**Project R19AC00053 Performance/Progress Report**
**Project Title**: Establishing a Framework for Robust Planning in the Colorado River Basin
**Phase III Report**: A framework to simulate, calculate, and visualize robustness of Lake Mead operation policies
**Authors**: Nathan Bonham, Joseph Kasprzyk, Edith Zagona

## 1. Report overview

This project seeks to build a modeling and analytical framework for improved planning on the Colorado River Basin (CRB), building on initial research in a prior funded grant. The framework combines optimization technology, robustness evaluation, statistical analysis to identify performance vulnerabilities, and interactive visualization tools. Specifically, the framework is building on advances in Many Objective Robust Decision Making (MORDM, Joseph Robert Kasprzyk et al., 2013).

### Stated goals

MORDM has four components: (i) conceptualization of the problem in terms of an optimization problem formulation (ii) multi-objective optimization to generate new planning alternatives, (iii) robustness analysis in which the alternatives are subjected to a wide range of uncertainties, and (iv) vulnerability analysis that identifies critical uncertainties that cause performance issues for the alternatives. The project is broken into four phases that address different elements of the MORDM process. Phase I is literature review on advancements in MORDM, uncertainty characterization, and the calculation of robustness. Phase II performs uncertainty quantification; Phase III performs robustness analysis, and Phase IV performs the vulnerability assessment.

Within this reporting period, we completed Phase III (robustness analysis). This reporting period included analyzing the considerable data generated within MORDM and building interactive web applications, which sets us up well to complete Phase IV in the next reporting period.

### Progress to date

Nathan Bonham, the graduate student funded by the project, continues to make excellent progress, hitting important project milestones, and exceeding our expectations on desired progress and communication. He has frequently met with project leaders Kasprzyk and Zagona, providing daily e-mail updates and periodically presenting extensive PowerPoint presentations that identify the newest literature, points of departure for the research, and results from data analysis and visualization. As part of Phase III robustness analysis, we created an interactive web application that has wide-ranging applicability to support Reclamation's MORDM investigations. We have met several times with BOR collaborators, incorporating their feedback on the app and how the analysis is carried out and communicated.

The desired outputs from Phase III documented in this report include robustness metric outputs for all planning alternatives and an interactive visualization tool to support analysis of the robustness results. Although we anticipated also sharing a small number of promising planning alternatives to be studied in Phase IV, we are continuing to work with our Reclamation partners to identify such alternatives. We anticipate that in Phase IV we will build analytical infrastructure that will enable analysis of all potential planning alternatives, to facilitate the most exploration possible.

The remainder of this document provides the formal Phase III Report. Robustness analysis is the process of simulating solutions in an ensemble of plausible future States of the World (SOW), then using statistical functions, called robustness metrics, to quantify the performance of each policy when tested in the SOW ensemble. Previous work, Phase II, characterized uncertainty in the Colorado River Basin (CRB) and developed a framework for sampling an efficient SOW ensemble.

In this report, we describe how the uncertainties in the SOW ensemble are incorporated in the Colorado River Simulation System (CRSS). Then, we calculate eight types of robustness metrics and establish an interactive visualization web application (app) to filter Lake Mead policies by robustness. Finally, we perform an example robustness analysis within the app while providing recommendations for robustness metrics and a workflow for identifying robust policies.

The concluding section of the report summarizes research contributions of this project Phase: a web-application built on the philosophy of robustness metric exploration and efficient filtering of solution alternatives, leveraging Pareto-dominance as an alternative to aggregating metrics, and overall a framework for Reclamation and CRB stakeholders to assess robustness of Lake Mead policies.

## 2. Methods

We provide an overview of the methods in Figure 1, with a black box denoting the new research tasks from Phase III. Reclamation has provided us with 463 Lake Mead policies for this analysis. The policies were identified with the Borg Multi-Objective Evolutionary Algorithm (MOEA), using eight objectives and 14 decision variables (DV) (Alexander, 2018; Hadka and Reed, 2013). Each policy is subjected to an ensemble of multiple States of the World (SOWs). Although the full set of methods for generating the SOW ensemble were documented in Phase II, Section 2.1 of this report briefly discusses the ensemble used in our analysis because the ranges of some uncertainties have changed. In Section 2.2, we discuss how the SOW ensemble was modeled in the Colorado River Simulation System (CRSS). The process of simulating each policy in every SOW creates a "stress testing" database describing the distribution of performance for each policy. Section 2.3 describes the types of robustness metrics we chose for the analysis. The robustness metrics are statistical functions that summarize distributions of performance as scalar values. In section 2.4, we introduce the CRSS robustness analysis app for interactive visualization of robustness.

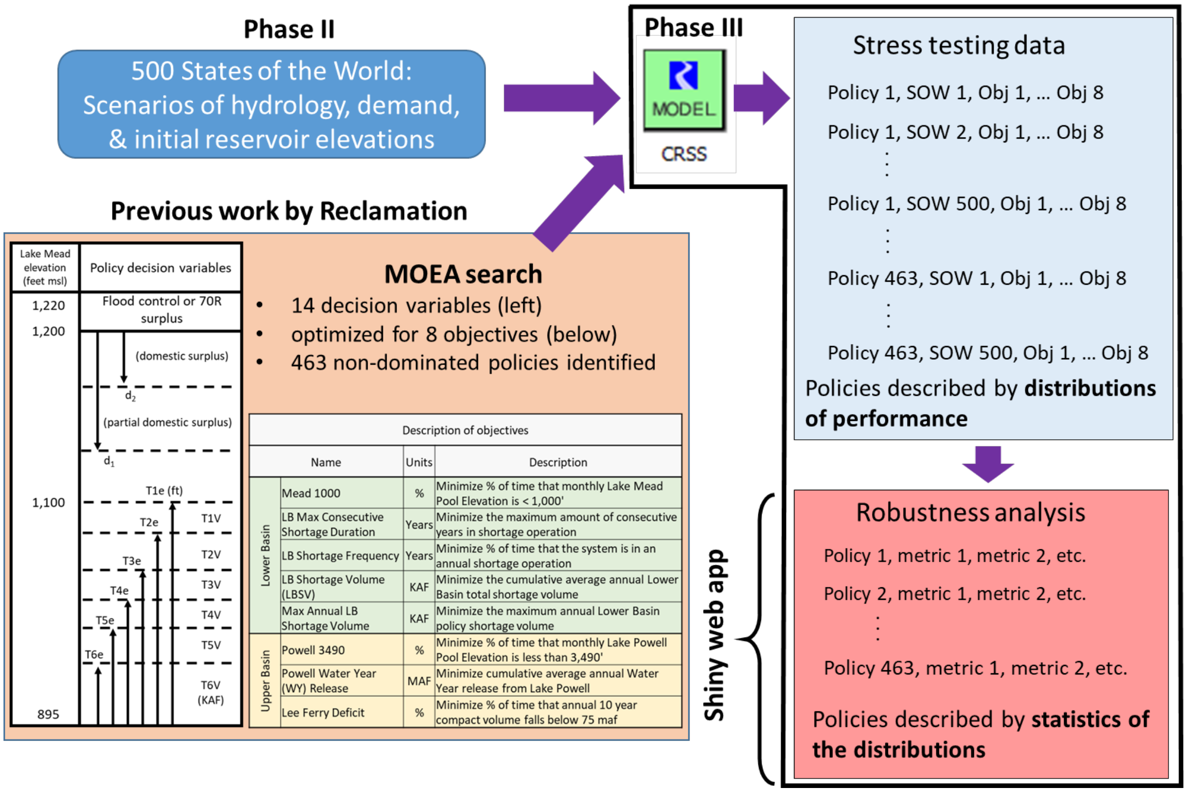


*Figure 1:* Overview of the methods. Reclamation has provided 463 Lake Mead policies discovered with MOEA search. In Phase II of the project, we established methods for creating a 500-member SOW ensemble for stress testing the policies. Phase III begins by simulating the policies in the SOW ensemble to create stress testing data. Lastly, we create a web application to visualize policy robustness.

## 2.1. Chosen state of the world ensemble

| SOW ensemble design choices | | |
|---|---|---|
| **Sampling algorithm:** | conditioned Latin Hypercube Sampling (cLHS) | |
| **Ensemble size:** | 500 SOW | |
| **Uncertainty metrics** | | |
| **Metric name** | **Description** | **Range** |
| Hydrology | Hydrology traces are sampled from Observed Resampled, Global Climate Model (GCM), Direct Paleo Natural Flow (DPNF), and Paleo Conditioned Natural Flow (PCNF) ensembles based on 48 annual flow statistics at Lees Ferry. | NA |
| Demand | total annual Upper Basin demand, minus AZ's upper basin apportionment | 4.2 to 6.0 MAF |
| Powell.PE | Lake Powell pool elevation at initialization of model run (feet above mean sea level) | 3450 to 3675 feet |
| Mead.PE | Lake Mead pool elevation at initialization of model run (feet above mean sea level) | 1000 to 1185 feet |

*Table 1:* Description of the SOW ensemble. Our SOW ensemble is created with the non-uniform version of cLHS, and samples uncertainty in hydrology, demand, and initial reservoir pool elevations. Note that MAF stands for million acre-feet.

The SOW ensemble samples plausible future scenarios of hydrology, Upper Basin demand, and pool elevations in Lake Mead and Lake Powell at the beginning of CRSS simulation. Table 1 summarizes the SOW design choices and uncertainty metrics used. We use an ensemble of 500 SOW sampled with the non-uniform version of conditioned Latin Hypercube Sampling (Minasny and McBratney, 2006, 2010). This results in a non-uniform distribution of hydrology scenarios and uniform distributions of demand and reservoir pool elevations. For details on the sampling method and sampling ranges, please refer to the Phase II report. Note that the lower limit on demand has been increased from 4.0 million-acre feet (MAF) in the Phase II report to 4.2 MAF based on feedback from Reclamation.

### 2.2. Implementation of SOW and policies in CRSS and RiverSmart

In order to implement the input of the SOW ensemble into CRSS for robustness analysis, we made several modifications to the model, which makes it deviate slightly from the model used by Reclamation for other purposes. We made these modifications in consultation with our Reclamation collaborators, who have seen detailed PowerPoint presentations on the matter. These changes are summarized briefly in this section.

The SOW ensemble includes uncertainties that have not been previously modelled in MOEA-CRSS optimization or robustness simulations, so we developed several Data Management Interfaces (DMIs), Initialization Rules (IR), object slots, and data objects to load a new SOW into CRSS for each robustness simulation. Lake Mead and Lake Powell pool elevations are initialized with a trace directory DMI that sets a scalar slot in the Mead and Powell reservoir objects. IR then transfer this value to the pool elevation series slots in the initialization time step. Hydrology traces are loaded via a single trace directory DMI. For a single DMI to handle DPNF, PCNF, GCM, and Observed Resampled datasets without causing warning messages in CRSS, dummy .rdf files were added to PCNF and DPNF trace folders that set climate-related slots to zero. Demand scenarios are created by linearly scaling the 2020 monthly demand values from the Current Trends projection in the 2012 Basin Study (Bureau of Reclamation, 2012). The 2020 monthly demand values are repeated from 2016 to 2060 and saved in new data objects as the baseline demand scenario. In another data object, the total UB demand value, from the SOW ensemble, is loaded, and a scaling coefficient is calculated. Lastly, IR set the original demand schedule objects by multiplying the baseline demand data objects by the scaling coefficient. Preexisting IR then transfer the demand schedules from the data objects to the water user objects. We have presented the CRSS development to Reclamation and have provided documentation.

We configured all DMIs in a single Multiple Run Manager (MRM) to accommodate straightforward scenario creation in RiverSmart. In our RiverSmart study, there is one event each for the MRM, the CRSS model file, the ruleset, and the Lake Mead DV. Generating scenarios results in 500 SOW x 463 policies = 231,500 scenarios, where each scenario is one model run. For details on the CRSS DMI and the RiverSmart event that handles Lake Mead DV, we refer the reader to Alexander 2018.

### 2.3. Robustness metrics

After simulating policies with the SOW ensemble, we calculate eight classes of robustness metrics. Table 2 describes the metrics according to the taxonomy established in McPhail et al. (2018). Performance value transformation describes how, or if, the data from simulations undergoes a mathematical transformation. Examples include regret or a binary transformation (e.g. meets or fails a performance threshold). Scenario subset selection indicates which SOW are used. For example, the satisficing metric considers all SOW, while maximin uses only the worst-case SOW. Lastly, the robustness metric calculation describes the statistic used to summarize the distribution of performance for a given policy as a scalar value. In the case where only one SOW is used in the calculation, no statistical function is needed (e.g. percent deviation and maximin). The remainder of metrics in this research take the average across the SOW ensemble although higher-order moments can be used (McPhail et al., 2018). For more information on robustness metrics and example calculations, we refer the reader to the PDF documentation on the *For Reference* page of the app (see Section 2.4).

The satisficing and satisficing deviation metrics require user-defined performance thresholds on performance objectives (see figure 1 for a brief description of the objectives). In practice, the thresholds indicate a hard delineation between acceptable and unacceptable performance. In the app, discussed in section 2.4, we have implemented the ability for the user to change the thresholds for each objective as they see fit. For demonstration purposes, this report uses the same thresholds as the original Alexander 2018 MORDM study. The thresholds are: Mead 1000 < 10%, Powell 3490 < 5%, and Lower Basin Shortage Volume < 600 thousand acre-feet (KAF).

| Metric | Reference | Performance value transformation | Scenario subset selection | Robustness metric calculation |
|---|---|---|---|---|
| satisficing (fraction) | Star (1963) and Schneller and Sphicas (1983)* | binary satisfaction of thresholds | all | mean |
| satisficing deviation | original to Phase III analysis: similar to regret type A from Alexander 2018 | regret from satisficing threshold divided by threshold | all | mean |
| regret from best | similar to Alexander 2018 | regret from best divided by maximum deviation | all | mean |
| percent deviation | Kasprzyk et al. (2013) | 90th percentile deviation divided by baseline performance | 90th percentile | identity |
| Laplace's principle of insufficient reason | Laplace and Simon (1951)* | identity | all | mean |
| Hurwicz optimisim-pessimism rule | Hurwicz (1953)* | identity | best and worst-case | mean |
| mean-variance | Hamarat et al. (2014)* | identity | all | mean x standard deviation** |
| maximin | Wald (1950)* | identity | worst-case | identity |

*Table 2:* Descriptions of robustness metrics using the taxonomy of McPhail et al. 2018. *reference taken directly from McPhail et al. (2018) **mean x standard deviation is used in the case of a minimization objective. In the case of maximization, mean/standard deviation is used, but in this research all objectives are minimized.

## 2.4. CRB robustness tradeoffs: a web app for interactive visualization and filtering

### 2.4.1. Language and implementation

Our goal is to create an interactive tool to visualize robustness tradeoffs, DV, and facilitate decision-making-related conversation. Further, we desire such a tool to be cost-effective (or free, ideally), intuitive to operate, and require no specialized software programs or operating systems. Lastly, Reclamation should be able to adapt, update, and share the tool as they see fit. Therefore, we built a web app with the R language while leveraging several packages to provide reactivity, intuitive web page layout, and powerful visualizations (R Core Team, 2021).

The app depends on Shiny, the linchpin package for building web apps in R. Shiny provides functions and data objects for reactive programming, which enables the linking of plots and pages plus response to user input (Chang et al., 2021). After building an app in Shiny, it can be deployed on the shinyapps.io server, then accessible to anyone with internet connection ("shinyapps.io," n.d.). Therefore, the CRB robustness tradeoffs app is built in R, depends on Shiny, and is currently deployed on shinyapps.io. It can be accessed through this link: https://nabocrb.shinyapps.io/CRB-Robustness-App-PhaseIII-Report/.

### 2.4.2. Design philosophy and overview

A screen capture of the app is shown in Figure 2. Throughout this report, we use the term "page" to indicate tabs on the blue ribbon and "tab" to indicate tabs within pages. To follow along with the analysis shown in the report, we recommend opening the app using the link above and navigating the pages, tabs, and buttons discussed below. Note that there is ongoing development of the app, so there may be new features added after the completion of this report.

The app is designed as a self-contained robustness analysis tool. The *For Reference* page includes Figure 1, providing important background information on methods and descriptions of performance objectives and DV. Moreover, this page includes a non-technical review of robustness metrics that can be downloaded and referenced as the user performs an analysis.

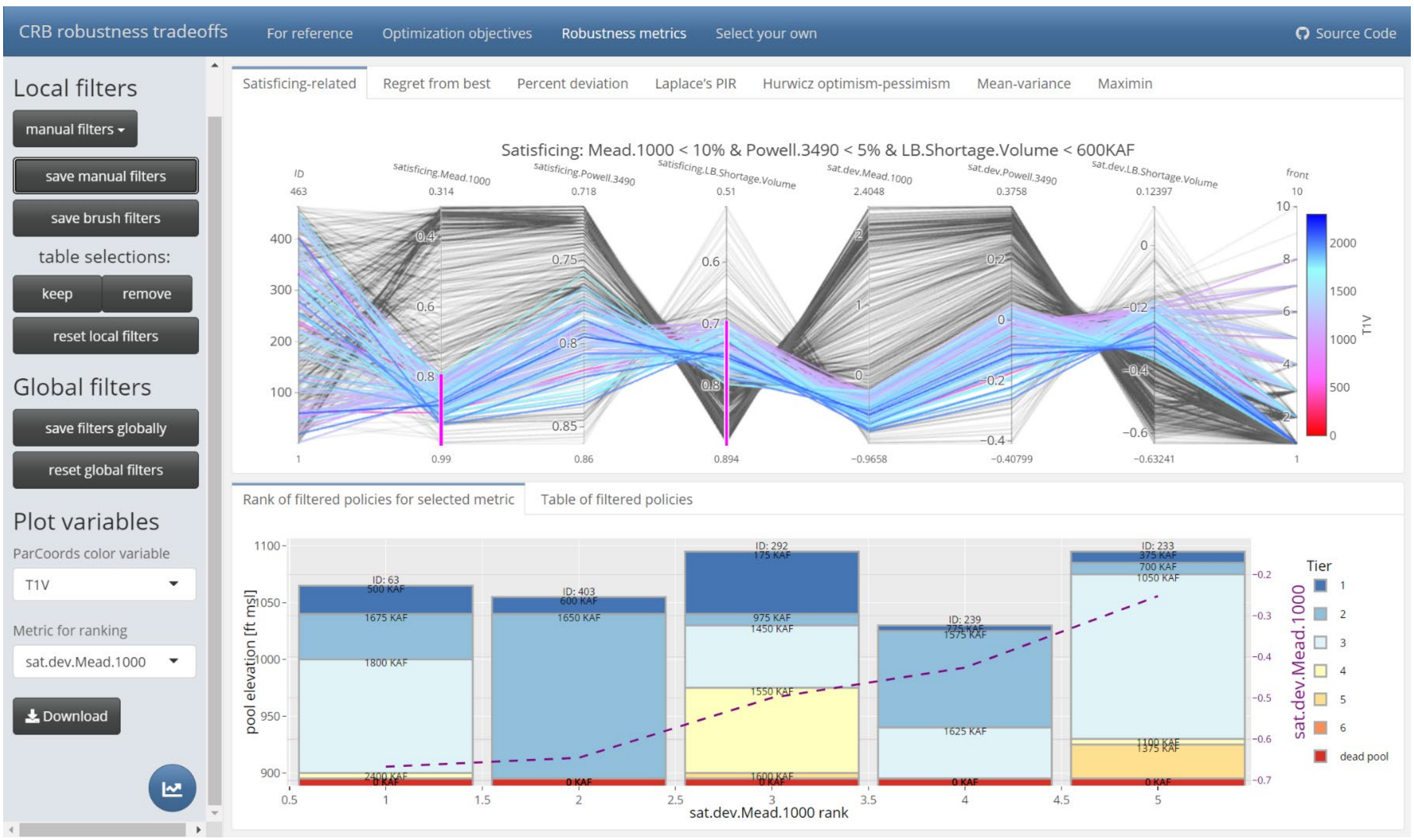

*Figure 2*: A screen capture of the CRB robustness tradeoffs web app. The app provides interactive visualization and filtering through the linking of parallel coordinate plots (top) and decision variable plots (bottom). The left sidebar provides multiple options for filtering policies, and multiple pages can be linked with global filters to successively filter by performance objectives, robustness metrics, and decision variables.

Three pages illustrate the performance, robustness, and DV of policies. The *Optimization objectives* page shows the results of the MOEA optimization used to obtain the archive of 463 policies. The *Robustness metrics* page reports the results of calculating the robustness metrics in Table 2. Lastly, the *Select your own* page empowers the user to visualize a combination of optimization objectives, robustness metrics, and DV of choice.

In each of these three pages, the user simultaneously views parallel coordinate (PC) plots of performance (objectives or robustness) and stacked histograms of DV. In the PC (top plot in Figure 2), each line represents a policy, and where the line crosses an axis is the performance of the policy for that objective or robustness metric (Inselberg, 2009). We designed the PC such that the desired direction for objectives and robustness is always downward, so the ideal policy would be a straight line across the bottom (assuming the axis spanned the full range of possible values). The only exception is that, in the *Select your own* page, DV can be added to the PC, in which case the direction is simply the magnitude of the DV and thus neither the downward or upward direction indicates better performance. The color of the lines map to an objective, robustness metric, or DV selected by the user under the "Plot options" section in the left sidebar.

Each vertical bar in the DV plot (bottom plot of Figure 2) describes shortage elevations and volumes of a policy. The y axis on the left is Lake Mead pool elevation, so the colored subsections of each bar depict the shortage elevations (T1e-T6e in Figure 1) and corresponding shortage volumes (T1V-T6V in Figure 1). The x axis is the rank of policies according to an objective, robustness metric, or DV selected under the "Plot options" in the left sidebar, where rank one is the best. In addition to rank, the y axis on the right shows the magnitude, which maps to the dashed purple line superimposed on the vertical bars. We include magnitude such that change in performance between similarly ranked policies is apparent. We revisit this point in Section 3.5.

### 2.4.3. Filtering options

The user has multiple mechanisms hosted in the left sidebar to filter policies by optimization objectives, robustness metrics, or DV (see Figure 2). Recall the DV represent decisions that define a solution, whereas optimization objectives and robustness metrics quantify performance of solutions. First, policies can be filtered manually by using the "Manual filters" drop down button, meaning the user can select from objectives, robustness metrics, or DV, then choose an inequality operator, and lastly type in a threshold. Second, policies can be filtered by interactively brushing on the PC plot then selecting "Save brush filters". Brushing is performed by dragging and dropping the mouse over a range on an axis, where the brushed region is highlighted in pink. Policies outside of the brushed range(s) are shown in light gray. Lastly, policies can be filtered by table selections. The user can select the "Table of filtered policies" tab above the DV plot to reveal an interactive data table of policies that can be sorted or searched. By clicking on a row in the table, which represents a policy, the corresponding policy in the PC plot will be highlighted. Then, other policies can be filtered out with the "keep table selections" button, or the policy can be removed with the "remove table selections" button. As filters are applied, the DV plot is updated to show only policies meeting the filter criteria. We refer to this connectivity between the PC and DV plot as "linking" (Kollat and Reed, 2007). Moreover, the user has the option within the "global plot options" page to either keep or remove policies from the PC plot that do not meet filter criteria ("global plot options" are accessed by the circle button to the right of the "Download" button). For example, the user could filter out policies with tier 1 elevation greater than 1080 feet mean sea level using a manual filter. Then, remaining policies could be filtered again with a brush filter on an objective such as Mead 1000. Lastly, the user could use table selections to remove all but a select few policies.

The user also has the option to link pages. Initially, all filters are saved "locally", meaning those filters are not enforced on other pages. But, local filters can also be saved "globally" by hitting the "save filters globally" button, meaning those filters are enforced on all pages. For example, policies could first be filtered according to objectives on the *Optimization objectives* page. After saving filters globally, the policies meeting optimization objectives criteria can be interrogated further according to robustness metrics on the *Robustness metrics* or *Select your own* pages, with policies that are no longer of interest removed from the plots. By linking the PC, the DV plot, and multiple pages, the app establishes an efficient and intuitive mechanism for exploring copious metrics while narrowing the scope of policies under consideration.

This app utilizes an additional tool for exploring metrics and filtering policies, namely Pareto-dominance. Pareto dominance is the concept used in place of optimality in the case where tradeoffs exist in a multi-objective optimization problem (Hadka and Reed, 2013). We define Pareto-dominance below. Consider two policies, $\boldsymbol{u}$ and $\boldsymbol{v}$, which represent vectors of $i$ DV. Then, $\boldsymbol{F(u)}$ and $\boldsymbol{F(v)}$ are their respective objective vectors of length $M$.

**Definition**

Assuming minimization of all $M$ objectives, policy $\boldsymbol{u} = (u_1, u_2, \dots, u_i)$ Pareto-dominates policy $\boldsymbol{v} = (v_1, v_2, \dots, v_i)$ if and only if for $\forall m \in \{1,2,\dots,M\}$, $F(u_m) \leq F(v_m)$ and $\exists j \in \{1,2,\dots,M\}$ such that $F(u_j) < F(v_j)$. In words, policy $\boldsymbol{u}$ dominates policy $\boldsymbol{v}$ if policy $\boldsymbol{u}$ is better or equal in all objectives compared to policy $\boldsymbol{v}$ while being better in at least one objective.

For example, the MOEA optimization performed previously by Reclamation (see Figure 1), utilizes Pareto-dominance. Policy vectors have length of 14 DV, which includes shortage tier elevations, shortage tier volumes, and distances to define surplus operations. Objective vectors have length of eight, including five Lower Basin (LB) objectives and three Upper Basin (UB) objectives. However, Pareto-dominance can be useful in applications besides multi-objective optimization. In the app, we use Pareto-dominance as an alternative to aggregating multiple robustness metrics. In other words, our policy vectors are the same as in the MOEA optimization, but the performance vectors now consist of robustness metrics.

The Pareto-dominance operator identifies "fronts". A policy belonging to front 1 is non-dominated, meaning no policy exists (among our 463 solutions) that Pareto-dominates it. A policy in front 2 is dominated by at least one policy in front 1. We illustrate an example with PC in Figure 3. Policy 124 (blue line) is dominated by policy 352 (red) with respect to two satisficing metrics. This is shown by policy 352 lying below policy 124 on the satisficing axes. Therefore, policy 124 belongs to a higher (inferior) front than policy 352. However, note policy 256, which also belongs to front 1. A tradeoff exists between the satisficing metrics between policy 256 and policies 352/124 (shown by crossing lines). However, policy 256 belongs to front 1 because no other policy dominates it. We calculated Pareto-dominance fronts for all classes of robustness metrics in Table 2, and they can be added as an axis in their respective PC using the "global plot options" button. Moreover, the user can calculate fronts after selecting their own combinations of objectives and robustness metrics under "Plot variables" on the *Select your own page* (not shown in Figure 2). Fronts provide an alternative mechanism to aggregation for evaluating robustness, and serves as an additional metric by which policies can be filtered.

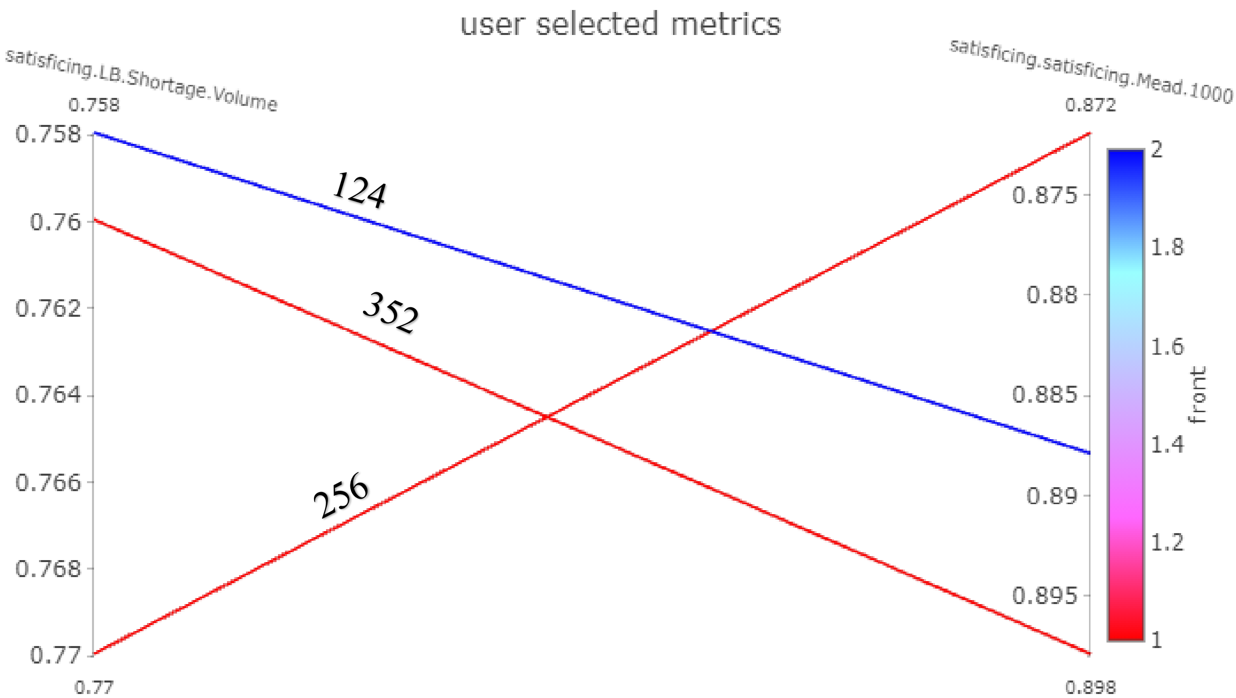


*Figure 3:* An example of Pareto-dominance fronts using two satisficing robustness metrics. In this parallel coordinates plot, each line represents one policy, and where the line crosses an axis is its values for that metric. Policy 124 is in front two because it is dominated by policy 352. Policy 256 belongs to front 1 because no policy is better in at least one of the satisficing metrics while being equal or better in the other. Pareto-dominance fronts are an alternative to aggregating metrics and another option for filtering policies.

#### 2.4.4. App summary

Overall, the app supports exploration of dozens of performance measures while empowering the user to efficiently reduce the number of policies of interest by using multiple filtering options. Behind this design philosophy is the idea that it is difficult to know which robustness metric(s) is (are) important for the problem, and instead the selection should be informed by interactive exploration. This claim will be revisited in section 3.4.

## 3. Example illustrative robustness analysis

This section details an example robustness analysis that highlights the utility of the app, demonstrates a potential workflow for selecting robust policies, and emphasizes how *a posteriori* robustness metric exploration can yield surprising insight into policy robustness.

.

### 3.1. Optimization objective thresholds

First, we filter policies on the *Optimization objectives* page according to the performance thresholds identified in Alexander 2018 (see section 2.3). Figure 4 shows these thresholds applied as brush filters, which means policies not meeting the thresholds are shown in light gray. This reduces the archive from 463 to 43 policies. We save these brushes locally then globally to explore them further with robustness metrics.

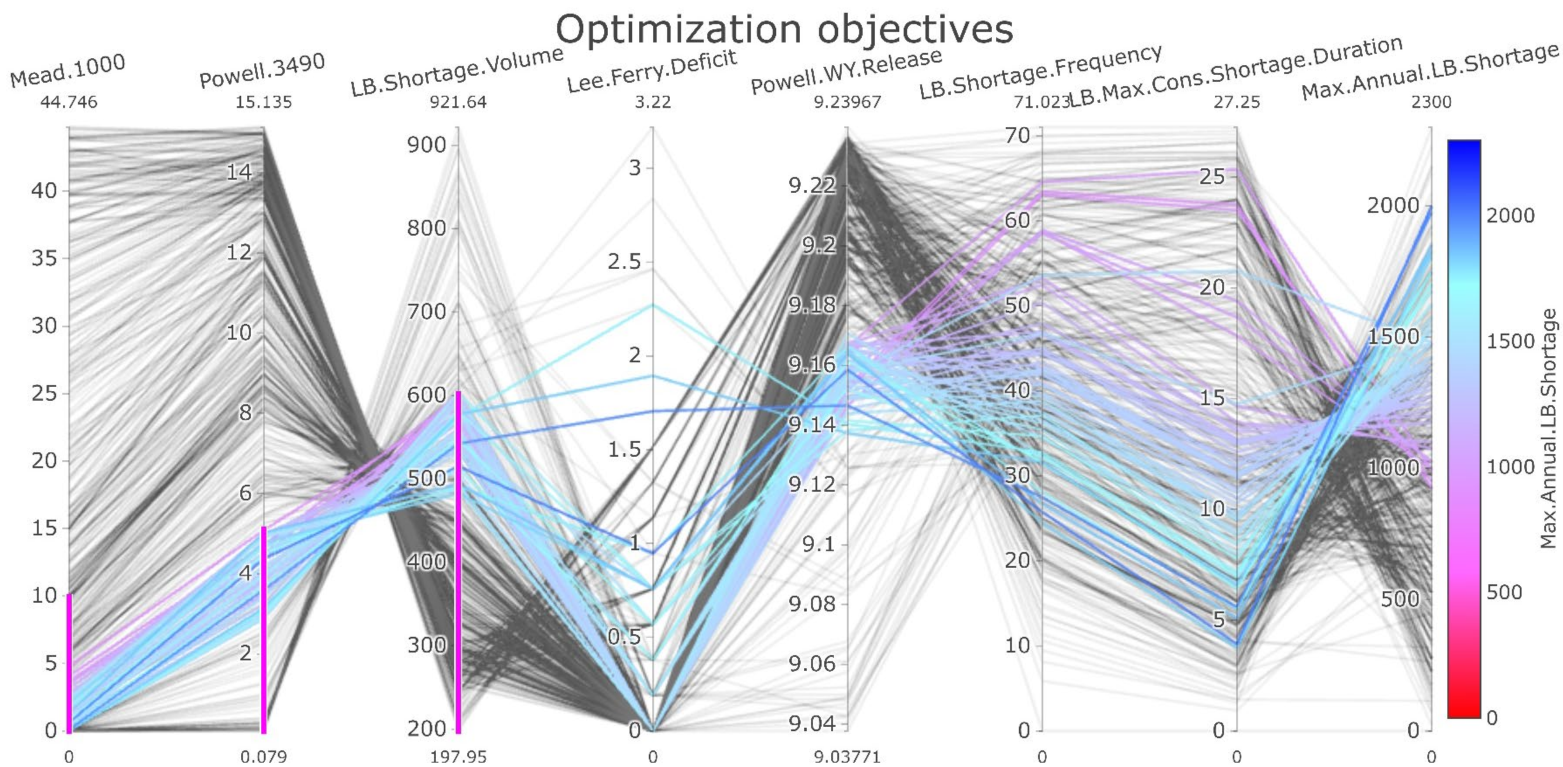


*Figure 4*: Using brushing to filter policies by thresholds on optimization objectives. Policies failing to meet the thresholds of Mead 1000 < 10%, Powell 3490 < 5%, and LB Shortage Volume < 600 KAF are removed. The color shows the max annual LB shortage. For units and descriptions of objectives, see Figure 1. 463 policies are reduced to 43.

### 3.2. Satisficing robustness thresholds

After saving the 43 policies globally, we interrogate them further according to satisficing metrics because performance thresholds have been identified. From Figure 5, we see that all policies remaining after application of optimization objectives filters have satisficing Mead 1000 and satisficing Powell 3490 exceeding 0.7. However, policies remain with satisficing Lower Basin Shortage Volume (LBSV) less than 0.7, so we remove those policies with a brush filter. Doing so reduces the number of policies from 43 to 37.

### 3.3. Selecting policies in the non-dominated front

The optimization objectives PC (Figure 4) shows that the three objectives used in satisficing metrics are in conflict with other objectives (shown by crossing lines between these objectives/robustness metrics). Further, the other LB objectives (LB shortage frequency, Max annual LB shortage, LB max consecutive shortage duration) have large ranges while the other UB objectives (Lee Ferry deficit, Powell water year release) do not. Therefore, we use the *Select your own* page to create a PC plot with all three satisficing metrics plus Laplace LB max shortage duration and Laplace max annual LB shortage (Figure 6, top). We exclude LB max consecutive

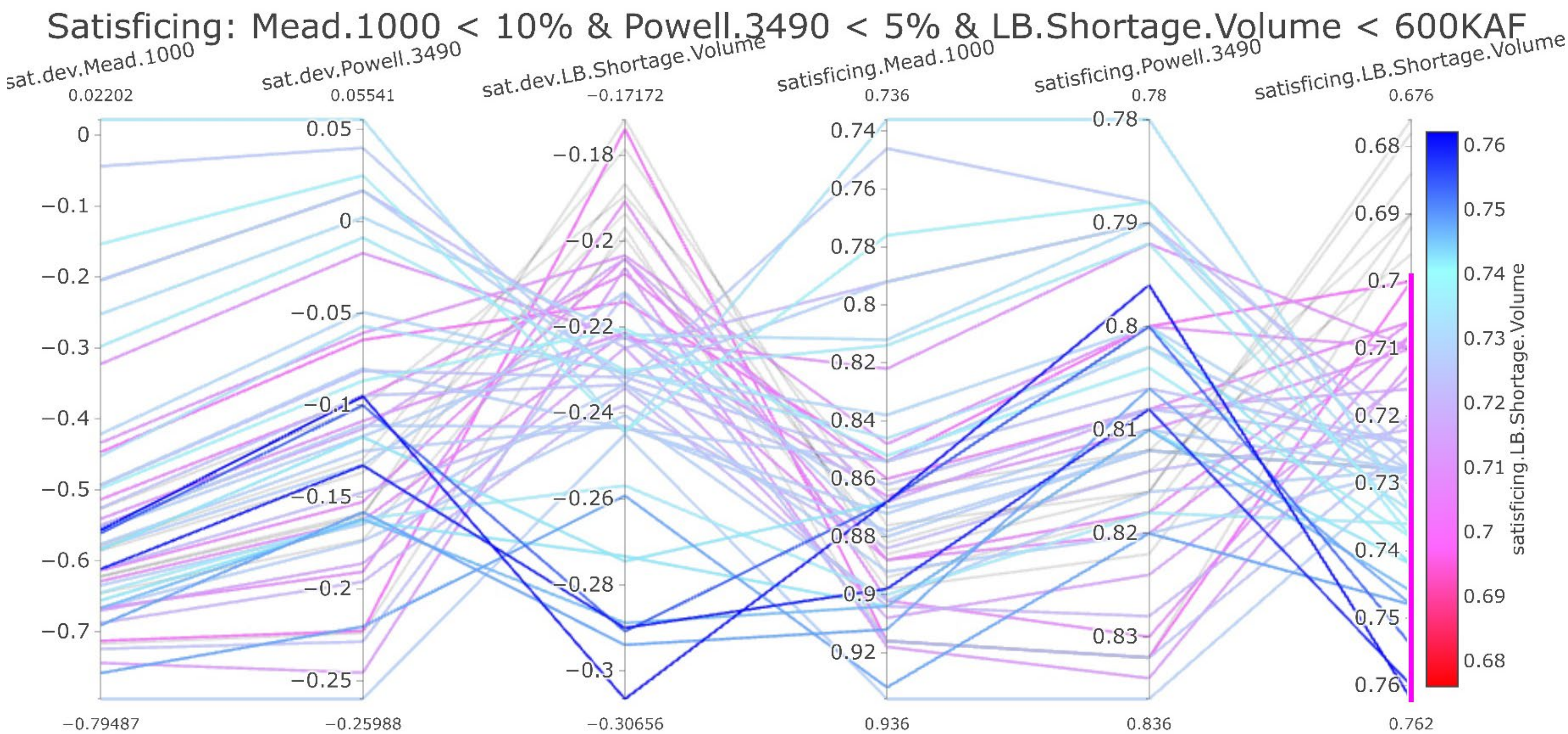


*Figure 5*: Applying a second round of filters using satisficing robustness metrics. After applying filters with optimization objective thresholds, the remaining policies demonstrate satisficing greater than 0.7 for both Mead 1000 and Powell 3490 objectives, but this is not true for Lower Basin Shortage Volume (LBSV). Therefore, we remove policies not meeting this threshold, shown as the traces in light gray. Note that satisficing indicates the fraction of SOW where performance thresholds are met. For example, satisficing.Mead.1000 = 0.75 means Mead 1000 is less than 10% in 75% of SOW. 43 policies are reduced to 37.

shortage duration because both the optimization objectives and Laplace PC figures suggest these objectives are harmonious (shown by lack of crossing lines). We calculate Pareto-dominated fronts within the app, then filter to front 1 (Figure 6, bottom). From Figure 6, we notice that three policies have notably greater Laplace Max annual LB shortage, so we apply a brush filter of < 1800 KAF. Ample policies meet these filters, so we further filter the policies by Laplace LB max consecutive shortage duration < 10 years. These filters reduce the archive of policies from 37 to 17.

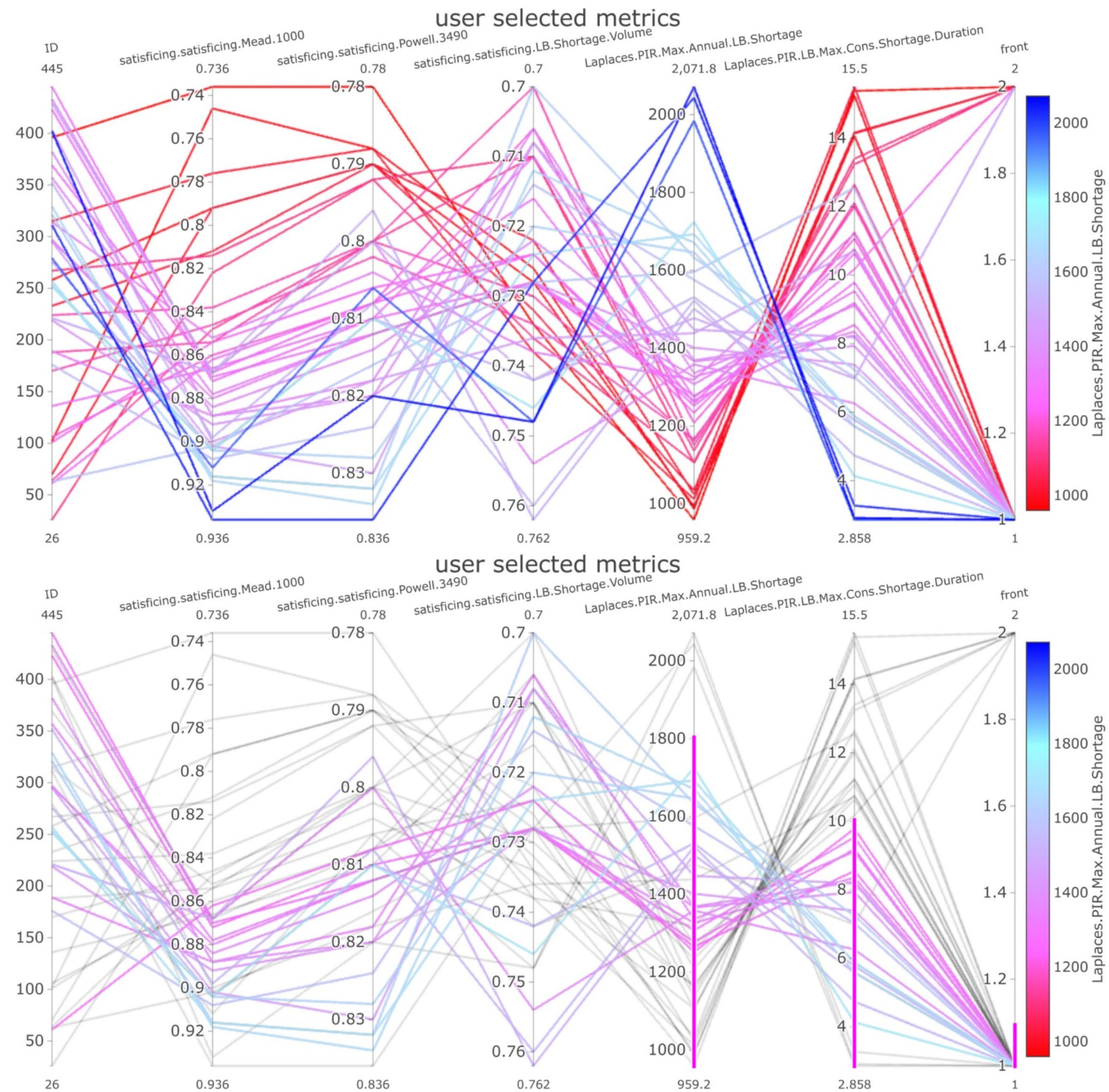


*Figure 6:* A third round of filtering with user-selected metrics and Pareto-dominance fronts. We create a custom parallel coordinates plot with the satisficing metrics plus Laplace's Principle of Insufficient Reason for two LB shortage objectives that demonstrated tradeoffs with the objectives used in the satisficing metrics (top). After calculating Pareto-fronts, we remove policies outside of the non-dominated front. Further, we exclude policies with Laplace max annual LB shortage greater than 1800 KAF and max consecutive shortage duration greater than 10 years (bottom). 37 policies are reduced to 17.

### 3.4. Robustness metric exploration: identifying surprising robustness results

Thus far, we have filtered policies using optimization objectives, satisficing robustness, and Laplace's Principle of Insufficient Reason robustness. Further, we recommend checking the robustness of policies according to other metrics; doing so has the potential to identify surprising robustness results that could influence which policies are included for further discussion or analysis. For example, we found that five policies contained in the global filter, although they met all previous filters, are some of the least robust according to the maximin metric for Max annual LB Shortage (see annotation in Figure 7, top). This means these policies exhibited some of the most severe shortages under their worst-case SOW. Therefore, we filter out these policies (Figure 7, bottom), dropping from 17 to 13 policies.

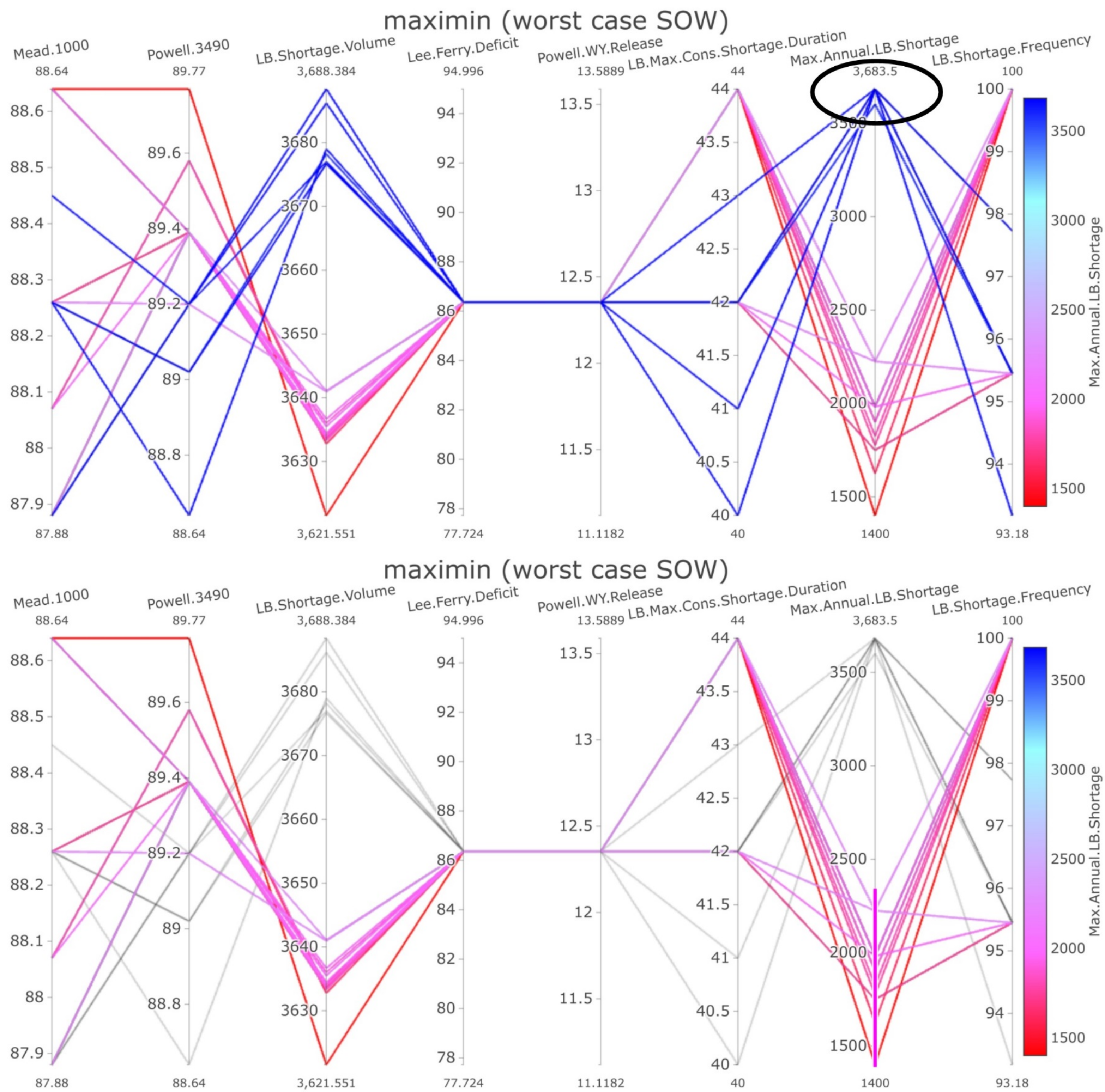


*Figure 7:* Removing policies with poor performance in worst-case SOW. After filtering by objectives, satisficing metrics, and Laplace metrics, we found through posterior exploration five of the remaining 17 policies demonstrated near the worst performance in max annual LB shortage under their worst-case SOW out of the entire archive of 463 policies (circled in top plot). We remove those policies from further consideration (bottom plot), with 13 remaining.

### 3.5. Decision variables of remaining policies

As we apply successive filters, the DV plot is continually updating to reflect the remaining policies. Figure 8 shows the remaining 13 policies ranked by maximin Max annual LB shortage. These policies enforce their first shortages starting at Lake Mead pool elevations between 1035 and 1065 feet. Further, the first shortage volumes are large, ranging from 1200 to 1825 KAF. However, the maximum shortage volume for all policies is less than 2000 KAF. No clear pattern is observed in the number of shortage tiers, ranging from one to five.

Considering both the ranking and magnitude of maximin Max annual LB shortage reveals where change in ranking corresponds to significant gains or losses in performance. For example, performance decreases significantly from rank one to three (steep slope of purple line), but is similar from rank three to five (flat slope). Likewise, performance from rank nine to eleven are very similar, then decreases sharply to rank twelve. Considering magnitude can reveal additional polices for consideration in the case that the change in magnitude is relatively small for a series of ranked policies.

## 4. Discussion & Conclusion

### 4.1. A framework for Colorado River Basin robustness analysis

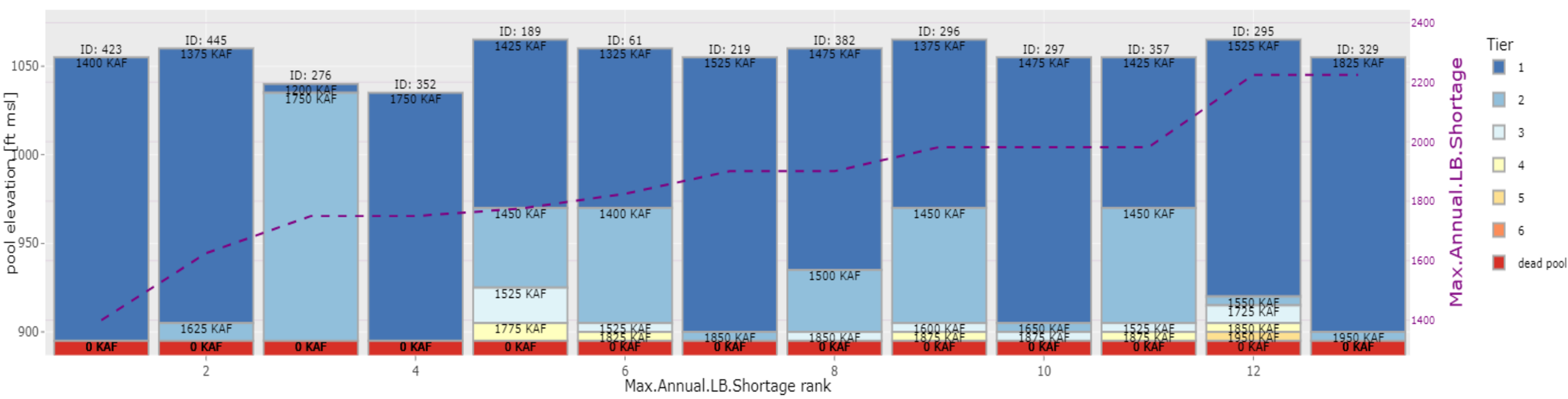


*Figure 8:* Decision variable plot of the remaining 13 policies. Each vertical bar represents one policy. The y axis on the left shows Lake Mead pool elevation, and the subsections represent shortage tiers, labeled with their respective shortage volumes. The x axis shows the rank of policies by maximin max annual LB shortage, and the dashed purple line and second y axis shows the magnitude

This research establishes a web application for Reclamation and/or CRB stakeholders to engage with multi-metric robustness analysis. The app is built in R, increasing the ability for Reclamation to adopt and adapt the app as desired. Further, the app is deployed in the shinyapps.io server, meaning it can be accessed by stakeholders through a web browser without specialized or proprietary software. The app utilizes parallel coordinates linked to DV plots to simultaneously view robustness and DV. Local and global filtering options enable the user to explore dozens of metrics while efficiently reducing the number of Lake Mead policies for consideration. Moreover, the example analysis identifies insightful and interpretable robustness metrics based on performance thresholds, then demonstrates how to use the app to identify other metrics of interest *a posteriori* through exploration. Indeed, the analysis identified four policies that fulfilled filter criteria in optimization objectives, satisficing robustness, and Laplace's PIR robustness, yet were some of the least robust according to performance in worst-case SOW.

### 4.2. Contribution to robustness analysis literature

This app is a novel contribution to decision science, in particular robustness analysis. To our knowledge, it is the only such interactive visualization application to provide the user with simultaneous exploration of a large library of robustness metrics. We demonstrate that a large solution set can be filtered to a manageable subset without aggregating multiple objectives or robustness metrics into a single metric. In other words, this app establishes a method to explicitly explore copious robustness tradeoffs while still arriving at a parsimonious quantity of policies for decision-making-related discussion. We accomplish this via the linking of PC plots of performance/robustness and DV plots with manual, brush, and table filtering combined with the option to globally link filters across multiple classes of metrics. Moreover, we provide the user with an additional tool for filtering polices by robustness, namely Pareto-dominance fronts. We believe this to be the first such application in a robustness analysis. We demonstrated the utility of such a tool by subsetting 463 Lake Mead policies to 13. If Reclamation or stakeholders wanted a smaller subset, the policies can be filtered further with additional robustness metrics or by harsher thresholds. Another option is to apply DV filters. For example, applying the constraint of T1V < 1400 KAF reduces 13 policies to four.

## 5. Future work

We are collaborating with Reclamation to select policies for use in Phase IV of this research. Together, we will identify objectives and thresholds in the case where satisficing metrics are used. Further, we will continue development of the app, incorporating feedback from joint sessions where our research team and Reclamation will thoroughly review the app and interrogate policies. In Phase IV, Scenario Discovery analysis, we will use statistical data mining algorithms to classify the characteristics of uncertainty that cause policies to fail performance thresholds. The analysis will facilitate exploration of all policies, with reported results focusing on a small set of policies we are in the process of selecting through collaboration with Reclamation.

## References

Alexander, E., 2018. Searching for A Robust Operation of Lake Mead (M.S.). University of Colorado, Boulder, CO.

Bureau of Reclamation, 2012. Colorado River Basin Water Supply and Demand Study (Summary Report). Bureau of Reclamation.

Hadka, D., Reed, P., 2013. Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework. Evol. Comput. 21, 231–259. https://doi.org/10.1162/EVCO_a_00075

Hamarat, C., Kwakkel, J.H., Pruyt, E., Loonen, E.T., 2014. An exploratory approach for adaptive policymaking by using multi-objective robust optimization. Simul. Model. Pract. Theory 46, 25–39. https://doi.org/10.1016/j.simpat.2014.02.008

Hurwicz, L., 1953. Optimality criterion for decision making under ignorance. Uncertain. Expect. Econ. Essays Honour GLS Shackle.

Inselberg, A., 2009. Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer Science & Business Media.

Kasprzyk, J.R., Nataraj, S., Reed, P.M., Lempert, R.J., 2013. Many objective robust decision making for complex environmental systems undergoing change. Environ. Model. Softw. 42, 55–71. https://doi.org/10.1016/j.envsoft.2012.12.007

Kollat, J.B., Reed, P., 2007. A framework for Visually Interactive Decision-making and Design using Evolutionary Multi-objective Optimization (VIDEO). Environ. Model. Softw. 22, 1691–1704. https://doi.org/10.1016/j.envsoft.2007.02.001

McPhail, C., Maier, H.R., Kwakkel, J.H., Giuliani, M., Castelletti, A., Westra, S., 2018. Robustness Metrics: How Are They Calculated, When Should They Be Used and Why Do They Give Different Results? Earths Future 6, 169–191. https://doi.org/10.1002/2017EF000649

Minasny, B., McBratney, A.B., 2010. Conditioned Latin Hypercube Sampling for Calibrating Soil Sensor Data to Soil Properties, in: Viscarra Rossel, R.A., McBratney, Alex B., Minasny, Budiman (Eds.), Proximal Soil Sensing. Springer Netherlands, Dordrecht, pp. 111–119. https://doi.org/10.1007/978-90-481-8859-8_9

Minasny, B., McBratney, A.B., 2006. A conditioned Latin hypercube method for sampling in the presence of ancillary information. Comput. Geosci. 32, 1378–1388. https://doi.org/10.1016/j.cageo.2005.12.009

Schneller, G.O., Sphicas, G.P., 1983. Decision making under uncertainty: Starr's Domain criterion. Theory Decis. 15, 321–336. https://doi.org/10.1007/BF00162111

Simon, P., Laplace, M. de, 1950. A Philosophical Essay on Probabilities.

Starr, M.K., 1963. Starr's Product Design and Decision Theory: 1963 Edition. Prentice-Hall.

Wald, A., 1950. Statistical Decision Functions. New York; Chapman & Hall: London.