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

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 water quality management problem at Echo Reservoir, Utah identifies numerous and flexible alternatives to manage the system and maintain close-to-optimal performance. This flexibility moves beyond the traditional optimal solution and limited alternatives generated by the Modelling to Generate Alternatives method.

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

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

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

Abstract: State-of-the-art systems analysis techniques unanimously 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 additionally 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 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 full extent of the near-optimal region to an optimization problem. Interactive plot controls allow users to explore region features of most interest. The controls also streamline the process to elicit un-modelled issues and update the model formulation in response to elicited issues. Use for a single-objective 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. This 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. This flexibility moves beyond the traditional optimal solution and limited alternatives generated by the MGA method.

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

1 INTRODUCTION

The best is the enemy of the good
Voltaire, *La Bégueule* [1772]

State-of-the-art systems analysis techniques emphasize efficiently finding optimal and *pareto*-optimal solutions [*Brown et al.*, 1997] to large, uncertain, real-world, 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 also address un-modelled objectives, preferences among objectives, limits, uncertainties, and other shortcomings in the original model formulation [*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 them. Not surprisingly, both challenges intensify as problem size grows and the modeller has less ability to qualitatively and quantitatively characterize the un-modelled issues.

Modelling to Generate Alternatives [MGA; *Brill et al.*, 1982; *Chang et al.*, 1982] was an early method to generate near-optimal alternatives. MGA first solved for the optimal solution, then identified 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. Next, they generated a second alternative that was maximally different from the optimal solution and first alternative. The process continued until no new alternatives were generated. Generated alternatives were few, located at select vertices of near-optimal region, addressed un-modelled issues typically related to substitutability among decision variables that aligned with the maximally-different search criterion, and offered limited opportunity to elicit further un-modelled issues.

Subsequent near-optimal work has also focused on alternative generation. *Burton et al.* [1987] enumerated the near-optimal region vertices for a small, linear farm management problem, but encountered problems with scaling [*Harrington and Gidley*, 1985; *Matheiss and Rubin*, 1980]. Problems increase exponentially with problem size because one must enumerate all extreme points where model constraints intersect, then retain only feasible points as vertices to the near-optimal region.

A second class of generation methods substituted different and sometimes randomly parameterized objectives for MGA's maximally-different criterion [e.g., *Harrington and Gidley*, 1985; *Kennedy and Quinn*, 1998; *Makowski et al.*, 2000]. These methods also generated a few alternatives—one for each new objective tested—located at select vertices of the near-optimal region.

A third and contrasting class of uniform Monte-Carlo Markov chain sampling methods generate points inside a bounded region [*Chen and Schmeiser*, 1993; *Kroese et al.*, 2011; *Liu*, 2001] and remain to be 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] and scale to large problems. The methods 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 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. Unfortunately, uniform sampling congregates samples towards the centroid of the region and away from the vertices and surfaces. The reason is local volumes near extremities are very small compared to the region volume so the probability is low to sample alternatives that span the full region defined by the original and near-optimal tolerance constraints.

A fourth class of methods use evolutionary algorithms to extend MGA to larger non-linear problems [*Loughlin et al.*, 2001; *Zechman and Ranjithan*, 2004; *Zechman and Ranjithan*, 2007]. But like MGA, these methods generate a few alternatives that are located close to near-optimal region boundary and address un-modelled issues that align with the maximally-different search criterion.

The above near-optimal methods have generated a few alternatives that address some existent un-modelled issues and shortcomings of the optimal solution. To address more numerous un-modelled issues, a tool must: (i) generate a more comprehensive set of alternatives from throughout the near-optimal region, and (ii) not overwhelm users. These dual challenges similarly confront narrower searches of *pareto* frontiers for preferred compromise solutions to multi-objective problems where all objectives are *a priori* known and quantified [*Castelletti et al.*, 2010; *Cohon and Marks*, 1975; *Marler and Arora*, 2004]. For example, alternative generation [e.g., *Deb*, 2008] and set-value [*Aufiero et al.*, 2001; *Nardini and Montoya*, 1995; *Orlovski et al.*, 1983] techniques produce and present a manager with a large number of *pareto* alternatives. In contrast, interactive and progressive articulation of preference methods [*Korhonen and Wallenius*, 1988; *Miettinen et al.*, 2008; *Steuer*, 1986; *Wierzbicki*, 1979] involve the manager to identify one or a few alternatives of most interest. More recent methods blend the alternative generation, visualization, and interaction approaches to leverage computational, cognitive, and learning strategies [*Castelletti et al.*, 2010; *Lotov et al.*, 2004]. For example, generate and visualize one or numerous alternatives for users, then the user guides further generation and visualization until reaching an acceptable end point. These blended methods for *pareto* search can also help identify and communicate the structure of larger near-optimal regions and allow users to explore the region for alternatives that address un-modelled issues.

This paper presents new blended algorithms and tools to generate, visualize, and interactively explore the near-optimal region of an optimization problem. Stratified Monte-Carlo Markov Chain sampling identifies a large number of near-optimal alternatives that comprehensively span the region through both the decision and objective spaces. A Parallel coordinate plot [*Inselberg*, 2009; *Wegman*, 1990] 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. Interactive controls direct plot rendering and help the user explore the near-optimal region as well as seek alternatives that address un-modelled issues. Together, the tools additionally (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 while sections 3-5 present the alternative generation, visualization, and interactive tools. The remaining sections show use of the tools for a water quality management problem at Echo Reservoir, Utah, discuss results, and present conclusions.

2 NEAR-OPTIMAL FORMULATION

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,

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

subject to

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

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, 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.

For maximization problems, 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 $\gamma = 1.8$. In the example, the optimal solution allocates water only to User 1 (blue circle, bottom right).

Managers may prefer near-optimal alternatives that perform close to the optimal solution on the stated objective but better address un-modelled issues (such as equity in allocation among users). In the Figure 1 example, the single near-optimal alternative generated by MGA still allocates all water to User 1 at higher cost than the optimal cost (Figure 1, purple triangle, bottom middle) and will likely be disregarded. In contrast, generating and communicating the *entire* near-optimal region (Figure 1, green shaded area) identifies numerous strategies to share water between users and still keep allocation cost near the optimal cost. Further, when a manager selects one of these alternatives, the systems analyst can elicit the reason for the manager's selection and use the new information to update the model formulation 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 the near-optimal regions for larger problems that cannot be represented on a Cartesian plot.

3 ALTERNATIVE GENERATION

The stratified sampling generates alternatives that comprehensively span the near-optimal region and extends uniform Monte-Carlo Markov chain sampling of high-dimensional, closed regions [*Gelfand and Smith*, 1990; *Hastings*, 1970; *Kroese et al.*, 2011; *Liu*, 2001; *Rubinstein*, 1986; *Smith*, 1984].

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 ($\hat{f}_j; j = 1 \text{ to } p/(n+1)$) between the optimal objective function and tolerable deviation values [f^*, $\gamma\, f^*$]. 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), samples per group (p/[n+1]), or samples per Monte Carlo Markov chain to achieve the desired sampling density or focus attention on particular sub-regions. Below, I present extensions for several 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 during each step. Testing of different 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 Gibbs sampling (re-coded from [*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 Monte-Carlo Markov Chain method [*Gelfand and Smith*, 1990]. During cycles through decision variables, do not sample a value but rather 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 bounding extents ensure generating only feasible near-optimal alternatives and 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 this integer alternative generation method for a water supply planning problem in Amman, Jordan. One interactive tool in Section 5 adapts the method to generate a single alternative.

3.3 Multi-objective problems

The alternative generation method also extends to multi-objective problems. For a problem with i=1 to d objectives defined *a priori*, form $n+d$ (rather than n+1) groups in step a. Later in step e, introduce a tolerable deviation for each objective (γ_i) and uniformly sample j=1 to $p/(n+d)$ values $\hat{f}_{i,j}$ for the first objective (c_1) within its range [f_i*, $\gamma_i f_i$*]. Using the first sampled objective function value ($\hat{f}_{1,1}$), identify the extents for the second objective (conditioned on the original, near-optimal tolerance, and $\hat{f}_{1,1} = c_1(x)$ constraints). Then uniformly sample a value for the second objective within its conditioned extents ($\hat{g}_{2,1}$), 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. This sampling strategy emulates Gibbs sampling but cycles through the objectives rather than decision variables. Next, add the *d* sampled objective function values as *d* equity constraints to the original and near-optimal constraints, and then Monte-Carlo Markov Chain sample decision variable values with the specified objective function values. Repeat these steps for the remaining $\hat{f}_{i,j}$ values (*j*=2 to *p*/(*n*+*d*)) and then again for *i*=2 to *d* objectives. The resulting near-optimal alternatives span the objective and decision spaces, the *pareto* front, as well as the region that lies under the front. Here, 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 (Figure 2, left) 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 (Figure 2, right, pink and blue line segments). Recent uses in water management studies include to visualize discrete points on *pareto*-optimal 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]. For example, linear relationships between an objective function and two decision variables (Figure 2, left, blue circles in a line) are indicated in parallel coordinates by the points where line segments intersect (right plot, blue segment cross between each pair of adjacent axes. The Parallel Coordinate plotting tool uses these correspondences to visualize high-dimensional near-optimal regions.

First lay out parallel axes for the objective function and each decision variable. Second, draw a set of connected, polygonal lines that cross the objective function and decision variable axes at the corresponding optimal objective function and decision variable values. These connected lines represent the optimal solution. Next, draw additional polygonal lines for each generated near-optimal alternative. 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 polygonal lines cross each axis. For multi-objective problems, add axes for the new objectives. And although axes order influences Parallel Coordinate plot rendering [*Edsall*, 2003], interactive controls allow users to specify (and re-specify) axes order while they explore the near-optimal region.

5 INTERACTIVE EXPLORATION

Interactive mouse-over, highlighting, grouping, axes re-ordering, check box, sliders (scrollbars), and other controls on and next to the plot allow the user to interactively explore the near-optimal region. This exploration serves to: (i) control the rendering and visualization of 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. The 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.

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 the updated 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:

- Individual alternative. Set a slider for one decision variable within its current extents, update extents for all other sliders, set a second slider within its extents, and continue updating extents and setting sliders until specifying all decision variables (this sequence also represents one pass through the integer generation method described in Section 3.2 with no enumeration).
- Family of alternatives. Set one to *n*-1 sliders on decision axes to values within their extents, then use the alternative generation algorithm (section 3) to identify values for the remaining (un-set) decisions. The 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 an individual or 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 has an opportunity to step in to 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. Solve the updated model for a new optimal solution (or *pareto* set), 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 new alternative generation, visualization, and interaction tools are coded in Matlab 2013b and available for download at https://github.com/dzeke/Blended-Near-Optimal-Tools. The code repository also makes available the data and model files for the example application as well as scripts and directions to generate each figure.

6 EXAMPLE APPLICATION

I illustrate the new near-optimal generation, visualization, and interaction tools for a 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 builds on a single-objective linear program for Echo Reservoir [*Alminagorta et al.*, 2013] that I helped develop. Below, I overview the model for Echo Reservoir and compare the prior-published optimal solution, near-optimal alternatives generated by MGA, and by the new tools. Discussion highlights un-modelled issues and management insights the new tools identify.

6.1 Model background

The single-objective linear program identifies the cost-effective phosphorus removal 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. The program considered 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 excluded point sources because the sources already adopted best available treatment technologies. The 39 decision variables represent 39 allowable combinations of practices, sources, and sub-watershed locations. The objective function minimized removal costs while a constraint required the phosphorus load decrease by 8,067 kg/year as set forth in the TMDL. Additional constraints specified dependencies and mutual exclusive relationships among removal practices as well as upper limits for phosphorus removal. These limits are due to existing phosphorus loads, land area, and stream bank length available to adopt practices in each sub-watershed. *Alminagorta et al.* [2013] present the linear program formulation; the input data, model file, and output are available on the GitHub repository.

The 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 property allowed formulating as a linear program and solving for a scenario used here that specified a global reduction target and allowed trading of load reductions across sub-watersheds.

6.2 Optimal and near-optimal comparisons

A parallel coordinate plot (Figure 3) compares the (i) optimal solution (thick black line) to (ii) five maximally-different alternatives generated by MGA (purple lines), and (iii) some 2,500 near-optimal alternatives generated by the new stratified Monte Carlo Markov Chain sampling method (green lines). 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 five phosphorus removal practices (manage agriculture nutrients in each sub-watershed; protect grazing land and stabilize stream banks in one sub-watershed) and concentrates actions in the Chalk Creek sub-watershed. Five MGA alternatives (purple lines) each cost $1.08 million (110% of the optimal removal cost), implement a mix of 13 phosphorus reduction practices at up to three levels per practice, and target implementation in the sub-watersheds above and below Wanship.

In contrast, near-optimal alternatives generated with the new tools (green lines) cross each decision axis at much wider ranges of values and show the full near-optimal region. Here, 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 2,700 kg phosphorus indicated by the MGA alternatives. In each sub-watershed, managers can also install grass filter strips, fence streams, cover crop, adopt conservation tillage, and irrigate with sprinklers. Interestingly, managers must remove between 2,000 and 3,000 kg phosphorus from manure sources by managing agricultural nutrients in the sub-watershed above Wanship. Together, the newly generated alternatives show managers can flexibly implement any phosphorus removal practice in any sub-watershed and keep removal costs within 110% of the optimal and cost-minimizing value.

6.3 Interaction to explore the near-optimal region

The interaction tools facilitate further exploration of the newly-generated alternatives (Figure 4, right panel). For example, 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 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. This sequence (Figure 4, checked axes) specifies a set of alternatives that (i) shift the location to stabilize stream banks, and (ii) increase the total phosphorus removed by the practice. Remaining sliders (unchecked in Figure 4) show the effects of the 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. Managers can still flexibly protect grazing land across the sub-watersheds but have limited ability to adopt other practices like grass filter strips, conservation tillage, and fence streams through sprinkler irrigation.

With the three slider settings for stream bank stabilization and no allowable deviation, the generate button automatically adds three new equity constraints to the original and near-optimal constraints, updates the model formulation, and generates a new family of alternatives with shifted, increased stream bank stabilization (Figure 4, orange lines). The resulting alternatives have removal costs at the $1.08 million limit defined by the near-optimal tolerance parameter (110% of optimal removal cost) and show a variety of strategies that include shifted, increased removal by stream bank stabilization.

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

If the manager selects one of the shifted, increased stream bank stabilization alternatives as preferred to the optimal solution, then there is an opportunity to elicit one or more un-modelled issues and improve the model formulation. Say, for example, the manager explains that their preference is motivated by a belief that the TMDL phosphorus removal target of 8,060 kg/year is inadequate—the manager wants to remove more phosphorus. This statement identifies phosphorus removed as a new issue to consider 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 (Figure 5, thick black lines on the main Parallel Coordinate plot and black triangles on the inset Cartesian plot) and 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 the middle right of the inset plot (high phosphorus removal, low removal cost) while the upward sloping line of peach crosses is the familiar *pareto*-tradeoff for the objectives. This tradeoff is also indicated on the main Parallel Coordinate plot by the peach 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 near-optimal 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 alternatives (purple lines in the Parallel Coordinate plot, purple circle, upper-left of inset) congregate at an undesirable point in objective space (high removal cost, low phosphorus removed) while numerous newly generated 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 new tolerable deviation value of 125% to significantly expand the near-optimal region and generating up to 2,500 new alternatives allows the region with removal costs up to $1,230,000 (Figure 6). 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 several varied phosphorus removal strategies that maintain total removal cost very near the optimal removal cost.

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 full extent of the near-optimal region. Use of the tools for a linear program with 39 decisions to select and locate phosphorus removal practices to meet a load reduction target for Echo Reservoir, Utah found that managers can implement any of the practices in any of the sub-watersheds that drain to the reservoir and keep total cost within 110% of the optimal cost. In contrast, the optimal solution recommended five removal practices while the MGA approach generated five near-optimal alternatives that together used 13 removal practices concentrated in two sub-watersheds. The new tools reveal managers have great flexibility to meet the phosphorus reduction target.

Results from the two methods differ markedly because MGA seeks alternatives that are maximally different in the decision space and located at select vertices of the near-optimal region whereas the new tools sample alternatives from throughout the linked decision and objectives spaces. This difference was more pronounced when updating the model formulation to add a second objective to increase the total phosphorus removed. In the multi-objective formulation, the MGA alternatives were still few and congregated at vertexes far from the *pareto* front. In contrast, the new tools generated numerous alternatives that were well distributed throughout the near-optimal decision and objective spaces and improved on the MGA alternatives in *each stated objective*.

Adding interactive slider, checkbox, axis reordering, and other controls to the plot allows users to further explore the near-optimal region and generate additional alternatives. One example sequentially set sliders to shift and increase the phosphorus removed by stabilizing stream banks and showed the effects on remaining practices. The outcome of this interaction depended on the sequence to set sliders similar to ordering axes on a Parallel Coordinate plot [*Edsall*, 2003]. At the same time, the slider and axis reordering tools allow managers to jump directly to the region features of most interest.

A second example 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 6 created contours of removal cost through the near-optimal region and simultaneously related removal cost to *all* other plot axes.

The tools also help streamline the process to select preferred alternatives, elicit the un-modelled issues that motivate the preference, and update the model formulation to respond to those issues. This streamlining is achieved by using slider controls to elicit user preferences for alternative components, automatically encode the slider settings as additional constraints on decision variables in the model formulation, then generate new near-optimal alternatives from the updated formulation. When an aspiration instead motivates a preference for an alternative, the analyst must intervene to help elicit the reason for the preference and translate the articulated aspiration into a new model objective. In the example that added an objective to remove more phosphorus, the tools helped the user to select the preferred alternative, generate new alternatives from the updated model formulation, and visualize the resulting near-optimal region.

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 helps fill the Parallel Coordinate plot and is part of the strategy to first generate a large number of alternatives and visualize them en-mass to communicate the near-optimal region structure and full extent. Then provide interactive tools for users to further explore the features of most interest. 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 only for select, prior-specified decision variables for which the user will focus initial interaction.

The current tools and applications (including 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. Or, use an evolutionary algorithm and define niching operations and a fitness function to distribute alternatives across the near-optimal region. 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. Still, even if a user stops exploration, 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 generation methods such as MGA.

8 CONCLUSIONS

New blended alternative generation, visualization, and interaction tools tackle the dual challenges to comprehensively identify and tractably communicate the structure and full 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 the linked objective and decision features of generated alternatives en mass. 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 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. This flexibility moves beyond the traditional optimal solution and limited alternatives generated by the MGA method.

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 full 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, four anonymous reviewers, the associated editor, and editor provided extensive comments on prior drafts.

10 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. <Go to ISI>://A1997YD16300003.

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.

Makowski, D., E. M. T. Hendrix, M. K. van Ittersum, and W. A. H. Rossing (2000), A framework to study nearly optimal solutions of linear programming models developed for agricultural land use exploration, *Ecological Modelling*, *131*(1), 65-77. http://www.sciencedirect.com/science/article/pii/S0304380000002490.

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. <Go to ISI>://000220714000001.

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. <Go to ISI>://A1994NL48500001.

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.

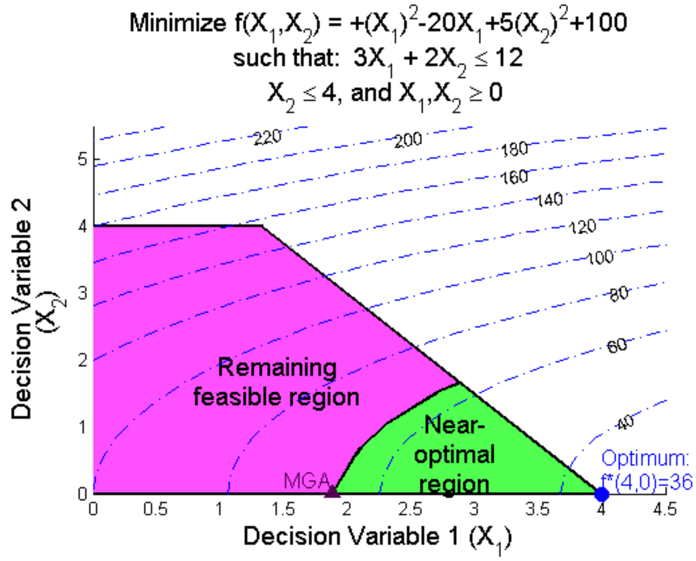


Figure 1. Optimal solution (blue circle), single maximally-different alternative produced by the Modelling to Generate Alternatives method (purple triangle), and entire near-optimal region (green shading) for an example two-decision problem with a quadratic objective function, four linear constraints, and tolerable deviation of γ=1.80.

Cartesian	Parallel
Point	Line
Line	Point
Plane	Pencil of points
Curve	Envelope of lines
Ellipse	Hyperbola
Convex region	Hyperbola
Interior points	Lines between brances

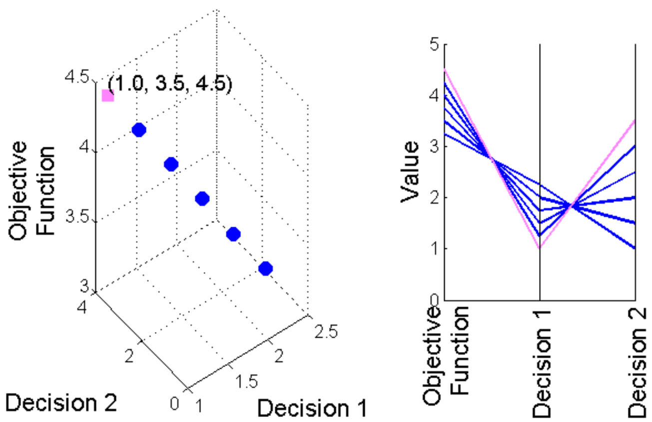


Figure 2. Point (pink), linear (blue), and other topological relations in Cartesian (left) and Parallel (right) coordinate systems.

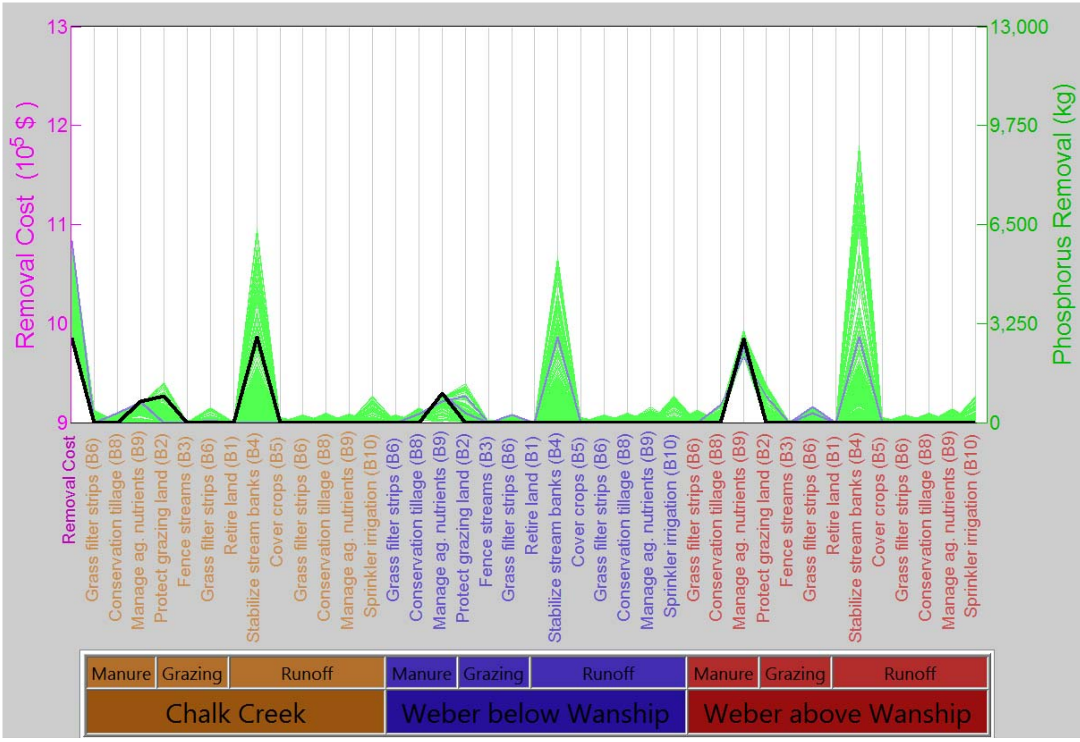


Figure 3. Phosphorus removal strategies for Echo Reservoir, Utah recommended by (i) the optimal solution (thick black line), (ii) five maximally-different Modelling to Generate Alternatives (purple lines) and (iii) some 2,500 stratified sampled near-optimal alternatives generated by the new tools (green lines).

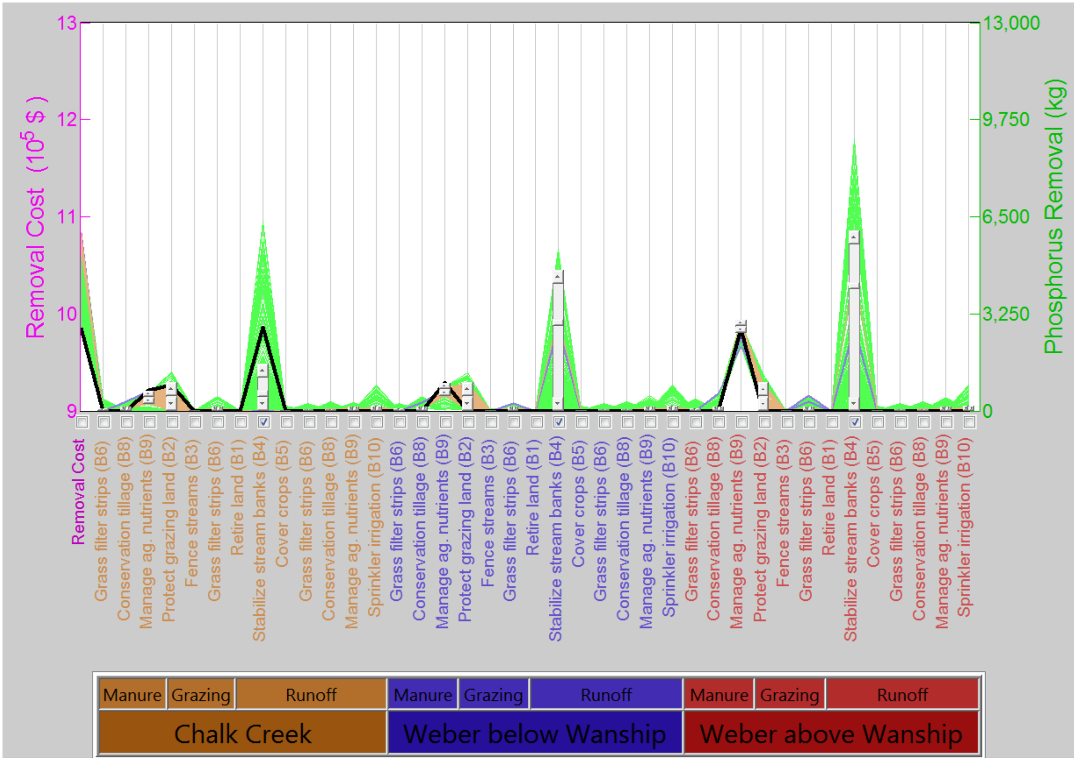


Figure 4. Exploring the effects to not stabilize stream banks in Chalk Creek (first checked axis and slider) but shift and increase phosphorus removal in the Weber above and below Wanship sub-watersheds (second and third checked axes and sliders) on ranges of allowable phosphorus removal for other practices (slider heights) and removal cost (orange lines).

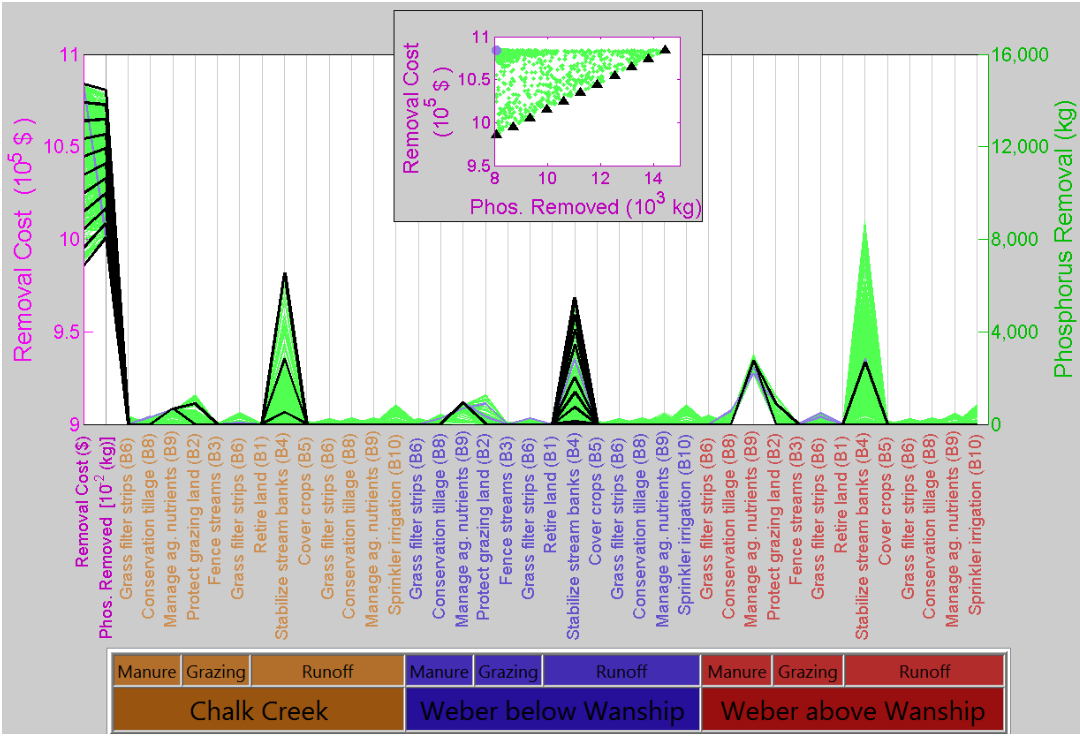


Figure 5. Comparing *pareto* solutions (thick black lines and black triangles), maximally-different Modelling to Generate Alternatives (purple lines and purple circle), and stratified sampled near-optimal alternatives (green lines and green dots) for an updated multi-objective formulation in linked objective-decision spaces (main parallel coordinate plot) and objective space (inset Cartesian plot).

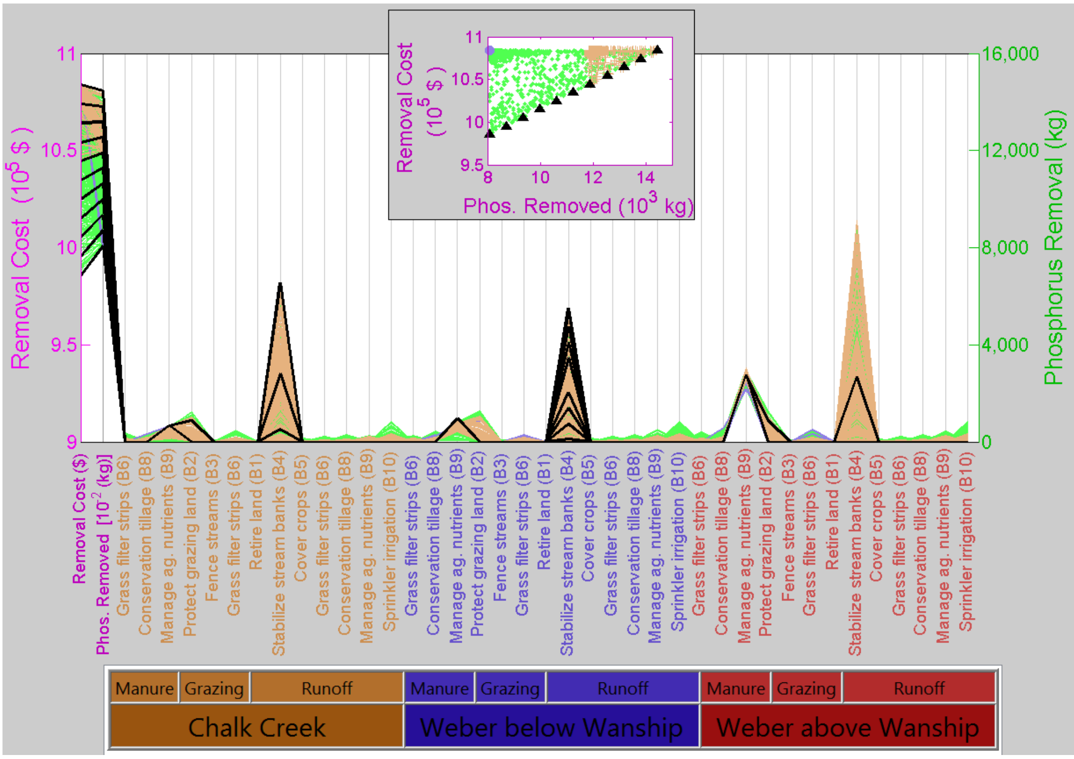
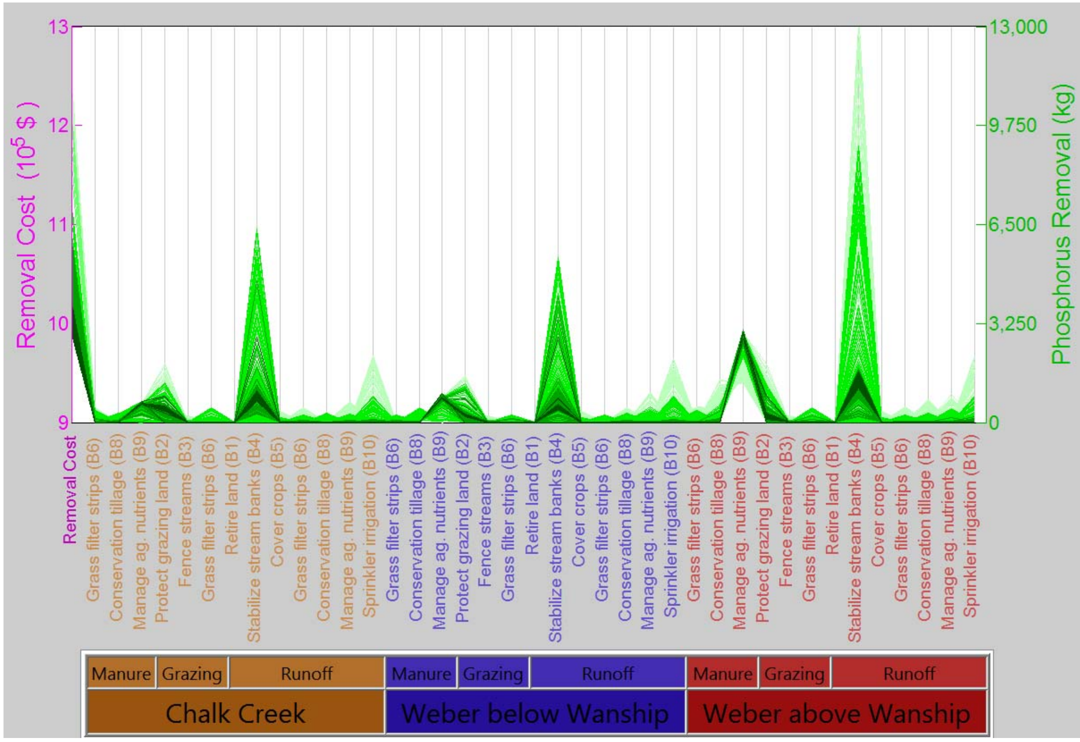


Figure 6. Sub-region of near-optimal alternatives that remove large amounts of phosphorus (orange lines and crosses) but use more varied locations and practices than either *pareto* solutions (thick black lines and black triangles) or MGA alternatives (purple lines and circles).

838 

839 **Figure 7. Progressive effects of relaxing the near-optimal tolerance parameter on**
840 **ranges of allowable phosphorus removal for each practice. Green shading denotes**
841 **removal cost and varies in 5% increments from dark green (optimal cost) to light**
842 **green (125% of optimal cost).**
843