William Wijaya Lab #1 3035992465 Due: Friday 2/12 at 11:59pm

**Lab 1: Water Supply Network Optimization**

**Problem 1**

After trials and errors using excel, I generated the following table using excel:

|  |  |
| --- | --- |
| Deliver route | Amount of Water (ML) |
| Source 1 to Appletown | 0 |
| Source 1 to Berrytown | 0 |
| Source 1 to Cherrytown | 0 |
| Source 1 to Grapetown | 0 |
| Source 1 to Mangotown | 15 |
| Source 2 to Appletown | 0 |
| Source 2 to Berrytown | 0 |
| Source 2 to Cherrytown | 0 |
| Source 2 to Grapetown | 0 |
| Source 2 to Mangotown | 5 |
| Source 3 to Appletown | 9 |
| Source 3 to Berrytown | 3 |
| Source 3 to Cherrytown | 15 |
| Source 3 to Grapetown | 6 |
| Source 3 to Mangotown | 18 |
| Source 4 to Appletown | 21 |
| Source 4 to Berrytown | 7 |
| Source 4 to Cherrytown | 35 |
| Source 4 to Grapetown | 14 |
| Source 4 to Mangotown | 2 |

Cost: $200,145

Here is how I decided to distribute the water from source into each town. I broke it into steps as shown below:

1. I see that the common pipe merges source 1 and source 2, meaning we can use source 1 and source 2 with the total of 20 ML. And, both source 1 and source 2 offers the lowest hardness of water and the lowest price. So, I can’t waste source 1 and source 2. I decided to use all source 1 and source 2 into Mangotown. Since that the cost to deliver water to Mangotown is the most expensive, I decided to spend all supply from source 1 and source 2 into Mangotown. This means that the amount of supply from source 3 and source 4 into Mangotown can be decreased drastically, allowing to lower the cost price drastically.
2. Next, after putting all the supply from source 1 and source 2 into Mangotown, I decided to do some trial and error in excel that makes the amount of supply satisfies all constraints. I decided to prioritize in using source 1 in Mangotown because it provides cheapest cost. We know that the maximum hardness from each town is 1200 kg/ML, so I tried to try combination of number from source 3 and 4 into each town such that it satisfies maximum hardness constraint. After that, I decided to adjust the number further until it satisfies all constraints. Since that we can input formulas in excel for each cell, it is easy to try different combinations until all constraints are satisfied.
3. I decided to recheck again. If I increase the amount of water from source 3 to each town by 1, it violates the maximum hardness constraints. So, I stop my trial-and-error process in excel and summarized my result in a table as shown above.

Here are all constraints provided in a table to check whether the constraints are satisfied are not:

|  |  |
| --- | --- |
| Town | Daily need |
| Appletown | 30 |
| Berrytown | 10 |
| Cherrytown | 50 |
| Grapetown | 20 |
| Mangotown | 40 |

|  |  |
| --- | --- |
| Town | Hardness |
| Appletown | 1180 |
| Berrytown | 1180 |
| Cherrytown | 1180 |
| Grapetown | 1180 |
| Mangotown | 1188.75 |

|  |  |
| --- | --- |
| Source | Supply amount |
| 1 | 15 |
| 2 | 5 |
| 3 | 51 |
| 4 | 79 |

Common pipe constraint: 20 ML. Just sum up supply amount from source 1 and source 2 (15+5=20 ML). You can see that the hardness from Appletown until Grapetown is 1180 kg/ML. That is the maximum hardness I can get with different combination of amount of water from source 3 and source 4 in each town. The hardness from Mangotown is different because I use supply from all supply sources.

**Problem 2**

1. I decided to use summation and double index variables to make formulation much easier.

Decision variable:

Let be amount of water in ML distributed from source “i” to town “j” daily in 10 years, where i and j. Here how I define “j”:

|  |  |
| --- | --- |
| j | Town |
| 1 | Appletown |
| 2 | Berrytown |
| 3 | Cherrytown |
| 4 | Grapetown |
| 5 | Mangotown |

Parameters:

1. Let be the cost of delivering water ($/ML) from source source “i” to town “j” for all “i” and “j”
2. Let be the supply limit per day (ML) from source “i” for all “i”
3. Let be the hardness from source “i” for all “i”
4. Let be the daily water need (ML) from town “j” for all “j”
5. Now, I formulate the linear programming using sum:

Objective function:

Subject to:

Explanation:

* To calculate the cost in the objective function, we need to sum up the product of cost in ($/ML) and the amount of water in (ML). Unit-wise it makes sense to multiply the cost with amount of water ($/ML\*ML=$). Just sum up all the produce of cost and amount of water and we get the total cost with the unit of $.
* For the daily need constraint, we need to satisfy the daily need for each town using all available source “i”. We can’t send water below the daily need or above the daily need. If we send it below the daily need, then the daily need of each town is not satisfied. If we send above the daily need, then we just waste our effort to send excess amount of water and the excess water will not be used anyway.
* For the supply limit constraint, we can’t send amount of water to each town exceeding the supply limit because simply the resources are not available. However, we can send below the supply limit because it is not mandatory to use all the supply available. So, inequality constraint () is what we need to formulate the constraint.
* For the common pipe constraint, the amount of water from source 1 and source 2 sent to each town need to be 20 ML. We don’t need to use all 20 ML from source 1 and source 2, so inequality constraint () will be useful to formulate the constraint.
* For the maximum hardness constraint, each source has its hardness in (kg/ML). The hardness of each town cannot exceed 1200, that’s why the notation ( is necessary. Hardness can be formulated as and I just move to right hand side to make the constraint neater.
* For the nonnegativity constraint, we can not send negative amount of water. So, the amount of water needs to be nonnegative.

1. We have 20 decisions variables and 20 constraints in total. I split the equality constraint into two inequalities constraint. Hence, is expected to be 1x20 column vector, A matrix is expected to be 20x20 matrix, and b is expected to be 20x1 row vector. In MATLAB I define b as 1x20 column vector, I can transpose b if needed. Here is how I define , A, and b according to the LP I have formulated:

c:

365 375 355 360 370 425 430 435 440 420 705 720 690 725 730 2005 2020 2000 1990 2025

A:

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

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

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

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

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

-1 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0

0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0 0

0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0

0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0

0 0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1 0 0 0 0 -1

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

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

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

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

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

250 0 0 0 0 200 0 0 0 0 2300 0 0 0 0 700 0 0 0 0

0 250 0 0 0 0 200 0 0 0 0 2300 0 0 0 0 700 0 0 0

0 0 250 0 0 0 0 200 0 0 0 0 2300 0 0 0 0 700 0 0

0 0 0 250 0 0 0 0 200 0 0 0 0 2300 0 0 0 0 700 0

0 0 0 0 250 0 0 0 0 200 0 0 0 0 2300 0 0 0 0 700

b:

30 10 50 20 40 -30 -10 -50 -20 -40 15 10 55 85 20 36000 12000 60000 24000 48000

It is hard to see if I copy and paste it into word document. Perhaps, I recommend to see how I define c,A, and b in the MATLAB workspace.

1. I summarized the optimal result in a table:

|  |  |
| --- | --- |
| Deliver route | Amount of Water (ML) |
| Source 1 to Appletown | 0 |
| Source 1 to Berrytown | 0 |
| Source 1 to Cherrytown | 0 |
| Source 1 to Grapetown | 0 |
| Source 1 to Mangotown | 15 |
| Source 2 to Appletown | 0 |
| Source 2 to Berrytown | 0 |
| Source 2 to Cherrytown | 0 |
| Source 2 to Grapetown | 0 |
| Source 2 to Mangotown | 5 |
| Source 3 to Appletown | 9.375 |
| Source 3 to Berrytown | 3.125 |
| Source 3 to Cherrytown | 15.625 |
| Source 3 to Grapetown | 6.25 |
| Source 3 to Mangotown | 18.28125 |
| Source 4 to Appletown | 20.625 |
| Source 4 to Berrytown | 6.875 |
| Source 4 to Cherrytown | 34.375 |
| Source 4 to Grapetown | 13.75 |
| Source 4 to Mangotown | 1.71875 |

The minimum transportation cost is $200,000.78

As you can see in the table, the difference between my prediction and the optimal solution is not that significant. The only difference is the program can display the optimal solution more accurate than my prediction. The objective function has $145 difference compared to my prediction, which is high if we consider that this is daily cost.

Now, for the constraint, I will show a table listing all constraints and show whether the constraint is active or inactive:

|  |  |  |
| --- | --- | --- |
| Constraint | Value | Active or Inactive |
| Daily need of Appletown (+) | 0 | active |
| Daily need of Berrytown (+) | 0 | active |
| Daily need of Cherrytown (+) | 0 | active |
| Daily need of Grapetown (+) | 0 | active |
| Daily need of Mangotown (+) | 0 | active |
| Daily need of Appletown (-) | 0 | active |
| Daily need of Berrytown (-) | 0 | active |
| Daily need of Cherrytown (-) | 0 | active |
| Daily need of Grapetown (-) | 0 | active |
| Daily need of Mangotown (-) | 0 | active |
| Supply limit of S1 | 0 | active |
| Supply limit of S2 | -5 | inactive |
| Supply limit of S3 | -2.34375 | inactive |
| Supply limit of S4 | -7.65625 | inactive |
| Common pipe | 0 | active |
| Maximum Hardness of Appletown | 0 | active |
| Maximum Hardness of Berrytown | 0 | active |
| Maximum Hardness of Cherrytown | -7.28E-12 | active |
| Maximum Hardness of Grapetown | 0 | active |
| Maximum Hardness of Mangotown | -7.28E-12 | active |

The value is obtained by A\*x-b. If the value is 0, then the constraint is active, otherwise it is inactive. Treat since this is a very small number. Remember that the equality constraint is split into two inequality constraints: (+) and (-).

Before continuing further, I want to tell how to run the MATLAB program. First, press run. Then, after the program finished running for each question, press enter to continue to proceed into the next question.

|  |
| --- |
| %% 2    % Data    clear all,clc;    % Maximum hardness data  h1=250; %Source 1  h2=200; %Source 2  h3=2300; %Source 3  h4=700; %Source 3    % Daily water need  n1=30; %Appletown  n2=10; %Berrytown  n3=50; %Cherrytown  n4=20; %Grapetown  n5=40; %Mangotown    % Supply limit per day  s1=15; %Source1  s2=10; %Source2  s3=55; %Source3  s4=85; %Source4    % Matrix  O=ones(1,5); %ones 1x5 matrix  Z=zeros(1,5); %zeros 1x5 matrix    % lower bound  lb=zeros(20,1);    % Constructing "c^T" matrix    %Cost of providing water from source 1  c1=[365,375,355,360,370];  %Cost of providing water from source 2  c2=[425,430,435,440,420];  %Cost of providing water from source 3  c3=[705,720,690,725,730];  %Cost of providing water from source 4  c4=[2005,2020,2000,1990,2025];  %Concatenate matrix to c^T  c=[c1,c2,c3,c4];    % Constructing "A" Matrix    % Daily need of Appletown  T\_A=[1,0,0,0,0];  N1=[T\_A,T\_A,T\_A,T\_A];  % Daily need of Berrytown  T\_B=[0,1,0,0,0];  N2=[T\_B,T\_B,T\_B,T\_B];  % Daily need of Cherrytown  T\_C=[0,0,1,0,0];  N3=[T\_C,T\_C,T\_C,T\_C];  % Daily need of Grapetown  T\_G=[0,0,0,1,0];  N4=[T\_G,T\_G,T\_G,T\_G];  % Daily need of Mangotown  T\_M=[0,0,0,0,1];  N5=[T\_M,T\_M,T\_M,T\_M];  % Concatenate Daily need constraint  N=[N1;N2;N3;N4;N5;-1.\*N1;-1.\*N2;-1.\*N3;-1.\*N4;-1.\*N5];    % Supply limit of Source 1  S1=[O,Z,Z,Z];  % Supply limit of Source 2  S2=[Z,O,Z,Z];  % Supply limit of Source 3  S3=[Z,Z,O,Z];  % Supply limit of Source 4  S4=[Z,Z,Z,O];  % Concatenate supply constraint  S=[S1;S2;S3;S4];    % Common pipe constraint  P=[O,O,Z,Z];    % Maximum Hardness for Appletown  H1=[h1.\*T\_A,h2.\*T\_A,h3.\*T\_A,h4.\*T\_A];  % Maximum Hardness for Berrytown  H2=[h1.\*T\_B,h2.\*T\_B,h3.\*T\_B,h4.\*T\_B];  % Maximum Hardness for Cherrytown  H3=[h1.\*T\_C,h2.\*T\_C,h3.\*T\_C,h4.\*T\_C];  % Maximum Hardness for Grapetown  H4=[h1.\*T\_G,h2.\*T\_G,h3.\*T\_G,h4.\*T\_G];  % Maximum Hardness for Mangotown  H5=[h1.\*T\_M,h2.\*T\_M,h3.\*T\_M,h4.\*T\_M];  % Concatenate Maximum Hardness constraint  H=[H1;H2;H3;H4;H5];    % Concatenate "A" Matrix  A=[N;S;P;H];    % Constructing "b" Matrix    % Daily need matrix  n=[n1,n2,n3,n4,n5];    % Supply limit matrix  s=[s1,s2,s3,s4];    % Maximum Hardness matrix  h=1200.\*n;    % Concatenate to "b" matrix  b=[n,-1.\*n,s,20,h];    % Solve Linear Programming  options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options);    % Constructing a table summarizing the answer    nameS1=["Source 1 to Appletown","Source 1 to Berrytown",...  "Source 1 to Cherrytown","Source 1 to Grapetown",...  "Source 1 to Mangotown"];  nameS2=["Source 2 to Appletown","Source 2 to Berrytown",...  "Source 2 to Cherrytown","Source 2 to Grapetown",...  "Source 2 to Mangotown"];  nameS3=["Source 3 to Appletown","Source 3 to Berrytown",...  "Source 3 to Cherrytown","Source 3 to Grapetown",...  "Source 3 to Mangotown"];  nameS4=["Source 4 to Appletown","Source 4 to Berrytown",...  "Source 4 to Cherrytown","Source 4 to Grapetown",...  "Source 4 to Mangotown"];  Deliver\_route=[nameS1,nameS2,nameS3,nameS4]';  Amount\_of\_water\_delivered=x;  fprintf("The result for no 2 is displayed below in a table:\n")  result\_2=table(Deliver\_route,Amount\_of\_water\_delivered)  mincost\_2=fval;  fprintf("The minimum transportation cost is $%4.2f \n\n",mincost\_2)    %Checking for active constraints  fprintf("The table for constraints is shown below: \n")  activeValue=A\*x-b';  nameC1=["Daily need of Appletown (+)","Daily need of Berrytown (+)",...  "Daily need of Cherrytown (+)","Daily need of Grapetown (+)",...  "Daily need of Mangotown (+)"];  nameC2=["Daily need of Appletown (-)","Daily need of Berrytown (-)",...  "Daily need of Cherrytown (-)","Daily need of Grapetown (-)",...  "Daily need of Mangotown (-)"];  nameC3=["Supply limit of S1","Supply limit of S2","Supply limit of S3",...  "Supply limit of S4"];  nameC4=["Common pipe"];  nameC5=["Maximum Hardness of Appletown","Maximum Hardness of Berrytown",...  "Maximum Hardness of Cherrytown","Maximum Hardness of Grapetown",...  "Maximum Hardness of Mangotown"];  constraint=[nameC1,nameC2,nameC3,nameC4,nameC5]';  active\_or\_inactive=["active","active","active","active","active",...  "active","active","active","active","active",...  "active","inactive","inactive","inactive","active",...  "active","active","active","active","active"]';  tableConstraint=table(constraint,activeValue,active\_or\_inactive) |

**Problem 3**

1. Value of decision variables:

|  |  |
| --- | --- |
| Deliver route | Amount of water (ML) |
| Source 1 to Appletown | 0 |
| Source 1 to Berrytown | 0 |
| Source 1 to Cherrytown | 0.902439024 |
| Source 1 to Grapetown | 0 |
| Source 1 to Mangotown | 14.09756098 |
| Source 2 to Appletown | 0 |
| Source 2 to Berrytown | 0 |
| Source 2 to Cherrytown | 0 |
| Source 2 to Grapetown | 0 |
| Source 2 to Mangotown | 5 |
| Source 3 to Appletown | 10.03125 |
| Source 3 to Berrytown | 3.34375 |
| Source 3 to Cherrytown | 16.97256098 |
| Source 3 to Grapetown | 5.75 |
| Source 3 to Mangotown | 18.90243902 |
| Source 4 to Appletown | 18.46875 |
| Source 4 to Berrytown | 6.15625 |
| Source 4 to Cherrytown | 29.625 |
| Source 4 to Grapetown | 13.25 |
| Source 4 to Mangotown | 0 |

The minimum transportation cost is $181877.56.

The minimum transportation is significantly lower if the demand is decreased by 5%, meaning that the daily need/demand constraint is very sensitive when perturbed a little. The objective function is expected to change since the demand constraint is active. Other than that, it seems that supply from source 1 is not solely spent on Mangotown, but some of them were spent into Cherrytown.

|  |
| --- |
| %% 3a    display("Press enter to continue no 3a:");pause;clc;    %Decrease the demand of water by 5% from each town  n=0.95.\*n;    b=[n,-1.\*n,s,20,h];  options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options);  Amount\_of\_water\_delivered=x;  fprintf("The result for no 2 is displayed below in a table:\n")  result\_3a=table(Deliver\_route,Amount\_of\_water\_delivered)  fprintf("The minimum transportation cost is $%4.2f \n",fval) |

1. The program stopped with no optimal solution generated, no feasible solution found. It means that the linear problem is infeasible, therefore no optimal solution will be generated

|  |
| --- |
| %% 3b  display("Press enter to continue no 3b:");pause;clc;    % Increase the demand of water by 5% from each town  n=[n1,n2,n3,n4,n5];  n=1.05.\*n;    b=[n,-1.\*n,s,20,h];  options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options)  fprintf("The result for no 3b is not feasible,")  fprintf(" i.e., no optimal sol found \n") |

1. The program stopped with no optimal solution generated. The program told us that no feasible solution found. This means that the Linear Problem is infeasible, meaning no optimal solution will be generated.

|  |
| --- |
| %% 3c  display("Press enter to continue no 3c:");pause;clc;  n=[n1,n2,n3,n4,n5];  s=[s1,s2,95,75];  b=[n,-1.\*n,s,20,h];  options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options)  fprintf("The result for no 3c is not feasible,")  fprintf(" i.e., no optimal sol found \n") |

**Problem 4**

First and foremost, we can see that the Kiwi Inc is located close to Berrytown. This means that the transportation cost to Kiwi Inc is roughly the same as Berrytown. So, we just have to know to cost of sending the water to Berrytown per 1 ML. Since Kiwi Inc offered a deal to buy 1ML for the price of $1500 per day and the county does not want to lose any money, the cost of sending the water to Berrywater per 1 ML should be less or equal to 1500.

After programming was done, it can be concluded that the cost of sending the water to Berrytown per 1 ML is $1613.75, which exceeds $1500. This means that it is not recommended for the county to accept the offer, since this generates loss to the county.

|  |
| --- |
| %% 4    % First, we note that sending water to Kiwi is the same as sending it to  % Berrytown. So, we just need to know what is the price/ML to send water to  % Berrytown. Now, from the table number 2, we can extract the amount of  % water delivered from source i to Berrytown (i=1,2,3,4)    display("Press enter to continue no 4:");pause;clc;  s=[s1,s2,s3,s4];  b=[n,-1.\*n,s,20,h];  options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options);    % Cost of sending water to Berrytown per ML  cost\_B=(x(2)\*c(2)+x(7)\*c(7)+x(12)\*c(12)+x(17)\*c(17))/n2;  fprintf("Cost of sending water to Berrytown per ML is $%4.2f \n",cost\_B)    if cost\_B<1500  fprintf("Accept Kiwi Inc's offer since this is a profit ")  fprintf("(price/ML<1500) \n")  else  fprintf("Reject Kiwi Inc's offer since this is a loss ")  fprintf("(price/ML>1500) \n")  end |

**Problem 5**

Value of decisions variable:

|  |  |
| --- | --- |
| Deliver route | Amount of Water (ML) |
| Source 1 to Appletown | 0 |
| Source 1 to Berrytown | 0 |
| Source 1 to Cherrytown | 0 |
| Source 1 to Grapetown | 0 |
| Source 1 to Mangotown | 17 |
| Source 2 to Appletown | 0 |
| Source 2 to Berrytown | 0 |
| Source 2 to Cherrytown | 0 |
| Source 2 to Grapetown | 0 |
| Source 2 to Mangotown | 3 |
| Source 3 to Appletown | 9.375 |
| Source 3 to Berrytown | 3.125 |
| Source 3 to Cherrytown | 15.625 |
| Source 3 to Grapetown | 6.25 |
| Source 3 to Mangotown | 18.21875 |
| Source 4 to Appletown | 20.625 |
| Source 4 to Berrytown | 6.875 |
| Source 4 to Cherrytown | 34.375 |
| Source 4 to Grapetown | 13.75 |
| Source 4 to Mangotown | 1.78125 |

The minimum transportation cost is $199995.42. Since source 1 constraint is active constraint, the objective function will change. By adding a small structure to source 1, we gain profit of $5.36/day ($200,000.78-$199995.42=$5.36). Since this is a profit, it is recommended to add a small structure to source 1 and therefore increasing the supply limit for source 1 and generated profit.

|  |
| --- |
| %% 5    display("Press enter to continue no 5:");pause;clc;  % The program is the same as no 2, but the only changes are:  % I increase s1 by 2  % And, I added the mincost\_5 by (5000/365)    s1=s1+2;  s=[s1,s2,s3,s4];  b=[n,-1.\*n,s,20,h];    % Solve Linear Programming  options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options);    % Constructing a table summarizing the answer    Amount\_of\_water\_delivered=x;  fprintf("The result for no 5 is displayed below in a table: \n")  result\_5=table(Deliver\_route,Amount\_of\_water\_delivered)  mincost\_5=fval+(5000/365);  fprintf("The minimum transportation cost is $%4.2f \n",mincost\_5)  difference=mincost\_2 - mincost\_5;  fprintf("By adding a small structure to source 1,")  fprintf(" we gain profit of $%4.2f \n",difference)  fprintf("Choose this option to gain the profit \n") |

**Problem 6**

Source 3 constraint is inactive, meaning the objective function won’t change when the source 3 constraint is perturbed a bit. Let’s say that we decided to increase supply limit from source 3 by 1. The program still generated the same optimal solution and the same objective function as number 2.

The values of decision variables:

|  |  |
| --- | --- |
| Deliver route | Amount of Water (ML) |
| Source 1 to Appletown | 0 |
| Source 1 to Berrytown | 0 |
| Source 1 to Cherrytown | 0 |
| Source 1 to Grapetown | 0 |
| Source 1 to Mangotown | 15 |
| Source 2 to Appletown | 0 |
| Source 2 to Berrytown | 0 |
| Source 2 to Cherrytown | 0 |
| Source 2 to Grapetown | 0 |
| Source 2 to Mangotown | 5 |
| Source 3 to Appletown | 9.375 |
| Source 3 to Berrytown | 3.125 |
| Source 3 to Cherrytown | 15.625 |
| Source 3 to Grapetown | 6.25 |
| Source 3 to Mangotown | 18.28125 |
| Source 4 to Appletown | 20.625 |
| Source 4 to Berrytown | 6.875 |
| Source 4 to Cherrytown | 34.375 |
| Source 4 to Grapetown | 13.75 |
| Source 4 to Mangotown | 1.71875 |

The minimum transportation cost for no 6 is $200000.78

The minimum transportation cost for no 2 is $200000.78

The minimum cost for no 2 and 6 are the same, with a difference of 0.00.

Hence, do not pay for any increase of supply limit 3 because the minimum cost will not change

|  |
| --- |
| %% 6  display("Press enter to continue no 6:");pause;clc;    % Maximum hardness data  h1=250; %Source 1  h2=200; %Source 2  h3=2300; %Source 3  h4=700; %Source 3    % Daily water need  n1=30; %Appletown  n2=10; %Berrytown  n3=50; %Cherrytown  n4=20; %Grapetown  n5=40; %Mangotown    % Supply limit per day  s1=15; %Source1  s2=10; %Source2  s3=56; %Source3, add this by one  s4=85; %Source4    s=[s1,s2,s3,s4];  n=[n1,n2,n3,n4,n5];  b=[n,-1.\*n,s,20,h];    options = optimoptions('linprog','Algorithm','dual-simplex');  [x,fval,exitflag,output,lambda]=linprog(c,A,b,[],[],lb,[],options);    Amount\_of\_water\_delivered=x;  fprintf("The result for no 6 is displayed below in a table:\n")  result\_6=table(Deliver\_route,Amount\_of\_water\_delivered)  mincost\_6=fval;  fprintf("The minimum transportation cost for no 6 is $%4.2f \n",mincost\_6)  fprintf("The minimum transportation cost for no 2 is $%4.2f \n",mincost\_2)  fprintf("The minimum cost for no 2 and 6 are the same,")  different= mincost\_6-mincost\_2;  fprintf(" with a difference of %4.2f \n", different)  fprintf("So, do not pay for any increase of supply limit 3 because the")  fprintf(" minimum cost will not change \n\n") |