Blended near-optimal alternative generation, visualization, and interaction for water resources decision making

David E. Rosenberg

Associate professor
Department of Civil & Environmental Engineering and Utah Water Research Laboratory,
4110 Old Main Hill
Utah State University,
Logan, UT, 84322-4110, USA

david.rosenberg@usu.edu
001 (435) 797-8689

Submitted to *Water Resources Research*

Submitted – September, 2013
Resubmitted – August, 2014
Resubmitted – December, 2014

Research Significance

New stratified sampling and parallel coordinate plotting tools generate and communicate the structure and extent of the near-optimal region to an optimization problem. Interactive controls guide exploration of features that most interest users. Controls also streamline the process to elicit un-modelled issues and update the model formulation with new information. Use for a linear-programming water quality management problem at Echo Reservoir, Utah identifies numerous and flexible alternatives to manage the system and maintain close-to-optimal performance. The new tools generate more numerous alternatives faster, more completely show the near-optimal region, and help elicit a larger set of un-modelled issues than prior Modelling to Generate Alternatives methods.

Key Points

1. New generation and visualization tools show the full near-optimal region
2. Interactive exploration can elicit un-modelled issues
3. The tools identify flexible management to maintain near-optimal performance

Blended near-optimal alternative generation, visualization, and interaction for water resources decision making

David E. Rosenberg
Associate professor, Dept. of Civil & Environmental Engineering and Utah Water Research Laboratory, Utah State University, USA (david.rosenberg@usu.edu)

Abstract: State-of-the-art systems analysis techniques focus on efficiently finding optimal solutions. Yet an optimal solution is optimal only for the modelled issues and managers often seek near-optimal alternatives that address un-modelled objectives, preferences, limits, uncertainties, and other issues. Early on, Modelling to Generate Alternatives (MGA) formalized near-optimal as performance within a tolerable deviation from the optimal objective function value and identified a few maximally-different alternatives that addressed select un-modelled issues. This paper presents new stratified, Monte Carlo Markov Chain sampling and parallel coordinate plotting tools that generate and communicate the structure and extent of the near-optimal region to an optimization problem. Interactive plot controls allow users to explore region features of most interest. Controls also streamline the process to elicit un-modelled issues and update the model formulation in response to elicited issues. Use for an example, single-objective, linear water quality management problem at Echo Reservoir, Utah identifies numerous and flexible practices to reduce the phosphorus load to the reservoir and maintain close-to-optimal performance. Flexibility is upheld by further interactive alternative generation, transforming the formulation into a multi-objective problem, and relaxing the tolerance parameter to expand the near-optimal region. Compared to MGA, the new blended tools generate more numerous

alternatives faster, more fully show the near-optimal region, and help elicit a larger set of un-modelled issues.

Keywords: optimize; near-optimal; Modelling to Generate Alternatives; water management; linear program; alternative generation; visualization; interaction; Echo Reservoir, Utah

1 INTRODUCTION

State-of-the-art systems analysis techniques emphasize efficiently finding optimal and *pareto*-optimal solutions [*Brown et al.*, 1997] to water supply and environmental decision problems [*Nemhauser*, 1994; *Sahinidis*, 2004; *Zhang and Li*, 2007]. Yet an optimal solution is optimal only with respect to the modelled issues and managers may prefer near-optimal alternatives that address un-modelled objectives, preferences among objectives, limits, uncertainties, or other shortcomings in the original model [*Brill et al.*, 1982; *Chang et al.*, 1982; *Harrington and Gidley*, 1985; *Rogers and Fiering*, 1986]. Providing managers with useful near-optimal alternatives requires solving two conflicting challenges: (i) generate alternatives that address the un-modelled issues, and (ii) tractably communicate alternatives. Both challenges intensify as problem size grows and the modeller has less ability to qualitatively and quantitatively characterize the un-modelled issues.

The Modelling to Generate Alternatives (MGA) Hop-Skip-Jump (HSJ) algorithm [*Brill et al.*, 1982; *Chang et al.*, 1982] was an early linear-programming method to generate near-optimal alternatives. The MGA-HSJ first solved for the optimal solution, then generated a new alternative that was maximally different in decision space from the optimal solution yet performed within a (near-optimal) tolerable deviation of the optimal objective function value. The new alternative minimized the sum of the non-zero decision variables in the optimal solution. Next, MGA-HSJ generated a second alternative that was maximally different from the optimal solution and first alternative and continued until either every variable became non-zero in at least one alternative or no new non-zero variables were found. Generated alternatives were always less than the number of decision variables in the problem, located at select vertices of near-optimal region, and addressed un-modelled issues typically related to decision variable substitutability.

Subsequent near-optimal work enumerated all near-optimal region vertices for a small farm management problem [*Burton et al.*, 1987], but encountered difficulty with problem scaling [*Harrington and Gidley*, 1985; *Matheiss and Rubin*, 1980]. Effort increased exponentially with problem size to enumerate all extreme points where model constraints intersect, then retain only feasible points as vertices to the region.

A third set of near-optimal alternative generation methods substituted different and sometimes randomly parameterized objectives for MGA-HSJ's maximally-different criterion [e.g., *Chang et al.*, 1982; *Harrington and Gidley*, 1985; *Kennedy and Quinn*, 1998]. These methods also generated a few alternatives—one for each objective tested.

More recent MGA extensions use distance metrics and evolutionary algorithms to generate near-optimal alternatives either in series (similar to MGA-HSJ) or simultaneously (all at once) for non-linear problems [*Loughlin et al.*, 2001; *Zechman and Ranjithan*, 2004; *Zechman and Ranjithan*, 2007]. The distance metrics maximize either the (i) aggregate difference between alternative decision components (L_1 norm), or (ii) distance to the nearest neighbour ($-L_\infty$ norm). In testing use of the $-L_\infty$ norm distance metric, *Zechman and Ranjithan* [2007] found MGA-Serial generates an alternative that is furthest from the optimal solution and prior alternatives whereas MGA-Simultaneous spreads alternatives through the near-optimal region. Both methods face scaling issues to increase the number of generated alternatives that involve making more numerous distance calculations and solving larger optimization problems. As a result, Evolutionary Algorithm applications have generated a small number of near-optimal alternatives (3, 3, and 9 in the examples) that offer limited opportunities to elicit un-modelled issues.

To generate a more comprehensive set of alternatives in reasonable time, uniform Monte-Carlo Markov chain samplers use conditional sampling to generate points inside bounded high-dimensional regions [*Chen and Schmeiser*, 1993; *Kroese et al.*, 2011; *Liu*, 2001] but have not yet been applied to near-optimal problems. Samplers include (a) Metropolis–Hastings [*Hastings*, 1970; *Kroese et al.*, 2011], (b) Gibbs [*Gelfand and Smith*, 1990], and (c) Hit-and-Run [*Smith*, 1984]; start at an arbitrary point within the region; then sample new alternatives by, respectively, (a) generating a new candidate alternative and accepting/rejecting it using a threshold probability, (b) cycling through variables to progressively sample each decision component, or (c) randomly picking a direction and moving a random distance in that direction from the current point towards the region boundary. And although fast, the uniform samplers undesirably cluster samples towards the region centroid and away from vertices and surfaces because local volumes near extremities are very small compared to the region volume.

The challenges of fast, comprehensive alternative generation and tractable communication also confront narrower searches of control policies for the multiple optima that have the same objective function value but different locations in decision space [*Aufiero et al.*, 2001; *Nardini and Montoya*, 1995; *Orlovski et al.*, 1983]. Similarly, the challenges confront *pareto* search using generation [e.g., *Deb*, 2008], interaction, or progressive articulation of preferences [*Korhonen and Wallenius*, 1988; *Miettinen et al.*, 2008; *Steuer*, 1986; *Wierzbicki*, 1979] for preferred compromise solutions to multi-objective problems where all objectives are specified *a priori* [*Castelletti et al.*, 2010; *Cohon and Marks*, 1975; *Marler and Arora*, 2004]. More recent *pareto* methods blend alternative generation, visualization, and interaction to leverage computational, cognitive, and learning strategies [*Castelletti et al.*, 2010; *Lotov et al.*, 2004]. Blending first generates and visualizes one or numerous alternatives, then the user guides further generation and visualization until reaching an acceptable end point. These blended methods for *pareto* search also hold promise to help identify, visualize, and explore larger near-optimal regions for alternatives that address un-modelled issues.

This paper presents new blended tools to more comprehensively generate, visualize, and interactively explore the near-optimal region of an optimization problem. Stratified Monte-Carlo Markov Chain sampling quickly identifies a large number of near-optimal alternatives that span the region through both the decision and objective spaces. A new Parallel Coordinate plot [*Inselberg*, 2009; *Wegman*, 1990] places axes for all objectives and decision variables side-by-side on a single page to show generated alternatives across the linked decision and objective spaces. Interactive controls allow the user to explore the region and seek alternatives that address un-modelled issues. Together, the tools (i) show the structure and full extent of the near-optimal region, (ii) identify numerous and flexible ways to manage the system to maintain near-optimal performance, and (iii) streamline the process to elicit un-modelled issues and update model formulations. Section 2 reviews the near-optimal formulation and prior MGA solution methods while sections 3-5 present the new alternative generation, visualization, and interactive tools. The remaining sections show and compare results for an example linear-programming water quality management problem at Echo Reservoir, Utah to prior MGA methods and present conclusions.

2 NEAR-OPTIMAL FORMULATION AND PRIOR SOLUTION METHODS

For the general problem to minimize objective *f* with objective function *c*, a vector of *n* decision variables **x**, *a* comprising *m* constraint functions, and *b* a vector of *m* bounds for the *m* constraints,

$$\underset{\mathbf{x}}{\mathrm{Min}}\ \ \mathrm{f} = \mathrm{c}(\mathbf{x}) \qquad \text{[Eq. 1a]}$$

subject to

$$\mathrm{a}(\mathbf{x}) \leq \mathrm{b} \text{ and } \mathbf{x} \geq 0, \qquad \text{[Eq. 1b]}$$

define the near-optimal region in three steps [*Brill et al.*, 1982; *Chang et al.*, 1982]:

1. Solve Eq. 1 for the optimal decision variable values ($\mathbf{x}^*$) using the appropriate programming technique (e.g., linear, mixed-integer, nonlinear, global, etc.),
2. Allow a tolerable deviation, γ (γ ≥ 1 [unitless]), from the optimal objective function value (f^*)

$$c(\mathbf{x}) \leq \gamma \cdot f^{*}\text{, and} \qquad \text{[Eq. 2]}$$

3. Compose the near-optimal region from the original (Eq. 1b) and objective function tolerance (Eq. 2) constraints.

If maximizing *f*, reverse the inequality signs in Step 2 (γ ≤ 1 and $c(\mathbf{x}) \geq \gamma \cdot f^{*}$).

In the near-optimal formulation, the tolerable deviation constraint (Eq. 2) partitions the feasible region (Eq. 1b) into two sub-regions: the near-optimal and remaining feasible region. Figure 1 shows this partitioning for an example allocation problem with two decisions such as water deliveries to Users 1 and 2 (X_1 and X_2), quadratic cost minimization objective, linear constraints that specify the available water, delivery efficiencies, and capacity, and near-optimal tolerance of γ = 1.8. In the example, the optimal solution allocates water only to User 1 (Figure 1, blue circle, bottom right).

When the optimal solution is not acceptable because of un-modelled issues, MGA-HSJ can generate near-optimal alternatives that perform close to the optimal solution on the stated objective but are maximally-different in their decision variable values [*Brill et al.*, 1982; *Chang et al.*, 1982]. MGA-HSJ solves a new problem that minimizes the sum (*y*) of the non-zero decision variables in the optimal solution (x_k, $k \in K$):

$$\begin{aligned} &\min_{\mathbf{x}} \; y = \sum_{k \in K} x_k \\ &s.t. [1b] \text{ and } [2] \end{aligned} \qquad \text{[Eq. 3]}$$

Problem [3] is the same size as Eqs. [1] and [2]. After solving [3] for a new alternative, MGA adds the new non-zero decision variables in the new alternative to the set *K*, resolves [3], and repeats until either no new decision variables are added to *K* or *K* comprises all decision variables [*Brill et al.*, 1982; *Chang et al.*, 1982]. MGA-HSJ can generate up to *n-1* alternatives depending on the number of non-zero variables in the optimal solution and number of variables added to *K* at each iteration.

In the Figure 1 example, MGA-HSJ generates a single near-optimal alternative that still allocates all water to User 1 but at higher cost (Figure 1, black square). This alternative represents movement along the X_1 axis towards the origin to minimize the single non-zero variable X_1 in the optimal solution. MGA-HSJ terminates after the first iteration because no new decision variable enters *K*.

In contrast, MGA-Serial [*Zechman and Ranjithan*, 2007] iteratively solves for a new maximally-different alternative ($\mathbf{x}^p$) by maximizing the distance between the new alternative, prior generated alternatives ($\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^{p-1}$), and the optimal solution ($\mathbf{x}^0$).

$$\begin{aligned} &\underset{\mathbf{x}^p}{\text{Max}}\ \ \mathrm{D} = \min_{y \le p-1}\left\{\sum_{k=1}^{n}\left|x_k^p - x_k^y\right|\right\} \\ &s.t. [1b] \text{ and } [2] \end{aligned} \qquad \text{[Eq. 4]}$$

Here, distance (*D*) is the smallest summed difference between decision components ($-L_\infty$ norm) and the absolute value operator makes the problem non-linear. Generating *p* alternatives means solving *p* problems in series, each of size *n* decisions and *m* constraints, but that grow in the number of distance calculations between the current and prior generated alternatives plus the optimal solution. The related MGA-simultaneous approach uses the same distance metric (*D*) but instead solves a single problem to generate *p* alternatives $\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^p$ all at once to more uniformly spread alternatives [*Zechman and Ranjithan*, 2007]. Generating *p* simultaneous alternatives requires $n \cdot p$ decision variables, $m \cdot p$ constraints, and evaluating $\frac{1}{2}(p+1)(p) \sim p^2$ distances between the alternatives and optimal solution.

Initially, these scaling issues are imperceptible to generate a small number of alternatives for small problems like in Figure 1. For example, solving for a single near-optimal alternative with MGA-Serial and Simultaneous using Matlab's 2013b global search and sqp non-linear solver generated the same alternative in under 1 s (Figure 1, black triangle, only MGA-Serial result shown). And while the generation density of 0.5 alternatives per decision variable may appear low, this density is actually 3 to 50 times *larger* than prior applications of the methods. Run times for both MGA methods rapidly increase with the number of alternatives generated and mean the methods can only generate a small number of alternatives for moderate sized problems. Further, the distance metrics require solution with global search or evolutionary algorithms that use randomized start points and do not necessarily generate the same alternatives on each run [*Zechman and Ranjithan*, 2007]. Thus, the MGA method's partial coverage of the near-optimal region shown in Figure 1 is emblematic of results for higher-dimensional problems and means there is benefit to more efficiently generate a larger number of alternatives that more fully communicate the near-optimal region (e.g., Figure 1, green area).

More complete coverage would identify areas near the upper vertex or optimal solution that offer alternatives that could address un-modelled allocation equity, flexibility, or other issues. Further, when the generation strategy is fast, a manager can select an alternative, the systems analyst can elicit the reason for the manager's selection, use the new information to update the model formulation, generate additional alternatives, and guide further exploration in the near-optimal region. The next three sections present new alternative generation, visualization, and interaction tools to identify, communicate, and explore near-optimal regions for larger problems that cannot be represented on a Cartesian plot.

3 ALTERNATIVE GENERATION

Because existing uniform Monte-Carlo Markov chain methods to sample high-dimensional, closed regions [*Gelfand and Smith*, 1990; *Hastings*, 1970; *Kroese et al.*, 2011; *Liu*, 2001; *Rubinstein*, 1986; *Smith*, 1984] congregate samples towards the centroids of regions, this paper introduces a stratified sampling approach to generate alternatives that more comprehensively span the entire near-optimal region.

a. Divide the p alternatives to be sampled into n+1 groups (associate n groups with the n decision variables and last group with the objective function).
b. Identify the minimum and maximal extent of each decision variable in the near-optimal region. To identify these extents, solve 2n optimization problems that separately and independently Minimize x_1, Maximize x_1, Minimize x_2, Maximize x_2, … , Minimize x_n, and Maximize x_n each subject to the same original and near-optimal tolerance constraints [Eqs. 1b and 2].
c. Select a decision variable and uniformly sample $p/(n+1)$ values from within the variable's maximal extents identified in step b. For each sampled value, reduce the problem dimensionality by one and use a Monte-Carlo Markov chain method to sample values for the remaining decision variables. Together, the $p/(n+1)$ sampled alternatives in the group are stratified along the chosen decision variable.

d. Repeat step c for each decision variable.

e. Finally, uniformly sample a set of $p/(n+1)$ objective function values ($\hat{f}_j$; $j = 1$ to $p/(n+1)$) between the optimal objective function and tolerable deviation values [f^*, γf^*]. For each sampled objective function value, add a constraint to fix the objective function at the sampled value, $\hat{f}_j = c(x)$. Then Monte-Carlo Markov Chain sample decision variable values (as in step c) with the specified objective function value.

Together, the n+1 groups of sampled alternatives are stratified along the extents of each decision variable and the objective function within the near optimal region. Adjust the total samples (p), stratified samples per group, or number of Monte-Carlo Markov chain samples per stratified sample (chain length) to achieve the desired sampling density or focus attention on particular sub-regions. The generation strategy solves 2(n+1) optimization problems that are n decisions and m constraints in size and requires drawing p Monte-Carlo Markov chain samples. The next subsections present extensions of the stratified sampling alternative generation strategy for three classes of optimization problems.

3.1 Linear programs

For linear problems with continuous decision variables, follow steps (a) through (e) above. Use vectors and matrixes of linear coefficients to efficiently pass model data. Testing of Monte-Carlo Markov Chain methods [*Gelfand and Smith*, 1990; *Hastings*, 1970; *Kroese et al.*, 2011; *Smith*, 1984] found each sampler worked fast and confirmed prior comparisons by *Chen and Schmeiser* [1993]. Here, the tools use and extend Gibbs sampling [coded by *Benham*, 2011].

3.2 Integer problems

For integer decision variable problems, restrict sampling in steps (c) and (d) to integer values. For small- and moderate-sized problems, instead generate all near-optimal integer alternatives in the region by modifying the Gibbs sampler [*Gelfand and Smith*, 1990] to instead during the cycle through decision variables enumerate all integer values within the bounding extents. Then for each integer value, continue to the next variable and again enumerate within the extents. The extents ensure generating only feasible near-optimal alternatives and further reduce the overall effort to a fraction of the exponential time required for brute force enumeration (which tests all variable value combinations). Rosenberg [2012] uses the technique on a two-stage, mixed-integer stochastic program with recourse that identifies 18 water supply and conservation actions to meet a probability distribution of shortages in Amman, Jordan. One interactive tool in Section 5 applies the method to generate a family of alternatives.

3.3 Multi-objective problems

For a multi-objective problem with i=1 to d objectives defined *a priori*, form $n+d$ (rather than n+1) groups in step a. Then in step e, introduce a tolerable deviation for each objective (γ_i) and uniformly and unconditionally sample j=1 to $p/(n+d)$ values $\hat{f}_{1,j}$ for the first objective (c_1) within its near-optimal range [f_1*, γ_1 f_1*]. Using the first sampled objective function value ($\hat{f}_{1,1}$), identify extents for the second objective (conditioned on the original, near-optimal tolerance and $\hat{f}_{1,1} = c_1(x)$ constraints). Next, uniformly sample a value for the second objective ($\hat{g}_{2,1}$) within its conditioned extents, identify the extents for the third objective (conditioned on the original, near-optimal tolerance, $\hat{f}_{1,1} = c_1(x)$, and $\hat{g}_{2,1} = c_2(x)$ constraints), and continue through the *d* objectives. Up to this point, the sampling strategy resembles the Lexicographic method [*Marler and Arora*, 2004] and emulates Gibbs sampling but cycles through the objectives rather than decision variables. From here, add the *d* sampled objective function values as *d* equity constraints to the original and near-optimal constraints and Monte-Carlo Markov Chain sample decision variable values with the specified objective function values. Together, the sampled objective function and decision variable values comprise an alternative associated with the first unconditionally sampled objective. Repeat the above steps to obtain *j*=2 to *p*/(*n*+*d*) additional alternatives associated with the remaining unconditionally sampled values $\hat{f}_{1,j}$ for the first objective.

Finally, repeat the above steps, but switch the order the objective functions are sampled so that each objective gives rise to *p*/(*n*+*d*) alternatives where that objective is sampled first and unconditionally. The resulting *p* alternatives span the objective and decision spaces, the *pareto* front, as well as the near-optimal region that lies under the front. Dissatisfaction with alternatives on the front in objective space motivate search in the larger region near the front.

4 PARALLEL COORDINATE VISUALIZATION

It is difficult to visualize high-dimensional near-optimal regions on Cartesian plots because they *orthogonally* arrange axes and can only show two or three dimensions (on a flat page) or possibly six or seven dimensions when using marker color, size, orientation, scroll bars, or other attributes [*Kasprzyk et al.*, 2009; *Kollat and Reed*, 2007; *Lotov et al.*, 2004]. In contrast, Parallel Coordinate plots [*Inselberg*, 2009; *Wegman*, 1990] arrange numerous axes in *parallel* and use polygonal lines that span axes to represent Cartesian points. Recent uses in water management studies include to visualize discrete points on *pareto* frontiers [e.g., *Kasprzyk et al.*, 2009; *Ortiz et al.*, 2011; *Shenfield et al.*, 2007; *Stummer and Kiesling*, 2009]. Beyond discrete points, parallel coordinate systems also uniquely represent line, curve, and other Cartesian concepts [*Inselberg*, 2009] and these correspondences can be used to visualize high-dimensional near-optimal regions.

First lay out parallel axes for the objective function and each decision variable (Figure 2). Second, draw a set of connected, polygonal lines to represent the optimal solution that cross the objective function and decision variable axes at the corresponding optimal objective function and decision variable values (Figure 2, blue line). Next, draw additional polygonal lines for each generated near-optimal alternative (Figure 2, green lines). The resulting parallel coordinate plot i) simultaneously links the objective and decision spaces on a single plot, and ii) displays a large number of generated near-optimal alternatives. The plot also iii) shows the extents for each variable (identified in section 3, step b) within the near-optimal region as the lower and upper limits where near-optimal polygonal lines cross each axis. For multi-objective problems, add axes for the new objectives. Although axes order influences Parallel Coordinate plot rendering [*Edsall*, 2003], interactive controls allow users to specify and re-order axes while they explore the near-optimal region.

5 INTERACTIVE EXPLORATION

Controls on and next to the Parallel Coordinate plot allow the user to interactively explore the near-optimal region. This exploration serves to: (i) render and visualize the near-optimal region, (ii) generate further alternatives of interest to the user, and (iii) streamline the process to elicit un-modelled issues and update the model formulation to respond to those issues. Below, text describes nine interactive tools six of which are demonstrated later in Section 6. A first set of rendering tools:

- Mouse-over an axis and alternative to read a value,
- Highlight individual or groups of alternatives on the plot,
- Specify the axes order from left to right across the plot, and
- Color plot traces in a ramp from dark to light to indicate the values traces cross one selected axis and allow simultaneous visual comparison between values on the selected axis and each other axes.

A second toolset places slider controls side-by-side on each axis as a control panel [*Inselberg*, 2009] and uses slider settings to dynamically guide exploration of the near-optimal region, update the model formulation, and generate new alternatives. Slider heights show the current allowable ranges for each decision variable and objective function as defined by the original and near-optimal tolerance constraints and slider settings. Setting a slider for a decision axis to a value (e.g., s_{value}) adds a pair of deviation constraints of the form $x_k \leq s_{value}+\varepsilon$ and $x_k \geq s_{value}-\varepsilon$ that specify an allowable deviation (ε) for the decision variable (x_k) above and below the set value. Similarly, set the slider for the objective function on axis *i* to add constraints of the form $c_i(x) \leq s_{value}+\varepsilon$ and $c_i(x) \geq s_{value}-\varepsilon$. And when the allowable deviation is zero (ε=0), setting a slider instead specifies an equity requirement ($x_k=s_{value}$ or $c_i(x)=s_{value}$). Thus, the user can generate:

- Family of alternatives. Set a slider for one decision variable within its current extents, update extents for all other sliders, set a second slider within its extents, continue updating extents and setting sliders for up to *n*-1 sliders. Then use the alternative generation algorithm (section 3) to identify values for the remaining (un-set) decisions. Generated alternatives define a sub-space of the near-optimal region.

A third toolset can redirect exploration into different parts of the near-optimal region.

- Change the order sliders are set when generating a family of near-optimal alternatives. Equivalently, reorder the axes and specify variable values in the new order.
- Relax the tolerable deviation parameter (γ), generate new alternatives, and explore in the expanded region further away from the optimal objective function value.

During exploration, a user may identify a preferred near-optimal alternative. In this situation, the analyst can help elicit the un-modelled issue(s) that make the alternative preferred. Elicitation will be stronger now because the analyst can work from the manager's revealed preference between the selected alternative, unsatisfactory optimal solution, and prior explored alternatives. For example, what un-modelled requirement or aspiration does the alternative satisfy or achieve that the optimal solution does not? If the requirement is quantifiable, then add a new constraint to the model. Similarly, if the aspiration is quantifiable, add a new objective and transform the single-objective problem into a multi-objective one. Two additional plot tools speed and automate the process to enter new constraint and objective function information, solve the updated model for a new optimal solution (or *pareto* set using the appropriate multi-objective method), generate the new near-optimal region, and continue exploring the updated region. Terminate exploration when the manager is satisfied with an optimal solution, near-optimal alternative, or insight gleaned from exploring.

The generation, visualization, and interaction tools are coded in Matlab 2013a and available for download at https://github.com/dzeke/Blended-Near-Optimal-Tools.

6 EXAMPLE APPLICATION

I illustrate the new near-optimal tools for an example water quality management problem to reduce the phosphorus load to Echo Reservoir in the Weber basin, Utah. Throughout the U.S., excess nutrients like phosphorus and nitrogen impair water bodies and managers must select and locate nutrient removal practices to meet water quality standards. Researchers have developed both simple and complex optimization programs to recommend practices [*Alminagorta et al.*, 2013; *Hsieh and Yang*, 2007; *Maringanti et al.*, 2009]. This near-optimal application first overviews an existing single-objective linear program for Echo Reservoir [*Alminagorta et al.*, 2013] then compares the prior-published optimal solution, near-optimal alternatives generated by several MGA methods, and by the new tools with the goal to demonstrate several key features of the new tools. All alternatives were generated on a Windows 2.7 Ghz machine with 8 GB RAM; the number of alternatives for the MGA-Serial and Simultaneous methods were set to make runtimes commensurate with the new tools. Although the application does not represent the total complexity of the Echo Reservoir phosphorus removal problem, the results and discussion highlight several methodological contributions and management insights gleaned by use of the new blended near-optimal tools.

6.1 Model background

The single-objective linear program identifies the cost-effective practices to reduce phosphorus loads from non-point sources to the level specified in a pending Total Maximum Daily Load (TMDL) program for Echo Reservoir [*Adams and Whitehead*, 2006]. The program considers ten practices such as installing grass filter strips, managing agricultural nutrients, stabilizing stream banks, adopting sprinkler irrigation, retiring grazing land, among others that target three non-point phosphorus sources (manure, grazing, and diffuse runoff) in three sub-watersheds that drain to the reservoir (Chalk Creek, Weber above and below Wanship). The program excludes point sources because sources already adopted best available treatment technologies. The 39 decision variables represent 39 allowable combinations of practices, sources, and sub-watershed locations. A single objective function minimizes removal costs and reflects the TMDL criteria of cost to select practices to reduce phosphorus loads. 78 constraints require separate phosphorus load reductions by source as set forth in the TMDL (for a total load reduction of 8,067 kg/year), specify dependencies and mutual exclusive relationships among removal practices, and set upper limits for phosphorus removal for each source in each sub-watershed. The linear program formulation is identical to *Alminagorta et* *al.* [2013] except that here all decision variables are expressed (scaled) in phosphorus mass removed (Appendix I).

Input data include implementation costs and phosphorus removal rates for each practice [*Horsburgh et al.*, 2009], land area and stream bank lengths available to implement practices, and load reduction targets across sources. Costs reflect the full cost for agricultural users to implement a practice and include opportunity costs (such as forgone benefits) for practices like retiring land. Prior work [*Adams and Whitehead*, 2006] simulated the major physical, chemical, and biological processes affecting total phosphorus and dissolved oxygen concentrations within the stream and reservoir, identified the permissible phosphorus load to the reservoir and corresponding reduction target, as well as verified that phosphorus loads from each sub-watershed strictly and linearly add to produce the total load to Echo Reservoir. This later property allowed formulating the model as a linear program with constraints that specify global reduction targets for each source and permit trading load reductions across sub-watersheds.

6.2 Optimal and near-optimal comparisons

A parallel coordinate plot (Figure 3) compares the (i) optimal solution to (ii) 13 alternatives generated by MGA-Serial, and (iii) some 2,500 near-optimal alternatives generated by the new stratified Monte Carlo Markov Chain sampling method. Samples were stratified 15%/85% between the objective function and decision variable groups with two Monte Carlo Markov chain samples drawn per stratified sample within the decision variable groups. Each near-optimal alternative was generated by allowing a 110% deviation from the optimal removal cost to permit a small, but still significant (i.e., 10%) increase in removal costs. In Figure 3, the left-most axis and left pink scale show removal costs for each alternative while the 39 axes to the right and right-most green scale show the phosphorus removed by each practice. These axes are further grouped and color-coded by sub-watershed and phosphorus source. The optimal solution (thick black line) costs $985,000, implements only three phosphorus removal practices (manage agriculture nutrients, protect grazing land, and stabilize stream banks) and concentrates removal in the Chalk Creek sub-watershed. Thirteen MGA-Serial alternatives (purple lines) generated in 27 s each cost $1.08 million (110% of the optimal removal cost) and implement up to seven practices per sub-watershed at somewhat more varied levels than the optimal solution.

In contrast, the new tools generate more numerous near-optimal alternatives (green lines) in 25 s that cross each decision axis at much wider ranges of values. These alternatives show managers can stabilize stream banks in the Chalk Creek sub-watershed at levels between 0 and 2.3 times the 2,820 kg phosphorus the optimal solution recommends to remove. Alternatively, managers can stabilize stream banks in the sub-watersheds above and below Wanship at levels much higher than the 2,130 kg phosphorus indicated by the MGA-Serial alternatives. In each sub-watershed, managers can also cover crop, install grass filter strips, fence streams, adopt conservation tillage, or manage agricultural nutrients from diffuse runoff. Further, the alternatives identify numerous strategies that vary removal cost between the cost-minimizing value and 110% threshold.

Additional testing (results not shown) found:

- Generating 25 alternatives increased MGA-Serial runtime above 100 s with no appreciable change in the phosphorus removal practices implemented or costs,
- MGA-HSJ generated 22 alternatives in under 1 s all with removal cost at the 110% threshold and terminated when all practices entered the non-zero set. Among the HSJ set, up to 6 practices per sub-watershed (15 of the 39 decision variables) had values significantly less than the full extents identified by stratified sampling. For all but one practice (stabilize stream banks in Chalk Creek), MGA-HSJ generated between one and four unique non-zero values. Also, MGA-HSJ terminated much earlier after alternatives 4 and 5 had the same 13 non-zero variables on a model version with different scaled land area and stream bank length decision variables [*Alminagorta et al.*, 2013].
- Five MGA-Simultaneous trials each generated ten alternatives in 100 to 500 s all at the 110% removal cost threshold. These alternatives implemented up to six phosphorus removal practices in each sub-watershed.

Each MGA method gave a somewhat different and incomplete view of the near-optimal region and together the MGA methods only found the full extents identified by the new tools for 25 of the 39 decision variables. Therefore, subsequent comparisons focus on MGA-Serial, which, among the three MGA methods tested, had a moderate run-time but identified a more diverse set of removal levels for a slightly smaller set of practices.

6.3 Interaction to explore the near-optimal region

Enabling sliders on the parallel coordinate plot shows the minimum and maximum extents of each decision variable permitted by the original and near-optimal tolerance constraints. Then for example, with no allowable deviation from slider settings, set the slider for stream bank stabilization in Chalk Creek to zero, dynamically update the extents for remaining sliders, and set the sliders for stream bank stabilization in the Weber below and above Wanship sub-watersheds to large values greater than 2,130 kg to explore a region area where MGA-Serial did not generate alternatives. This sequence (Figure 4, checked axes) specifies an alternative set that (i) shifts the location to stabilize stream banks, and (ii) increases total phosphorus removed by the practice. Remaining sliders (unchecked in Figure 4) show the effects of slider settings and exploration. Sliders for manage agricultural nutrients from manure update to positive, non-zero ranges and indicate this practice must now be implemented in *each* sub-watershed while managers can still flexibly protect grazing land across the sub-watersheds.

With the three slider settings for stream bank stabilization and no allowable deviation, a generate button automatically adds three new equity constraints to the original and near-optimal constraints, updates the model formulation, and generates a new alternative set with shifted, increased stream bank stabilization (Figure 4, orange lines). Removal costs range between $1.05 and $1.08 million (110% of optimal removal cost) and show diverse strategies to both shift and increase phosphorus removal by stream bank stabilization.

6.4 Eliciting an un-modelled issue and updating the model formulation

Selecting one of the shifted, increased stream bank stabilization alternatives as preferred to the optimal solution offers an opportunity to elicit one or more un-modelled issues and improve the model formulation. For example, if the manager explains that his/her preference is motivated by a belief that the TMDL phosphorus removal target of 8,067 kg/year is inadequate and s/he wants to remove more phosphorus, then the manager identifies phosphorus removed as a new issue to include with a new objective rather than a requirement or constraint.

To address, reformulate the problem with a new, second objective (maximize phosphorus removed), solve for the *pareto* alternatives, and define a tolerable deviation for the new objective (here, 56% from the 14,400 kg/year phosphorus removed when maximizing the new objective separately and independently to still allow the prior optimal solution). Then, add the new objective as a new parallel axis to the right of the removal cost axis and present the *pareto* set generated by a standard multi-objective method such as, here, the constraint method (Figure 5, thick black lines on the main Parallel Coordinate plot and black triangles on the inset Cartesian plot). Finally, add numerous newly generated alternatives that are stratified through the objective and decision spaces of the near-optimal region (Figure 5, green lines and dots). Note the ideal point for the two modelled objectives is at high phosphorus removal, low removal cost (bottom right of inset plot) while the upward sloping line of black triangles is the familiar *pareto*-tradeoff for the stated objectives. This tradeoff is also indicated on the main Parallel Coordinate plot by the black line segments that cross from the first to the second axes and converge towards a point to the right of the second phosphorus removed axes. Typical multi-objective analyses try to identify a preferred non-dominated alternative from the *pareto* set. Here, the near-optimal alternative generation and visualization show that the non-dominated alternatives (thick black lines) concentrate in a sub-region of the decision space where *pareto* improvements in the two objectives are achieved mainly by stabilizing longer or shorter lengths of stream banks, particularly in the sub-watershed below Wanship. Interestingly and problematically, the MGA-Serial alternatives (purple lines in the Parallel Coordinate plot, purple circles in the inset) all fall on one boundary of the objective space (high removal cost) whereas numerous stratified-sampled alternatives improve on the MGA alternatives in *both stated objectives*. Highlighting the newly generated alternatives that remove more than 12,000 kg of phosphorus identifies multiple alternatives that remove large quantities of phosphorus from varied practices spread across the three sub-watersheds (Figure 6, orange lines on the parallel coordinate plot and orange crosses in the inset). Managers can select one of these alternatives as preferred to the *pareto* solutions and terminate interaction. Or the analyst can again elicit the un-modelled issue(s) that motivate the preference (e.g., more equitably site phosphorus removal practices across sub-watersheds), add a new equalization objective, generate new alternatives, and re-plot the new alternatives with the equity measure as a new objective.

6.5 Expanding the near-optimal region

Entering a single new tolerable deviation value of 125% significantly expands the near-optimal region and generates some 2,500 new alternatives with removal costs up to $1,230,000 (Figure 7). Here, green shading redundantly indicates removal cost and is ramped in intervals of 5% tolerable deviation from light green (125%) to dark green (100%, optimal cost, plotted on top) to show how the tolerable deviation effects the range of allowable phosphorus removal for each practice. As expected, larger deviations permit larger ranges of removal for many practices like protect grazing lands, cover crop, grass filter strips, conservation tillage, manage agricultural nutrients, sprinkler irrigation, and stabilize stream banks that target grazing and runoff sources (lightest green lines visible at extents). For other practices such as stabilize stream banks in Chalk Creek and the Weber below Wanship sub-watershed, the lightest green lines are not visible at the extents (hidden underneath) and the tolerable deviation does not change the allowable range. For these practices, a system constraint—rather than the near-optimal tolerable deviation—limits implementation. Lastly, the darkest green lines highlight combinations of manage agricultural nutrients, protect grazing land, and stabilize stream banks that include the problem's multiple global optima or maintain total removal cost very near the optimal removal cost. To highlight just the multiple global optima, set the tolerable deviation parameter value to 1.0 and resample (results not shown).

7 DISCUSSION

New stratified Monte Carlo Markov Chain sampling and Parallel Coordinate visualization tools generate a large number of near-optimal alternatives that perform within a tolerable deviation of the optimal objective function value and communicate the structure and extent of the near-optimal region. Use of the tools to select and locate phosphorus removal practices to meet load reduction targets for Echo Reservoir, Utah found that managers can implement any practice in any of the sub-watersheds that drain to the reservoir at costs between 100% and 110% of the optimal cost. In contrast, the optimal solution recommended five removal practices while 22, 13, and 10 alternatives generated by, respectively, the MGA- HSJ, Serial, and Simultaneous methods recommended less varied removal practices and levels all at the 110% cost threshold. The new tools take less time to generate more numerous alternatives that more completely show the near-optimal region. Results reveal managers have great flexibility to meet the phosphorus reduction target.

The stratified sampling and MGA results differ markedly for scaling, generation criterion, and methodological reasons. In terms of scaling, the stratified-sampling method solves a larger number of smaller optimization problems to identify decision variable maximum extents, whereas MGA- Serial and Simultaneous solve fewer, larger problems to maximize distance between alternatives (Table 1). The maximum extent problems are also linear in the objective whereas the MGA-Serial and Simultaneous distance metrics are non-linear, discontinuous, and must be solved with more computationally intensive methods. Further, the MGA-Serial and Simultaneous problems grow in size with the number of generated alternatives whereas stratified sampling simply adds Monte-Carlo Markov chain samples within prior-identified maximum extents. Thus, stratified sampling generates more alternatives much faster than the MGA-Serial and Simultaneous even for Echo Reservoir problem with 39 decision variables.

MGA-HSJ generates alternatives more quickly up to a density of one alternative per decision variable, but it’s maximally different search criteria does not necessary identify near-optimal region extents for all decision variables. Minimizing the sum the non-zero decision variables does not control which and how many variables *enter* the non-zero set and at what level(s). Thus, one or multiple variables enter at a low or arbitrary level on a particular iteration, MGA-HSJ drives those variables back to zero for remaining iterations, and never identifies their true (larger) extents. Even a density of one alternative per decision variable gives an incomplete view of the region for moderate and large problems. In contrast, the new tools achieve a higher alternative density by identifying the extents of each decision variable and objective, then sampling alternatives from the boundaries and throughout the linked decision and objectives spaces of the region.

Methodologically, MGA uses an optimization model and an objective function to express a narrow preference for where in the near-optimal region the analyst seeks new alternatives to address un-modelled issues. This optimizing may generate useful alternatives for select issues that align with the objective function parameterization. But for other un-modelled issues that do not align, useful alternatives could lie *anywhere* in the near-optimal region. (If the analyst has information on where to find useful alternatives, the analyst should model the underlying issue, tune the MGA objective function, and better direct generation towards the more useful alternatives). Because the analyst does not generally have *a priori* information on the un-modelled issues, the new tools instead use a sweep method to first stratify sample alternatives from throughout the near-optimal region. Then visualize them *en mass* and offer interactive plot controls to help streamline the process to select preferred alternatives, elicit un-modelled issues that motivate the preference, update the model formulation to respond to those issues, and generate new near-optimal alternatives from the updated formulation.

The several interactive examples illustrate the differences. For example, the more complete coverage of stratified sampled alternatives revealed options to set sliders to shift and increase phosphorus removal by stream bank stabilization (Figure 4) and prompted updating the model formulation to add a second objective to increase the total phosphorus removed. In the resulting multi-objective formulation, MGA alternatives clustered on an objective region boundary of large removal cost (Figure 5), whereas numerous stratified sampled alternatives were distributed throughout the near-optimal decision and objective spaces, showed numerous ways to spread implementation across sub-watersheds, and improve on the MGA alternatives in *each stated objective*.

Further interaction relaxed the tolerable deviation parameter to expand the near-optimal region. Results showed the progressive effects of expansion on region composition and ranges of allowable phosphorus removal for each practice. Typically, Parallel Coordinate plots only permit comparisons between *adjacent* axes [*Edsall*, 2003]. Here, ramping color along the removal cost axis in Figure 7 created contours of removal cost through the near-optimal region, simultaneously related removal cost to *all* other plot axes, and also allows the user to compare the problem's multiple global optima to near-optimal alternatives.

Admittedly, the outcome of interactions to specify a preferred alternative depend on the sequence to set sliders similar to ordering axes on a Parallel Coordinate plot [*Edsall*, 2003]. At the same time, the controls allow managers to jump directly to the region features of most interest. All of the examples presented are demonstrative and important work remains to test the interactive process with managers and decision makers. Towards this end, my research group is now working with local stakeholders to apply the near-optimal methods to a model that allocates water for environmental and ecological objectives in the lower Bear River, Utah [*Alafifi and Rosenberg*, 2014].

A key aspect to test is whether the large number of generated alternatives—some 2,500 alternatives for the Echo Reservoir problem—overwhelm users. The large number fills the Parallel Coordinate plot, allows the user to view generated alternatives together *en mass* as a single more comprehensive near-optimal region, and serves as a reference frame to guide subsequent interactive exploration of the region for preferred alternatives. The problem structure and size determine the appropriate number of alternatives to initially generate. Minimally, this number should be $2(n+d)$ to separately and independently show the minimum and maximum extents for each decision variable and objective function in the near-optimal region. But the number can be reduced to 2, 4, 6, … to identify extents for a sub-set of decision variables on which the user will focus initial interaction.

The current tools and applications (including two-staged stochastic water supply planning in Amman, Jordan [*Rosenberg*, 2012]) feature single- and multi-objective linear and mixed-integer problems that have convex, bounded feasible regions. Important work remains to extend the tools to non-linear problems. To extend, pass model data in structures that can describe non-linear problems. Further, generate alternatives with Monte-Carlo Markov Chain samplers such as Hit-and-Run [*Kroese et al.*, 2011] that work over non-convex regions. My research group is also starting to make these extensions.

Finally, the new tools help users explore the near-optimal region and streamline the process to elicit un-modelled issues but do not guarantee a user will find a preferred alternative to the optimal solution. First, a preferred alternative may not lie in the near-optimal region and second a user may stop exploring before they find an alternative. Even should a user stop exploring, the blended tools offer new insights on the region and problem structure and can elicit more varied un-modelled issues than either the optimal solution or prior MGA generation methods.

8 CONCLUSIONS

New blended alternative generation, visualization, and interaction tools tackle the challenges to quickly and comprehensively identify and tractably communicate the structure and extent of the near-optimal region for an optimization problem. Stratified Monte-Carlo Markov Chain sampling generates a large number of alternatives spread through the objective and decision spaces of the near-optimal region. A Parallel Coordinate plot visualizes generated alternatives *en mass* as a region. Interactive sliders, checkboxes, re-sampling and other controls allow users to further explore region features of most interest. The tools also help streamline the process to elicit un-modelled issues and update model formulations in response to the issues.

Use of the tools for a linear programming water quality management problem at Echo Reservoir, Utah showed managers can flexibly implement any phosphorus removal practice in any sub-watershed and keep removal cost within 110% of the optimal cost. This flexibility was upheld by further interaction and exploration, using elicited information to transform the original single-objective formulation into a multi-objective problem, and relaxing the near-optimal tolerance parameter to explore an expanded region farther from the optimum. Compared to three MGA methods, the new tools generate more numerous alternatives faster, more completely show the near-optimal region, and can help elicit a larger set of un-modelled issues.

Important work remains to test managers and decision makers use of the tools as well as extend to non-linear problems. Here, demonstrations for linear and mixed-integer problems: (i) communicate the structure and extent of the near-optimal region, (ii) guide further exploration within the region, (iii) help streamline the process to elicit un-modelled issues and update model formulations, and (iv) show the numerous and flexible ways to manage a system and maintain near-optimal performance.

9 ACKNOWLEDGMENTS

This work was funded by NSF grant #1149297. Ayman Alafifi, five anonymous reviewers, the associated editor, and editor provided extensive comments on prior drafts. All model, data, and script files, as well as directions to generate each figure and all results presented in the paper are available for download at https://github.com/dzeke/Blended-Near-Optimal-Tools.

10 APPENDIX I– PHOSPHORUS REMOVAL MODEL FORMULATION

This appendix presents the linear programming model formulation for the Echo Reservoir example. The formulation follows identically from *Alminagorta et al.* [2013] except here the sole decision variables are phosphorus mass removed by practice *i* targeting phosphorus source *s* in sub-watershed *w* (P_{iws}; kg). This specification eliminates the equity constraints that relate land area and stream bank length decisions to phosphorus mass. The single objective minimizes total removal costs (Z):

$$Min \ \ Z = \sum_{iws} u_i P_{iws} \qquad \text{[Eq. A1]}$$

where u_i = unit cost to remove phosphorus by practice i ($/kg).

Constraints are:

- Phosphorus removal across the three sub-watersheds must meet or exceed global load reduction targets for each source s (p_{ws}; kg)

$$\sum_{iw} c_{is} P_{iws} \geq \sum_{w} p_{ws}, \ \forall s \qquad \text{[Eq. A2]}$$

  where c_{is} = a binary parameter that takes the value 1 to indicate phosphorus removal practice i can be applied to source s (0 otherwise).

- Phosphorus removal is limited by the available land area and stream bank length resources in each sub-watershed (b_{gw}; km^2 or km)

$$\sum_{iw} \frac{c_{is} x_{gi}}{e_i} P_{iws} \leq b_{gw}, \ \forall g, w \qquad \text{[Eq. A3]}$$

  where x_{gi} = a binary parameter that takes the value 1 to indicate phosphorus removal practice i precludes implementing other practices on the same resource g (0 otherwise) and e_i = removal practice efficiency (kg P/km^2 or kg P/km).

- Phosphorus removal cannot exceed the existing load in each sub-watershed for each source s (l_{sw}; kg)

$$\sum_{iw} c_{is} P_{iws} \leq l_{ws}, \ \forall w, s \qquad \text{[Eq. A4]}$$

- All decision variables are non-negative, $P_{iws} \geq 0, \ \forall i, w, s$ [Eq. A5].

Eq. [A1] subject to [A2-A5] is solved for the optimal cost removal (Z*) before moving onto near-optimal analysis.

11 REFERENCES

Adams, C., and J. Whitehead (2006), Echo reservoir TMDL water quality study*Rep.*, Utah Dept. of Environmental Quality—Division of Water Quality, Salt Lake City, Utah. http://www.waterquality.utah.gov/TMDL/Echo_Reservoir_TMDL.pdf.

Alafifi, A. H., and D. E. Rosenberg (2014), Rethinking Riverine Habitat Quality: Integrated Systems Modeling to Improve Watershed Habitat Management and Decision Making in *7th Intl. Congress on Env. Modelling and Software,*, edited by D. P. Ames, N. W. T. Quinn and A. E. Rizzoli, pp. 1-8, International Environmental Modelling and Software Society San Diego, CA. http://www.iemss.org/sites/iemss2014/papers/iemss2014_submission_261.pdf.

Alminagorta, O., B. Tesfatsion, D. E. Rosenberg, and B. Neilson (2013), Simple Optimization Method to Determine Best Management Practices to Reduce Phosphorus Loading in Echo Reservoir, Utah, *Journal of Water Resources Planning and Management*, *139*(1), 122-125. http://ascelibrary.org/doi/abs/10.1061/%28ASCE%29WR.1943-5452.0000224.

Aufiero, A., R. S. Sessa, and E. Weber (2001), Set-valued control laws in minmax control problem, paper presented at Modeling and control in environmental issues, International Federation of Automatic Control, Yokohama, Japan. http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCsQFjAA&url=ftp%3A%2F%2Fhome.dei.polimi.it%2Foutgoing%2FAndrea.Francesco.Castelletti%2Fpapers%2Fsoncini_set_val.pdf&ei=FMmoUtjgFozwrAHUvYBg&usg=AFQjCNH7Ur4bx6gSI5ixfupvzbvjroWL7w&bvm=bv.57799294,d.aWM.

Benham, T. (2011), Uniform distribution over a convex polytope, edited, Matlab Central. http://www.mathworks.com/matlabcentral/fileexchange/34208.

Brill, E. D., Jr., S.-Y. Chang, and L. D. Hopkins (1982), Modeling to Generate Alternatives: The HSJ Approach and an Illustration Using a Problem in Land Use Planning, *Management Science*, *28*(3), 221-235. http://www.jstor.org/stable/2630877.

Brown, G. G., R. F. Dell, and R. K. Wood (1997), Optimization and persistence, *Interfaces*, *27*(5), 15-37. http://pubsonline.informs.org/doi/abs/10.1287/inte.27.5.15.

Burton, R. O., J. S. Gidley, B. S. Baker, and K. J. Reda-Wilson (1987), Nearly Optimal Linear Programming Solutions: Some Conceptual Issues and a Farm Management Application, *American Journal of Agricultural Economics*, *69*(4), 813-818. http://ajae.oxfordjournals.org/content/69/4/813.abstract.

Castelletti, A., A. V. Lotov, and R. Soncini-Sessa (2010), Visualization-based multi-objective improvement of environmental decision-making using linearization of response surfaces, *Environmental Modelling & Software*, *25*(12), 1552-1564. http://www.sciencedirect.com/science/article/pii/S1364815210001775.

Chang, S.-Y., E. D. Brill, Jr., and L. D. Hopkins (1982), Use of mathematical models to generate alternative solutions to water resources planning problems, *Water Resources Research*, *18*(1), 58-64. http://dx.doi.org/10.1029/WR018i001p00058.

Chen, M.-H., and B. Schmeiser (1993), Performance of the Gibbs, Hit-and-Run, and Metropolis Samplers, *Journal of Computational and Graphical Statistics*, *2*(3), 251-272. http://www.tandfonline.com/doi/abs/10.1080/10618600.1993.10474611.

Cohon, J. L., and D. H. Marks (1975), A review and evaluation of multiobjective programing techniques, *Water Resources Research*, *11*(2), 208-220. http://dx.doi.org/10.1029/WR011i002p00208.

Deb, K. (2008), Introduction to Evolutionary Multiobjective Optimization, in *Multiobjective Optimization*, edited by J. Branke, K. Deb, K. Miettinen and R. Słowiński, pp. 59-96, Springer Berlin Heidelberg. http://link.springer.com/chapter/10.1007/978-3-540-88908-3_3.

Edsall, R. M. (2003), The parallel coordinate plot in action: design and use for geographic visualization, *Computational Statistics & Data Analysis*, *43*(4), 605-619. http://www.sciencedirect.com/science/article/pii/S0167947302002955.

Gelfand, A. E., and A. F. M. Smith (1990), Sampling-Based Approaches to Calculating Marginal Densities, *Journal of the American Statistical Association*, *85*(410), 398-409. http://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10476213.

Harrington, J. J., and J. S. Gidley (1985), The Variability of Alternative Decisions in a Water Resources Planning Problem, *Water Resources Research*, *21*(12), 1831-1840. http://dx.doi.org/10.1029/WR021i012p01831.

Hastings, W. K. (1970), Monte Carlo Sampling Methods Using Markov Chains and Their Applications, *Biometrika*, *57*(1), 97-109. http://www.jstor.org/stable/2334940.

Horsburgh, J. S., N. O. Mesner, D. K. Stevens, A. Caplan, T. Glover, and B. T. Neilson (2009), USEPA targeted watersheds grant Bear River Basin*Rep. Project # WS-97807301*, 292 pp, Utah State University, Logan, Utah. http://bearriverinfo.org/files/publications/publication/pub__8578631.pdf.

Hsieh, C.-D., and W.-F. Yang (2007), Optimal nonpoint source pollution control strategies for a reservoir watershed in Taiwan, *Journal of Environmental Management*, *85*(4), 908-917. http://www.sciencedirect.com/science/article/pii/S0301479706003446.

Inselberg, A. (2009), *Parallel Coordinates: visual multidimensional geometry and its applications*, 554 pp., Springer, New York.

Kasprzyk, J. R., P. M. Reed, B. R. Kirsch, and G. W. Characklis (2009), Managing population and drought risks using many-objective water portfolio planning under uncertainty, *Water Resources Research*, *45*(12), W12401. http://dx.doi.org/10.1029/2009WR008121.

Kennedy, K. T., and J. T. Quinn (1998), Multiple Sourcing Alternatives Using Nearly Optimal Programming, *Journal of Education for Business*, *73*(4), 206-210. http://dx.doi.org/10.1080/08832329809601631.

Kollat, J. B., and P. M. Reed (2007), A Framework for Visually Interactive Decision-making and Design using Evolutionary Multiobjective Optimization (VIDEO), *Environmental Modelling & Software*, *22*(12), 1691-1704. http://www.coe.psu.edu/water/images/3/36/Kollet_and_Reed_2007_VIDEO_Paper.pdf.

Korhonen, P., and J. Wallenius (1988), A pareto race, *Naval Research Logistics (NRL)*, *35*(6), 615-623. http://dx.doi.org/10.1002/1520-6750(198812)35:6<615::AID-NAV3220350608>3.0.CO;2-K.

Kroese, D. P., T. Taimre, and Z. I. Botev (2011), *Handbook of Monte Carlo Methods*, 772 pp., John Wiley and Sons, New York.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, 343 pp., Springer, New York.

Lotov, A. V., V. A. Bushenkov, and G. K. Kamenev (2004), *Interactive Decision Maps Approximation and Visualization of Pareto Frontier*, Springer-Verlag, Heidelberg.

Loughlin, D. H., S. R. Ranjithan, E. D. Brill, and J. W. Baugh (2001), Genetic algorithm approaches for addressing unmodeled objectives in optimization problems, *Engineering Optimization*, *33*(5), 549-569. http://dx.doi.org/10.1080/03052150108940933.

Maringanti, C., I. Chaubey, and J. Popp (2009), Development of a multiobjective optimization tool for the selection and placement of best management practices for nonpoint source pollution control, *Water Resources Research*, *45*(6), W06406. http://dx.doi.org/10.1029/2008WR007094.

Marler, R. T., and J. S. Arora (2004), Survey of multi-objective optimization methods for engineering, *Structural and Multidisciplinary Optimization*, *26*(6), 369-395. http://dx.doi.org/10.1007/s00158-003-0368-6.

Matheiss, T. H., and D. S. Rubin (1980), A Survey And Comparison Of Methods For Finding All Vertices Of Convex Polyhedral-Sets, *Mathematics of Operations Research*, *5*(2), 167-185. http://www.jstor.org/stable/10.2307/3689148.

Miettinen, K., F. Ruiz, and A. P. Wierzbicki (2008), Introduction to Multiobjective Optimization: Interactive Approaches, in *Multiobjective Optimization*, edited by J. Branke, K. Deb, K. Miettinen and R. Słowiński, pp. 27-57, Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-540-88908-3_2.

Nardini, A., and D. Montoya (1995), Remarks on a Min-Max Optimization Technique for the Management of a Single Multiannual Reservoir Aimed at Hydroelectric Generation and Water Supply, *Water Resources Research*, *31*(4), 1129-1135. http://dx.doi.org/10.1029/94WR02705.

Nemhauser, G. L. (1994), The Age Of Optimization - Solving Large-Scale Real-World Problems, *Operations Research*, *42*(1), 5-13. http://www.jstor.org/discover/10.2307/171518?sid=21105945757843&uid=3739256&uid=70&uid=2&uid=4&uid=2129&uid=3739928.

Orlovski, S., S. Rinaldi, and R. Soncini-Sessa (1983), A min-max approach to storage control problems, *Applied Mathematics and Computation*, *12*(2–3), 237-254. http://www.sciencedirect.com/science/article/pii/0096300383900103.

Ortiz, M. C., L. A. Sarabia, M. S. Sanchez, and D. Arroyo (2011), Improving the visualization of the Pareto-optimal front for the multi-response optimization of chromatographic determinations, *Analytica Chimica Acta*, *687*(2), 129-136.

Rogers, P. P., and M. B. Fiering (1986), Use of Systems Analysis in Water Management, *Water Resources Research*, *22*(9), 146S - 158S.

Rosenberg, D. E. (2012), Near-optimal water management to improve multi-objective decision making, paper presented at 2012 International Congress on Environmental Modelling and Software: Managing Resources of a Limited Planet: Pathways and Visions under Uncertainty, Sixth Biennial Meeting, International Environmental Modelling and Software

Society (iEMSs), Leipzig, Germany, July 1-5. http://www.iemss.org/sites/iemss2012//proceedings/A2_0656_Rosenberg.pdf.

Rubinstein, R. Y. (1986), *Monte Carlo optimization, simulation, and sensitivity of queueing networks*, John Wiley & Sons, New York, NY.

Sahinidis, N. V. (2004), Optimization under uncertainty: state-of-the-art and opportunities, *Computers & Chemical Engineering*, *28*(6-7), 971-983. http://www.sciencedirect.com/science/article/B6TFT-49YH97T-1/2/f15875aad97740410effc526416289aa.

Shenfield, A., P. J. Fleming, and M. Alkarouri (2007), Computational steering of a multi-objective evolutionary algorithm for engineering design, *Engineering Applications of Artificial Intelligence*, *20*(8), 1047-1057. http://www.sciencedirect.com/science/article/pii/S0952197607000085.

Smith, R. L. (1984), Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions, *Operations Research*, *32*(6), 1296-1308. http://www.jstor.org/stable/170949.

Steuer, R. E. (1986), *Multi Criteria Optimization: Theory, Computation, and Application*, John Wiley and Sons, New York.

Stummer, C., and E. Kiesling (2009), A multicriteria decision support system for competence-driven project portfolio selection, *International Journal of Information Technology & Decision Making*, *8*(2), 379-401. http://web.ebscohost.com/ehost/detail?vid=3&hid=122&sid=da5cb4f0-dd00-4817-8135-a079d8b26112%40sessionmgr115&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#db=cph&AN=53872988.

Wegman, E. J. (1990), Hyperdimensional Data Analysis Using Parallel Coordinates, *Journal of the American Statistical Association*, *85*(411), 664-675. http://www.jstor.org/stable/2290001.

Wierzbicki, A. (1979), The use of reference objectives in multiobjective optimization*Rep.*, 36 pp, International Institute for Applied Systems Analysis, Laxenburg, Austria. http://www.iiasa.ac.at/publication/more_WP-79-066.php.

Zechman, E. M., and S. R. Ranjithan (2004), An evolutionary algorithm to generate alternatives (EAGA) for engineering optimization problems, *Engineering Optimization*, *36*(5), 539-553. http://dx.doi.org/10.1080/03052150410001704863.

Zechman, E. M., and R. Ranjithan (2007), Generating Alternatives Using Evolutionary Algorithms for Water Resources and Environmental Management Problems, *Journal of Water Resources Planning and Management*, *133*(2), 156-165. http://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9496%282007%29133%3A2%28156%29.

Zhang, Q., and H. Li (2007), MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, *IEEE Transactions on Evolutionary Computation*, *11*(6), 712-731. http://dx.doi.org/10.1109/TEVC.2007.892759.

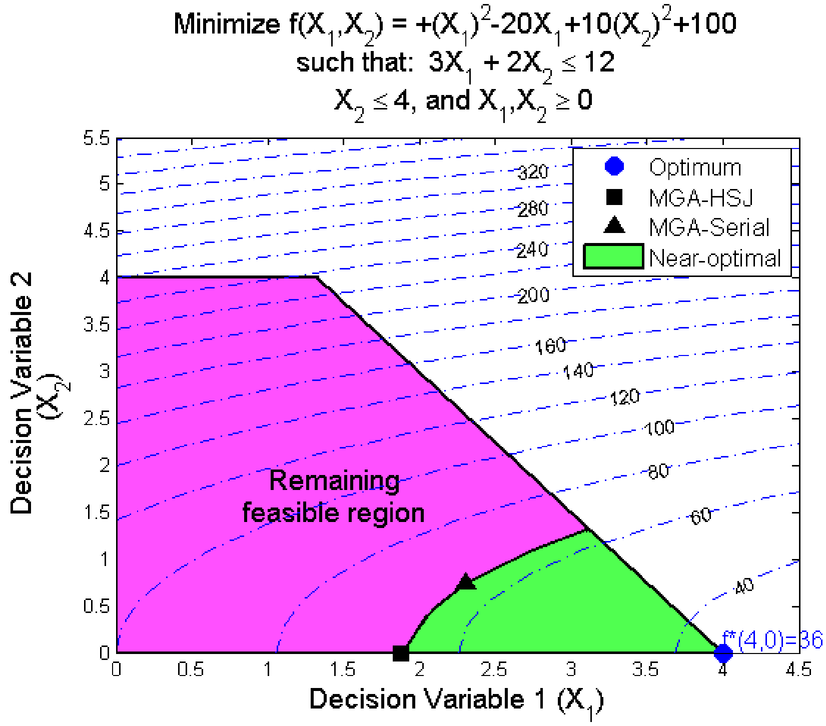

Figure 1. Comparison of the optimal solution and Modelling to Generate Alternatives (MGA) results to the near-optimal region for an example two-decision problem with a quadratic objective function, four linear constraints, tolerable deviation of γ=1.80, and generation density of 0.5 alternatives per decision variable for the MGA-Serial method that is at least three times larger than densities reported in prior applications.

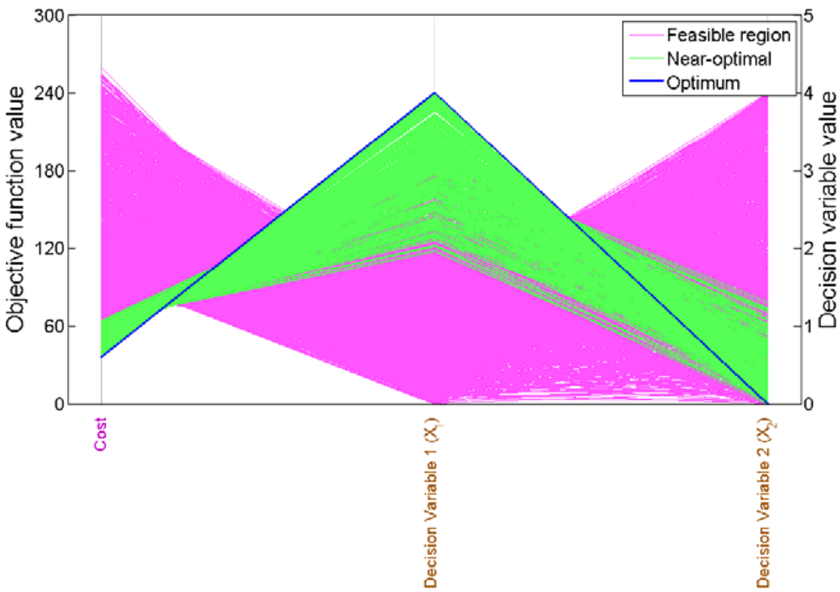

Figure 2. Parallel Coordinate plot that shows the optimal solution, feasible, and near-optimal regions across the linked objective (left most axis and left scale) and decision (right two axes and right scale) spaces for the example problem in Figure 1.

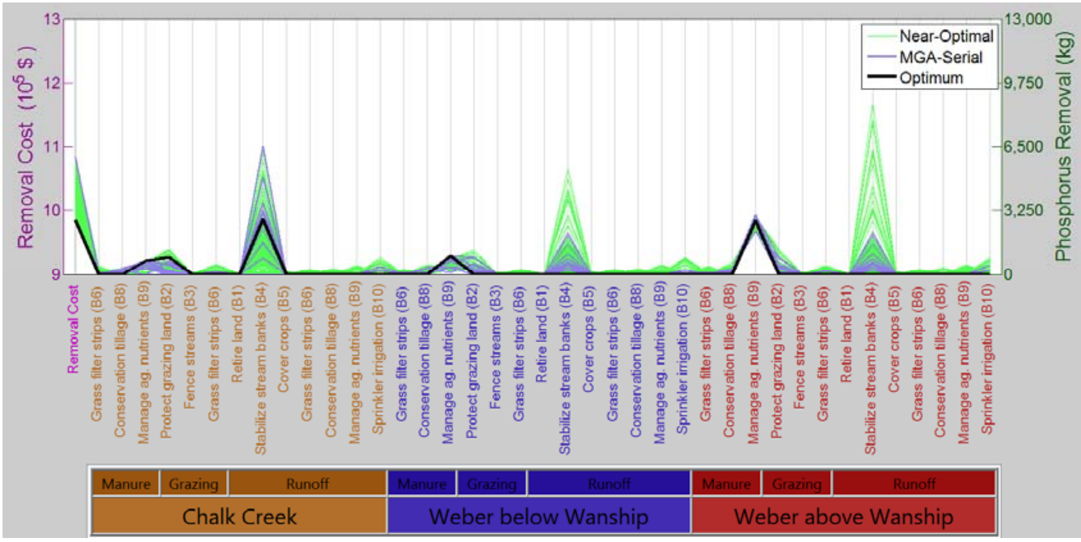

Figure 3. Some 2,500 stratified-sampled near-optimal phosphorus removal strategies for Echo Reservoir, Utah (green lines) more completely cover in shorter run time the region within 110% of the optimal removal cost than 13 alternatives serially generated by the Modelling to Generate Alternatives method (purple lines) or the optimal solution (black line).

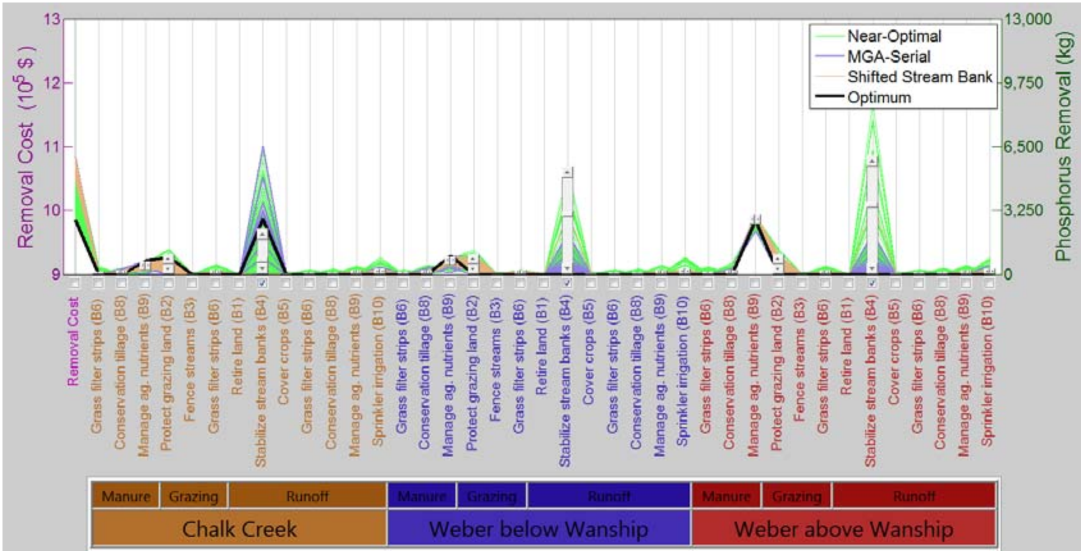

Figure 4. Exploring the effects (orange lines and slider heights) to not stabilize stream banks in Chalk Creek (first checked axis and slider) but shift and increase phosphorus removal in the Weber above and below Wanship sub-watersheds (second and third checked axes and sliders) in a sub-space of the near-optimal region (green lines) where MGA-Serial did not generate alternatives (purples lines).

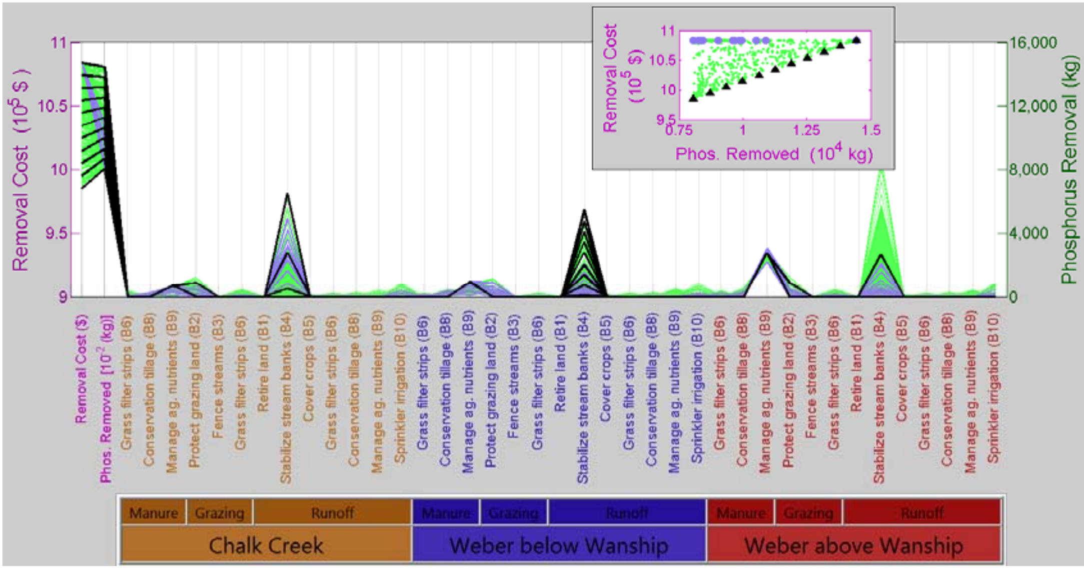

Figure 5. Comparing *pareto* solutions generated by the constraint method (thick black lines and black triangles), 13 alternatives generated by MGA-Serial (purple lines and purple circle), and stratified sampled near-optimal alternatives (green lines and green dots) for an updated multi-objective formulation in linked objective-decision spaces (main parallel coordinate plot) and the objective space (inset Cartesian plot).

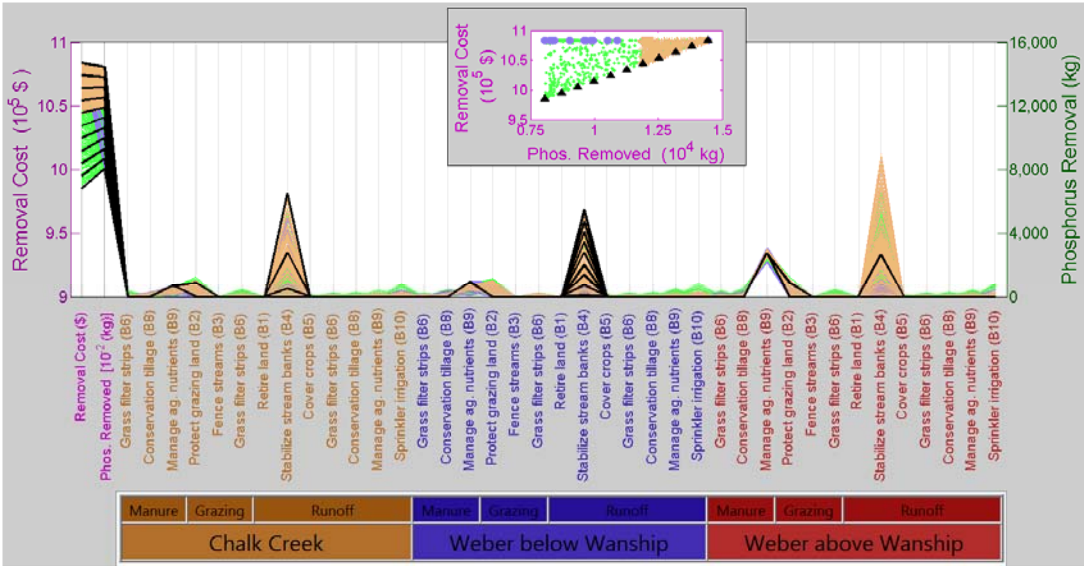

Figure 6. Sub-region of near-optimal alternatives that remove large amounts of phosphorus (orange lines and crosses) but use more varied locations and practices than either *pareto* solutions (thick black lines and black triangles) or MGA-Serial alternatives (purple lines and circles).

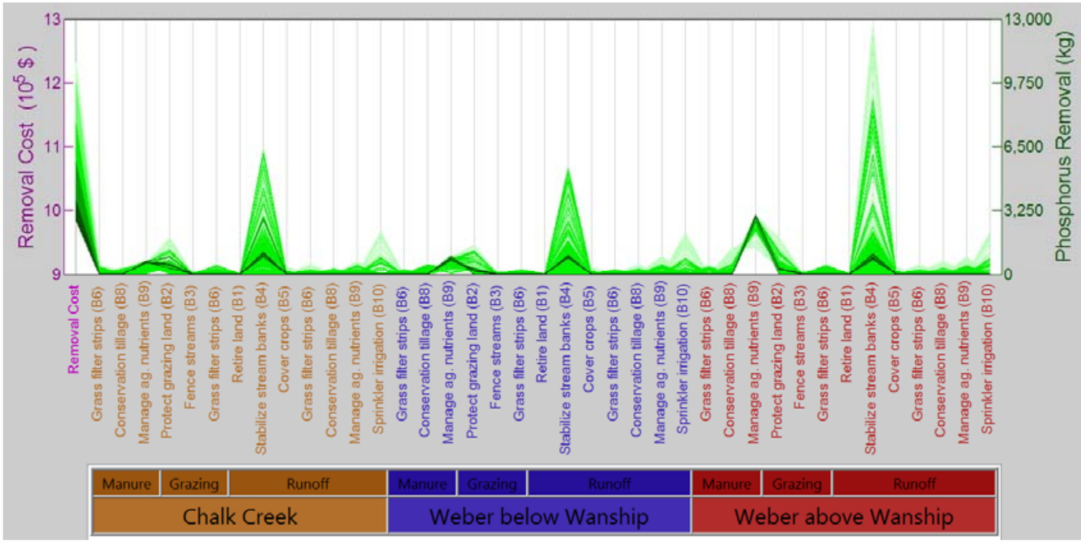

Figure 7. Progressive effects of relaxing the near-optimal tolerance parameter on ranges of allowable phosphorus removal for each practice. Green shading denotes removal cost and varies in 5% increments from dark green (optimal cost including multiple global optima) to light green (125% of optimal cost).

Table 1. Comparison of problem size, solution effort, and performance of near-optimal methods[a]

Method	Problem Size			Upper Bound on Number of Solves	Performance on Echo Reservoir Problem		
	Decision Variables	Constraints	Type[b]		Alternatives Generated	Time (sec)	Alt. Density[c]
MGA-HSJ	n	m	LP	n-1	22[d]	0.5	0.6
MGA-Serial	n	m	NLP	p	13	26.6	0.3
MGA-Simultaneous	p·n	p·m	NLP	1	10	101 to 553	0.3
Stratified sampling	n	m	LP	2(n+d)	2,483	25.4	63.7

a. To generate *p* near-optimal alternatives for an optimization problem with *n* decision variables, *m* constraints, and *d* objectives.
b. LP = Linear program, NLP = Non-linear program
c. Measured as alternatives per decision variable
d. More favorably scaled model variant