Jan 7, 2015

Workshop

Blended Near-Optimal Alternative Generation, Visualization, and Interaction for Water Resources Decision Making

Tarbiat Modares University

Instructor:	Dr. David E. Rosenberg, Utah State University, Utah, USA
Contact:	david.rosenberg@usu.edu
Workshop Length:	4 hours
Language:	English
Prerequisites:	Linear Algebra and Water Resources Systems Analysis (e.g., USU's MATH 2270 and CEE 6410 or equivalents)

Overview

State-of-the-art systems analysis techniques unanimously focus on efficiently finding optimal or pareto-optimal solutions. Yet water resources managers rather need decision aides that show multiple, promising, near-optimal alternatives. In this workshop, participants will use blended tools to generate, visualize, and interactively explore near-optimal alternatives to optimization problems. These near-optimal alternatives perform within a tolerable deviation from the optimal stated objective function value of the original optimization problem. Participants will additionally use the tools to generate one or more near-optimal alternatives to an optimization problem of their choosing. The workshop will allow participants to generate, visualize, and explore near-optimal alternatives to water management problems.

Learning Objectives

Upon successfully completing the workshop, participants will be able to:

- Formulate a near-optimal problem,
- Generate, visualize, and explore near-optimal alternatives to an optimization problem,
- Apply near-optimal tools to an optimization problem of their choosing,
- Help identify how the near-optimal tools can be improved.

Computer Requirements and Setup

Prior to the beginning of the workshop, participants must:

1. Install Matlab version 2013 or higher on the computer they will use
2. Install the General Algebraic Modeling System (GAMS) version 24.3.3 or more recent.
    a. Download from www.gams.com.

   b. In Matlab, add the directory where you installed GAMS to the Matlab Path. On the Ribbon, select **Home** tab=>**Environment**=>**Set Path**. In the Set Path window, click the **Add with subfolders…** button and navigate to the directory where you installed GAMS.

3. Download all code for the Blended Near-Optimal Tools from the GitHub repository at https://github.com/dzeke/Blended-Near-Optimal-Tools.
   a. Click the button labeled **Download ZIP** at the far right bottom.
   b. Unzip the folder
   c. Add the location where you unzipped the folder to the Matlab Path as in Step 2b.
4. Read the manuscript available on the code repository that describes use of the tools. The manuscript is also available in the unzipped folder /1-Documentation/Rosenberg-NearOptimalWaterResourcesDecisionMaking-Dec2014.pdf.

Additional Questions

Please direct additional questions to either Dr. David E. Rosenberg at david.rosenberg@usu.edu or Dr. Saeid Morid at s_morid@hotmail.com.

Workshop Activities

Below are the steps to follow to use the near-optimal tools in three directed activiteis. Instructions in **Bold** or *Italic* refer to Matlab or tool items (windows, menus, tabs, input items, etc.). *Italicized, underlined sentences indicate questions that you should write the answer to before proceeding to the next step.* If any point you have a question, consult the documentation materials (.txt files) that accompany each folder you downloaded in Step 3 or seek help from a facilitator.

If you have not already done so, install the required materials as directed above in Steps 1 to 4 (Computer Requirements and Setup).

Activity #A. Verify the Installation

5. Test use of the generation and visualization tools on a small 5-decision variable, linear optimization program. In Matlab, run the script LoadYourOwnModel.m. The script finds the optimal solution, generates 150 stratified-sampled alternatives from the feasible region of the problem, shows the vertices of the feasible region, and plots the results in parallel coordinates. More specifics of the problem are provided as comments in Lines 37-40 of the script file. At the Matlab command prompt, enter:

```
>> LoadYourOwnModel
```

You should see Figure 1. If yes, move on to Step 6, if not, seek help.

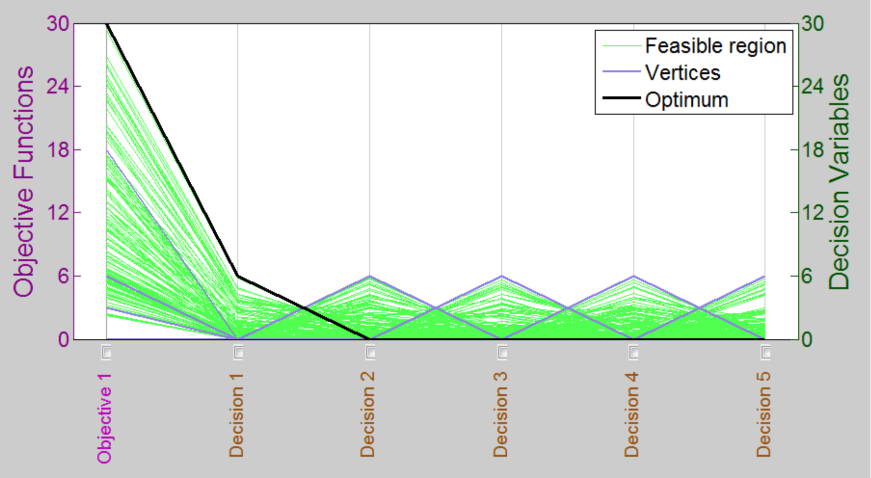

Figure 1. Output of LoadYourOwnModel.m

6. Test use of the generation and visualization tools on the much larger model for Echo Reservoir, Utah used in the manuscript. In Matlab:

   a. Change the Current Folder to \4-EchoReservoirApplication (or the folder where you unzipped the file WQNE_outG6.gdx in step 3b.
   b. Enter the command:

   ```
   >> LoadEchoGamsResultsMGAComp('WQNE_outG6.gdx',3,2500,0,[2 0 50]);
   ```

   c. This command will read the Echo Reservoir model data from GAMS, stratify-sample 2,500 near-optimal alternatives, generate additional alternatives by the Modelling to Generate Alternatives-Serial Method in about the same run time as it took the stratified-sampling, and plot all the results in Parallel Coordinates. You should see Figure 2 below (which is also Figure 3 in the manuscript).

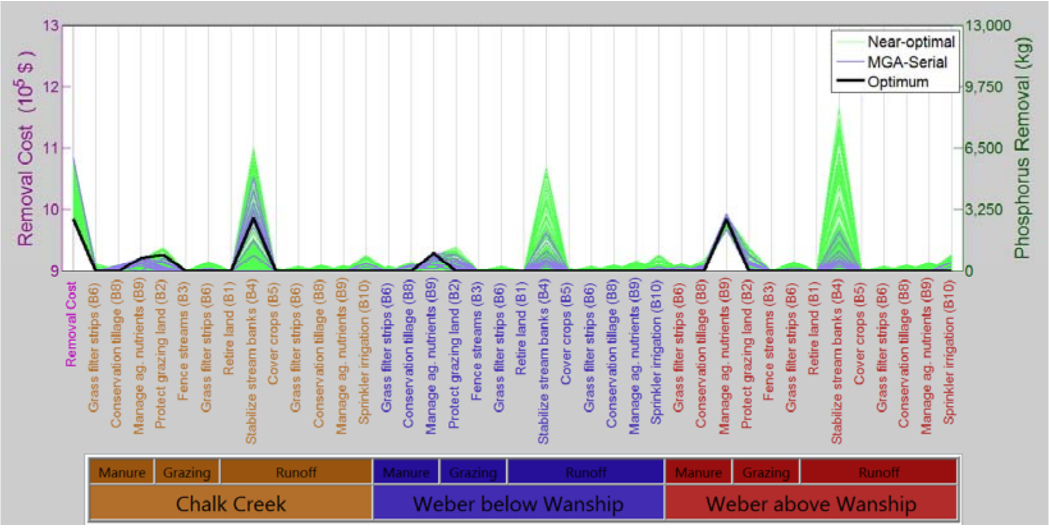

Figure 2. Optimal solution and near-optimal alternatives generated by the stratified sampling and MGA methods for the Echo Reservoir Phosphorus removal problem.

d. If you generated Figure 2, congratulations, you have successfully installed the blended near-optimal tools. If not, please seek help.

e. Look at Figure 2. *How do near-optimal alternatives generated by the stratified sampling method differ from alternatives generated by the MGA-Serial method? Which method generates alternatives that better cover the near-optimal region?*

Activity #B. Explore the Near-Optimal Region

7. *What near-optimal alternatives increase the phosphorus removed by stabilizing stream banks and shift implementation into the Weber below and above sub-watersheds?* To answer:

a. Use Matlab to open the file Fig_GenForNearOptPaper.m in the folder you created in Step 3.

b. Scroll down to Line 96 and begin following actions #2-10 listed as comments in the script (proceeded by %). Work from the figure you generated in Step 6d.

c. After finishing action #10 in the script, you should see:

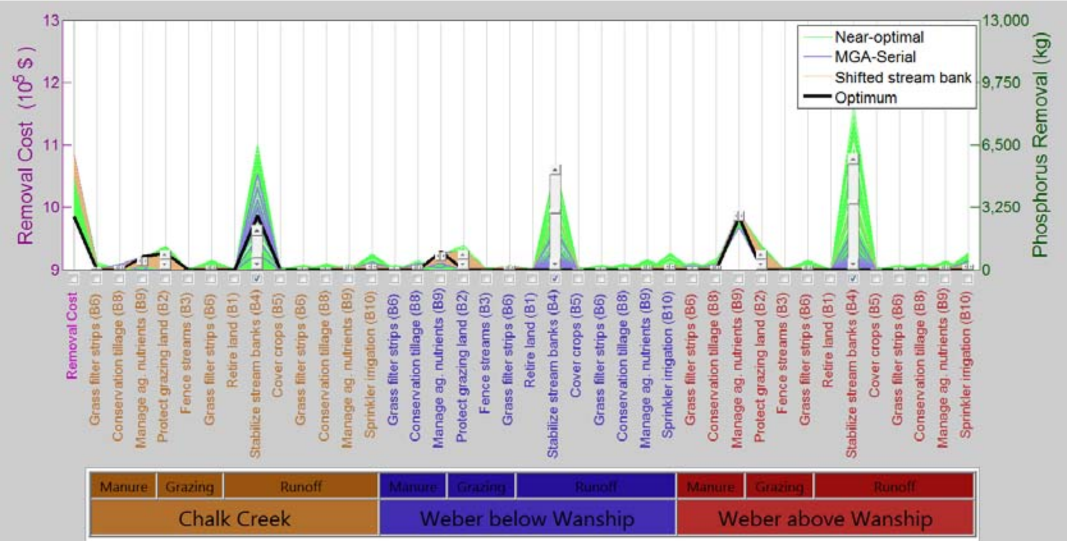

Figure 3. Near-optimal alternatives generated by shifting and increasing stream bank stabilization to the Weber below and above sub-watersheds.

d. To highlight particular alternatives you generated

i. On the **Display** tab in the **Group Traces** box, uncheck all the groups except for the 'Shifted stream bank' group you just created. Click the **Show Checked Groups** button.

ii. On the **Controls** menu, select **Alternatives**=>**Show current**. The current alternative will highlight in green.

iii. To select other alternatives, click the **Interact** tab. In the **Filter Existing Alternatives** box, use the arrows to scroll through the alternatives.

e. *What near-optimal alternative did you select as preferred? Why?*

8. Look through the interactive steps listed in Fig_GenForNearOptPaper.m for Figures 5, 6 or 7 (starting at lines 126, 231, or 269). Follow one of these interactive sequences or choose your own interaction.

a. *How did you explore the near-optimal region?*

b. *What alternative(s) did you generate and why?*

Activity #C. Generate Near-Optimal Alternatives from your Own Model Data

9. Collect your optimization model data into the objective function, decision variables, and constraints.

10. If your model is a linear program in the form

Maximize cX
Subject to: $AX \leq b$, $X \geq 0$

you can use the LoadYourOwnModel.m script with some small modifications.

a. Save the file LoadYourOwnModel.m a as a new file to make your changes.
b. On line 60, enter the number of decision variables for your problem.
c. On line 61, enter the row vector of objective function coefficients for your maximization problem. If you have a minimization problem, add a negative sign to each coefficient.
d. On line 63, enter the m x n matrix of constraint coefficients for the m model constraints. (Ax ≤ b). If you have constraints of the form Gx ≥ h, reverse the direction so they can be entered as –Gx ≤ -h.
e. On line 65, enter the m x 1 column vector of right hand side values for the constraints
f. On line 66, enter the number of near-optimal alternatives to generate.
g. On line 68, set mVert to dummy values of zero. mVert=zeros(1,n);
h. On line 75, set the Tolerance value to a number between 0 < tv < 1 for a maximization problem or > 1 for a minimization problem.
i. Run the script.
j. *How do the near-optimal alternatives differ from the optimal solution?*
k. Use the checkbox, slider, and other tools from Steps #7 and 8 to explore in the near-optimal region.

i. *What additional alternatives did you generate?*
ii. *Why?*

11. If your problem is non-linear:

a. The linprog and stratgibbs functions on lines 71 and 84 of LoadYourOwnModel.m will not work.
b. Consult with Dr. Rosenberg and we will develop strategies to generate a few near-optimal alternatives.