Optimizing LRGV Water Allocation

based on:

Characklis, Gregory W., Ronald C. Griffin, and Philip B. Bedient. “Improving the Ability of a Water Market to Efficiently Manage Drought.” *Water Resources Research* 35 (March 1999): 823-31.

Set-Up

We begin by establishing demand, cost, and loss parameters for all sectors. Water units throughout are 1000s of af. Prices and costs and values are per 1000af, except in the very last results table. We solve for unknown "A_i" coefficients using presumed elasticities and known demand points. p represents price at the expansion point, which may or may not be the same as C which represents constant mc of processing for the forthcoming opt analysis. Five sectors are present. ALL data should be entered in the **pm** matrix.

```
In[1]:= lbl1 = {{"Sector", "Ai", "ϵi", "qi", "pi", "Li", "Ci"}};
        pm = {{"muni-sm",   0., -.32, 40.0, 440 000., 0.2, 440 000.},
             {"muni-lg",   0., -.32, 100.0, 400 000., 0.1, 400 000.},
             {"ag-field",  0., -.7, 547.5, 20 000., 0.20, 20 000.},
             {"ag-veget",  0., -.4, 371.9, 24 000., 0.20, 24 000.},
             {"ag-citrus", 0., -.4, 227.5, 22 000., 0.20, 20 000.}};
        Do[
          pm[[i, 2]] = pm[[i, 4]] * (1. - pm[[i, 6]]) / (pm[[i, 5]] ^ pm[[i, 3]]),
          {i, 1, 5}
         ];
        lbled1 = Join[lbl1, pm];
        TableForm[lbled1]
```

```
Out[5]//TableForm=
        Sector       Ai          ϵi       qi       pi          Li      Ci
        muni-sm      2046.7      -0.32    40.      440 000.    0.2     440 000.
        muni-lg      5583.44     -0.32    100.     400 000.    0.1     400 000.
        ag-field     448 947.    -0.7     547.5    20 000.     0.2     20 000.
        ag-veget     16 811.3    -0.4     371.9    24 000.     0.2     24 000.
        ag-citrus    9932.09     -0.4     227.5    22 000.     0.2     20 000.
```

Compute Demand, Total Benefit, and Marginal Net Benefit Functions

Some intermediate functionals are computed here for verification or later use. The q_i function is simple demand for processed water by sector i. The remaining functions all relate to raw water. p_i is inverse demand and is expressed as a function of Q rather than Q_i for ease of later plotting. Due to nonconvergence of the integral under the demand function (demand is asymptotic to price axis), it is not possible to wholy express total benefits as a function of Q. Only the TB and TC columns are used is the forthcoming optimization work.

```
In[6]:= lbl2 = {{"Sector", "q_i", "p_i", "TB(Q_i)-Trunc(i)", "TC(Q_i)", "MNB(Q_i)"}};
        unkn = {{Q1, Q2, Q3, Q4, Q5}};
        funct = Table[0., {i, 5}, {j, 6}];
        Do[
            {funct[[i, 1]] = pm[[i, 1]],
            funct[[i, 2]] = pm[[i, 2]] * p^pm[[i, 3]],
            funct[[i, 3]] = ((1. - pm[[i, 6]]) * Q /
                            pm[[i, 2]])^(1. / pm[[i, 3]]),
            funct[[i, 4]] = pm[[i, 2]]^(-1. / pm[[i, 3]]) * (pm[[i, 3]] / (pm[[i, 3]] + 1.)) *
              (unkn[[1, i]] * (1. - pm[[i, 6]]))^((pm[[i, 3]] + 1.) / pm[[i, 3]]),
            funct[[i, 5]] = pm[[i, 7]] * unkn[[1, i]] * (1. - pm[[i, 6]]),
             funct[[i, 6]] = D[funct[[i, 4]], unkn[[1, i]]] - pm[[i, 7]] * (1. - pm[[i, 6]])
            }, {i, 1, 5}
          ];
        lbled2 = Join[lbl2, funct];
        TableForm[lbled2]
```

```
Out[11]//TableForm=
     Sector       q_i        p_i              TB(Q_i)-Trunc(i)       TC(Q_i)        MNB(Q_i)

                  2046.7     4.46571×10^10      1.68121×10^10                                     3.57256×1
     muni-sm      ------     -------------    - -------------      352 000. Q1    - 352 000. + -----------
                  p^0.32     Q^3.125            Q1^2.125                                          Q1^3.125

                  5583.44    7.11312×10^11      3.01261×10^11                                     6.40181×1
     muni-lg      -------    -------------    - -------------      360 000. Q2    - 360 000. + -----------
                  p^0.32     Q^3.125            Q2^2.125                                          Q2^3.125

                  448 947.   1.63308×10^8       3.04841×10^8                                      1.30646×10^8
     ag-field     --------   ------------     - ------------       16 000. Q3     - 16 000. +  ------------
                  p^0.7      Q^1.42857          Q3^0.428571                                       Q3^1.42857

                  16 811.3   6.40143×10^10      3.41409×10^10                                     5.12114×10
     ag-veget     --------   -------------    - -------------      19 200. Q4     - 19 200. +  ----------
                  p^0.4      Q^2.5              Q4^1.5                                            Q4^2.5

                  9932.09    1.71742×10^10      9.15957×10^9                                      1.37393×10
     ag-citrus    -------    -------------    - ------------       16 000. Q5     - 16 000. +  ----------
                  p^0.4      Q^2.5              Q5^1.5                                            Q5^2.5
```

Optimize

The outputed table for this section corresponds to WRR Table 3. Here we have to ignore the resultant values provided for the objective function below. They exclude some total benefits due to nonconvergence of integral under Cobb - Douglas demands. The omission of a portion of total benefits does not invalidate the optimization results. This code pertains to situations in which no sector becomes water - satiated. Absent a choice of "Method" in the NMaximize command, the procedure does not converge. Available Methods include "NelderMead", "DifferentialEvolution", "SimulatedAnnealing", and "RandomSearch". Only the last two of these work here.

In[12]:=

```
obj = funct[[1, 4]] - funct[[1, 5]] + funct[[2, 4]] - funct[[2, 5]] + funct[[3, 4]] -
    funct[[3, 5]] + funct[[4, 4]] - funct[[4, 5]] + funct[[5, 4]] - funct[[5, 5]];
excesswater = Q1 + Q2 + Q3 + Q4 + Q5 - availQ;
wvector = {Q1, Q2, Q3, Q4, Q5};
results =
   {{"Avail Q", "MNB", "Q1-sm", "Q2-lg", "Q3-fld", "Q4-veg", "Q5-cit", "summd Q"}};
Do[
{answ = NMaximize[{obj, {excesswater == 0, Q1 ≥ 0., Q2 ≥ 0., Q3 ≥ 0., Q4 ≥ 0., Q5 ≥ 0.}},
      wvector, Method → SimulatedAnnealing],
     answ2 = { {availQ, funct[[3, 6]] / 1000., Q1, Q2, Q3, Q4, Q5, Q1 + Q2 + Q3 + Q4 + Q5}} /.
      Join[Part[answ, 2]],
  results = Join[results, answ2]},
     {availQ, 800, 1300, 100}
];
TableForm[results]
```

Out[17]//TableForm=

```
Avail Q    MNB          Q1-sm     Q2-lg     Q3-fld    Q4-veg    Q5-cit    summd Q
800        31.0859      38.9313   97.3845   257.184   253.028   153.472   800.
900        19.8481      39.304    98.2973   311.273   279.967   171.159   900.
1000       12.2373      39.565    98.936    367.867   305.329   188.303   1000.
1100       6.84491      39.7542   99.3991   426.693   329.193   204.961   1100.
1200       2.88374      39.8957   99.745    487.536   351.641   221.183   1200.
1300       -0.112292    40.0041   100.01    550.206   372.773   237.007   1300.
```

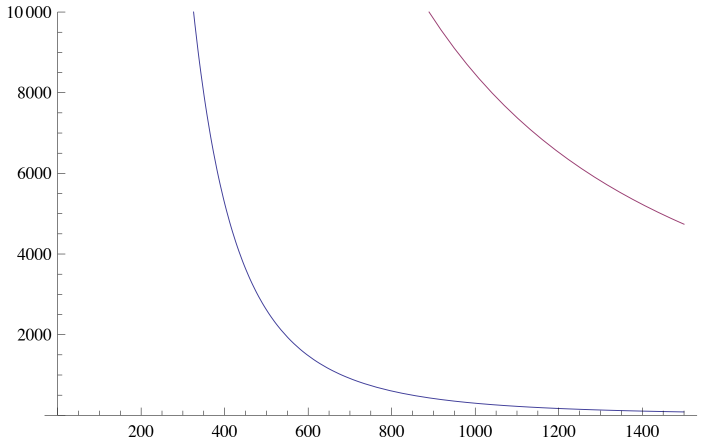
Plot

Demands

In[18]:=

```
Plot[{funct[[2, 3]], funct[[3, 3]]}, {Q, 0, 1500},
              PlotRange -> {0, 10 000}]
```

Out[18]=



Marginal Net Benefits

In[19]:=
```
Block[
    {Q1 = Q,
        Q2 = Q,
        Q3 = Q,
        Q4 = Q,
        Q5 = Q},
        Plot[{funct[[1, 6]], funct[[2, 6]],
                funct[[3, 6]], funct[[4, 6]],
                funct[[5, 6]]},
            {Q, 0, 600},
          PlotRange -> {0, 40 000}]
    ]
```

Out[19]=