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

David E. Rosenberg

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

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

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. The new tools generate more numerous alternatives faster, more completely show the near-optimal region, and help elicit a larger set of un-modelled issues than Modelling to Generate Alternatives methods.

Key Points

1. New generation and visualization tools show the full near-optimal region

2. Interactive exploration can elicit un-modelled issues

3. The tools identify flexible management to maintain near-optimal performance

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

David E. Rosenberg

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

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

74 **Keywords**: optimize; near-optimal; water management; linear program; alternative

75 generation; visualization; interaction; Echo Reservoir, Utah

76 **1 INTRODUCTION**

77 State-of-the-art systems analysis techniques emphasize efficiently finding optimal and

78 *pareto*-optimal solutions [*Brown et al.*, 1997] to water supply and environmental decision

79 problems [*Nemhauser*, 1994; *Sahinidis*, 2004; *Zhang and Li*, 2007]. Yet an optimal

80 solution is optimal only with respect to the modelled issues and managers may prefer

81 near-optimal alternatives that address un-modelled objectives, preferences among

82 objectives, limits, uncertainties, or other shortcomings in the original model [*Brill et al.*,

83 1982; *Chang et al.*, 1982; *Harrington and Gidley*, 1985; *Rogers and Fiering*, 1986].

84 Providing managers with useful near-optimal alternatives requires solving two

85 conflicting challenges: (i) generate alternatives that address the un-modelled issues, and

86 (ii) tractably communicate them. Both challenges intensify as problem size grows and the

87 modeller has less ability to qualitatively and quantitatively characterize the un-modelled

88 issues.

89

90 The Modelling to Generate Alternatives (MGA) Hop-Skip-Jump (HSJ) algorithm [*Brill*

91 *et al.*, 1982; *Chang et al.*, 1982] was an early method to generate near-optimal

92 alternatives. The MGA-HSJ first solved for the optimal solution, then generated a new

93 alternative that was maximally different in decision space from the optimal solution yet

94 performed within a (near-optimal) tolerable deviation of the optimal objective function

95 value. The new alternative minimized the sum of the non-zero decision variables in the

96 optimal solution. Next, MGA-HSJ generated a second alternative that was maximally

97 different from the optimal solution and first alternative and continued until either every

98 variable became non-zero in at least one alternative or no new non-zero variables were

99   found. Generated alternatives were always less than the number of decision variables in

100 the problem, located at select vertices of near-optimal region, and addressed un-modelled

101 issues typically related to decision variable substitutability.

102

103 Subsequent near-optimal work enumerated all near-optimal region vertices for a small,

104 linear farm management problem [*Burton et al.*, 1987], but encountered difficulty with

105 problem scaling [*Harrington and Gidley*, 1985; *Matheiss and Rubin*, 1980]. Effort

106 increased exponentially with problem size to enumerate all extreme points where model

107 constraints intersect, then retain only feasible points as vertices to the region.

108

109 A third set of near-optimal alternative generation methods substituted different and

110 sometimes randomly parameterized objectives for MGA-HSJ's maximally-different

111 criterion [e.g., *Chang et al.*, 1982; *Harrington and Gidley*, 1985; *Kennedy and Quinn*,

112 1998]. These methods also generated a few alternatives—one for each objective tested.

113

114 More recent MGA extensions use distance metrics and evolutionary algorithms to

115 generate near-optimal alternatives either in series (similar to MGA-HSJ) or

116 simultaneously (all at once) for non-linear problems [*Loughlin et al.*, 2001; *Zechman and*

117 *Ranjithan*, 2004; *Zechman and Ranjithan*, 2007]. The distance metrics maximize either

118 the (i) aggregate difference between alternative decision components (L_1 norm), or (ii)

119 distance to the nearest neighbour (-L_∞ norm). In testing use of the -L_∞ norm distance

120 metric, *Zechman and Ranjithan* [2007] found MGA-Serial generates an alternative that is

121 furthest from the optimal solution and prior alternatives whereas MGA-Simultaneous

122 spreads alternatives through the near-optimal region. Both methods face scaling issues to

123 increase the number of generated alternatives that include making more numerous

124 distance calculations and solving larger optimization problems. As a result, Evolutionary

125 Algorithm applications have generated a small number of near-optimal alternatives (3, 3,

126 and 9 in the examples) that offer limited opportunities to elicit un-modelled issues.

127

128 A contrasting set of uniform Monte-Carlo Markov chain samplers use conditional

129 sampling to efficiently generate points inside bounded high-dimensional regions [*Chen*

130 *and Schmeiser*, 1993; *Kroese et al.*, 2011; *Liu*, 2001]. Samplers include (a) Metropolis–

131 Hastings [*Hastings*, 1970; *Kroese et al.*, 2011], (b) Gibbs [*Gelfand and Smith*, 1990], and

132 (c) Hit-and-Run [*Smith*, 1984]; start at an arbitrary point within the region; then sample

133 new alternatives by, respectively, (a) generating a new candidate alternative and

134 accepting/rejecting it using a threshold probability, (b) cycling through variables to

135 progressively sample each decision component, or (c) randomly picking a direction and

136 moving a random distance in that direction from the current point towards the region

137 boundary. Undesirably, uniform sampling clusters samples towards the region centroid

138 and away from the vertices and surfaces because local volumes near extremities are very

139 small compared to the region volume. This small ratio means a low probability to sample

140 alternatives that span the full region defined by the original and near-optimal tolerance

141 constraints. These samplers remain to be applied to near-optimal problems.

142

143 The above near-optimal methods have generated a few alternatives that address some

144 existent un-modelled issues and shortcomings of optimal solutions. To address more

145 numerous un-modelled issues in reasonable time, a tool must: (i) generate a more

146 comprehensive set of alternatives from throughout the near-optimal region, yet (ii) not

147 overwhelm users. These challenges similarly confront narrower searches of control

148 policies for the multiple optima that have the same objective function value but different

149 locations in decision space [*Aufiero et al.*, 2001; *Nardini and Montoya*, 1995; *Orlovski et*

150 *al.*, 1983]. The challenges also confront *pareto* search for preferred compromise solutions

151 to multi-objective problems where all objectives are specified *a priori* [*Castelletti et al.*,

152 2010; *Cohon and Marks*, 1975; *Marler and Arora*, 2004]. For example, alternative

153 generation [e.g., *Deb*, 2008] techniques produce and present a large number of *pareto*

154 alternatives to managers. In contrast, interactive and progressive articulation of

155 preference methods [*Korhonen and Wallenius*, 1988; *Miettinen et al.*, 2008; *Steuer*, 1986;

156 *Wierzbicki*, 1979] involve the manager to identify a few alternatives of most interest.

157 More recent methods blend alternative generation, visualization, and interaction to

158 leverage computational, cognitive, and learning strategies [*Castelletti et al.*, 2010; *Lotov*

159 *et al.*, 2004]. Blending first generates and visualizes one or numerous alternatives, then

160 the user guides further generation and visualization until reaching an acceptable end

161 point. These blended methods for *pareto* search hold promise to help identify, visualize,

162 and explore the larger near-optimal region for alternatives that address un-modelled

163 issues.

164

165 This paper presents new blended tools to more comprehensively generate, visualize, and

166 interactively explore the near-optimal region of an optimization problem. Stratified

167 Monte-Carlo Markov Chain sampling quickly identifies a large number of near-optimal

168 alternatives that span the region through both the decision and objective spaces. A new

169 Parallel Coordinate plot [*Inselberg*, 2009; *Wegman*, 1990] places axes for all objectives

170 and decision variables side-by-side on a single page to show generated alternatives across

171 the linked decision and objective spaces. Interactive controls allow the user to explore the

172 region and seek alternatives that address un-modelled issues. Together, the tools (i) show

173 the structure and full extent of the near-optimal region, (ii) identify numerous and

174 flexible ways to manage the system to maintain near-optimal performance, and (iii)

175 streamline the process to elicit un-modelled issues and update model formulations.

176 Section 2 reviews the near-optimal formulation and prior MGA solution methods while

177   sections 3-5 present the new alternative generation, visualization, and interactive tools.

178   The remaining sections show results for a water quality management problem at Echo

179   Reservoir, Utah, compare to prior MGA methods, and present conclusions.

180   **2   NEAR-OPTIMAL FORMULATION AND PRIOR SOLUTION METHODS**

181   For the general problem to minimize objective f with objective function c, a vector of n

182   decision variables $\mathbf{x}$, a comprising m constraint functions, and b a vector of m bounds for

183   the m constraints,

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

184   subject to

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

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

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

187   programming technique (e.g., linear, mixed-integer, nonlinear, etc.),

188   2. Allow a tolerable deviation, γ ($\gamma \geq 1$ [unitless]), from the optimal objective function

189   value (f^*)

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

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

191   tolerance (Eq. 2) constraints.

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

193

194   In the near-optimal formulation, the tolerable deviation constraint (Eq. 2) partitions the

195   feasible region (Eq. 1b) into two sub-regions: the near-optimal and remaining feasible

196   region. Figure 1 shows this partitioning for an example allocation problem with two

197   decisions such as water deliveries to Users 1 and 2   (X_1 and X_2), quadratic cost

198   minimization objective, linear constraints that specify the available water, delivery

199   efficiencies, and capacity, and near-optimal tolerance of γ = 1.8. In the example, the

200   optimal solution allocates water only to User 1 (Figure 1, blue circle, bottom right).

201

202   When the optimal solution is not acceptable because of un-modelled issues, MGA-HSJ

203   can generate near-optimal alternatives that perform close to the optimal solution on the

204   stated objective but are maximally-different in their decision variable values [*Brill et al.*,

205   1982; *Chang et al.*, 1982]. MGA-HSJ solves a new problem that minimizes the sum (*y*)

206   of the non-zero decision variables in the optimal solution (*x_k, $k \in K$)*:

$$\underset{\mathbf{x}}{\text{Min}} \quad y = \sum_{k \in K} x_k \qquad \qquad \text{[Eq. 3]}$$
$$s.t. [1b] \text{ and } [2]$$

207   Problem [3] is the same size as Eqs. [1] and [2]. After solving [3] for a new alternative,

208   MGA adds the new non-zero decision variables in the new alternative to the set *K*,

209   resolves [3], and repeats until either no new decision variables are added to *K* or *K*

210   comprises all decision variables [*Brill et al.*, 1982; *Chang et al.*, 1982]. MGA-HSJ can

211   generate up to *n-1* alternatives depending on the number of non-zero variables in the

212   optimal solution and number of variables added to *K* at each iteration.

213

214   In the Figure 1 example, MGA-HSJ generates a single near-optimal alternative that still

215   allocates all water to User 1 but at higher cost (Figure 1, black square). This alternative

216   represents movement along the X_1 axis towards the origin to minimize the single non-

217   zero variable X_1 in the optimal solution. MGA-HSJ terminates after the first iteration

218   because no new decision variable enters *K*.

219

220   In contrast, MGA-Serial [*Zechman and Ranjithan*, 2007] iteratively solves for a new

221   maximally-different alternative ($\mathbf{x}^p$) by maximizing the distance between the new

222   alternative, prior generated alternatives ($\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^{p-1}$), and the optimal solution ($\mathbf{x}^0$).

$$\underset{\mathbf{x}^p}{\text{Max}} \quad D = \min_{y \leq p-1} \left\{ \sum_{k=1}^{n} \left| x_k^p - x_k^y \right| \right\}$$
$$s.t. [1b] \text{ and } [2]$$

[Eq. 4]

223   Here, distance (D) is the smallest summed difference between decision components ($-L_\infty$

224   norm) and the absolute value operator makes the problem non-linear. Generating p

225   alternatives means solving p problems in series, each of size n decisions and m

226   constraints, but that grow in the number of distance calculations between the current and

227   prior generated alternatives plus the optimal solution. The related MGA-simultaneous

228   approach uses the same distance metric (D) but instead solves a single problem to

229   generate p alternatives $\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^p$ all at once to more uniformly spread alternatives

230   [*Zechman and Ranjithan*, 2007]. Generating p simultaneous alternatives requires $n \cdot p$

231   decision variables, $m \cdot p$ constraints, and evaluating $\frac{1}{2}(p+1)(p) \sim p^2$ distances between the

232   alternatives and optimal solution.

233

234   Initially, these scaling issues are imperceptible to generate a small number of alternatives

235   for small problems like in Figure 1. For example, solving for a single near-optimal

236   alternative with MGA-Serial and Simultaneous using Matlab's 2013b global search and

237   sqp non-linear solver generated the same alternative in under 1 s (Figure 1, black

238   triangle, only MGA-Serial result shown). And while the generation density of 0.5

239   alternatives per decision variable may appear low, it is actually 3 to 50 times *larger* than

240   prior applications of the methods. Run times for both MGA methods rapidly increase

241   with the number of alternatives generated and mean the methods can only generate a

242   small number of alternatives for moderate sized problems. Further, the distance metrics

243   require solution with global search or evolutionary algorithms that use randomized start

244   points and do not necessarily generate the same alternatives on each run [*Zechman and*

245   *Ranjithan*, 2007]. Thus, the MGA method's partial coverage of the near-optimal region

246   shown in Figure 1 is emblematic of results for higher-dimensional problems and means

247   there is benefit to more efficiently generate a larger number of alternatives that

248   communicate the *entire* near-optimal region (e.g., Figure 1, green area).

249

250   More complete coverage would identify areas near the upper vertex or optimal solution

251   that offer alternatives that could address un-modelled allocation equity, flexibility, or

252   other issues. Further, when the generation strategy is fast, a manager can select an

253   alternative, the systems analyst can elicit the reason for the manager's selection, use the

254   new information to update the model formulation, generate additional alternatives, and

255   guide further exploration in the near-optimal region. The next three sections present new

256   alternative generation, visualization, and interaction tools to identify, communicate, and

257   explore near-optimal regions for larger problems that cannot be represented on a

258   Cartesian plot.

259   **3   ALTERNATIVE GENERATION**

260   The stratified sampling approach extends uniform Monte-Carlo Markov chain sampling

261   of high-dimensional, closed regions [*Gelfand and Smith*, 1990; *Hastings*, 1970; *Kroese et*

262   *al.*, 2011; *Liu*, 2001; *Rubinstein*, 1986; *Smith*, 1984] to generate alternatives that more

263   comprehensively span the near-optimal region.

264   a. Divide the p alternatives to be sampled into $n+1$ groups (associate n groups with

265   the n decision variables and last group with the objective function).

266   b. Identify the minimum and maximal extent of each decision variable in the near-

267   optimal region. To identify these extents, solve $2n$ optimization problems that

268   separately and independently Minimize x_1, Maximize x_1, Minimize x_2, Maximize

269   x_2, … , Minimize x_n, and Maximize x_n each subject to the same original and near-

270   optimal tolerance constraints [Eqs. 1b and 2].

271   c. Select a decision variable and uniformly sample $p/(n+1)$ values from within the

272   variable's maximal extents identified in step b. For each sampled value, reduce the

273   problem dimensionality by one and use a Monte-Carlo Markov chain method to

274   sample values for the remaining decision variables. Together, the $p/(n+1)$ sampled

275   alternatives in the group are stratified along the chosen decision variable.

276   d. Repeat step c for each decision variable.

277   e. Finally, uniformly sample a set of $p/(n+1)$ objective function values

278       $(\hat{f}_j; j = 1 \text{ to } p/(n+1))$ between the optimal objective function and tolerable

279   deviation values [f^*, γ f^*]. For each sampled objective function value, add a

280   constraint to fix the objective function at the sampled value, $\hat{f}_j = c(x)$. Then

281   Monte-Carlo Markov Chain sample decision variable values (as in step c) with the

282   specified objective function value.

283 Together, the $n+1$ groups of sampled alternatives are stratified along the extents of each

284 decision variable and the objective function within the near optimal region. Adjust the

285 total samples (p) or samples per group ($p/[n+1]$) to achieve the desired sampling density

286 or focus attention on particular sub-regions. The generation strategy solves $2(n+1)$

287 optimization problems that are n decisions and m constraints in size and requires drawing

288 p Monte-Carlo Markov chain samples. The next subsection present extensions for

289 different classes of optimization problems.

290 **3.1   Linear programs**

291   For linear problems with continuous decision variables, follow steps (a) through (e)

292   above. Use vectors and matrixes of linear coefficients to efficiently pass model data.

293   Testing of Monte-Carlo Markov Chain methods [*Gelfand and Smith*, 1990; *Hastings*,

294   1970; *Kroese et al.*, 2011; *Smith*, 1984] found each sampler worked fast and confirmed

295   prior comparisons by *Chen and Schmeiser* [1993]. Here, the tools use Gibbs sampling

296   (re-coded from [*Benham*, 2011]).

297   3.2   Integer problems

298   For integer decision variable problems, restrict sampling in steps (c) and (d) to integer

299   values. For small- and moderate-sized problems, instead generate all near-optimal integer

300   alternatives in the region by modifying the Gibbs sampler [*Gelfand and Smith*, 1990] to

301   instead during the cycle through decision variables enumerate all integer values within

302   the bounding extents. Then for each integer value, continue to the next variable and again

303   enumerate within the extents. The extents ensure generating only feasible near-optimal

304   alternatives and further reduce the overall effort to a fraction of the exponential time

305   required for brute force enumeration (which tests all variable value combinations).

306   Rosenberg [2012] uses the technique to identify 18 water supply and conservation actions

307   to meet shortages in Amman, Jordan. One interactive tool in Section 5 adapts the method

308   to generate a single alternative.

309   3.3   Multi-objective problems

310   For a multi-objective problem with $i=1$ to d objectives defined *a priori*, form $n+d$ (rather

311   than $n+1$) groups in step a. Then in step e, introduce a tolerable deviation for each

312   objective (γ_i) and uniformly and unconditionally sample $j=1$ to $p/(n+d)$ values $\hat{f}_{1,j}$ for the

313   first objective (c_1) within its near-optimal range [f_1*, γ_1 f_1*]. Using the first sampled

314   objective function value ($\hat{f}_{1,1}$), identify extents for the second objective (conditioned on

315   the original, near-optimal tolerance and $\hat{f}_{1,1} = c_1(x)$ constraints). Next, uniformly sample a

316   value for the second objective ($\hat{g}_{2,1}$) within its conditioned extents, identify the extents

317   for the third objective (conditioned on the original, near-optimal tolerance, $\hat{f}_{1,1} = c_1(x)$,

318   and $\hat{g}_{2,1} = c_2(x)$ constraints), and continue through the d objectives. Up to this point, the

319   sampling strategy resembles the Lexicographic method [*Marler and Arora*, 2004] and

320   emulates Gibbs sampling but cycles through the objectives rather than decision variables.

321   From here, add the d sampled objective function values as d equity constraints to the

322   original and near-optimal constraints and Monte-Carlo Markov Chain sample decision

323   variable values with the specified objective function values. Together, the sampled

324   objective function and decision variable values comprise an alternative associated with

325   the first unconditionally sampled objective. Repeat the above steps to obtain j=2 to

326    $p/(n+d)$ additional alternatives associated with the remaining unconditionally sampled

327   values $\hat{f}_{1,j}$ for the first objective.

328

329   Finally, repeat the above steps, but switch the order the objective functions are sampled

330   so that each objective gives rise to $p/(n+d)$ alternatives where that objective is sampled

331   first and unconditionally. The resulting p alternatives span the objective and decision

332   spaces, the *pareto* front, as well as the near-optimal region that lies under the front.

333   Dissatisfaction with alternatives on the front in objective space motivate search in the

334   larger region near the front.

335   **4   PARALLEL COORDINATE VISUALIZATION**

336   It is difficult to visualize high-dimensional near-optimal regions on Cartesian plots

337   because they *orthogonally* arrange axes and can only show two or three dimensions (on a

338   flat page) or possibly six or seven dimensions when using marker color, size, orientation,

339   scroll bars, or other attributes [*Kasprzyk et al.*, 2009; *Kollat and Reed*, 2007; *Lotov et al.*,

340   2004]. In contrast, Parallel Coordinate plots [*Inselberg*, 2009; *Wegman*, 1990] arrange

341   numerous axes in *parallel* and use polygonal lines that span axes to represent Cartesian

342   points. Recent uses in water management studies include to visualize discrete points on

343   *pareto* frontiers [e.g., *Kasprzyk et al.*, 2009; *Ortiz et al.*, 2011; *Shenfield et al.*, 2007;

344   *Stummer and Kiesling*, 2009]. Beyond discrete points, parallel coordinate systems also

345   uniquely represent line, curve, and other Cartesian concepts [*Inselberg*, 2009] and these

346   correspondences can be used to visualize high-dimensional near-optimal regions.

347

348   First lay out parallel axes for the objective function and each decision variable (Figure 2).

349   Second, draw a set of connected, polygonal lines to represent the optimal solution that

350   cross the objective function and decision variable axes at the corresponding optimal

351   objective function and decision variable values (Figure 2, blue line). Next, draw

352   additional polygonal lines for each generated near-optimal alternative (Figure 2, green

353   lines). The resulting parallel coordinate plot i) simultaneously links the objective and

354   decision spaces on a single plot, and ii) displays a large number of generated near-

355   optimal alternatives. The plot also iii) shows the extents for each variable (identified in

356   section 3, step b) within the near-optimal region as the lower and upper limits where

357   near-optimal polygonal lines cross each axis. For multi-objective problems, add axes for

358   the new objectives. Although axes order influences Parallel Coordinate plot rendering

359   [*Edsall*, 2003], interactive controls allow users to specify and re-order axes while they

360   explore the near-optimal region.

361   **5   INTERACTIVE EXPLORATION**

362   Interactive mouse-over, highlighting, grouping, axes re-ordering, check box, sliders

363   (scrollbars), and other controls on and next to the plot allow the user to interactively

364   explore the near-optimal region. This exploration serves to: (i) control the rendering and

365   visualization of the near-optimal region, (ii) generate further alternatives of interest to the

366   user, and (iii) streamline the process to elicit un-modelled issues and update the model

367   formulation to respond to those issues. The first set of rendering tools:

368   • Mouse-over an axis and alternative to read a value,

369   • Highlight individual or groups of alternatives on the plot,

370   • Specify the axes order from left to right across the plot.

371

372   A second toolset places slider controls side-by-side on each axis as a control panel

373   [*Inselberg*, 2009] and uses slider settings to dynamically guide exploration of the near-

374   optimal region, update the model formulation, and generate new alternatives. Slider

375   heights show the current allowable ranges for each decision variable and objective

376   function as defined by the original and near-optimal tolerance constraints and slider

377   settings. Setting a slider for a decision axis to a value (e.g., s_{value}) adds a pair of deviation

378   constraints of the form $x_k \le s_{value} + \varepsilon$ and $x_k \ge s_{value} - \varepsilon$ that specify the updated allowable

379   deviation (ε) for the decision variable (x_k) above and below the set value. Similarly, set

380   the slider for the objective function on axis *i* to add constraints of the form $c_i(x) \le s_{value} + \varepsilon$

381   and $c_i(x) \ge s_{value} - \varepsilon$. And when the allowable deviation is zero ($\varepsilon = 0$), setting a slider instead

382   specifies an equity requirement ($x_k = s_{value}$ or $c_i(x) = s_{value}$). Thus, the user can generate:

383   • Individual alternative. Set a slider for one decision variable within its current

384   extents, update extents for all other sliders, set a second slider within its extents,

385   and continue updating extents and setting sliders until specifying all decision

386   variables (this sequence also represents one pass through the integer generation

387   method described in Section 3.2 with no enumeration).

388   • Family of alternatives. Set one to n-1 sliders on decision axes to values within their

389   extents, then use the alternative generation algorithm (section 3) to identify values

390   for the remaining (un-set) decisions. The generated alternatives define a sub-space

391   of the near-optimal region.

392

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

394   • Change the order sliders are set when generating an individual or family of near-

395   optimal alternatives. Equivalently, reorder the axes and specify variable values in

396   the new order.

397   • Relax the tolerable deviation parameter (γ), generate new alternatives, and explore

398   in the expanded region further away from the optimal objective function value.

399

400   During exploration, a user may identify a preferred near-optimal alternative. In this

401   situation, the analyst has an opportunity to step in to help elicit the un-modelled issue(s)

402   that make the alternative preferred. Elicitation will be stronger now because the analyst

403   can work from the manager's revealed preference between the selected alternative,

404   unsatisfactory optimal solution, and prior explored alternatives. For example, what un-

405   modelled requirement or aspiration does the alternative satisfy or achieve that the optimal

406   solution does not? If the requirement is quantifiable, then add a new constraint to the

407   model. Similarly, if the aspiration is quantifiable, add a new objective and transform the

408   single-objective problem into a multi-objective one. Solve the updated model for a new

409   optimal solution (or *pareto* set using the appropriate multi-objective method), generate

410   the new near-optimal region, and continue exploring the updated region. Terminate

411 exploration when the manager is satisfied with an optimal solution, near-optimal

412 alternative, or insight gleaned from exploring.

413

414 The new alternative generation, visualization, and interaction tools are coded in Matlab

415 2013b and available for download at https://github.com/dzeke/Blended-Near-Optimal-

416 Tools. The repository also makes available all model, data, and script files and directions

417 to generate each figure in this paper.

418 **6   EXAMPLE APPLICATION**

419 I illustrate the new near-optimal tools for a water quality management problem to reduce

420 the phosphorus load to Echo Reservoir in the Weber basin, Utah. Throughout the U.S.,

421 excess nutrients like phosphorus and nitrogen impair water bodies and managers must

422 select and locate nutrient removal practices to meet water quality standards. Researchers

423 have developed both simple and complex optimization programs to recommend practices

424 [*Alminagorta et al.*, 2013; *Hsieh and Yang*, 2007; *Maringanti et al.*, 2009]. This near-

425 optimal application first overviews an existing single-objective linear program for Echo

426 Reservoir [*Alminagorta et al.*, 2013] then compares the prior-published optimal solution,

427 near-optimal alternatives generated by several MGA methods, and by the new tools. All

428 alternatives were generated on a Windows 2.7 Ghz machine with 8 GB RAM. The

429 number of alternatives for the MGA-Serial and Simultaneous methods were set to make

430 runtimes commensurate with the new tools. Discussion highlights characteristics of

431 generated alternatives and management insights the new tools identify.

432 **6.1   Model background**

433 The single-objective linear program identifies the cost-effective practices to reduce

434 phosphorus loads from non-point sources to the level specified in a pending Total

435 Maximum Daily Load (TMDL) program for Echo Reservoir. The program considers ten

436 practices such as installing grass filter strips, managing agricultural nutrients, stabilizing

437 stream banks, adopting sprinkler irrigation, retiring grazing land, among others that target

438 three non-point phosphorus sources (manure, grazing, and diffuse runoff) in three sub-

439 watersheds that drain to the reservoir (Chalk Creek, Weber above and below Wanship).

440 The program excludes point sources because sources already adopted best available

441 treatment technologies. The 39 decision variables represent 39 allowable combinations of

442 practices, sources, and sub-watershed locations. The objective function minimizes

443 removal costs. 78 constraints require separate phosphorus load reductions by source as

444 set forth in the TMDL (for a total load reduction of 8,067 kg/year), specify dependencies

445 and mutual exclusive relationships among removal practices, and set upper limits for

446 phosphorus removal for each source in each sub-watershed. The linear program

447 formulation is identical to *Alminagorta et al.* [2013] except that here all decision

448 variables are expressed (scaled) in phosphorus mass removed (Appendix I). Model input

449 data, files, and output are provided on the GitHub repository.

450

451 Input data include implementation costs and phosphorus removal rates for each practice

452 [*Horsburgh et al.*, 2009], land area and stream bank lengths available to implement

453 practices, and load reduction targets across sources. Costs reflect the full cost for

454 agricultural users to implement a practice and include opportunity costs (such as forgone

455 benefits) for practices like retiring land. Prior work [*Adams and Whitehead*, 2006]

456 simulated the major physical, chemical, and biological processes affecting total

457 phosphorus and dissolved oxygen concentrations within the stream and reservoir,

458 identified the permissible phosphorus load to the reservoir and corresponding reduction

459 target, as well as verified that phosphorus loads from each sub-watershed strictly and

460 linearly add to produce the total load to Echo Reservoir. This property allowed

461   formulating as a linear program with constraints that specify global reduction targets for

462   each source and permit trading load reductions across sub-watersheds.

463   **6.2   Optimal and near-optimal comparisons**

464   A parallel coordinate plot (Figure 3) compares the (i) optimal solution to (ii) 13

465   alternatives generated by MGA-Serial, and (iii) some 2,500 near-optimal alternatives

466   generated by the new stratified Monte Carlo Markov Chain sampling method. Each near-

467   optimal alternative was generated by allowing a 110% deviation from the optimal

468   removal cost to permit a small, but still significant (i.e., 10%) increase in removal costs.

469   In Figure 3, the left-most axis and left pink scale show removal costs for each alternative

470   while the 39 axes to the right and right-most green scale show the phosphorus removed

471   by each practice. These axes are further grouped and color-coded by sub-watershed and

472   phosphorus source. The optimal solution (thick black line) costs $985,000, implements

473   only three phosphorus removal practices (manage agriculture nutrients, protect grazing

474   land, and stabilize stream banks) and concentrates removal in the Chalk Creek sub-

475   watershed. Thirteen MGA-Serial alternatives (purple lines) generated in 35 s each cost

476     $1.08 million (110% of the optimal removal cost) and implement up to seven practices

477   per sub-watershed at somewhat more varied levels than the optimal solution.

478

479   In contrast, the new tools generate more numerous near-optimal alternatives (green lines)

480   in under 29 s that cross each decision axis at much wider ranges of values. These

481   alternatives show managers can stabilize stream banks in the Chalk Creek sub-watershed

482   at levels between 0 and 2.3 times the 2,820 kg phosphorus the optimal solution

483   recommends to remove. Alternatively, managers can stabilize stream banks in the sub-

484   watersheds above and below Wanship at levels much higher than the 2,130 kg

485   phosphorus indicated by the MGA-Serial alternatives. In each sub-watershed, managers

486   can also cover crop, install grass filter strips, fence streams, adopt conservation tillage, or

487   manage agricultural nutrients from diffuse runoff. Further, the alternatives identify

488   numerous strategies that vary removal cost between the cost-minimizing value and 110%

489   threshold.

490

491   Additional testing (results not shown) found:

492   • Generating 25 alternatives increased MGA-Serial runtime to 105 s with no

493   appreciable change in the phosphorus removal practices implemented or costs,

494   • MGA-HSJ generated 22 alternatives in under 1 s all with removal cost at the 110%

495   threshold and terminated when all practices entered the non-zero set. Among the

496   HSJ set, up to 6 practices per sub-watershed (15 of the 39 decision variables) had

497   values significantly less than the full extents identified by stratified sampling. For

498   all but one practice (stabilize stream banks in Chalk Creek), MGA-HSJ generated

499   between one and four unique non-zero values. Also, MGA-HSJ terminated much

500   earlier after alternatives 4 and 5 had the same 13 non-zero variables on a model

501   version with different scaled land area and stream bank length decision variables

502   [*Alminagorta et al.*, 2013].

503   • Five MGA-Simultaneous trials each generated ten alternatives in 100 to 500 s all at

504   the 110% removal cost threshold. These alternatives implemented up to six

505   phosphorus removal practices in each sub-watershed.

506   Each MGA method gave a somewhat different and incomplete view of the near-optimal

507   region and together they only found the full extents identified by the new tools for 25 of

508   the 39 decision variables. Therefore, subsequent comparisons focus on MGA-Serial,

509   which, among the three MGA methods tested, had a moderate run-time but identified a

510   more diverse set of removal levels for a slightly smaller set of practices.

511 **6.3   Interaction to explore the near-optimal region**

512   Enabling sliders on the parallel coordinate plot shows the minimum and maximum

513   extents of each decision variable permitted by the original and near-optimal tolerance

514   constraints. Then for example, with no allowable deviation from slider settings, set the

515   slider for stream bank stabilization in Chalk Creek to zero, dynamically update the

516   extents for remaining sliders, and set the sliders for stream bank stabilization in the

517   Weber below and above Wanship sub-watersheds to large values greater than 2,130 kg to

518   explore a region area where MGA-Serial did not generate alternatives. This sequence

519   (Figure 4, checked axes) specifies an alternative set that (i) shifts the location to stabilize

520   stream banks, and (ii) increases total phosphorus removed by the practice. Remaining

521   sliders (unchecked in Figure 4) show the effects of slider settings and exploration. Sliders

522   for manage agricultural nutrients from manure update to positive, non-zero ranges and

523   indicate this practice must now be implemented in *each* sub-watershed while managers

524   can still flexibly protect grazing land across the sub-watersheds.

525

526   With the three slider settings for stream bank stabilization and no allowable deviation, the

527   generate button automatically adds three new equity constraints to the original and near-

528   optimal constraints, updates the model formulation, and generates a new alternative set

529   with shifted, increased stream bank stabilization (Figure 4, orange lines). Removal costs

530   range between $1.05 and $1.08 million (110% of optimal removal cost) and show diverse

531   strategies to both shift and increase phosphorus removal by stream bank stabilization.

532   **6.4   Eliciting an un-modelled issue and updating the model formulation**

533   Selecting one of the shifted, increased stream bank stabilization alternatives as preferred

534   to the optimal solution offers an opportunity to elicit one or more un-modelled issues and

535   improve the model formulation. For example, if the manager explains that his/her

536   preference is motivated by a belief that the TMDL phosphorus removal target of 8,067

537   kg/year is inadequate and s/he wants to remove more phosphorus, then the manager

538   identifies phosphorus removed as a new issue to include with a new objective rather than

539   a requirement or constraint.

540

541   To address, reformulate the problem with a new, second objective (maximize phosphorus

542   removed), solve for the *pareto* alternatives, and define a tolerable deviation for the new

543   objective (here, 56% from the 14,400 kg/year phosphorus removed when maximizing the

544   new objective separately and independently to still allow the prior optimal solution).

545   Then, add the new objective as a new parallel axis to the right of the removal cost axis

546   and present the *pareto* set generated by a standard multi-objective method such as, here,

547   the constraint method (Figure 5, thick black lines on the main Parallel Coordinate plot

548   and black triangles on the inset Cartesian plot). Finally, add numerous newly generated

549   alternatives that are stratified through the objective and decision spaces of the near-

550   optimal region (Figure 5, green lines and dots). Note the ideal point for the two modelled

551   objectives is at high phosphorus removal, low removal cost (bottom right of inset plot)

552   while the upward sloping line of black triangles is the familiar *pareto*-tradeoff for the

553   stated objectives. This tradeoff is also indicated on the main Parallel Coordinate plot by

554   the black line segments that cross from the first to the second axes and converge towards

555   a point to the right of the second phosphorus removed axes. Typical multi-objective

556   analyses try to identify a preferred non-dominated alternative from the *pareto* set. Here,

557   the near-optimal alternative generation and visualization show that the non-dominated

558   alternatives (thick black lines) concentrate in a sub-region of the decision space where

559   *pareto* improvements in the two objectives are achieved mainly by stabilizing longer or

560   shorter lengths of stream banks, particularly in the sub-watershed below Wanship.

561   Interestingly and problematically, the MGA-Serial alternatives (purple lines in the

562   Parallel Coordinate plot, purple circles in the inset) all fall on one boundary of the

563   objective space (high removal cost) whereas numerous stratified-sampled alternatives

564   improve on the MGA alternatives in *both stated objectives*. Highlighting the newly

565   generated alternatives that remove more than 12,000 kg of phosphorus identifies multiple

566   alternatives that remove large quantities of phosphorus from varied practices spread

567   across the three sub-watersheds (Figure 6, orange lines on the parallel coordinate plot and

568   orange crosses in the inset). Managers can select one of these alternatives as preferred to

569   the *pareto* solutions and terminate interaction. Or the analyst can again elicit the un-

570   modelled issue(s) that motivate the preference (e.g., more equitably site phosphorus

571   removal practices across sub-watersheds), add a new equalization objective, generate

572   new alternatives, and re-plot the new alternatives with the equity measure as a new

573   objective.

574   **6.5   Expanding the near-optimal region**

575   Entering a single new tolerable deviation value of 125% significantly expands the near-

576   optimal region and generates some 2,500 new alternatives with removal costs up to

577    $1,230,000 (Figure 7). Here, green shading redundantly indicates removal cost and is

578   ramped in intervals of 5% tolerable deviation from light green (125%) to dark green

579   (100%, optimal cost, plotted on top) to show how the tolerable deviation effects the range

580   of allowable phosphorus removal for each practice. As expected, larger deviations permit

581   larger ranges of removal for many practices like protect grazing lands, cover crop, grass

582   filter strips, conservation tillage, manage agricultural nutrients, sprinkler irrigation, and

583   stabilize stream banks that target grazing and runoff sources (lightest green lines visible

584   at extents). For other practices such as stabilize stream banks in Chalk Creek and the

585   Weber below Wanship sub-watershed, the lightest green lines are not visible at the

586   extents (hidden underneath) and the tolerable deviation does not change the allowable

587 range. For these practices, a system constraint—rather than the near-optimal tolerable

588 deviation—limits implementation. Lastly, the darkest green lines highlight combinations

589 of manage agricultural nutrients, protect grazing land, and stabilize stream banks that

590 maintain total removal cost very near the optimal removal cost.

591 **7   DISCUSSION**

592 New stratified Monte Carlo Markov Chain sampling and Parallel Coordinate

593 visualization tools generate a large number of near-optimal alternatives that perform

594 within a tolerable deviation of the optimal objective function value and communicate the

595 structure and full extent of the near-optimal region. Use of the tools to select and locate

596 phosphorus removal practices to meet load reduction targets for Echo Reservoir, Utah

597 found that managers can implement any practice in any of the sub-watersheds that drain

598 to the reservoir at costs between 100% and 110% of the optimal cost. In contrast, the

599 optimal solution recommended five removal practices while 22, 13, and 10 alternatives

600 generated by, respectively, the MGA- HSJ, Serial, and Simultaneous methods

601 recommended less varied removal practices and levels all at the 110% cost threshold. The

602 new tools take less time to generate more numerous alternatives that more completely

603 show the near-optimal region and reveal managers have great flexibility to meet the

604 phosphorus reduction target.

605

606 The stratified sampling and MGA results differ markedly for scaling, generation

607 criterion, and methodological reasons. In terms of scaling, the stratified-sampling method

608 solves a larger number of smaller optimization problems to identify decision variable

609 maximum extents, whereas MGA- Serial and Simultaneous solve fewer, larger problems

610 to maximize distance between alternatives (Table 1). The maximum extent problems are

611 also linear in the objective whereas the MGA-Serial and Simultaneous distance metrics

612   are non-linear, discontinuous, and must be solved with more computationally intensive

613   methods. Further, the Serial and Simultaneous problems grow in size with the number of

614   generated alternatives whereas stratified sampling simply adds Monte-Carlo Markov

615   chain samples within prior-identified maximum extents. Thus, stratified sampling

616   generates more alternatives much faster than the MGA-Serial and Simultaneous even for

617   Echo Reservoir problem with 39 decision variables.

618

619   MGA-HSJ generates alternatives more quickly up to a density of one alternative per

620   decision variable, but it's maximally different search criteria does not necessary identify

621   near-optimal region extents for all decision variables. Minimizing the sum the non-zero

622   decision variables does not control which and how many variables *enter* the non-zero set

623   and at what level(s). Thus, one or multiple variables enter at a low or arbitrary level on a

624   particular iteration, MGA-HSJ drives those variables back to zero for remaining

625   iterations, and never identifies their true (larger) extents. Even a density of one

626   alternative per decision variable gives an incomplete view of the region for moderate and

627   large problems. In contrast, the new tools achieve a higher alternative density by

628   identifying the extents of each decision variable and objective, then sampling alternatives

629   from the boundaries and throughout the linked decision and objectives spaces of the

630   region.

631

632   Methodologically, MGA uses an optimization model and an objective function to express

633   a narrow preference for where in the near-optimal region the analyst seeks new

634   alternatives to address un-modelled issues. This optimizing may generate useful

635   alternatives for select issues that align with the objective function parameterization. But

636   for other un-modelled issues that do not align, useful alternatives could lie *anywhere* in

637   the near-optimal region. (If the analyst has information on where to find useful

638   alternatives, the analyst can model the underlying issue and tune the MGA objective

639   function to better direct generation towards the more useful alternatives). Because the

640   analyst does not generally have *a priori* information on the un-modelled issues, the new

641   tools instead use a sweep method to first stratify sample alternatives from throughout the

642   near-optimal region. Then visualize them en mass and offer interactive plot controls to

643   help streamline the process to select preferred alternatives, elicit un-modelled issues that

644   motivate the preference, update the model formulation to respond to those issues, and

645   generate new near-optimal alternatives from the updated formulation.

646

647   The several interactive examples illustrate the differences. For example, the more

648   complete coverage of stratified sampled alternatives revealed options to set sliders to

649   shift and increase phosphorus removal by stream bank stabilization (Figure 4) and

650   prompted updating the model formulation to add a second objective to increase the total

651   phosphorus removed. In the resulting multi-objective formulation, MGA alternatives

652   clustered on an objective region boundary of large removal cost (Figure 5), whereas

653   numerous stratified sampled alternatives were distributed throughout the near-optimal

654   decision and objective spaces, showed numerous ways to spread implementation across

655   sub-watersheds, and improve on the MGA alternatives in *each stated objective*.

656

657   Further interaction relaxed the tolerable deviation parameter to expand the near-optimal

658   region. Results showed the progressive effects of expansion on region composition and

659   ranges of allowable phosphorus removal for each practice. Typically, Parallel Coordinate

660   plots only permit comparisons between *adjacent* axes [*Edsall*, 2003]. Here, ramping color

661   along the removal cost axis in Figure 7 created contours of removal cost through the

662   near-optimal region and simultaneously related removal cost to *all* other plot axes.

663

664   Admittedly, the outcome of interactions to specify a preferred alternative depend on the

665   sequence to set sliders similar to ordering axes on a Parallel Coordinate plot [*Edsall*,

666   2003]. At the same time, the controls allow managers to jump directly to the region

667   features of most interest. All of the examples presented are demonstrative and important

668   work remains to test the interactive process with managers and decision makers. Towards

669   this end, my research group is now working with local stakeholders to apply the near-

670   optimal methods to a model that allocates water for environmental and ecological

671   objectives in the lower Bear River, Utah [*Alafifi and Rosenberg*, 2014].

672

673   A key aspect to test is whether the large number of generated alternatives—some 2,500

674   alternatives for the Echo Reservoir problem—overwhelm users. The large number fills

675   the Parallel Coordinate plot, allows the user to view generated alternatives together *en*

676   *mass* as a single more comprehensive near-optimal region, and serves as a reference

677   frame to guide subsequent interactive exploration of the region for preferred alternatives.

678   The problem structure and size determine the appropriate number of alternatives to

679   initially generate. Minimally, this number should be $2(n+d)$ to separately and

680   independently show the minimum and maximum extents for each decision variable and

681   objective function in the near-optimal region. But the number can be reduced to 2, 4, 6,

682   … to identify extents for a sub-set of decision variables on which the user will focus

683   initial interaction.

684

685   The current tools and applications (including water supply planning in Amman, Jordan

686   [*Rosenberg*, 2012]) feature single- and multi-objective linear and mixed-integer problems

687   that have convex, bounded feasible regions. Important work remains to extend the tools

688   to non-linear problems. To extend, pass model data in structures that can describe non-

689   linear problems. Further, generate alternatives with Monte-Carlo Markov Chain samplers

690   such as Hit-and-Run [*Kroese et al.*, 2011] that work over non-convex regions. My

691   research group is also starting to make these extensions.

692

693   Finally, the new tools help users explore the near-optimal region and streamline the

694   process to elicit un-modelled issues but do not guarantee a user will find a preferred

695   alternative to the optimal solution. First, a preferred alternative may not lie in the near-

696   optimal region and second a user may stop exploring before they find an alternative.

697   Even should a user stop exploring, the blended tools offer new insights on the region and

698   problem structure and can elicit more varied un-modelled issues than either the optimal

699   solution or prior MGA generation methods.

700   **8   CONCLUSIONS**

701   New blended alternative generation, visualization, and interaction tools tackle the dual

702   challenges to comprehensively identify and tractably communicate the structure and full

703   extent of the near-optimal region for an optimization problem. Stratified Monte-Carlo

704   Markov Chain sampling generates a large number of alternatives spread through the

705   objective and decision spaces of the near-optimal region. A Parallel Coordinate plot

706   visualizes generated alternatives en mass as a region. Interactive sliders, checkboxes, re-

707   sampling and other controls allow users to further explore region features of most

708   interest. The tools also help streamline the process to elicit un-modelled issues and

709   update model formulations in response to the issues.

710

711   Use of the tools for a water quality management problem at Echo Reservoir, Utah

712   showed managers can flexibly implement any phosphorus removal practice in any sub-

713   watershed and keep removal cost within 110% of the optimal cost. This flexibility was

714   upheld by further interaction and exploration, using elicited information to transform the

715   original single-objective formulation into a multi-objective problem, and relaxing the

716   near-optimal tolerance parameter to explore an expanded region farther from the

717   optimum. Compared to three MGA methods, the new tools generate more numerous

718   alternatives faster, more completely show the near-optimal region, and can help elicit a

719   larger set of un-modelled issues.

720

721   Important work remains to test managers and decision makers use of the tools as well as

722   extend to non-linear problems. Here, demonstrations for linear and mixed-integer

723   problems: (i) communicate the structure and full extent of the near-optimal region, (ii)

724   guide further exploration within the region, (iii) help streamline the process to elicit un-

725   modelled issues and update model formulations, and (iv) show the numerous and flexible

726   ways to manage a system and maintain near-optimal performance.

727   **9   ACKNOWLEDGMENTS**

730   **10 APPENDIX I– PHOSPHORUS REMOVAL MODEL FORMULATION**

731   This appendix presents the linear programming model formulation for the Echo Reservoir

732   example. The formulation follows identically from *Alminagorta et al.* [2013] except here

733   the sole decision variables are phosphorus mass removed by practice i targeting

734   phosphorus source s in sub-watershed w (P_{iws}; kg). This specification eliminates the

735   equity constraints that relate land area and stream bank length decisions to phosphorus

736   mass. The single objective minimizes total removal costs (Z):

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

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

738   Constraints are:

739   • Phosphorus removal across the three sub-watersheds must meet or exceed global

740   load reduction targets for each source s (p_{ws}; kg)

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

741   where c_{is} = a binary parameter that takes the value 1 to indicate phosphorus

742   removal practice i can be applied to source s (0 otherwise).

743   • Phosphorus removal is limited by the available land area and stream bank length

744   resources in each sub-watershed (b_{gw}; km^2 or km)

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

745   where x_{gi} = a binary parameter that takes the value 1 to indicate phosphorus

746   removal practice i precludes implementing other practices on the same resource g

747   (0 otherwise) and e_i = removal practice efficiency (kg P/km^2 or kg P/km).

748   • Phosphorus removal cannot exceed the existing load in each sub-watershed for each

749   source s (l_{sw}; kg)

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

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

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

752   onto near-optimal analysis.

753   **11   REFERENCES**

754   Adams, C., and J. Whitehead (2006), Echo reservoir TMDL water quality study*Rep.*,
755   Utah Dept. of Environmental Quality—Division of Water Quality, Salt Lake City,
756   Utah. http://www.waterquality.utah.gov/TMDL/Echo_Reservoir_TMDL.pdf.
757   Alafifi, A. H., and D. E. Rosenberg (2014), Rethinking Riverine Habitat Quality:
758   Integrated Systems Modeling to Improve Watershed Habitat Management and

759   Decision Making in *7th Intl. Congress on Env. Modelling and Software,*, edited by
760   D. P. Ames, N. W. T. Quinn and A. E. Rizzoli, pp. 1-8, International Environmental
761   Modelling and Software Society San Diego, CA.
762   http://www.iemss.org/sites/iemss2014/papers/iemss2014_submission_261.pdf.
763 Alminagorta, O., B. Tesfatsion, D. E. Rosenberg, and B. Neilson (2013), Simple
764   Optimization Method to Determine Best Management Practices to Reduce
765   Phosphorus Loading in Echo Reservoir, Utah, *Journal of Water Resources Planning*
766   *and Management*, *139*(1), 122-125.
767   http://ascelibrary.org/doi/abs/10.1061/%28ASCE%29WR.1943-5452.0000224.
768 Aufiero, A., R. S. Sessa, and E. Weber (2001), Set-valued control laws in minmax
769   control problem, paper presented at Modeling and control in environmental issues,
770   International Federation of Automatic Control, Yokohama, Japan.
771   http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&
772   ved=0CCsQFjAA&url=ftp%3A%2F%2Fhome.dei.polimi.it%2Foutgoing%2FAndre
773   a.Francesco.Castelletti%2Fpapers%2Fsoncini_set_val.pdf&ei=FMmoUtjgFozwrAH
774   UvYBg&usg=AFQjCNH7Ur4bx6gSI5ixfupvzbvjroWL7w&bvm=bv.57799294,d.a
775   WM.
776 Benham, T. (2011), Uniform distribution over a convex polytope, edited, Matlab Central.
777   http://www.mathworks.com/matlabcentral/fileexchange/34208.
778 Brill, E. D., Jr., S.-Y. Chang, and L. D. Hopkins (1982), Modeling to Generate
779   Alternatives: The HSJ Approach and an Illustration Using a Problem in Land Use
780   Planning, *Management Science*, *28*(3), 221-235.
781   http://www.jstor.org/stable/2630877.
782 Brown, G. G., R. F. Dell, and R. K. Wood (1997), Optimization and persistence,
783   *Interfaces*, *27*(5), 15-37. <Go to ISI>://A1997YD16300003.
784 Burton, R. O., J. S. Gidley, B. S. Baker, and K. J. Reda-Wilson (1987), Nearly Optimal
785   Linear Programming Solutions: Some Conceptual Issues and a Farm Management
786   Application, *American Journal of Agricultural Economics*, *69*(4), 813-818.
787   http://ajae.oxfordjournals.org/content/69/4/813.abstract.
788 Castelletti, A., A. V. Lotov, and R. Soncini-Sessa (2010), Visualization-based multi-
789   objective improvement of environmental decision-making using linearization of
790   response surfaces, *Environmental Modelling & Software*, *25*(12), 1552-1564.
791   http://www.sciencedirect.com/science/article/pii/S1364815210001775.
792 Chang, S.-Y., E. D. Brill, Jr., and L. D. Hopkins (1982), Use of mathematical models to
793   generate alternative solutions to water resources planning problems, *Water*
794   *Resources Research*, *18*(1), 58-64. http://dx.doi.org/10.1029/WR018i001p00058.
795 Chen, M.-H., and B. Schmeiser (1993), Performance of the Gibbs, Hit-and-Run, and
796   Metropolis Samplers, *Journal of Computational and Graphical Statistics*, *2*(3), 251-
797   272. http://www.tandfonline.com/doi/abs/10.1080/10618600.1993.10474611.
798 Cohon, J. L., and D. H. Marks (1975), A review and evaluation of multiobjective
799   programing techniques, *Water Resources Research*, *11*(2), 208-220.
800   http://dx.doi.org/10.1029/WR011i002p00208.
801 Deb, K. (2008), Introduction to Evolutionary Multiobjective Optimization, in
802   *Multiobjective Optimization*, edited by J. Branke, K. Deb, K. Miettinen and R.
803   Słowiński, pp. 59-96, Springer Berlin Heidelberg.
804   http://link.springer.com/chapter/10.1007/978-3-540-88908-3_3.
805 Edsall, R. M. (2003), The parallel coordinate plot in action: design and use for
806   geographic visualization, *Computational Statistics & Data Analysis*, *43*(4), 605-619.
807   http://www.sciencedirect.com/science/article/pii/S0167947302002955.

808   Gelfand, A. E., and A. F. M. Smith (1990), Sampling-Based Approaches to Calculating
809   Marginal Densities, *Journal of the American Statistical Association*, *85*(410), 398-
810   409. http://www.tandfonline.com/doi/abs/10.1080/01621459.1990.10476213.
811   Harrington, J. J., and J. S. Gidley (1985), The Variability of Alternative Decisions in a
812   Water Resources Planning Problem, *Water Resources Research*, *21*(12), 1831-1840.
813   http://dx.doi.org/10.1029/WR021i012p01831.
814   Hastings, W. K. (1970), Monte Carlo Sampling Methods Using Markov Chains and Their
815   Applications, *Biometrika*, *57*(1), 97-109. http://www.jstor.org/stable/2334940.
816   Horsburgh, J. S., N. O. Mesner, D. K. Stevens, A. Caplan, T. Glover, and B. T. Neilson
817   (2009), USEPA targeted watersheds grant Bear River Basin*Rep. Project # WS-
818   97807301*, 292 pp, Utah State University, Logan, Utah.
819   http://bearriverinfo.org/files/publications/publication/pub__8578631.pdf.
820   Hsieh, C.-D., and W.-F. Yang (2007), Optimal nonpoint source pollution control
821   strategies for a reservoir watershed in Taiwan, *Journal of Environmental
822   Management*, *85*(4), 908-917.
823   http://www.sciencedirect.com/science/article/pii/S0301479706003446.
824   Inselberg, A. (2009), *Parallel Coordinates: visual multidimensional geometry and its
825   applications*, 554 pp., Springer, New York.
826   Kasprzyk, J. R., P. M. Reed, B. R. Kirsch, and G. W. Characklis (2009), Managing
827   population and drought risks using many-objective water portfolio planning under
828   uncertainty, *Water Resources Research*, *45*(12), W12401.
829   http://dx.doi.org/10.1029/2009WR008121.
830   Kennedy, K. T., and J. T. Quinn (1998), Multiple Sourcing Alternatives Using Nearly
831   Optimal Programming, *Journal of Education for Business*, *73*(4), 206-210.
832   http://dx.doi.org/10.1080/08832329809601631.
833   Kollat, J. B., and P. M. Reed (2007), A Framework for Visually Interactive Decision-
834   making and Design using Evolutionary Multiobjective Optimization (VIDEO),
835   *Environmental Modelling & Software*, *22*(12), 1691-1704.
836   http://www.coe.psu.edu/water/images/3/36/Kollet_and_Reed_2007_VIDEO_Paper.p
837   df.
838   Korhonen, P., and J. Wallenius (1988), A pareto race, *Naval Research Logistics (NRL)*,
839   *35*(6), 615-623. http://dx.doi.org/10.1002/1520-6750(198812)35:6<615::AID-
840   NAV3220350608>3.0.CO;2-K.
841   Kroese, D. P., T. Taimre, and Z. I. Botev (2011), *Handbook of Monte Carlo Methods*,
842   772 pp., John Wiley and Sons, New York.
843   Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, 343 pp., Springer, New
844   York.
845   Lotov, A. V., V. A. Bushenkov, and G. K. Kamenev (2004), *Interactive Decision Maps
846   Approximation and Visualization of Pareto Frontier*, Springer-Verlag, Heidelberg.
847   Loughlin, D. H., S. R. Ranjithan, E. D. Brill, and J. W. Baugh (2001), Genetic algorithm
848   approaches for addressing unmodeled objectives in optimization problems,
849   *Engineering Optimization*, *33*(5), 549-569.
850   http://dx.doi.org/10.1080/03052150108940933.
851   Maringanti, C., I. Chaubey, and J. Popp (2009), Development of a multiobjective
852   optimization tool for the selection and placement of best management practices for
853   nonpoint source pollution control, *Water Resources Research*, *45*(6), W06406.
854   http://dx.doi.org/10.1029/2008WR007094.
855   Marler, R. T., and J. S. Arora (2004), Survey of multi-objective optimization methods for
856   engineering, *Structural and Multidisciplinary Optimization*, *26*(6), 369-395. <Go to
857   ISI>://000220714000001.

858 Matheiss, T. H., and D. S. Rubin (1980), A Survey And Comparison Of Methods For
859   Finding All Vertices Of Convex Polyhedral-Sets, *Mathematics of Operations*
860   *Research*, *5*(2), 167-185. http://www.jstor.org/stable/10.2307/3689148.
861 Miettinen, K., F. Ruiz, and A. P. Wierzbicki (2008), Introduction to Multiobjective
862   Optimization: Interactive Approaches, in *Multiobjective Optimization*, edited by J.
863   Branke, K. Deb, K. Miettinen and R. Słowiński, pp. 27-57, Springer Berlin
864   Heidelberg. http://dx.doi.org/10.1007/978-3-540-88908-3_2.
865 Nardini, A., and D. Montoya (1995), Remarks on a Min-Max Optimization Technique for
866   the Management of a Single Multiannual Reservoir Aimed at Hydroelectric
867   Generation and Water Supply, *Water Resources Research*, *31*(4), 1129-1135.
868   http://dx.doi.org/10.1029/94WR02705.
869 Nemhauser, G. L. (1994), The Age Of Optimization - Solving Large-Scale Real-World
870   Problems, *Operations Research*, *42*(1), 5-13. <Go to ISI>://A1994NL48500001.
871 Orlovski, S., S. Rinaldi, and R. Soncini-Sessa (1983), A min-max approach to storage
872   control problems, *Applied Mathematics and Computation*, *12*(2–3), 237-254.
873   http://www.sciencedirect.com/science/article/pii/0096300383900103.
874 Ortiz, M. C., L. A. Sarabia, M. S. Sanchez, and D. Arroyo (2011), Improving the
875   visualization of the Pareto-optimal front for the multi-response optimization of
876   chromatographic determinations, *Analytica Chimica Acta*, *687*(2), 129-136.
877 Rogers, P. P., and M. B. Fiering (1986), Use of Systems Analysis in Water Management,
878   *Water Resources Research*, *22*(9), 146S - 158S.
879 Rosenberg, D. E. (2012), Near-optimal water management to improve multi-objective
880   decision making, paper presented at 2012 International Congress on Environmental
881   Modelling and Software: Managing Resources of a Limited Planet: Pathways and
882   Visions under Uncertainty, Sixth Biennial Meeting, International Environmental
883   Modelling and Software Society (iEMSs), Leipzig, Germany, July 1-5.
884   http://www.iemss.org/sites/iemss2012//proceedings/A2_0656_Rosenberg.pdf.
885 Rubinstein, R. Y. (1986), *Monte Carlo optimization, simulation, and sensitivity of*
886   *queueing networks*, John Wiley & Sons, New York, NY.
887 Sahinidis, N. V. (2004), Optimization under uncertainty: state-of-the-art and
888   opportunities, *Computers & Chemical Engineering*, *28*(6-7), 971-983.
889   http://www.sciencedirect.com/science/article/B6TFT-49YH97T-
890   1/2/f15875aad97740410effc526416289aa.
891 Shenfield, A., P. J. Fleming, and M. Alkarouri (2007), Computational steering of a multi-
892   objective evolutionary algorithm for engineering design, *Engineering Applications*
893   *of Artificial Intelligence*, *20*(8), 1047-1057.
894   http://www.sciencedirect.com/science/article/pii/S0952197607000085.
895 Smith, R. L. (1984), Efficient Monte Carlo Procedures for Generating Points Uniformly
896   Distributed Over Bounded Regions, *Operations Research*, *32*(6), 1296-1308.
897   http://www.jstor.org/stable/170949.
898 Steuer, R. E. (1986), *Multi Criteria Optimization: Theory, Computation, and Application*,
899   John Wiley and Sons, New York.
900 Stummer, C., and E. Kiesling (2009), A multicriteria decision support system for
901   competence-driven project portfolio selection, *International Journal of Information*
902   *Technology & Decision Making*, *8*(2), 379-401.
903   http://web.ebscohost.com/ehost/detail?vid=3&hid=122&sid=da5cb4f0-dd00-4817-
904   8135-
905   a079d8b26112%40sessionmgr115&bdata=JnNpdGU9ZWhvc3QtbGl2ZQ%3d%3d#
906   db=cph&AN=53872988.

907 Wegman, E. J. (1990), Hyperdimensional Data Analysis Using Parallel Coordinates,
908   *Journal of the American Statistical Association*, *85*(411), 664-675.
909   http://www.jstor.org/stable/2290001.
910 Wierzbicki, A. (1979), The use of reference objectives in multiobjective
911   optimization*Rep.*, 36 pp, International Institute for Applied Systems Analysis,
912   Laxenburg, Austria. http://www.iiasa.ac.at/publication/more_WP-79-066.php.
913 Zechman, E. M., and S. R. Ranjithan (2004), An evolutionary algorithm to generate
914   alternatives (EAGA) for engineering optimization problems, *Engineering*
915   *Optimization*, *36*(5), 539-553. http://dx.doi.org/10.1080/03052150410001704863.
916 Zechman, E. M., and R. Ranjithan (2007), Generating Alternatives Using Evolutionary
917   Algorithms for Water Resources and Environmental Management Problems, *Journal*
918   *of Water Resources Planning and Management*, *133*(2), 156-165.
919   http://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-
920   9496%282007%29133%3A2%28156%29.
921 Zhang, Q., and H. Li (2007), MOEA/D: A Multiobjective Evolutionary Algorithm Based
922   on Decomposition, *IEEE Transactions on Evolutionary Computation*, *11*(6), 712-
923   731. http://dx.doi.org/10.1109/TEVC.2007.892759.
924
925
926

927
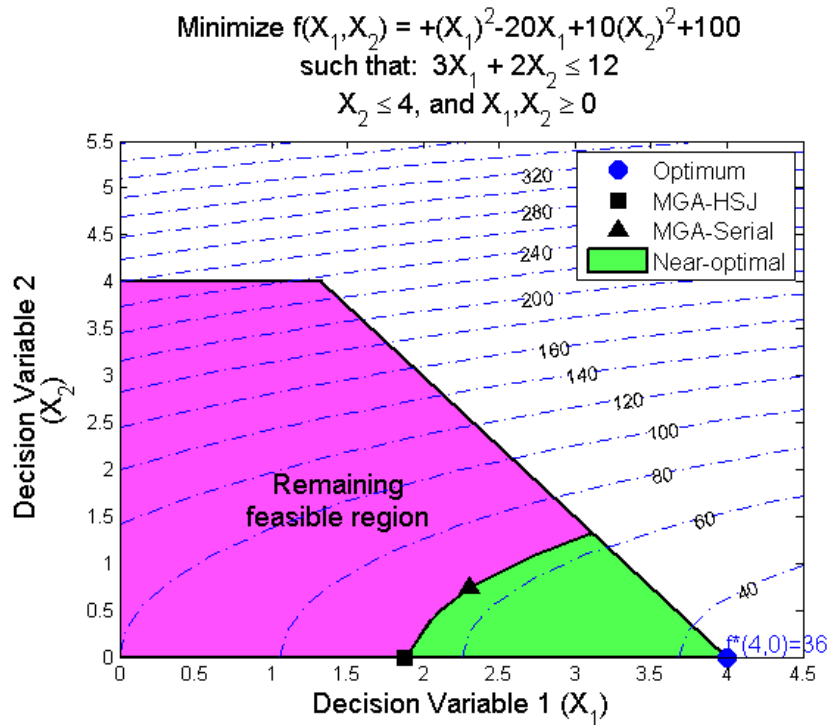
928 **Figure 1. Comparison of the optimal solution and Modelling to Generate**

929 **Alternatives (MGA) results to the near-optimal region for an example two-decision**

930 **problem with a quadratic objective function, four linear constraints, tolerable**

931 **deviation of γ=1.80, and generation density of 0.5 alternatives per decision variable**

932 **for the MGA-Serial method that is at least three times larger than densities reported**

933 **in prior applications.**

934

935
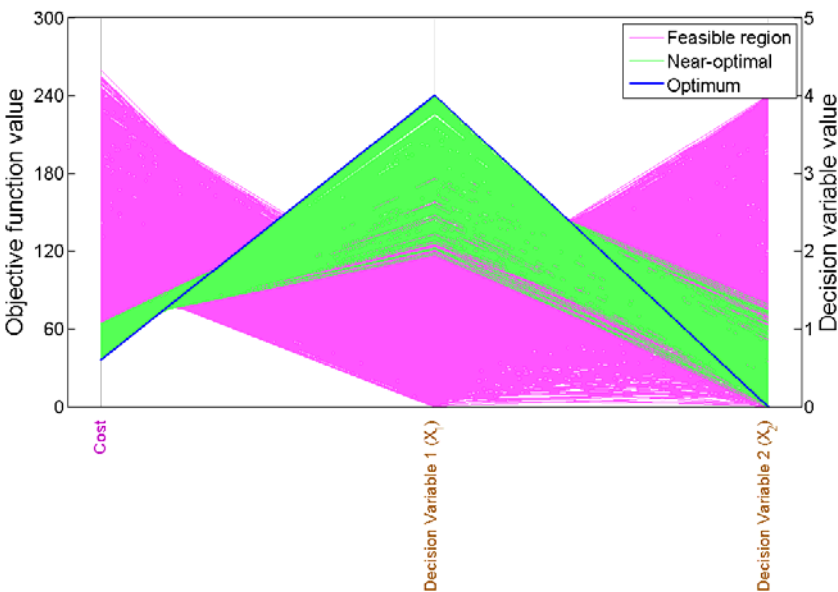
936   **Figure 2.   Parallel Coordinate plot that shows the optimal solution, feasible, and**

937   **near-optimal regions across the linked objective (left most axis and left scale) and**

938   **decision (right two axes and right scale) spaces for the example problem in Figure 1.**

939
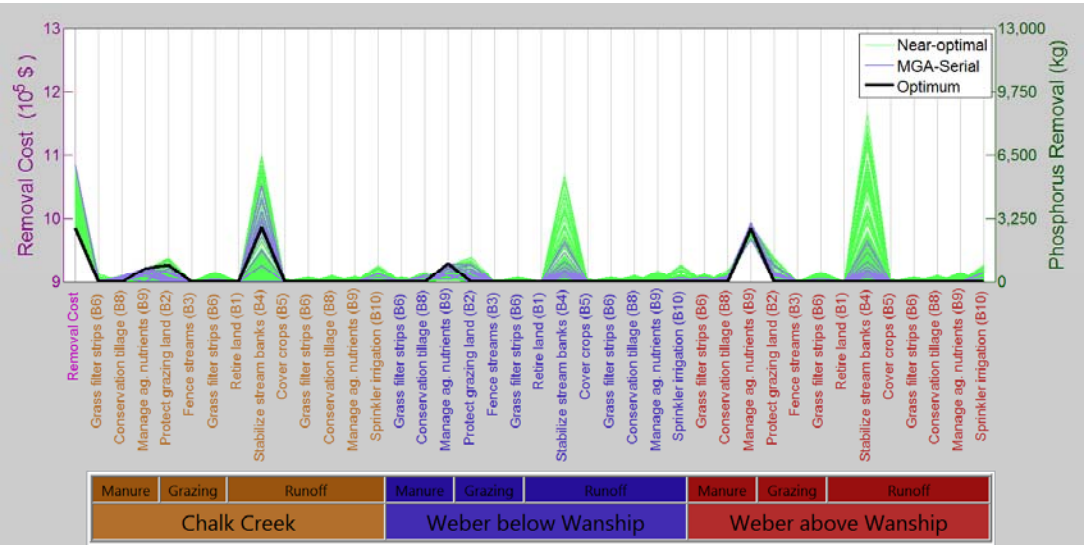
Figure 3.   Some 2,500 stratified-sampled near-optimal phosphorus removal strategies for Echo Reservoir, Utah (green lines) more completely cover in shorter run time the region within 110% of the optimal removal cost than 13 alternatives serially generated by the Modelling to Generate Alternatives method (purple lines) or the optimal solution (black line).

946
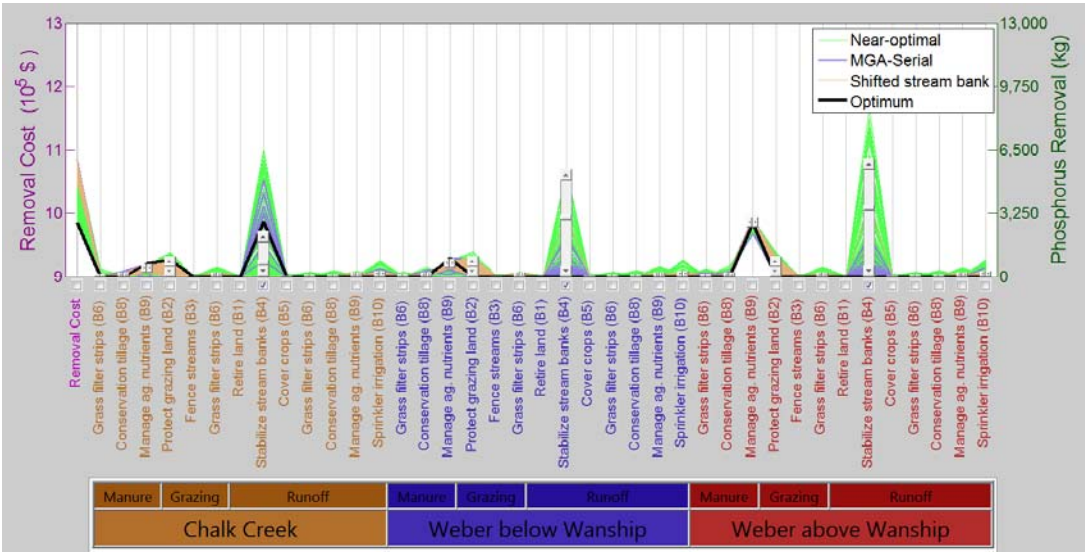
Figure 4. Exploring the effects (orange lines and slider heights) to not stabilize stream banks in Chalk Creek (first checked axis and slider) but shift and increase phosphorus removal in the Weber above and below Wanship sub-watersheds (second and third checked axes and sliders) in a sub-space of the near-optimal region (green lines) where MGA-Serial did not generate alternatives (purples lines).
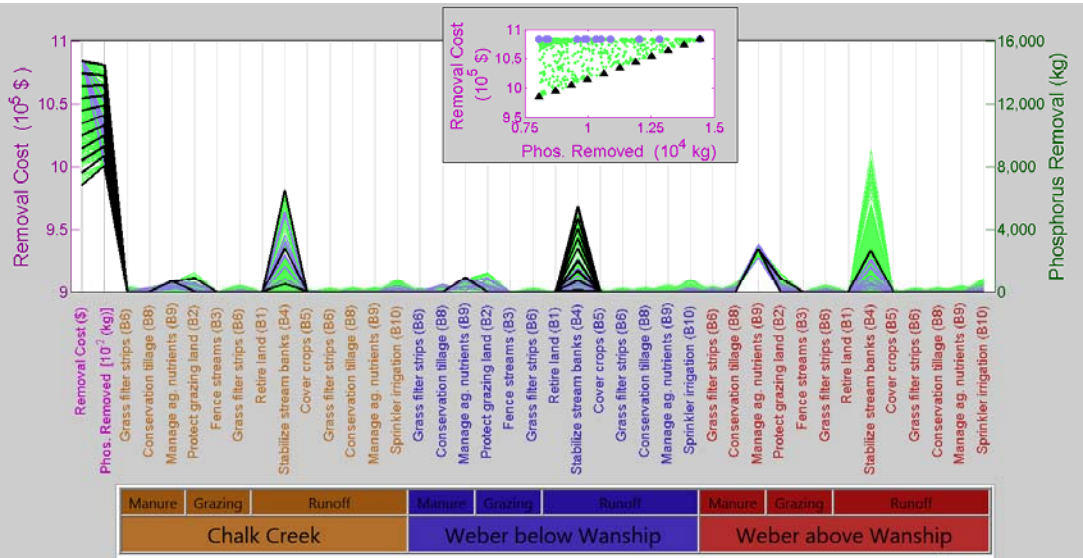
947

948

949

950

951

952

Figure 5.   Comparing *pareto* solutions generated by the constraint method (thick black lines and black triangles), 13 alternatives generated by MGA-Serial (purple lines and purple circle), and stratified sampled near-optimal alternatives (green lines and green dots) for an updated multi-objective formulation in linked objective-decision spaces (main parallel coordinate plot) and objective space (inset Cartesian plot).
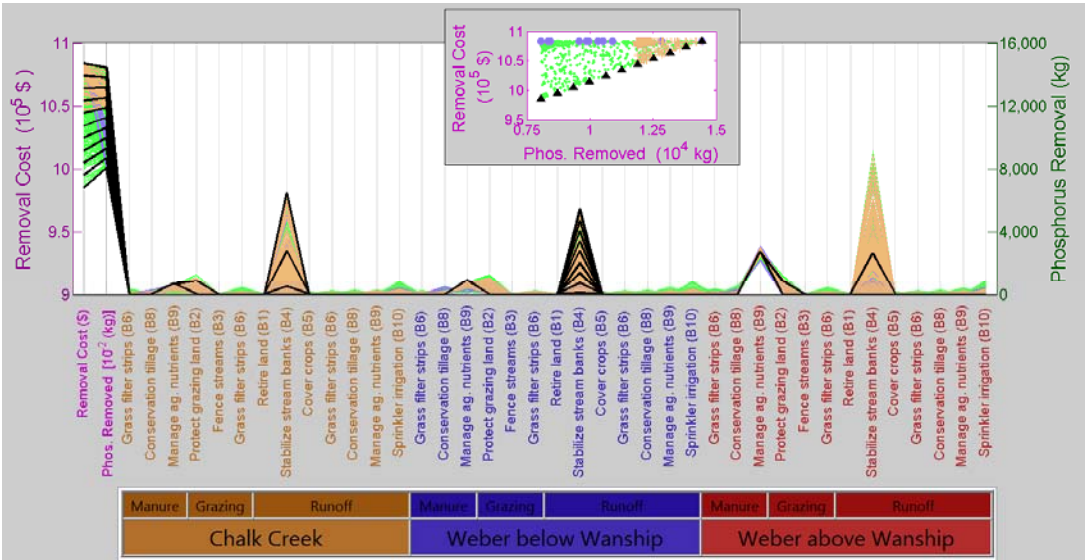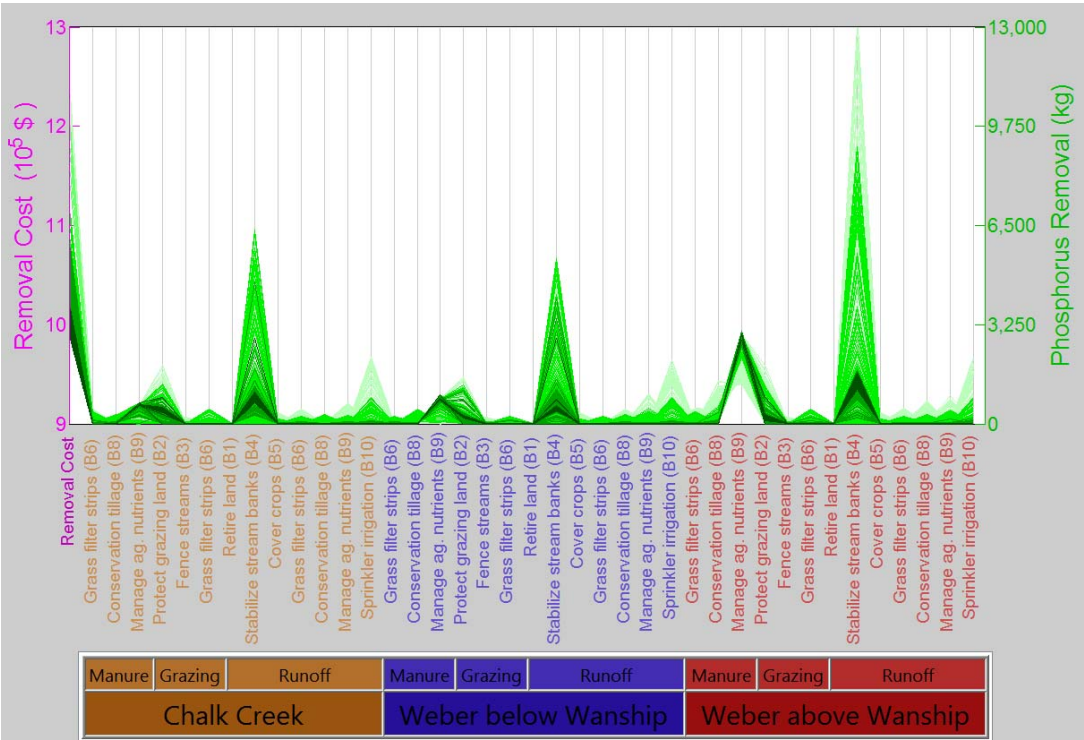
961

962 **Figure 6.   Sub-region of near-optimal alternatives that remove large amounts of**

963 **phosphorus (orange lines and crosses) but use more varied locations and practices**

964 **than either *pareto* solutions (thick black lines and black triangles) or MGA-Serial**

965 **alternatives (purple lines and circles).**

966

967

968

Figure 7.   Progressive effects of relaxing the near-optimal tolerance parameter on

ranges of allowable phosphorus removal for each practice. Green shading denotes

removal cost and varies in 5% increments from dark green (optimal cost) to light

green (125% of optimal cost).

973

974 **Table 1. Comparison of problem size, solution effort, and performance of near-**

975 **optimal methods[a]**

Method	Problem Size			Upper Bound on Number of Solves	Performance on Echo Reservoir Problem		
	Decision Variables	Constraints	Type[b]		Alternatives Generated	Time (sec)	Alt. Density[c]
MGA-HSJ	n	m	LP	n-1	22[d]	0.5	0.6
MGA-Serial	n	m	NLP	p	13	35	0.3
MGA-Simultaneous	p·n	p·m	NLP	1	10	101 to 553	0.3
Stratified sampling	n	m	LP	2(n+d)	2,500	25.7	64.1

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