Blended Near-Optimal Tools

==============================

This repository stores the Matlab 2013a source code and (1) Documentation for blended near-optimal tools that (2) generate alternatives, (3) visualize alternatives, and allow a user to interactively explore the near-optimal region from which alternatives are generated. The repository also contains the data and model files for a (4) linear programming example application to manage water quality for Echo Reservoir, Utah, (5) mixed-integer programming example application to manage water supply and demands in Amman, Jordan, and (6) multi-objective linear programming reservoir operations problem.

Near-optimal alternatives perform within a (near-optimal) tolerable deviation of the optimal objective function value and are of interest to managers and decision makers because they can address un-modelled objectives, preferences, limits, uncertainties, or issues that are not considered by the original optimization model or it's optimal solution. Mathematically, the region of near-optimal alternatives is defined by the constraints for the original optimization model as well as a constraint that limits alternatives to those with objective function values that are within a tolerable deviation of the optimal objective function value. The code and tools within this repository allow users to generate and visualize the structure and full extent of the near-optimal region to an optimization problem. The tools also allow users to interactively explore region features of most interest, streamline the process to elicit un-modelled issues, and update the model formulation with new information. The tools and their use are described here for generating, visualizing, and interactively exploring near-optimal alternatives to optimization problems, but the tools are general and can be used to generate and visualize points within any high-dimensional, closed, bounded region that can be defined by a system of constraints. The parallel coordinate visualization and several interaction tools can also be used for any high-dimensional data set.

Below is listing of the repository contents by folder as well as additional information on citation, licensing, bug reports and authorship.

==============================

1) DOCUMMENTATION (Steps to get started, Paper, and Scripts)

Steps to Get Started:

a. Install Matlab on your computer (versions 2013a through 2015b)

b. Install the General Algebraic Modeling System (GAMS) version 24.3.3 or more recent.

i. Download from www.gams.com.

ii. 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.

c. Download all code for the Blended Near-Optimal Tools from the GitHub repository at https://github.com/dzeke/Blended-Near-Optimal-Tools.

i. Click the button labeled Download ZIP at the far right bottom.

ii. Unzip the folder

iii. Add the location where you unzipped the folder to the Matlab Path as in Step b2.

d. At the Matlab command prompt, enter:

>> FigGenForNearOptPaper

e. The script will generate Figures 1, 2, 3, and 5 in the submitted manuscript and prompt you whether to generate additional results discussed but not presented in the paper (select no).

f. See the file Documentation-readme.txt and comments in FigGenForNearOptPaper.m for further instructions on how to interactively generate figures 4, 5, 6, and 7 from Figure 3.

Further Use:

Open the file NearOptimalManagement-Lab.pdf (in the 1-Documentation folder) and follow the step-by-step directions and instructions to use the near-optimal tools on existing Echo Reservoir phosphorus removal (See Section 4 below) and reservoir operations problems (see Section 6 below) (approximately 2-hours).

Advanced Use (load your own data):

Open the file LoadYourOwnModel.m (in the 1-Documentation folder) and follows the directions and example for how to load your own model data into the near-optimal tools. Specific directions for linear programs, more general directions for mixed-integer programs.

The documentation also provides copies of the pre-publication version of the paper describing the tools, draft manuscripts, and responses to reviewer comments. The final, open access published version of the paper is also available at http://dx.doi.org/10.1002/2013WR014667.

==============================

2) GENERATE ALTERNATIVES

Uses stratified Monte-Carlo Markov Chain Gibbs sampling to identify a large number of alternatives that comprehensively span the near-optimal region through both the decision and objective spaces. To get started:

a. Complete steps a-c above in the 1-Documentation section

d. At the Matlab prompt, invoke the function

>> stratgibbs(...)

e. See additional directions in the file GenerateAlternatives-readme.txt for use of the stratgibbs function.

==============================

3) PARALLEL COORDINATE VISUALIZATION and INTERACTION

Parallel coordinate plot places axes for all objectives and decision variables side-by-side on a single page and shows the generated alternatives across the decision and objective spaces. Interaction controls on and next to the plot:

- Render generated alternatives (mouse-over to read a value, highlight individual or groups of alternatives on the plot, and specify the axes order from left to right across the plot).

- Allow the user to explore the near-optimal region (set sliders on axes to specify features, dynamically update the model formulation, and generate individual or families of new alternatives with specified features)

- Direct exploration into different parts of the region (re-order axes or relax the constraint that specifies the size of the region.

- Save figure settings to the Matlab e workspace and use to recreate the figure

- Optional parameters/settings to control how the plot is displayed and labeled (varargin)

To get started:

a. Complete steps a-c above in the 1-Documentation section

b. At the Matlab command prompt, invoke the following function:

>> nearoptplotmo2(...)

c. Additional example commands to run the interactive parallel coordinate plot tool are in ExampleTestPlots.m

d. See additional directions in the file InteractiveParallelPlot-readme.txt and in the header of the nearoptplotmo2.m for use of the function.

==============================

4) ECHO RESERVOIR APPLICATION (Linear program to manage water quality)

Illustrates use of the tools for a linear program to identify the cost-effective phosphorus removal practices to reduce the phosphorus load to Echo Reservoir in the Weber basin, Utah to a level specified in a pending Total Maximum Daily Load (TMDL) program for the reservoir. Includes all the data and model files and the script that moves data from the models to the near-optimal tools as well as generate comparison results from one or more Modelling to Generate Alternatives (MGA) methods.

To use:

a. Complete steps a-c above in the 1-Documentation section

d. At the Matlab command prompt, enter:

>> LoadEchoGamsResultsMGAComp('WQNE\_outG6.gdx',3,2500,0,[2 0 50]);

e. Matlab will read optimal results from the file WQNE\_outG6.gdx. Matlab will also generate 2,500 near-optimal alternatives (see Section 2 above) as well as approximately 13 comparison MGA alternatives by the serial method that uses a distance metric and plot everything using the Interactive Parallel Plotting tool (see Section 3 above).

f. See additional directions in the file EchoReservoirApplication-readme.txt.

==============================

5) AMMAN, JORDAN APPLICATION (Mixed-integer program to manage water supply/demand)

Illustrates use of the tools for a mixed-integer program to identify the cost-effective combination of new water supply and conservation strategies to balance water supplies and demands in Amman, Jordan through 2020. Includes all the data and model files and scripts to move data from the models to the near-optimal tools and to run the model from within the Interactive Parallel Coordinate Plotting tool.

To use:

a. Complete steps a-c above in the 1-Documentation section

d. At the Matlab command prompt, enter:

>> LoadAmmanJordan(AmmanJordanUtilOptNear.gdx)

e. Matlab will read optimal and near-optimal results from the file AmmanJordanUtilOptNear.gdx and plot the results in the Interactive Parallel Plotting tool (see Section 3 above).

f. For interactive use, see additional directions in the file AmmanJordanApplication-readme.txt.

==============================

6) RESERVOIR OPERATIONS PROBLEM (Hydro-economic example)

Illustrates use of the tools for a linear programming hydroeconomic reservoir operations problem over six time periods that maximizes benefits from hydropower generation and irrigation supply while respecting an in-stream flow requirement. To use:

a. Complete steps a-c above in the 1-Documentation section

d. At the Matlab prompt, enter the command:

>> ReservoirOperationProblem([])

e. Follow further instructions in the Lab exercise 1-Documentation/NearOptimalManagement-Lab.pdf.

==================

CITATION

David E. Rosenberg (2015). "Blended near-optimal alternative generation, visualization, and interaction for water resources decision making". Water Resources Research. doi:10.1002/2013WR014667. http://onlinelibrary.wiley.com/doi/10.1002/2013WR014667/full.

LICENSING

Materials in this GitHub repository are disturbed under a BSD 3-Clause license (see <https://github.com/dzeke/Blended-Near-Optimal-Tools/blob/master/LICENSE>). For alternative licensing arrangements, contact David E. Rosenberg directly.

BUG REPORTS and FEEDBACK

This code is possibly laden with bugs so bug reports and feedback are much appreciated. Please submit via the the issue tracker for this repository. And note, that while much appreciated, there is no promise of when--or if--the bug will be fixed.

===================

AUTHOR and CONTACT

David E. Rosenberg

Department of Civil & Environmental Engineering and Utah Water Research Lab

Utah State University

Email: david.rosenberg@usu.edu