Near-optimal management to improve water resources decision making

David E. Rosenberg

Assistant 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*

August 2013

Research Significance

I present a new interactive solution algorithm, parallel coordinate visualization, and automation tools to identify and explore all near-optimal solutions to an optimization problem. These near-optimal solutions are within a tolerable deviation from the optimal objective function value. I demonstrate the approach for existing, high-dimension, linear and mixed-integer water quality and supply management problems for Echo Reservoir, Utah and Amman, Jordan. For each problem, results identify numerous, promising, and varied near-optimal strategies beyond the single, prior-published, optimal solution.

Key Points

1. New algorithm and tools identify all near-optimal solutions
2. Apply to existing linear and mixed-integer water quality and supply problems
3. Results show many promising near-optimal strategies beyond the single optimal

Near-optimal management to improve water resources decision making

David E. Rosenberg
Assistant 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, water resources managers rather need decision aides that show multiple, promising, near-optimal alternatives. Why near-optimal? Because an optimal solution is optimal only for modelled issues; un-modelled preferences, limits, uncertainties, or other considerations persist. Early work mathematically formalized near-optimal as performance within a tolerable deviation from the optimal objective function value but found computational difficulties to describe near-optimal regions for large linear programs. Here, I present a new interactive solution algorithm and tools to identify, visualize, and explore all near-optimal solutions for high-dimension optimization problems. First, describe the near-optimal region from the original optimization constraints and objective function tolerance. Second, determine the maximal extents of each decision variable within the region and plot the extents in parallel coordinates as the lower and upper bounds on parallel axes where each axis represents a decision variable. Third, choose a value for one decision variable within its maximal extent, reduce the problem dimensionality by one degree, find the allowable range for the next variable, and repeat. This process identifies a sub-region. Extensions automate the process and simultaneously allow water managers to explore sub-regions of interest and/or incorporate previously un-modelled issues. I demonstrate the approach for both linear and mixed-integer water quality and supply management problems involving Echo Reservoir, Utah and Amman, Jordan. Results identify

numerous, promising, and varied near-optimal strategies beyond prior, singular optimal solutions.

Keywords: optimize; near-optimal; water management; linear program; mixed-integer program; Echo Reservoir, Utah; Amman, Jordan.

1 INTRODUCTION

State-of-the-art systems analysis techniques emphasize finding optimal solutions efficiently [*Brown et al.*, 1997] that consider uncertainties in model parameters [*Sahinidis*, 2004] for large, real-world, water supply and environmental problems [*Nemhauser*, 1994; *Zhang and Li*, 2007]. Yet, water managers rather need decision aides that embrace (instead of react to or reduce) uncertainty [*Tsoukiàs*, 2008]. Managers need options beyond single-modelled optimal or *pareto* sets to solution sets that encompass a variety of near-optimal results [*Brill et al.*, 1982; *Chang et al.*, 1982]. Why near-optimal? Because a modelled optimal solution is optimal only with respect to modelled issues; un-modelled preferences, limits, uncertainties, or additional considerations persist.

Brill's Modelling to Generate Alternatives [MGA; *Brill et al.*, 1982; *Chang et al.*, 1982] was an early approach to mathematically formalize the near-optimal notion. They solve for the optimal solution, then solve a reformulated model to identify a new solution that is maximally different from the prior one yet performs within a (near-optimal) tolerable deviation of the optimal objective function value. Finally, they repeat until no new solutions are generated. MGA identifies a few alternatives from the near-optimal region. Subsequent work described the near-optimal region for small, linear farm, land, and forest management problems [*Burton et al.*, 1987; *Makowski et al.*, 2000; *Mendoza et al.*, 1987], but found computational difficulties to completely enumerate plus identify the extreme points of the convex polytopes that represent near-optimal regions for large linear programs [*Burton et al.*, 1987; *Matheiss and Rubin*, 1980]. Computations increase quadratic ally with problem size (rather than linearly as with the Simplex method [*Dantzig*, 1963]) because one must evaluate all decision and slack variable value combinations both on and outside the polytope comprising the near-optimal region at points where constraints intersect. Today, near-optimal regions remain uncharacterized for larger linear, mix-integer, and other problems.

Here I present a new, fast, interactive solution algorithm, parallel coordinate plot visualization [*Inselberg*, 2009; *Wegman*, 1990], and automation tools to identify, visualize, and explore near-optimal regions for large linear and mixed-integer optimization problems. Sections 2–4 review the near-optimal formulation, present the interactive solution algorithm, parallel coordinate visualization, and automation tools. Section 5 demonstrates the approach and tools for existing linear and mixed-integer, 39- and 18- decision variable, water quality and supply management optimization problems for (i) Echo Reservoir, Utah, and (ii) Amman, Jordan. Section 6 discusses the results and identifies several important areas for further work. Section 7 concludes.

2 NEAR-OPTIMAL METHOD

For a general minimization problem (Box 1), existing enumeration algorithms [*Burton et al.*, 1987; *Chang et al.*, 1982; *Makowski et al.*, 2000] identify the near-optimal region in three steps:

1. Solve for the optimal decision variable values ($\mathbf{x}^*$)

Box 1. General Optimization Formulation

Minimize f= c($\mathbf{x}$),

such that: a($\mathbf{x}$) $\leq$ b and $\mathbf{x} \geq 0$

Where:
f = objective to be minimized,
c = objective function,
$\mathbf{x}$ = vector of *n* decision variables,
a = constraint functions, and
b = vector of *m* bounds for the *m* constraints.

using a standard linear [*Dantzig*, 1963], mix-integer, or other programming technique;

2. Allow a tolerable deviation, γ (γ ≥ 1 [unitless]), from the optimal objective function value (f^*):

$$c(\mathbf{x}) \leq \gamma \cdot f^* \qquad \text{[Eq. 1]}$$

3. Identify feasible extreme points of the near-optimal region defined by the original (Box 1) and objective function tolerance (Eq. 1) constraints.

Extreme points represent constraint intersections, are locations where slack variables are zero (for the associated constraints), and are enumerated as $\binom{m+n}{n}$ combinations with *m* decision and slack variables nonzero (basic) and *n* zero (non-basic). Then test each extreme point to see whether it satisfies the original and objective function tolerance constraints.

The objective function tolerance constraint (eq. 1) partitions the feasible region into two sub-regions: the near-optimal and remaining feasible regions (Figure 1). In Figure 1, the optimal point (black circle, bottom right) includes only X_1 whereas near-optimal solutions (green shading) include both X_1 *and* X_2 which managers will likely find preferable when equity, diversification, or other un-modelled issues exist. Enumeration to identify and test the feasibility of extreme points (purple triangles and circles in Figure 1) works fine when there are a small number of decision variables and constraints but is computationally intractable for larger, high-dimensional, water resources problems.

3 PARALLEL COORDINATE VISUALIZATION AND INTERACTIVE SOLUTION

For larger problems, it is also difficult to visualize the decision, objective, and near-optimal spaces. We typically plot data in Cartesian coordinates that *orthogonally* arrange axes (Figure 2, left). This arrangement can accommodate up to 3 axis (on a flat page), or possibly 6 or 7 axes if we cleverly use marker color, size, orientation, and other attributes to represent additional dimensions [*Kasprzyk et al.*, 2009; *Kollat and Reed*, 2007].

Parallel coordinate plots [*Inselberg*, 2009; *Wegman*, 1990] rather arrange axes in *parallel*, use polygonal lines that span axes to represent Cartesian points, and can accommodate large dimensions (Figure 2, right). Several recent water management studies use parallel coordinates to visualize discrete points on multi-objective *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 coordinates also behave so that line, curve, and other Cartesian concepts are uniquely recognizable in parallel coordinates [*Inselberg*, 2009]. The interactive solution algorithm uses this correspondence to identify high-dimension, near-optimal regions.

First, mathematically describe the near-optimal region from the original optimization program and objective function tolerance constraints (see section 2). Second, find the maximal extents for the *n* decision variables within the region by solving 2*n* optimization programs to separately and independently minimize and maximize each variable value subject to the original and near-optimal tolerance constraints. Then plot the maximal extents in parallel coordinates where each parallel axis represents a decision variable. For example, in Figure 3, Min X_1, Max X_1, Min X_2, Max X_2, ..., Min X_4, Max X_4 (each subject to the same hypersphere constraint $X_1^2 + X_2^2 + X_3^2 + X_4^2 \leq 4$) yields X_1 maximum extents 0.2 to 2 and X_2, X_3, X_4 extents -2 to 2. Third, choose a value for one decision variable within its maximal extents (e.g., dashed lines cross the X_1 axis at 1.6), reduce the problem dimensionality by one degree, and solve for the extents and allowable range of the next variable, say X_2, given the value selected for X_1. Choose a value for X_2 within the identified range, and repeat for all successive variables. This process shows a sub-region (such as X_1=1.6, X_3=X_4=0, and X_2 ε [-1.2, 1.2]). Finally, repeat step 3 for different combinations of decision variable values. Together, the combinations represent the near-optimal region (green lines in Figure 3) and plot within the outer envelopes of lines that span each parallel axes (these envelopes represent the bounding hypersphere constraint).

Visually, this solution algorithm is analogous to a control panel with *n* sliders positioned side-by-side on each axis, one for each decision variable [*Inselberg*, 2009]. Adjust and set the slider for one variable within its extents, then recalculate the feasible ranges for remaining sliders. The algorithm is fast and computes in *O*(n) time (representing the 2*n* optimization programs solved to find the extents for *n* variables). The algorithm is also interactive and allows the user to dynamically explore (by setting sliders to values) the near-optimal region. At the same time, the algorithm can also identify high-dimension linkages between the objective *and* decision spaces. Simply add an objective axis parallel to the decision axes (Figure 3, *f(x)* [pink] at the left) or multiple objective axes for multi-objective problems. Additionally, rearrange axes to start with any decision variable or focus on correlations among subsets of axes. This flexibility allows a water manager to jump directly to a decision or objective function or value of interest and further promotes interactive use of the tool.

4 EXTENSIONS FOR LINEAR AND MIXED-INTEGER PROBLEMS

Steps 2 and 3 of the interactive algorithm present the water manager with the maximal extents for each decision variable and allow him/her to select a value for a decision variable, update extents for the next variable, and repeat. For large problems this interactive process may require lengthy interactions to find even one near-optimal solution. However, we can automate the process to first show the user the near-optimal region. Then, the user can further explore sub-regions of interest using the interactive method. The automation method depends on the problem structure.

4.1 Linear problems

For linear problems with continuous variables, there are an infinite number of variable-value combinations within the near-optimal region and it is not possible to enumerate the region with a finite set of solutions chosen by selecting variable values. Two options are possible. First, accept this limitation and use the interactive algorithm as described in Section 3 to identify and explore the portion(s) of the near-optimal region of management interest. Or second, random sample a large number of solutions from within the near-optimal region to communicate the composition, breadth, and depth of solutions in the region and guide the manager to further explore the region using the interactive algorithm. Later, after the user has selected values for one or a more decision variables, reduce the problem dimensionality and repeat—random sample solutions from the sub-region to aide further exploration.

Many methods exist to uniformly and randomly sample solutions from within a polytope defined by a bounded system of linear constraints [*Chen and Schmeiser*, 1993; *Kroese et al.*, 2011; *Liu*, 2001; *Smith*, 1984]. These methods include (i) rejection sampling, (ii) vertex identification [*Matheiss and Rubin*, 1980] combined with simplex decomposition (using a method like Delaunay triangulation), unit simplex sampling [*Rubinstein*, 1986], and weighting by simplex volume, or (iii) Monte-Carlo Markov chain methods like (a) Metropolis–Hastings [*Hastings*, 1970; *Kroese et al.*, 2011], (b) Gibbs [*Gelfand and Smith*, 1990], and (c) Hit-and-Run [*Smith*, 1984]. The latter Monte-Carlo Markov chain methods start at an arbitrary point within the polytope and find new points by, respectively, (a) generating a new candidate point and then accepting/rejecting it using an acceptance probability, (b) cycling through coordinate directions, or (c) picking a random direction and moving a random distance in that direction along the line between the current point and polytope boundaries. My testing on a 5-year old, 2.5 GHz Intel Dual core processor in Matlab 2012 found that, like *Chen and Schmeiser* [1993], most methods (except for rejection sampling) work sufficiently fast for small problems (of up to 7 decision variables). For larger problems, the Monte Carlo Markov chain methods were practical and Gibbs sampling (as coded by [*Benham*, 2011]) gave fast, uniform results.

However, for large problems it is undesirable to uniformly sample from the polytope because the samples tend to congregate towards the polytope centroid and away from the vertices and faces (e.g., Figure 4, top). This undesirable behaviour occurs because there is a very low probability to sample one variable value close to its extreme value (3 in the example) and all other values close to zero. Put another way, the volume of the polytope in the neighbourhood of the vertices is very small compared to the overall polytope volume; thus the likelihood to uniformly sample a solution from sub-regions near a vertex is also very low. Yet water managers are likely interested in *all* feasible near-optimal solutions and we need a sampling approach that communicates the composition, breadth, and depth of near-optimal solutions including solutions on or near edges and vertices that also represent the boundaries associated with the original optimization and objective function tolerance constraints.

A stratified random sampling approach (Figure 4, bottom) can generate a more complete range of near-optimal solutions. The method is:

a. Divide the p random solutions 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 maximal extents of each decision variable as in Step 2 of the interactive algorithm described in Section 3.
c. Select a decision variable and uniformly and randomly sample $p/(n+1)$ values from within the variable's maximal extents. For each sampled value in the group, reduce the problem dimensionality by one degree and Gibbs sample to obtain values for the remaining decision variables. Together, the $p/(n+1)$ random sampled solutions in the group are stratified along the chosen decision variable.
d. Repeat step c for each decision variable.
e. Similarly, repeat step c for the objective function. Uniformly and randomly 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^*, γ f^*]. For each sampled objective function value, add a constraint that fixes the objective function at the sampled value, $\hat{f}_j = c(x)$, and Gibbs sample decision variable values from within the subspace.

Together, the n+1 groups of random-sampled solutions are stratified along the extents of each decision variable and the objective function within the near optimal region. Obviously, the stratification (number of uniformly sampled values per group, $p/(n+1)$ in steps c and e, above) can be adjusted to focus attention on particular sub-regions. Or decrease the number of uniformly sampled values per group by a factor of two, three, four, or more and correspondingly increase by the same factor the number of Gibbs samples drawn for each uniformly sampled value. The sampling method also readily scales to multi-objective problems with i=1 to d objectives. Instead form $n+d$ (rather than $n+1$) groups, introduce a tolerable deviation for each objective (γ_i), and repeat step e for each objective by uniformly sampling objective function values within their ranges [f_i^*, $\gamma_i f_i^*$]. Because the underlying problem is linear, the stratified sampling codes use vectors and matrixes of linear coefficients for the objective function, objective function tolerance, and original constraints to efficiently pass model data during the various sampling steps. Overall, stratified sampling provides a more comprehensive basis to let water managers view and further explore solutions in the near-optimal region.

4.2 Integer problems

For integer variable problems, the near-optimal region has a finite number of variable-value combinations and corresponding sub-regions. For these problems, we can automate the interactive algorithm described in Section 3 by enumerating at each step all integer variable values within the identified minimum and maximum ranges. Then present the identified solutions to the user to view and further explore using the interactive algorithm. Note that by automating the interactive algorithm, we first identify bounding minimum and maximum values, only consider feasible near-optimal solutions within these bounding values, and reduce the overall solution effort to a fraction of the exponential time required for brute force enumeration (which must test all possible feasible and unfeasible solution combinations).

5 EXAMPLE APPLICATIONS

I now demonstrate the near-optimal approach and tools for prior-published linear water quality [*Alminagorta et al.*, 2013] and mixed-integer water shortage management [*Rosenberg and Lund*, 2009] optimization problems. I review the problem motivating each application, present the decision variables, and compare prior-published optimal and new near-optimal results.

5.1 Linear program to reduce phosphorus load to Echo Reservoir, UT

Echo Reservoir is located in the Weber Basin, Utah and suffers from eutrophication caused by excess phosphorus inputs. *Alminagorta et al.* [2013] developed a linear program to identify the cost-effective best management practices (BMPs) to reduce phosphorus loads to the level specified in a pending Total Maximum Daily Load (TMDL) program for the reservoir. The linear program considers ten BMPs such as installing grass filter strips, managing agricultural nutrients, stabilizing stream banks, adopting sprinkler irrigation, among others. The BMPs target three non-point phosphorus sources (manure, grazing, and diffuse runoff) in three sub-watersheds (Chalk Creek, Weber below, and Weber above Wanship) that drain to Echo reservoir. Point sources are excluded from the analyses because they have already adopted best available treatment technologies. Combined, there are 39 decision variables representing the 39 allowable combinations of BMPs, sources, and implementation locations. *Alminagorta et al.* [2013] describe the BMPs, their implementation costs, phosphorus removal rates, land area and stream bank lengths available to implement BMPs, load reduction targets, and the linear program used to identify the cost-minimizing mix of BMPs to reach the phosphorus reduction targets. They formulated and presented optimal, cost minimizing results for two scenarios: (i) separate reduction targets in each sub-watershed as listed in the TMDL, and (ii) a global reduction target that allows trading of load reductions among sub-watersheds. Here, I use the second scenario with global reduction targets for the near-optimal analysis.

A parallel coordinate plot (Figure 5) compares the prior-published optimal solution (thick black line) and 2,500 stratified, random-sampled, near-optimal solutions (light green lines) generated using a tolerable deviation of 110% from the optimal removal cost. I use a tolerable deviation of 110% to allow for a small, but still significant (i.e., 10%) increase in removal costs. On Figure 5, the left-most axis and left pink scale show removal costs for each solution while the 39 axes to the right and right-most green scale show the phosphorus removed by each BMP. The BMP axes and axis labels are grouped and color-coded by sub-watershed and phosphorus source. The optimal solution implements only 5 BMPs (manage agriculture nutrients from manure sources in each of the 3 sub-watersheds, protect grazing land, and stabilize stream banks in the Chalk Creek sub-watershed) at a cost of $985,000. In contrast, green lines correspond to near-optimal solutions, cross each decision axis at a much wider range of values, and show that a much broader mix of BMPs aimed at each source in each sub-watershed can also achieve the phosphorus reduction target at removal costs of, at most, 110% of the optimal and cost-minimizing value. Near-optimal solutions cross each axis at lower and upper values that correspond to the minimum and maximum extents for each associated decision variable. For example, 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. Or managers can shift stream bank stabilization to the Weber above or below Wanship sub-watersheds. Similarly for the BMP that protects grazing land.

The near-optimal results show there is broad flexibility to implement nearly every BMP at levels between zero and a value that equals or far exceeds the optimal implementation level and still maintain total removal costs within 110% of the optimal cost. The one exception is managing agricultural nutrients in the Weber above Wanship sub-watershed. This BMP must be implemented at levels that remove between 2,170 and 3,050 kg of phosphorus. The interactive solution technique also shows further restrictions to implement this BMP in other sub-watersheds. For example, setting the implementation level to zero (minimum extent) in the Chalk Creek sub-watershed means the BMP must remove at least 735 kg of phosphorus in the Weber below Wanship sub-watershed (Figure 6, dark green line). More generally, selecting and repositioning the manage agricultural nutrient axes adjacent to each other shows that this BMP must be implemented in at least two sub-watersheds. Despite this requirement, there is still flexibility to implement (or not implement) other BMPs like installing grass filter strips.

We can further use the interactive solution algorithm to explore solutions in sub-regions of management interest or that incorporate previously un-modelled issues. For example, should reservoir managers prefer some BMPs over others (for reasons beyond phosphorus removal costs), they could set implementation levels for the preferred or undesirable actions, then stratify random sample to identify additional near-optimal solutions with those settings. Managers would likely have such a preference if they suspect farmers will resist managing agricultural nutrients in their fields. In this case, managers could limit use of this BMP and set the associated implementation levels to the minimum extents of zero in Chalk Creek and the corresponding minimum required 735 kg in Weber below Wanship sub-watershed. Subsequent resampling identifies a near-optimal sub-region of limited agricultural nutrient management (Figure 7, purple lines; note that lines pass through the specified values on the checked axes). Within this sub-region, conservation tillage, protect grazing land, and stabilize stream banks BMPs are potentially implemented to compensate for reduced management of agricultural nutrients. Other

BMPs are not implemented. With limited agricultural nutrient management, removal costs rise to the largest allowable tolerable deviation – 110% of the minimum cost value.

Alternatively, managers may prefer to implement one or multiple BMPs for reasons besides cost. For example, managers may know farmers already have a favourable impression of sprinkler irrigation because a local irrigation company has started selling systems in the area and a government office can provide farmers technical assistance to install them. In this case, managers can set the implementation level for sprinkler irrigation to its maximum extent in one of the sub-watersheds like Chalk Creek and resample solutions (Figure 8, light blue lines; again blue lines pass through the specified value on the checked axis). With this setting, there is a reduced need to stabilize stream banks in the Chalk Creek sub-watershed compared to the optimal solution. Managers also have more flexibility to manage agricultural nutrients in each sub-watershed (compared to the limited agricultural nutrient management case) and can further or alternatively implement other BMPs like protect grazing land and/or stabilize stream banks in the Weber below and Weber above sub-watersheds. Configurations that use sprinkler irrigation similarly have removal costs at the upper limit of the near-optimal tolerance range (Figure 8, blue line at far left).

In the two above examples, the interactive solution algorithm allows managers to incorporate issues that were previously un-modelled and generate further near-optimal solutions that both (i) respond to the un-modelled issues, and (ii) achieve the phosphorus reduction target at removal costs of only 110% of the optimal, cost-minimizing value. More generally, the near-optimal results show there is much more flexibility in strategies to reduce phosphorus loads than the five BMPs offered by the optimal analysis. There is flexibility in the selection of BMPs, sources of phosphorus to target, and sub-watersheds where BMPs are implemented.

5.2 Mixed-integer program to meet water shortages in Amman, Jordan

This mixed-integer program identifies new supply and conservation projects to counteract chronic and expanding water shortages in Amman, Jordan through 2020 [*Rosenberg and Lund*, 2009]. 18 potential long-term projects include 10 supply options like tapping new or expanding local surface-, ground-, or brackish-water supplies, importing distant groundwater via the Disi conveyor, the Red-Dead Sea desalination project, treating and reusing wastewater, plus others. Eight long-term conservation projects either target water appliance retrofits to customers who will save the most water, offer rebates, reduce physical leaks, or re-price water, among others. The program also considers a variety of additional short-term and emergency coping measures that I do not consider in this near-optimal analysis. *Rosenberg and Lund* [2009] describe the long- and short- term projects, their costs, water volumes potentially gained or saved, interdependencies among projects, the shortage volumes and likelihoods of the water availability events, and the two-stage, stochastic, mixed-integer program formulation. The program selects projects to minimize expected long- and short-term costs to meet a probability distribution of water availability events.

A parallel coordinate plot (Figure 9) compares the prior-published optimal solution (thick black line) to near-optimal alternatives (light green lines). The optimal solution recommends implementing 5 new supply (blue text labels at the plot bottom) and 7 conservation actions (red text labels). The plot also shows as light green lines the approximately 2,200 near-optimal solutions that were generated by the near-optimal analysis and automated extension with an objective function tolerable deviation of 115%.

This tolerance deviation was selected to be larger than the value used in Echo Lake example. Near-optimal solutions have expected costs that vary between $40.7 and $46.8 million per year (pink lines, far left) with $46.8 million representing 115% of the optimal, cost minimizing $40.7 million value. I order the projects and axes in Figure 9 to highlight three important results from the near-optimal analysis.

First, the optimal and near-optimal solutions all cross the three left-most decision axes at the same positive values and maximum extents. Thus, all solutions expand the Zai pumping plant, adopt targeted conservation programs, and re-price water. Second, axes to the right of re-pricing water (new local groundwater through meter illegal connections) portray actions whose minimum and maximum extents differ. Near-optimal solutions implement these projects at varying levels; green lines cross the axes at regular intervals to indicate either the integer (e.g., new local groundwater or distant brackish water) or binary (e.g., Zara-Maeen project or reduce physical leakage) nature of project implementation. Third, the optimal and near-optimal solutions all pass through zero on the three far-right axes and indicate that the Red-Dead, Disi conveyor, and wastewater reuse mega-projects are never implemented within the near-optimal region.

Further inspection of the lines between adjacent axes reveals a fourth important management insight: lines that span the Zara-Maeen brackish desalination and reduce physical leaks axes all cross between the two axes. Thus, a near-optimal solution that reduces physical leaks does not build the Zara-Maeen project and *vice-versa*.

The 2,200 near-optimal solutions are much smaller than the approximately 506 million potential solutions that arise by brute force enumeration and show the automated algorithm for mixed-integer programs can be efficient. The near-optimal analysis can also justify several actions the Amman utility recently took that an optimal analysis might not fully explain. For example, the Amman utility very quickly expanded the Zai pumping and treatment plant capacity by 45 MCM/year to 90 MCM/year. The Jordanian government also pursued the Instituting Water Demand Management (IDARA) project from 2007 to 2012 to promote targeted water conservation campaigns. The optimal and near-optimal analyses both show the expansion and targeted conservation programs were extremely warranted. Further, in 2007, the Amman utility built and started operating the Zara-Maeen brackish water desalination project before fully fixing distribution system leaks [*Mohsen and Gammoh*, 2010]. This implementation counters the optimal solution which recommended the reverse but fits with many near-optimal alternatives that implement the Zara-Maeen project but do not fix leakage. At the same time, the near-optimal analysis does not explain Jordan's recent efforts to build the Disi conveyor or promote the Red-Dead project [*Kais*, 2013]. These projects are very controversial and the Jordanian government's efforts to advance them suggest the government is pursuing a different objective besides minimizing the expected cost for the Amman utility to cope with water shortages as considered in this analysis.

6 DISCUSSION

The near-optimal analysis, interactive solution algorithm, and automation techniques provide new tools to identify, visualize, and explore all water management strategies that perform within a tolerable deviation of the optimal objective function value. Use of the tools for the linear, water quality management problem to reduce the phosphorus load to Echo Reservoir, Utah shows managers can implement a much broader mix of best management practices that target each phosphorus source in every sub-watershed than is recommended by the optimal solution. This flexible implementation can be achieved with phosphorus reduction costs of up to 110% of the optimal, minimum cost. For the mixed-integer water shortage problem in Amman, Jordan, near-optimal results show that there are sub-sets of new supply and conservation actions that are always, never, and flexibly implemented and that maintain total expected costs at or below 115% of the optimal expected cost. Implementing certain flexible actions precludes implementing others, and vice versa. These near-optimal solutions can help explain actions the Jordanian Government has recently undertaken and that differ from the optimal recommendations.

The tools also allow managers to interactively include issues that were not previously considered in the optimization model objection function or constraints as well as generate near-optimal solutions that respond to these issues. For example, results from the Echo Reservoir phosphorus reduction example included cases where managers specified (i) reduced implementation of agricultural nutrient management because they suspected farmers would resist implementing this BMP, and (ii) full implementation of sprinkler irrigation for the opposite reason. Obviously, managers could specify a much larger number of interactive solutions and we neither intend nor need to show all the cases. Rather, managers should interactively explore cases of interest to them—particularly cases where preferences or constraints were not included in the original optimization model. The near-optimal method and tools provide an array of sliders, checkboxes, axis reordering, scaling, grouping, highlighting, re-sampling, and other plotting features (several of which are shown on Figures 5 to 9) to define cases and explore generated solutions. Set levels for decision variables, show solutions within the sub-region, compare results, and repeat. Together, these features facilitate and speed interactive exploration of the near-optimal region and sub-regions.

The Echo Reservoir, Utah and Amman, Jordan applications demonstrate near-optimal methods and tools for linear and mix-integer problems that have closed, convex, bounded solution regions. Important work remains to extend use of the tools to non-linear problems. Like linear and mixed-integer problems, this extension will also first solve for the optimal solution, second define a tolerable deviation from the optimal objective function value, and third use the interactive algorithm to identify the maximum extents and feasible ranges of each decision variable in turn. Modifications and extensions require automating how solutions are random sampled to communicate the composition, breadth, and depth of the non-linear near-optimal region. To do this, we must (1) modify the existing vector and matrix data formats used by the existing linear sampling codes to accommodate vectors of non-linear functions. And (2) replace the Gibbs sampler with a hit-and-run sampler that can sample over arbitrarily open regions [*Kroese et al.*, 2011]. Together, these changes and extensions will allow us to sample from and communicate the composition, breadth and depth of near-optimal solutions for non-linear problems.

The results presented in Section 4 are for single-objective problems and only considered tolerable deviations of 110% and 115% from the optimal objective function values. Further work must consider additional tolerable deviations as well as deviations from multiple objectives. There is likely a tradeoff between the (i) tolerable deviation value, and (ii) resulting size and shape of the associated near-optimal region. Further work must develop metrics to quantify the region shape, tie that shape to important management considerations like solution flexibility or substitutability, quantify the tradeoff, and provide guidance on appropriate tolerable deviation values to use. To include multiple objectives, we must (a) add additional objective axes to the parallel coordinate plot (one for each new objective), and (b) introduce separate tolerance levels for each objective (e.g., γ_i) as previously and briefly noted in Sections 3 and 4.1 that describe the interactive solution algorithm and stratified random sampling approach. Multi-objective near-optimal analysis holds promise both to identify all the solutions on and within specified tolerances of the *pareto*-optimal frontier and simultaneously link high dimension objective and decision spaces.

The above suggested extensions are important areas for further work which my research group is now tackling as part of a funded project to improve water management for environmental purposes in the Lower Bear River basin, Utah. We are working with an advisory group of water and environmental managers in the basin to identify and quantify multiple non-linear performance metrics for wetland and riparian areas. We have collected data over two summer field seasons [*Piiparinen et al.*, 2013] and are using the collected data to build and embed performance metrics in a water management model that determines reservoir operations and water allocations to users to maximize wetland and riparian performance. We look forward to share the completed model and near-optimal analysis tools with our advisory group members, observe how they use the tools during workshops, and see whether they identify promising near-optimal strategies. The project offers many avenues to extend the near-optimal methods I introduce and apply here to existing linear and mixed-integer problems.

7 CONCLUSIONS

An interactive solution algorithm, parallel coordinate plotting, and automation provide fast, new tools to identify, visualize, and explore all near-optimal solutions that perform within the tolerable deviation of the optimal objective function value. Stratified random sampling and other extensions automate the solution process for linear and mixed-integer problems. The approach simultaneously maps objective function performance to large decision spaces, allows managers to interactively include issues that were not previously considered in the optimization model objection function or constraints, and generate near-optimal solutions that respond to these issues.

Use for a prior linear, water quality optimization program to reduce phosphorus loading to Echo Reservoir, Utah identifies a much broader mix of near-optimal best management practices that target each phosphorus source in every sub-watershed than was recommended by the optimal solution of five BMPs. This flexible implementation can be achieved with a tolerable deviation of up to 110% from the optimal, minimum phosphorus removal cost. Further, managers can use the interactive solution algorithm to identify solutions that incorporate preferences or requirements not previously modelled like (i) suspecting farmers will resist implementing the BMP to manage agricultural nutrients, or (ii) ready adoption of sprinkler irrigation. In the mixed-integer application to manage water shortages in Amman, Jordan, a near-optimal analysis with a 115% tolerable deviation from the optimal expected cost shows: 1) the Zai project expansion, targeted conservation programs, and re-pricing water are always implemented, 2) three mega supply enhancement projects are never implemented, 3) there is flexibility to implement the remaining actions, and 4) building the Zara-Maeen brackish water desalination project foregoes the need to reduce physical leaks in the distribution system and *vice-versa*. Together, the applications identify numerous, promising, and varied near-optimal strategies beyond the prior, published singular optimal solutions.

Future work must include non-linear problems, additional tolerable deviations from the optimal objective function value, a larger number of objectives, and tolerable deviations from combinations of objectives. These developments are now in progress as part of a project to manage water for wetland and riparian objectives in the lower Bear River basin, Utah and will further enhance the existing near-optimal tools and capabilities introduced here. It will also be important to test whether water managers can use the near-optimal methods and tools, identify promising near-optimal solutions, and use identified near-optimal solutions to inform their water system management.

8 ACKNOWLEDGMENTS

This work was funded by NSF grant #1149297. Ayman AlAfifi commented on a prior draft.

9 REFERENCES

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.

Benham, T. (2011), Uniform distribution over a convex polytope, edited, Matlab Central.

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.

Brown, G. G., R. F. Dell, and R. K. Wood (1997), Optimization and persistence, *Interfaces*, *27*(5), 15-37.

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.

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.

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.

Dantzig, G. B. (1963), *Linear Programming and Extensions*, Princeton University Press, Santa Monica, CA.

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.

Hastings, W. K. (1970), Monte Carlo Sampling Methods Using Markov Chains and Their Applications, *Biometrika*, *57*(1), 97-109.

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

Kais, R. (2013), Jordan, Israel in advanced talks on water deal, in *Ynet News*, edited, Yedioth Media Group, Tel Aviv, Israel.

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.

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.

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.

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.

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.

Mendoza, G. A., B. B. Bare, and G. E. Campbell (1987), Multiobjective Programming For Generating Alternatives - A Multiple-Use Planning Example, *Forest Science*, *33*(2), 458-468.

Mohsen, M. S., and S. Gammoh (2010), Performance evaluation of reverse osmosis desalination plant: A case study of Wadi Ma'in, Zara and Mujib Plant, *Desalination and Water Treatment*, *14*(1-3), 265-272.

Nemhauser, G. L. (1994), The Age Of Optimization - Solving Large-Scale Real-World Problems, *Operations Research*, *42*(1), 5-13.

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.

Piiparinen, L., S. Stander, R. Babb, A. AlAfifi, and D. E. Rosenberg (2013), Measuring the eco-hydrological performance of the Lower Bear River Basin through experiential learning – the Bear River Fellows Program, in *USU Spring Runoff Conference*, edited, Logan, UT.

Rosenberg, D. E., and J. R. Lund (2009), Modeling integrated water utility decisions with recourse and uncertainties: Amman, Jordan, *Water Resources Management*, *23*(1), 85-115.

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.

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.

Smith, R. L. (1984), Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions, *Operations Research*, *32*(6), 1296-1308.

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.

Tsoukiàs, A. (2008), From decision theory to decision aiding methodology, *European Journal of Operational Research*, *187*(1), 138-161.

Wegman, E. J. (1990), Hyperdimensional Data Analysis Using Parallel Coordinates, *Journal of the American Statistical Association*, *85*(411), 664-675.

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

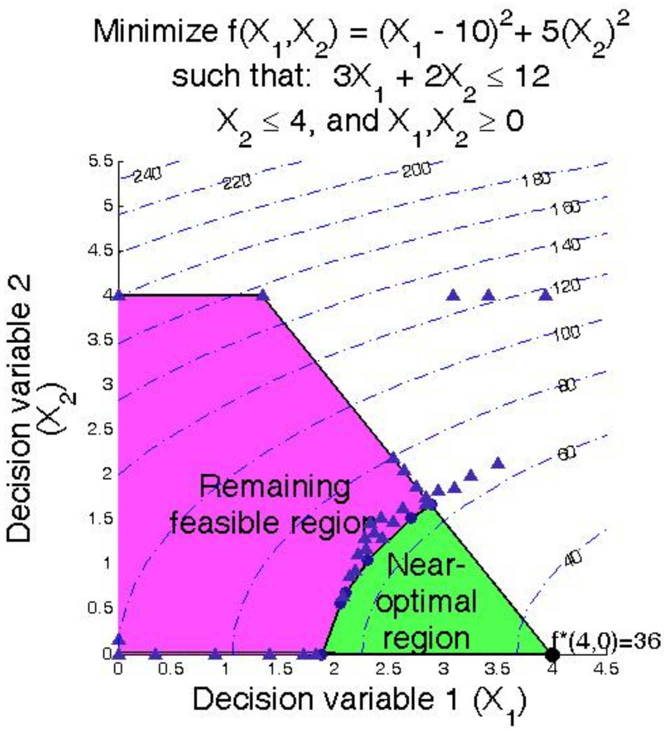


Figure 1. Near optimal region for a two-decision problem with a non-linear objective and 4 linear constraints. A tolerable deviation (γ=1.80) corresponds to an objective function value and contour of 64.8 (1.8*36). This contour also partitions the near-optimal and remaining feasible regions. Six purple circles and many triangles show, respectively, feasible and infeasible extreme points of the near-optimal region.

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

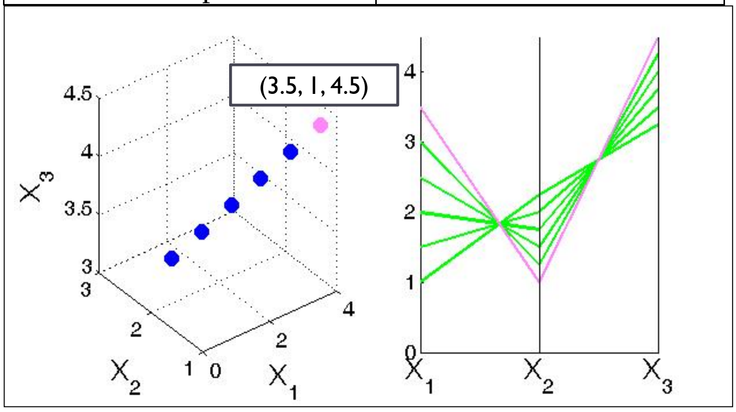


Figure 2. Coordinate system mapping can render complex geometric objects. Cartesian points (left) are connected line segments on a Parallel coordinate plot (right); conversely, a Cartesian line is the point where line segments intersect on a parallel coordinate plot.

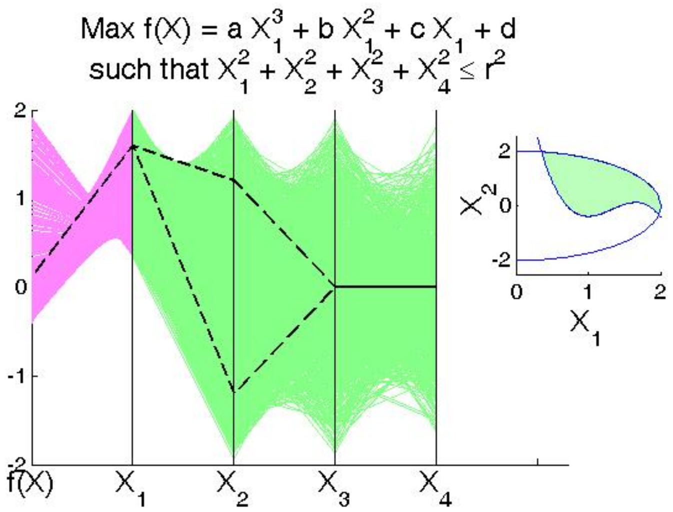


Figure 3. Near-optimal hyperspace in parallel coordinates for a 4-decision, nonlinear optimization problem. Solid lines represent points in the interior region and fall within outer envelopes between axes. Pin a value on the X_1 axes to narrow ranges on axes to the right (dashed black lines). Inset graph projects the near optimal hyperspace into the X_1-X_2 plane. Pink lines simultaneously map the objective function (f[X]; left axis) to decision space and show f(X) is inversely correlated to X_1.

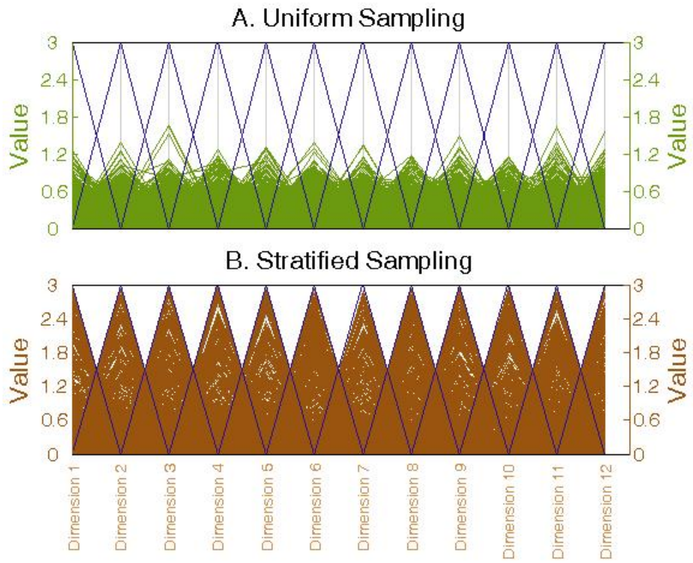
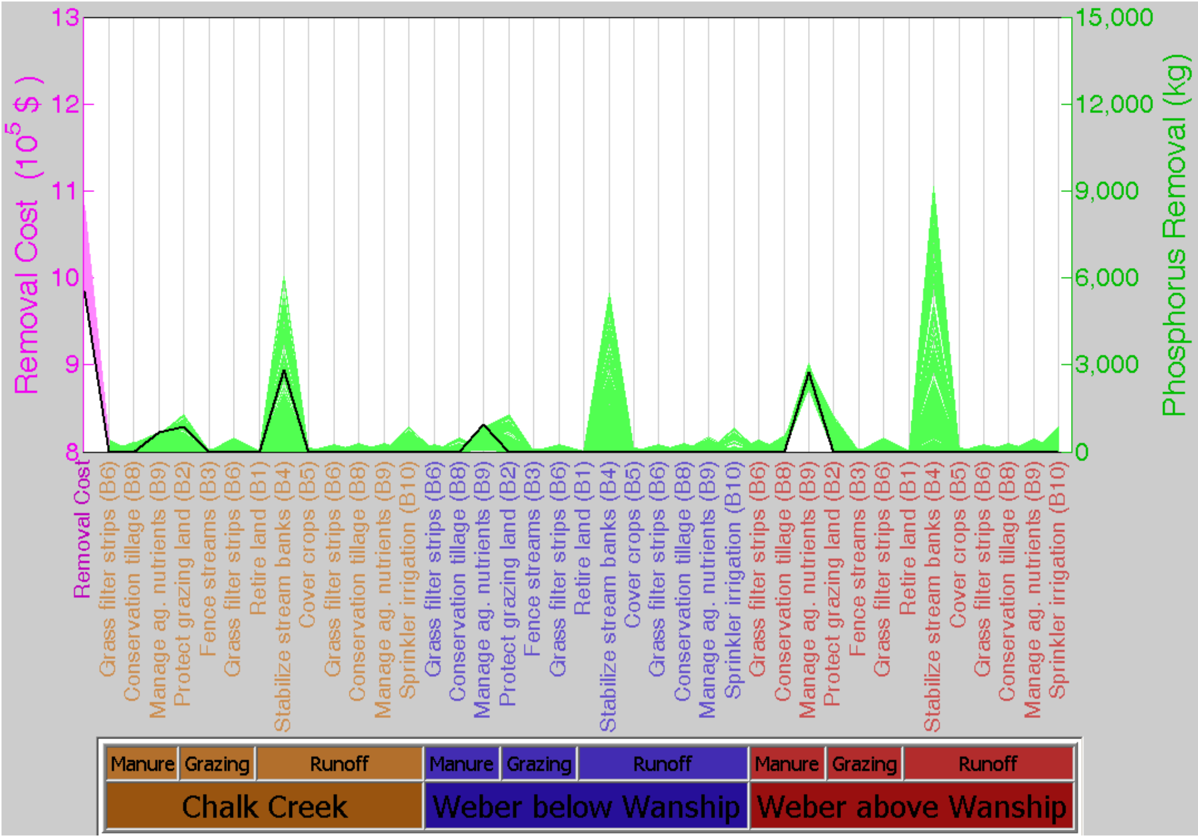


Figure 4. Parallel coordinate plots compare 1,500 uniform (top, green) and stratified (bottom, brown) random samples drawn from a 12-dimensional simplex with edge length 3 defined by the inequalities ($X_i \geq 0, \forall i; \sum_i X_i \leq 3$). Vertices are plotted in blue. Uniform samples cluster away from the vertices while stratified samples cover the entire polytope.

646 

647 **Figure 5. Parallel coordinate plot compares optimal (thick black) and near-optimal**
648 **(pink and green) solutions to reduce phosphorus inputs to Echo Reservoir, Utah.**
649 Removal costs are plotted in pink on the left scale and implementation levels for 39
650 BMPs are plotted in green on the right scale. Near-optimal solutions offer a much wider
651 mix of BMPs targeted at different sources in different sub-watersheds at up to 110% of
652 the removal cost of the optimal solution.

653

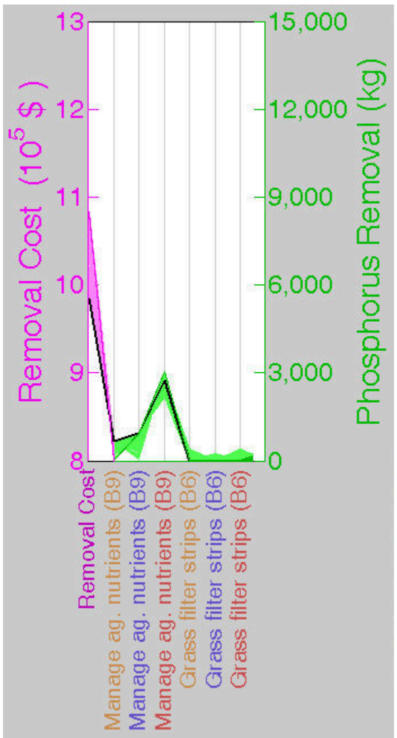


Figure 6. Selecting and repositioning the BMP axes from Figure 5 identifies trade-offs among locations where BMPs are implemented. Orange, Purple, and Red text labels correspond to the same Chalk Creek, Weber below, and Weber above Wanship sub-watershed locations as in Figure 5. Not managing agricultural nutrients in Chalk Creek requires doing so in the Weber below Wanship sub-watershed (dark green line). In contrast, grass filter strips need not be implemented in any sub-watersheds.

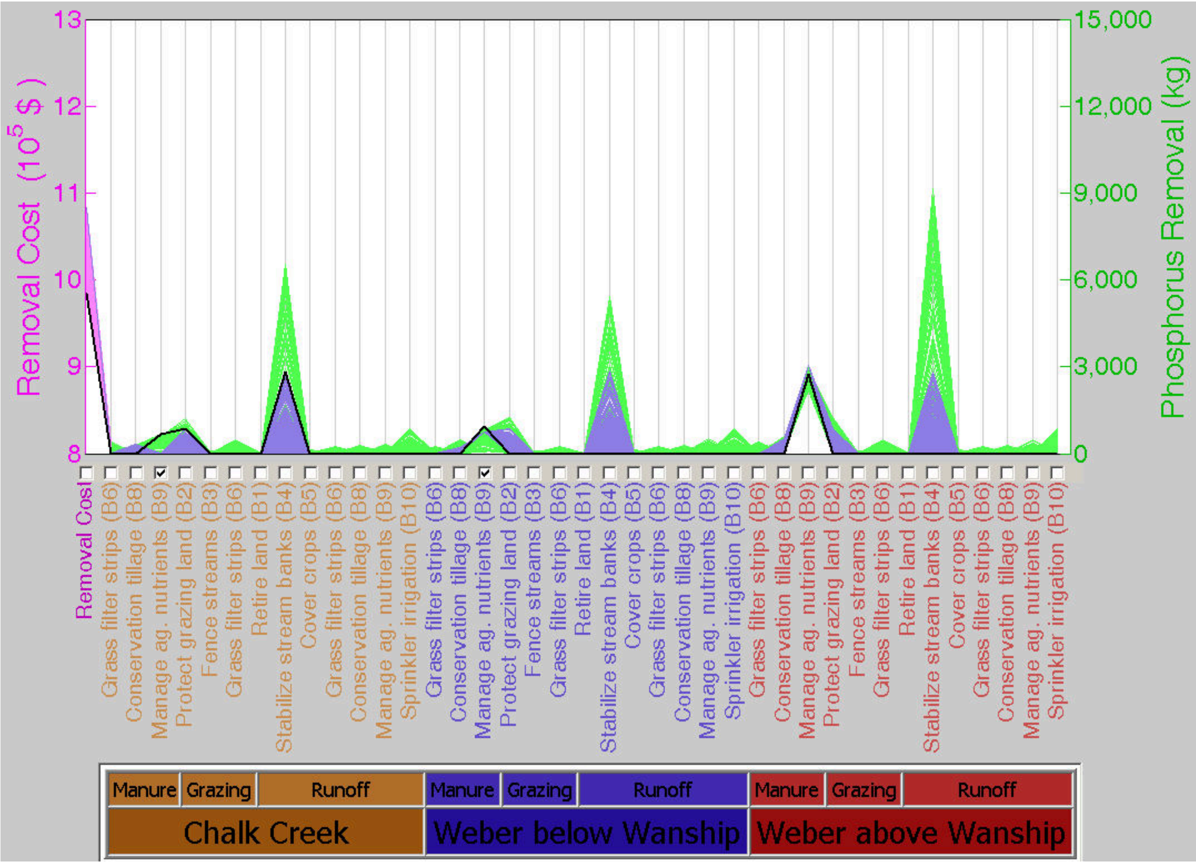


Figure 7. A snapshot of the interactive solution algorithm that highlights near-optimal solutions with limited agricultural nutrient management (purple lines). First a manager sets decision variable values associated with the checked axes to zero and 735 kg phosphorus removed. Then, reduce the problem dimension by two and repeat the stratified sampling to generate the sub-space (purple lines). Finally, observe that limited agricultural management still allows implementing a flexible mix of conservation tillage, protect grazing land, and stabilize stream bank BMPs.

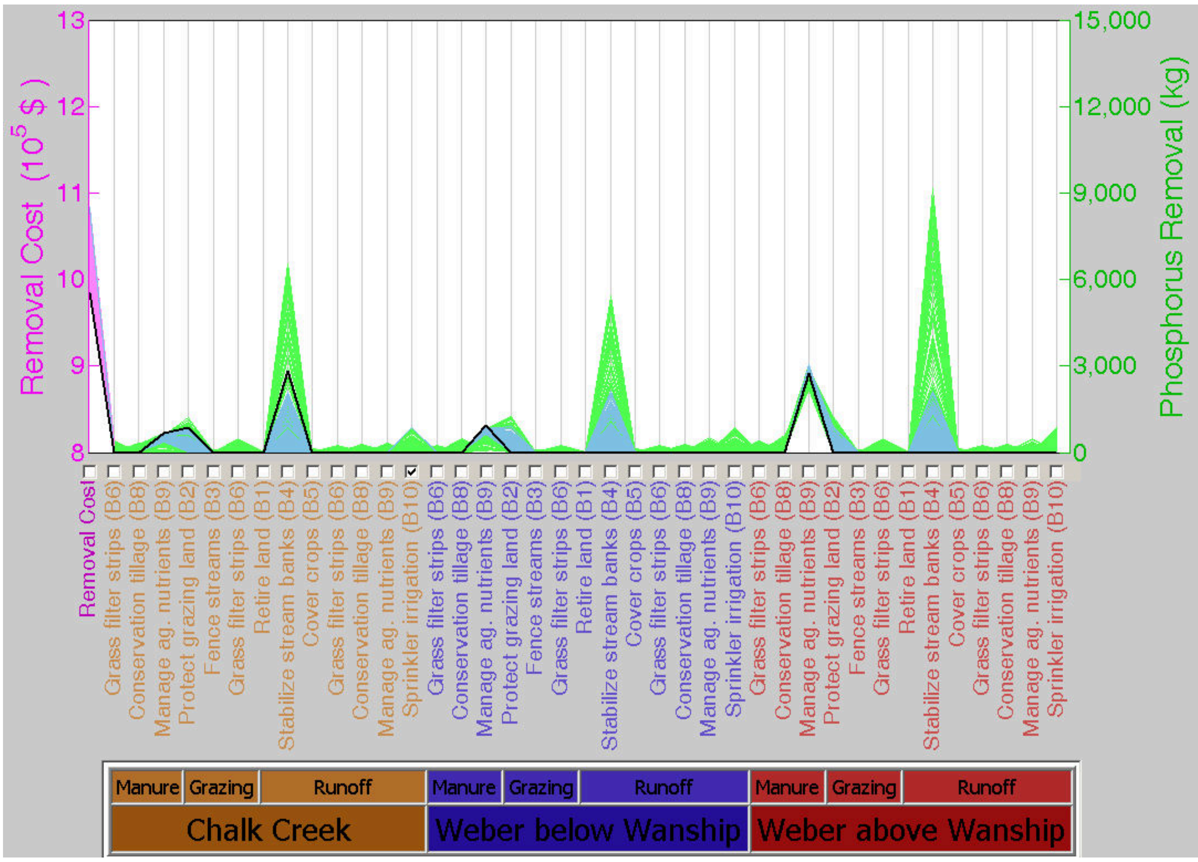


Figure 8. A snapshot of the interactive solution algorithm that highlights near-optimal solutions that implement sprinkler irrigation (checked axes) at its maximum extent (blue lines).

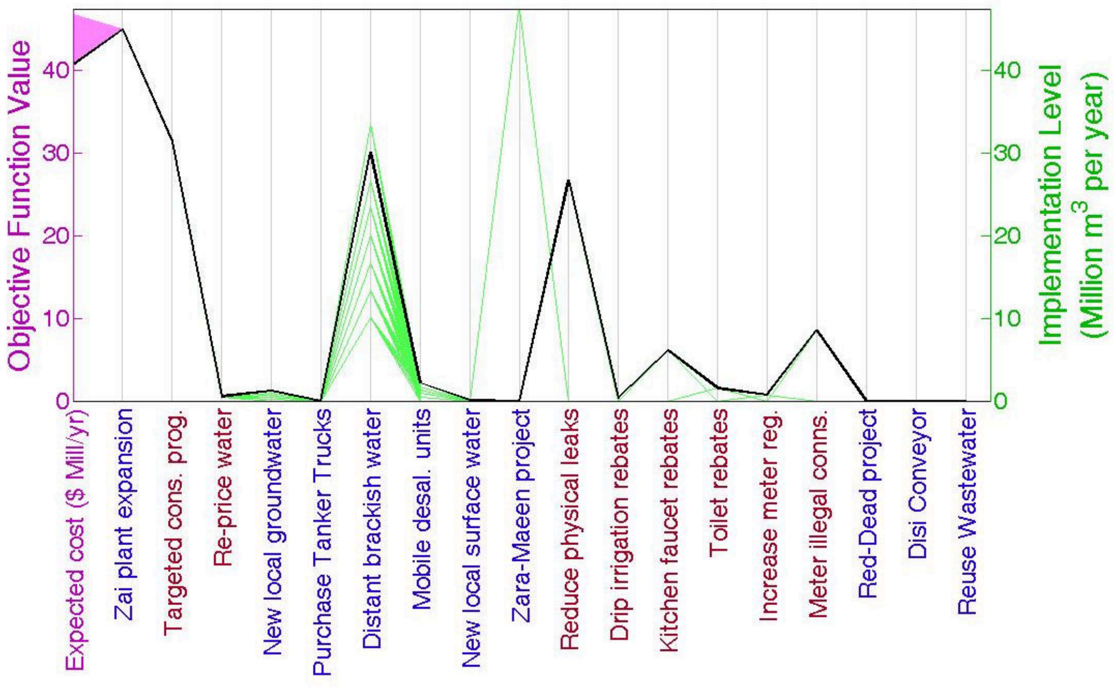


Figure 9. Parallel coordinate plot for the Amman, Jordan water management problem compares the optimal (thick black line) and near-optimal solutions whose expected cost is within 115% of the optimal cost (pink and green lines). Expected costs for solutions are plotted in pink on the left scale while implementation levels for 18 projects are shown in light green on the right scale (blue and red axis labels indicate, respectively, new supply and conservation projects). Optimal and near-optimal solutions all recommend to expand the Zai plant, target conservation programs, and reprice water while *not* implement 3 mega-supply projects. Near-optimal solutions offer great flexibility to implement 12 other new supply and conservation actions.