200	300
11	5	3	7.95477886	2472300	200	200	300

3. Parallel Merging of Links 5 and equivalent link for 2,3&4

keq3	keq2	k5	Pressure Drop(m)	Cost	Dia3	Dia4	Dia2	Dia5
1	1	1	62.58602864	1566500	125	125	200	100
2	2	1	58.33838241	1629500	125	150	200	100
3	3	1	51.06295522	1713500	150	150	200	100
4	4	1	49.3149784	1849700	150	200	200	100
5	5	1	39.53293793	1862300	125	125	250	100
6	5	2	34.08763839	1921400	125	125	250	125
7	6	2	29.83999216	1984400	125	150	250	125
8	7	2	22.56456497	2068400	150	150	250	125
9	8	2	20.81658815	2204600	150	200	250	125
10	9	2	14.02762818	2384600	150	150	300	125
11	10	2	13.33543931	2520800	150	200	300	125
12	10	3	12.27965135	2583800	150	200	300	150
13	11	3	7.954778855	2765400	200	200	300	150

4. Series Merging of Links 1 and equivalent link for 2,3,4&5

Since we know that H1>=85, we can first check if the discrete pipe diameters for link 1 are feasible solutions. We find that for d=250mm, H1= 65.96572 and hence infeasible. We can proceed with the other 2 feasible diameters for link 1; d=300mm and d=350mm

All combinations:

keq3\k1	2		3	
	Pressure Drop(m)	Cost	Pressure Drop(m)	Cost
1	82.76445	4017500	72.11084	4574500
2	78.5168	4080500	67.86319	4637500

3	71.24137	4164500	60.58776	4721500
4	69.4934	4300700	58.83979	4857700
5	59.71136	4313300	49.05775	4870300
6	54.26606	4372400	43.61245	4929400
7	50.01841	4435400	39.3648	4992400
8	42.74298	4519400	32.08937	5076400
9	40.99501	4655600	30.3414	5212600
10	34.20605	4835600	23.55244	5392600
11	33.51386	4971800	22.86025	5528800
12	32.45807	5034800	21.80446	5591800
13	28.1332	5216400	17.47959	5773400

Merge list:

keq4	**k1**	**keq3**	**Pressure Drop(m)**	**Cost**	**Dia1**	**Dia2**	**Dia3**	**Dia4**	**Dia5**
1	2	1	82.76444766	4017500	300	200	125	125	100
2	2	2	78.51680143	4080500	300	200	125	150	100
3	2	3	71.24137424	4164500	300	200	150	150	100
4	2	4	69.49339741	4300700	300	200	150	200	100
5	2	5	59.71135695	4313300	300	250	125	125	100
6	2	6	54.26605741	4372400	300	250	125	125	125
7	2	7	50.01841118	4435400	300	250	125	150	125
8	2	8	42.74298399	4519400	300	250	150	150	125
9	2	9	40.99500716	4655600	300	250	150	200	125
10	**2**	**10**	**34.2060472**	**4835600**	**300**	**300**	**150**	**150**	**125**
11	2	11	33.51385833	4971800	300	300	150	200	125
12	2	12	32.45807037	5034800	300	300	150	200	150
13	3	8	32.08937387	5076400	350	300	200	200	150
14	3	9	30.34139704	5212600	350	250	150	200	125
15	2	13	28.13319787	5216400	300	300	200	200	150
16	3	10	23.55243707	5392600	350	300	150	150	125
17	3	11	22.8602482	5528800	350	300	150	200	125
18	3	12	21.80446025	5591800	350	300	150	200	150
19	3	13	17.47958775	5773400	350	300	200	200	150

The first 9 solutions are infeasible as the net pressure drop does not satisfy the condition, $\Delta H_{net} \le 35$. Therefore the optimal discrete pipe diameters is the solution corresponding to keq=10.

D1=300mm
D2=300mm
D3=150mm
D4=150mm
D5=125mm

PART 2:

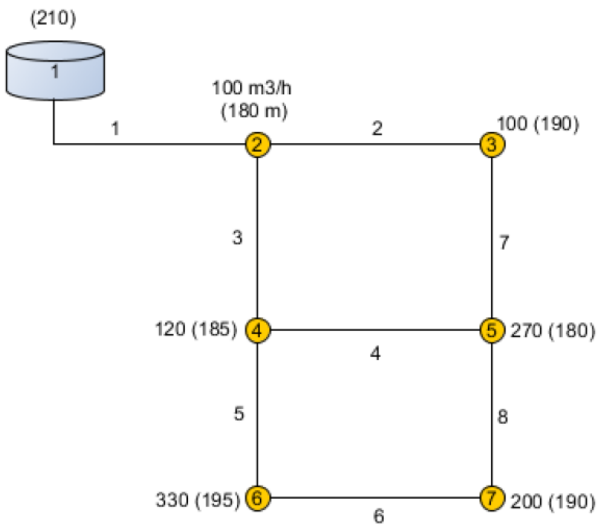
Optimization of water network [For DISCRETE diameters]

Introduction:

It is a gravity driven network with eight pipes for which optimal diameters are to be found.

The Hazen Williams formula for head loss calculations

$$h_f = \frac{162.9\ L\ Q^{1.85}}{130^{1.85} D^{4.87}}$$



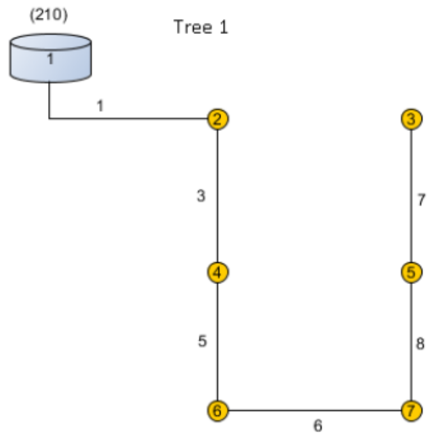
Specs:

Length of each pipe=1000m

The available pipe diameters in inches are given below with their cost per unit length in parenthesis: 1 inch (2 units), 2 (5), 3 (8), 4(11), 6(16), 8(23), 10(32), 12(50), 14(60), 16(90), 18(130), 20(170), 22(300) and 24 (550)

Procedure:

1. Finding the optimal diameters for **spanning tree 1**: The discrete merge method is used to obtain the optimal solution satisfying the demands and minimum pressure drops.



The minimum total flow rate is considered, i.e the sum of the demand at all the nodes. With this the flowrates are:

Link	Q(m3/h)
1	1120
3	1020
5	900
6	570
8	370
7	100

- **Discrete merge method:** Consecutive series merges are done for this tree, i.e links 7 & 8 ; then (7,8) & 6 ; then (7,8,6) & 5 ; then (7,8,6,5) & 3 ; then (7,8,6,5,3) & 1.
- **Constraints:**
  1. Total Pressure Drop over the network<=20 m (H0-H3=210-190)
  2. H6>=195

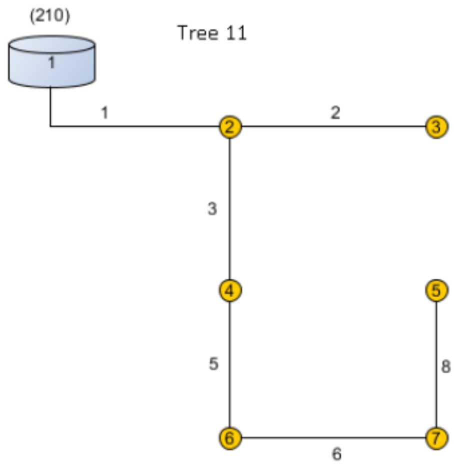
  The minimum head constraints of other nodes are automatically satisfied as flow is from H0 to H3 and the minimum heads for the other nodes is lesser than that of node 3. Constraints are checked for the final merge list from the top, and the first solution to satisfy them is the optimal solution.

D7=10 in
D8=14 in
D6=16 in
D5=18 in
D3=20 in
D1=20 in
Optimal Cost = 652000 units

H2 = 205.9620 m
H4 = 202.5656 m
H6 = 198.0647 m
H7 = 194.6337 m
H5 = 191.6780 m
H3 = 190.3255 m

2. Pick a loop, move clockwise and see if the next tree gives a better solution. If it does not, move anticlockwise and check for any improvement in cost. If both the directions does not give any improvement, pick the other loop and repeat the procedure.

Finding optimal diameters for **spanning tree 11**: Obtained by picking the bigger loop, including chord 2 and moving clockwise from spanning tree 1, thus removing branch 7.



The minimum total flow rate is considered, i.e the sum of the demand at all the nodes. With this the flowrates are:

Link	Q(m3/h)
1	1120
3	920
5	800
6	470
8	270
2	100

- **Discrete merge method:** Consecutive series merges are done starting from node 5 till node 2, at which a parallel merge is done and finally series merge of the equivalent branch is done with branch 1.
- **Constraints:** Check if each of the head constraints at the nodes is satisfied. Need not check at node 2 as it will be automatically satisfied if the others are satisfied.

D8=14 in
D6=14 in
D5=18 in
D3=18 in
D2=8 in
D1=20 in
Optimal Cost = 573000 units. We can see that the cost for tree 11 is less than that of tree 1 (652000 units). Hence this spanning tree will be the starting point for the next iteration.

H2 = 205.9620 m
H3 = 201.9526m

H4 = 201.2744m
H6 = 197.6547m
H7 = 193.0536m
H5 = 191.4035m

3. Starting with spanning tree 11 now, the above procedure is repeated iteratively to find a spanning tree with a lower cost. The final spanning tree will give the optimal solution.

CODE FOR DISCRETE SEARCH METHOD:


```
clc
clear all
close all
```

PART 2 - SPANNING TREE 1

VARIABLE INITIALIZATION

```
D=[1, 2, 3 , 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24];

Q=[100,370,570,900,1020,1120];

k1=1:length(D);
for i=1:length(D)
    PD1(i)=(162.9*1000*(Q(1)/130)^(1.85))/D(i)^(4.87);
end
Cost1= 1000*[2,5,8,11,16,23,32,50,60,90,130,170,300,550];

k2=1:length(D);
for i=1:length(D)
    PD2(i)=162.9*1000*(Q(2)/130)^(1.85)/D(i)^(4.87);
end
Cost2=1000*[2,5,8,11,16,23,32,50,60,90,130,170,300,550];

k3=1:length(D);
for i=1:length(D)
    PD3(i)=162.9*1000*(Q(3)/130)^(1.85)/D(i)^(4.87);
end
Cost3=1000*[2,5,8,11,16,23,32,50,60,90,130,170,300,550];

k4=1:length(D);
for i=1:length(D)
    PD4(i)=162.9*1000*(Q(4)/130)^(1.85)/D(i)^(4.87);
end
Cost4=1000*[2,5,8,11,16,23,32,50,60,90,130,170,300,550];

k5=1:length(D);
for i=1:length(D)
    PD5(i)=162.9*1000*(Q(5)/130)^(1.85)/D(i)^(4.87);
```

```
end
Cost5=1000*[2,5,8,11,16,23,32,50,60,90,130,170,300,550];

k6=1:length(D);
for i=1:length(D)
    PD6(i)=162.9*1000*(Q(6)/130)^(1.85)/D(i)^(4.87);
end
Cost6=1000*[2,5,8,11,16,23,32,50,60,90,130,170,300,550];
```

SERIES MERGES

```
% SERIES MERGE OF 7&8
[PDeq1, Costeq1, Diaeq1]= SeriesMerge(k1,k2,D',D',PD1,PD2,Cost1,Cost2);

% SERIES MERGE OF (7,8)&6
keq1=1:length(Diaeq1);
[PDeq2, Costeq2, Diaeq2]= SeriesMerge(keq1,k3,Diaeq1,D',PDeq1,PD3,Costeq1,Cost3);

% SERIES MERGE OF (7,8,6)&5
keq2=1:length(Diaeq2);
[PDeq3, Costeq3, Diaeq3]= SeriesMerge(keq2,k4,Diaeq2,D',PDeq2,PD4,Costeq2,Cost4);

% SERIES MERGE OF (7,8,6,5)&3
keq3=1:length(Diaeq3);
[PDeq4, Costeq4, Diaeq4]= SeriesMerge(keq3,k5,Diaeq3,D',PDeq3,PD5,Costeq3,Cost5);

% SERIES MERGE OF (7,8,6,5,3)&1
keq4=1:length(Diaeq4);
[PDeq5, Costeq5, Diaeq5]= SeriesMerge(keq4,k6,Diaeq4,D',PDeq4,PD6,Costeq4,Cost6);
```

CHECKING HEAD CONSTRAINTS OF FINAL OPTIONS

```
for i=1:length(PDeq5)
    TotalPD(i)=PDeq5(i);
    if(TotalPD(i)<=20 && TotalPD(i)>0)
        Sol=i;
        OptDia=Diaeq5(Sol,:)
        OptPD=PDeq5(Sol)
        OptCost=Costeq5(Sol)

        H2=210-(162.9*1000*(Q(6)/130)^(1.85))/OptDia(6)^(4.87)
        H4=H2-(162.9*1000*(Q(5)/130)^(1.85))/OptDia(5)^(4.87)
        H6=H4-(162.9*1000*(Q(4)/130)^(1.85))/OptDia(4)^(4.87)
        H7=H6-(162.9*1000*(Q(3)/130)^(1.85))/OptDia(3)^(4.87)
        H5=H7-(162.9*1000*(Q(2)/130)^(1.85))/OptDia(2)^(4.87)
        H3=H5-(162.9*1000*(Q(1)/130)^(1.85))/OptDia(1)^(4.87)

% if(H6>=195 && H3>=190 && H4>=185 && H5>=180 && H7>= 190) % H2 is always satisfied if
these are satisfied -Use here for any general spanning tree after removing above if
        if(H6>=195)
            break;
        end
```

```
    end
end
```

```
OptDia =

    10    14    16    18    20    20


OptPD =

   19.6745


OptCost =

      652000


H2 =

  205.9620


H4 =

  202.5656


H6 =

  198.0647


H7 =

  194.6337


H5 =

  191.6780


H3 =

  190.3255
```

FUNCTION FOR SERIES MERGE:

```
function [PressureDropFinal, CostFinal, DiaFinal]=
SeriesMerge(k1,k2,D1,D2,PD1,PD2,Cost1,Cost2)
count=1;
for i=1:length(k1)
    for j=1:length(k2)
        Dia(count,:)=[D1(i,:),D2(j,:)];
```

```
        PDSer(count)=PD1(i)+PD2(j);
        CostSer(count)=Cost1(i)+Cost2(j);
        count=count+1;
    end
end
[PDord,PDind]=sort(PDSer','descend');
PDNew(1)=PDord(1);
CostNew(1)=CostSer(PDind(1));
DiaNew=Dia(PDind(1),:);
counter=2;

for i=1:length(PDind)
    Costord(i)=CostSer(PDind(i)); % Arranging cost
    if(i~=1)
        if(Costord(i)>CostNew(counter-1))
                   PDNew(counter)=PDord(i);
CostNew(counter)=CostSer(PDind(i));
DiaNew(counter,:)=Dia(PDind(i),:);
counter=counter+1;
        else % Delete i-1
            for l=counter-1:-1:1
                 if(Costord(i)<=CostNew(l))
     PDNew(l)=PDord(i);
CostNew(l)=CostSer(PDind(i));
DiaNew(l,:)=Dia(PDind(i),:);
counter=l+1;
                 end
            end
        end
    end
end

PressureDropFinal=PDNew';
CostFinal=CostNew';
DiaFinal=DiaNew;
end
```

FUNCTION FOR PARALLEL MERGE:

```
function [PressureDropFinal, CostFinal, DiaFinal]=
ParallelMerge(k1,k2,D1,D2,PD1,PD2,Cost1,Cost2)
i=1;
j=1;
count=1;
while (i<=length(k1) && j<=length(k2))
    if(PD1(i)>PD2(j))
        PDed(count)=PD1(i);
        ind=1;
    else
        PDed(count)=PD2(j);
        ind=2;
    end
        CostSer(count)=Cost1(i)+Cost2(j);
         Dia(count,:)=[D1(i,:),D2(j,:)];
         if(ind==1)
             i=i+1;
         else
```

```
            j=j+1;
        end
        count=count+1;

end
PressureDropFinal=PDed';
CostFinal=CostSer';
DiaFinal=Dia;
end
```

Published with MATLAB® R2014a

The required variables, order and type of merging are changed for the other spanning trees in the code and run again.