        - Data source inflow = None
    - Hit "ok"
- Connect the nodes with arcs (arcs should follow the river).
    - click on the black arc arrow and then the nodes you want to connect, upstream node first, then downstream.
    - name them as you'd like. You can use canal numbers (C-31 and C-35). We often use convention [up_node].[down_node], such as 100.120.
    - you may want to select "hide name" to keep the schematic from getting too crowded.
- Save! (also saves when run).

Step 3: Add reservoir operations
- Add two target statements to the ocl file which specify the desirable storage in each Lake.
- Storage is used instead of stage because the lp solves for storage, flows, and deliveries.
- For Lake Toho (dstorage110), set the target stage to 55 ft all year (condition: default).
    - The "d" is used in front of "storage" to specify a *decision* variable, a variable that will be solved for in the lp.
    - You can convert elevation to storage using "Elev_to_Stor{ [NODE], [ELEVATION] }." See the OCL menu and manual pg 233.
- For East Lake Toho (dstorage100):
    - 58 ft from 1/1 to 6/1 (julian = 1 to 153, julian days include 2/29)
    - 56.5 ft from 6/2 to 11/1 (julian = 154 to 306)
    - 58 ft from 11/2 to 12/31 (julian = 155 to 366)
    - Note: Later, we will see how to make smooth transitions with pattern tables, but here we're practicing conditional target statements.
- Priority = 1 – Priorities are used for multiple solves (tells OASIS which simulation commands to include in each solve). OKISS, the LP solver, solves once for each time step, so priority should *always* be 1.
- You need penalties for going above and below this stage. Since these are currently the only weights in the program, you only need to worry about setting them relative to each other. One suggestion: give East Lake Toho's target penalties of 20 and Lake Toho's 10. In this way, if one of the stages must deviate from the target, it'll be Lake Toho's.
- Naming the target is optional but helps in debugging (special output files use these names, as we'll discuss later).
- Use the OCL menu in vedit and pg 207 of the manual. Working target statements are given at the end of this worksheet.
- Save the file.

Step 4: Run the model and view output.
- Run the model (under run menu or button on set up tab).
- Create plots to view the results.

- Open "Quickview" either from output menu or button on "setup" tab.
- Select the following options in the dialog box.
  - Choose elevation for East Lake Toho.
  - Check "Save with alternate file name" and enter name for plot file.
  - Hit "Display" button.
  - Scroll through the plot to see entire record.
- Repeat the process for Lake Toho (with a different file name).

Step 5: Make a new run with a maximum flow limit.

- Copy the current run under the file menu – you must close "_main.ocl" to copy run.
- Set a maximum flow on arc 110.130.
  - On the "arc" button on the "arc" tab, set max flow for this arc to "pattern" (you may have to scroll right).
  - On the maximum flow button on the arc tab set the maximum flow by entering the following:

US Number	DS Number	Units	Month	Day	Max Flow
110	130	cfs	1	1	3875
110	130		12	31	3875

- Run the program.
- Verify that the arcflow does not exceed the maximum.
  - In Quickview, create a plot of the flow in this arc.
    - Convert units from acft (which is acre-feet/day in the case of flow) to cfs.
    - be sure to name file.
  - Open "Tables" dialog box under the Output menu.
    - Highlight the file you just created and click "Edit File(s)".
    - This is the onevar file that creates plots and tables. It uses many of the same functions as OCL. We are going to add a second line to this plot: a horizontal line at 3875 cfs.
    - Copy and paste from "Table" to "}" right beneath it (before :END:).
    - Change "Convert_Units{
      flow110.130,
      ACFT, CFS }"
      to "3875"
      Note the "convert_units" function, which comes in handy in OCL
      and onevar files.
    - Save it.
  - Open the "plots" dialog box, highlight the file, and click "view output" button.
    - make sure the flow never exceeds the max.
    - make any changes to the formatting of the plot you like and save.

- See the difference in the two runs by plotting them together.
  - Open the "plots" dialog box, highlight both runs, and all three plots.
  - View the results. Note that in the arc flow plot, only the first line (arc flow) is shown (not the max flow).
  - To see lines that are over-plotted, change the line thicknesses.

Step 6: Change the weights
- Create a new run.
- Open _*main.ocl* and change the weights on the target statements so that the East Toho weights are *less than* the Toho weights.
- Run and program and view results in the plot files you have created.

If you have additional time, play: add new operating rules, minimum flow requirements, changes to the SAE table, etc. For example, make the target stages seasonal. Make changes in a new run so you can compare the results with the runs you've done.

SET AND TARGET COMMANDS:

```
Set: inflow100
{
  condition: default  // do it all the time
     value: timesers(100/inflow) + timesers(Hart_to_EastToho/arcflow)
}
```

OR

```
Set: inflow100  {value: timesers(100/inflow) + timesers(Hart_to_EastToho/arcflow)}
```

```
Target TohoStage: dstorage110
{
  condition: default
   priority: 1
   penalty+: 10
   penalty-: 10
     value: Elev_to_Stor{ 110, 55 }
}
```

```
Target EastTohoStage: dstorage100
{
  condition: julian < 154
   priority: 1
   penalty+: 20
   penalty-: 20
     value: Elev_to_Stor{ 100, 58 }
```

```
  condition: julian < 306
   priority: 1
   penalty+: 20
   penalty-: 20
      value: Elev_to_Stor{ 100, 56.5 }

  condition: default
   priority: 1
   penalty+: 20
   penalty-: 20
      value: Elev_to_Stor{ 100, 58 }
}
```

OR

```
Target EastTohoStage: dstorage100
{
  condition: (julian > 153) and (julian < 306)
   priority: 1
   penalty+: 20
   penalty-: 20
      value: Elev_to_Stor{ 100, 56.5 }

  condition: default
   priority: 1
   penalty+: 20
   penalty-: 20
      value: Elev_to_Stor{ 100, 58 }
}
```

<u>Part 2: Changing a Flow Target</u>

During the model simulation, OASIS processes through a set of operating rules and makes decisions based on the rules. Some of the rules require constant values that may be found in the program under the various tabs. Under the OCL tab, for example, there are constants used for direct substitutions into OCL code and constants used by lookup functions that use a constant based on the specified input. Currently, the flow target at Montague is 1750 cfs. Let's experiment with the flow target and see how our manipulation of the operating rules translates into quantifiable change in the behavior of the model.

1. We want to compare output for the changes we make with output for the initial settings, so run the Simbase run as is to make sure we have output for the initial settings.
    a. Make sure the open run is "Simbase"
    b. "Run" → "Run OASIS Model"

2. Copy the "Simbase" run into "Training1" so that if we make a huge mistake, we won't hurt anything other than our training run. Copying runs is always a good idea before making substantial changes because over multiple copies you build a series of reference points that are useful for reverting back or comparing performance.
   a. "File" → "Copy Run" for Simbase into "Training1"

3. Modify the normal conditions flow for Montague to 1850 cfs from 1/1 to 12/31.
   a. "OCL" tab → select "OCL Pattern" → find "MntaguNormal_Cfs"

4. Save your changes and run the model.

5. Now, so we can see the impact our changes made, let's plot the flow at Montague for the period of time the model ran.
   a. "Output" → "Quick View"
      i. Arc Output for 235.992
      ii. Convert units from MG to CFS
      iii. Save with alternate filename "flow_at_montague"
      iv. "Display"

6. Now that we see the flow for our Training1 run, let's compare it with Simbase.
   a. "Output" → "PLOTS"
   b. Hold "Ctrl" key and select Training1 and SimBase to compare them for the plot "flow_at_montague.mdb" → "View Output"
   c. The flows are similar, so in order to see both lines clearly:
      i. "Edit" → "SimBase" → "LINE ATTRIBUTES…"
         1. Set "Width" to 3

7. It would be helpful to verify the Montague flow target is set to 1850, so we will add another variable onto the chart by modifying the .1v file that determines what variables are to be displayed and how to display them. Save and close the chart.
   a. "Output" → "TABLES" → select "flow_at_montague.mdb" → "Edit File(s)"
   b. Select and copy from "Table {" to corresponding "}" and paste between the "}" and ":END:"
   c. Replace the value with "convert_units { _Montaguetarget , mg , cfs }" to plot the flow target at Montague in cfs. Save and close the file.

8. Let's view our addition, but instead of viewing the chart, we will look directly at the data.
   a. "Output" → "TABLES"
   b. Hold "Ctrl" key and select Training1 and SimBase to compare them for the table "flow_at_montague.1v"
   c. The keystroke "Ctrl" + "End" will take you to the last row of data, where you can see the average, min, and max for each column.
   d. If the Montague flow target column in Training1 is not 1850 for the entire period, something has been done incorrectly.

9. It would be practical to check on how the reservoir storage has been affected by our changes.
    a. We will make a cumulative frequency distribution for the storage
        i. Quickview node 100 for the storage variable
            1. Set the plot sorting from TimeSeries to Probability
            2. Save with alternate filename "storage"
    b. Open storage.1v for editing from the TABLES window
    c. Edit the value so that it includes the other two reservoirs. This should be a sum of storage100, storage120, and storage215
    d. View the plot "storage.1v" for "Training1" and "Simbase"

10. Prepare the chart and copy into a document.
    a. Give the chart a title, legend, and verify the axis labels are correct.
        i. Use the Edit menu, or double click onto objects on the chart to make your changes.
    b. "File" → "Copy to Clipboard"
    c. Paste into Word / Open Office Writer / etc.

11. Answer the following question:
    a. Does the chart make sense considering the modification we made? Explain.

<u>Part 3: Changing Reservoir Pool Elevations</u>

Flood pools are maintained in reservoirs to mitigate flood damage. When large precipitation events occur, the voids purposefully left in the reservoir will fill from runoff and inflows. In theory, the amount of empty space filled is the amount of water kept from spilling over the top of the reservoir, had it been full, and flowing downstream. The water that is prevented from spilling may have worsened flooding had it flowed downstream. Some explanation of reservoir terminology is necessary because we will be modifying the levels of desired storage in the reservoir. Figure 3 illustrates the important reservoir water levels to be aware of.

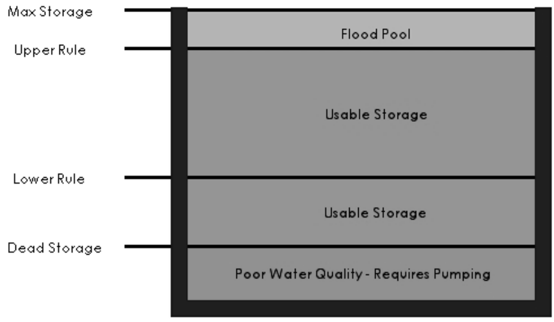

Figure 3. Reservoir storage levels and zones defined in OASIS model.

We will be lowering the values that determine the Upper Rule in order to increase the size of the Flood Pool.

1. Copy the "Training1" run into "Training2".

2. Lower the Upper Rule for reservoirs 100, 120, and 215.
    a. "Node" tab → select "Reservoir Rules"
    b. Decrease the upper rule by 25% for nodes 100, 120, and 215 from 1/1 to 12/31.

3. Save and run the model.

4. Prove to yourself that the upper rules have changed.
    a. Quickview node 100 for the upper rule variable.
    b. Open Quickview.1v for editing and add tables to display the upper rule for nodes 120 and 215.
    c. View the table output for Quickview.1v, the upper rules should reflect the changes made.

5. It would be prudent to make sure that decreasing the upper rules does not leave the reservoirs at risk for running too low on water. Compare Training1 and Training2 for the plot "storage" that was created earlier. Zoom in to see if there are any differences between the frequencies with which reservoir storages are at their lowest.
    a. "Edit" → "X Axis" → "Axis"
        i. From 0 to 10
        ii. Step of 1

6. Prepare the chart and copy into a document.

7. Answer the following question:
    a. Does the chart make sense considering the modification we made? Explain.

8. Since our goal was flood mitigation, check the flow for a downstream arc and see if the flood pulses have reduced. Flood pulses produce spikes in the flow record.
    a. QuickView arc output for 135.140, which is an area that has experienced flooding.
        i. Save as "flood_control_flow"
    b. Plot "flood_control_flow" to show Training1 and Training2.
        i. Show the period from 6/26/96 to 6/21/97.
        ii. Increase the width of the line in the background so that both lines are clearly visible.
        iii. Prepare the chart and copy into a document.
    c. Quickview node 100 for the variable storage.
        i. Convert MG to BG.
        ii. Save as "flood_control_storage"
        iii. Edit the .1v file:

1. To show the sum of storages for nodes 100, 120, and 215.
2. Add a table that sums the upper rule for nodes 100, 120, and 215.

d. Plot "flood_control_storage" for Training1.
   i. Show the period from 6/26/96 to 6/21/97.
   ii. Prepare the chart and copy into a document. Be sure that the title identifies the run you have plotted.

e. Plot "flood_control_storage" for Training2.
   i. Show the period from 6/26/96 to 6/21/97.
   ii. Prepare the chart and copy into a document.

9. Answer the following questions:
   a. What storage is being utilized when the storage line rises above the upper rule line?
   b. Looking at the highest flow event during the period 6/26/96 – 6/21/97, how much water was caught in the reservoirs?
   c. Which set of operating rules generally results in lower flood pulses?
      i. Use the plots to explain how you came to your conclusion.

<u>Part 4: Setting Minimum Flows</u>

The trout fisheries on the Delaware River owe their existence to cold water releases from reservoirs in the Catskills system. Trout thrive in waters low in temperature and turbidity. Ensuring that a minimum quantity of water is released every day is necessary to maintain low in-stream temperatures which preserve trout habitat. We will be setting releases to maintain a cold water minimum at all times.

1. Copy the "Training2" run into "Training3".

2. Tell the model that the min flow will be set in the OCL code.
   a. "Arc" tab → "Arc"
      i. For "PepactonRel" set Min Flow to OCL

3. Add a weight for the arc 100.105
   a. "Arc" tab → "Arc Weights"
   b. Add a weight of 4050 for upstream 100 and downstream 105 with a priority of 1.
      i. Weights and their importance are explained in the OASIS manual on page 20.
      ii. Priority refers to when the weight is used by the model. When there are multiple solves for each time step (in our model this is **not** the case) those weights with priority 1 are used for the first solve. Other weights are included in solves 2,…, *n*.

4. Set the min_flow for node 100.105 to come from a previously specified pattern.
   a. "OCL" tab → double click on "set_min_flows.ocl"

   b. At the end of the file, add a Set command (OASIS manual page 210) that sets the value to be the pattern "PepctnR1Norm_Cfs"
      i. Use nearby set commands as a guide for proper syntax.

5. Save and run the model.

6. Plot the flow for arc 100.105 and compare Training2 with Training3.

7. Keep in mind that because of the modifications made, the minimum flow for "Training3" must be released, where previously it may or may not have been released.
   a. Find a period where differing behavior of the two lines is exhibited.
   b. Prepare the chart and copy into a document.

8. Answer the following questions:
   a. What difference do you see between Training2 flow and Training3 flow?
   b. What do you think is the significance of the difference?

<u>Part 5: Evaluating Alternative Operation Plans with Performance Measures</u>

Now that you are proficient with the OASIS program and its plot making capabilities, create plots of your own performance measures. Prepare the plots and copy into a document. Post the document to the class discussion board. Note that it may not be possible to convert all performance measures from Excel® to OASIS (some will need Excel®).

References

Cardwell, H. E. and Lorie, M.A. (2006). "Collaborative Modeling for Water Management," *Southwest Hydrology*, 5(4), 26-27.

Keys, A.M. and Palmer, R.N. (1995). "An assessment of shared vision model effectiveness in water resources planning," *Integrated Water Resources Planning for the 21st Century*, Proceedings of the 22nd Annual Water Resources Planning and Management Conference. M.F. Dominica, ed. ASCE: Washington, D.C., 532-535.

Loucks, D.P. (1990). "Analytical Aids to Conflict Management," in *Managing Water-Related Conflicts: The Engineers Role*, W. Viessman and T.T. Smerdon, eds., ASCE: NY, 23-37.

National Research Council (2006). *Review of the Lake-Ontario-St. Lawrence River Studies*. Washington, D.C.: The National Academies Press. 148 pp.

Palmer, R.N. and Keys, A.M. (1993). "Empowering stakeholders through simulation in water resources planning," In *Water Management in the '90s*, Proceedings of the 20th Annual Water Resources Planning and Management Conference, K. Hon, ed. ASCE: Washington, D.C., 451-454.

Palmer, R.N., Werick, W.J., MacEwan, A., and Woods, A.W. (1999). “Modeling Water Resources Opportunities, Challenges and Trade-Offs: The Use of Shared Vision Modeling for Negotiation and Conflict Resolution,” In *Preparing for the 21st Century, Proceedings of the 29th Annual Water Resources Planning and Management Conference*. E.M. Wilson, ed. ASCE: Washington, D.C., 1.

Sheer, D.P., Baeck, M.L., and Wright, J.R. (1989). “The Computer as Negotiator,” *JAWWA*, 81(2), 68-73.

Theissen, E.M. and Loucks, D.P. (1992), “Computer assisted negotiation of multiobjective water resources conflicts,” *Water Resources Bulletin*, 28(1), 163-177.

Werick, W.J. and Whipple, W. (1994). *Managing Water for Drought*, IWR Report 94-NDS-8, Institute for Water Resources, U.S. Army Corps of Engineers, Alexandria, VA. 210 pp.