Empowering Participatory Decision Making Under Deep Uncertainty: Novel Algorithms and Interactive Tools applied to Colorado River Post-2026 Negotiations

A Dissertation Presented

By

Nathan Bonham

B.S. West Virginia University, 2019

M.S. University of Colorado Boulder, 2023

Submitted to the Graduate School of the

University of Colorado Boulder in partial fulfillment

of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Department of Civil, Architectural, and Environmental Engineering

University of Colorado Boulder

November 2023

ABSTRACT

Decision Making Under Deep Uncertainty (DMDU) is an emerging field of model-based decision support methods. Deep uncertainty exists when decision-makers disagree on how to prioritize conflicting performance goals and when there is no consensus on which assumptions to make about uncertain future conditions, such as climate and human population. DMDU tests policy alternatives in many plausible futures, prioritizes policies that perform well in many futures (robust policies), and discovers conditions causing poor performance (vulnerability analysis).

There is growing demand for participation by stakeholders in decision-making. In DMDU-based decision support, participation includes the modelling process since methodological decisions can have unexpected (and undesirable) impacts on policy recommendations. Participation requires that stakeholders and decision-makers can interpret complex relationships between policies, plausible futures, and system performance. Participation also requires that the analysis can be iterated based on their feedback.

The goal of this thesis is to address four barriers to participatory DMDU: large computing requirements, choosing robustness metrics considering tradeoffs, choosing policies considering conflicting goals, and choosing methods for vulnerability analysis that are interpretable for decision-making.

This thesis contributes novel algorithms and interactive tools to address these barriers. Chapter 2 contributes a framework to test the number of futures required to accurately identify the most robust policies compared to testing more futures, potentially reducing computing requirements. Chapter 3 contributes a framework for *a posteriori* choosing of robustness metrics, which enables stakeholders to refine their choice of robustness metrics after seeing robustness tradeoffs. Chapter 4 uses the Self-Organizing Map, a machine learning method, to create a negotiation framework that helps decision-

makers identify compromise policies. Chapter 5 contributes a review of vulnerability analysis methods and identifies best practices for choosing interpretable methods.

The research contributions are demonstrated using a case study of reservoir operation policy in the Colorado River Basin (CRB). Since 2000, extended periods of low inflow have depleted storage in Lakes Mead and Powell, threatening hydropower infrastructure and water deliveries. Current policies expire in 2026, thereafter a new policy takes effect. This research evaluates a set of Lake Mead policies in 500 futures of streamflow and demand conditions. The robustness of these policies is discussed in Chapters 2-4, and Chapter 5 uses the CRB to illustrate purposes and methods for vulnerability analysis.

DEDICATION

For my family, and for my Lord and Savior, Jesus Christ.

"In God we trust. All others must bring data."

- Dr. W. Edwards Deming: CU Boulder alumni (1925) known for his leadership in Japan's remarkable post-WWII economic recovery

ACKNOWLEDGEMENTS

I want to thank my advisors, Drs. Joseph Kasprzyk and Edith Zagona. They have spent numerous hours brainstorming with me on things like methods, titles, presentation slides, and word choice. Your joint creativity, work ethic, and commitment to actionable science are inspiring and clearly impactful. I am honored to have been advised by you, especially in the years preceding Colorado River post-2026 negotiations. Because of your mentorship, I am sincerely hopeful this research will be beneficial for the Colorado River and other water resources systems.

Next, I want to thank my committee members Drs. Jon Herman, Balaji Rajagopalan, and Rebecca Smith. Two of Jon's papers have been particularly influential, one helping me grasp the foundations of DMDU (Herman et al. 2015) and the other being a key motivator of Chapter 3 (Herman et al. 2014). Balaji's graduate courses initiated the programming and statistics skills that contributed to every chapter of this thesis. Rebecca's leading efforts in adopting DMDU for Colorado River post-2026 negotiations have empowered much of this research, including funding, data, and research questions.

Many people with Reclamation have helped me in different capacities during my graduate school years. The Colorado River Basin Research and Modelling Team (Rebecca Smith, Alan Butler, Jim Prairie, Sarah Baker, Elliot Alexander, Carly Jerla, and others) provided Lake Mead policies, streamflow data, and feedback during the development of the robustness framework/web tool. Specifically, two years of this research were funded by the Lower Colorado Region. The motivation for Chapter 5 originated from an internship with the Technical Service Center (Marketa McGuire and Subhrendu Gangopadhyay) working on a project with Columbia Pacific Northwest (Michael Poulos and Jennifer Johnson).

Thank you to the staff at CADSWES. I learned how to use RiverWare from Mitch Clement and David Neumann. Jim Pasquotto has provided years of timely IT support. Kathryn Baack has saved me much headache by booking itineraries for conference travel. Thank you to Gwen Miller for keeping the CADSWES calendar and for many pleasant conversations before work.

I would also like to thank the National Science Foundation for funding through the Graduate Research Fellowship Program.

Lastly, I want to thank my wife, my parents, and my church community. Emily has helped me during the preparation of many presentations, so much so that she can discuss shortage and delivery tradeoffs in the Colorado River Basin and explain a Self-Organizing Map. My parents have done everything in their power to provide me with opportunities to succeed while modelling hard work and integrity. Friends and mentors from my church propped me up and kept me going during the hard times while reminding me to put my energy towards greater goals. Thank you all for your unwavering support.

TABLE OF CONTENTS

TABLES

FIGURES

1 Introduction

1.1 Motivation

By the end of the century, the global risk of water management disputes could increase by 40% (Farinosi *et al.*, 2018, p. 293). This risk is due in part to future imbalances between water supply and demand, driven by changing climate and growing populations (Reclamation, 2012a; Kasprzyk *et al.*, 2013; Herman *et al.*, 2014). Facing water supply deficits, it can be impossible for freshwater systems to provide the same benefits to stakeholders as historically expected. Decision-makers must then choose how to prioritize competing uses of the available supply, e.g., how much water to allocate for different populations and economic sectors (Jafino and Kwakkel, 2021).

Water resources systems are managed with policies to provide benefits to stakeholders. Policies refer to specific rules defined by values of decision levers (Lempert, Popper and Bankes, 2003). In river systems, for example, reservoirs are operated according to policies, and the decision levers determine levels of storage that trigger actions such as delivery reductions or flood releases (Alexander, 2018; Quinn *et al.*, 2018). The purpose of these policies is to ensure the sustainability of the system while providing benefits to stakeholders (e.g., water deliveries, hydropower). Due to complex policies and system behavior, it can be difficult to know the impacts of policies on stakeholder benefits. To provide decision support, policies are tested in simulation models. The model quantifies costs and benefits of the policy using performance metrics, such as the percent of time demands are met or average hydropower production (Alexander, 2018).

A challenge, however, is the lack of an objectively 'best' policy, i.e., no policy achieves maximum benefits for all stakeholders. Rather, policy alternatives are characterized by tradeoffs, meaning a policy that improves performance in one metric diminishes performance in another (Kasprzyk *et al.*, 2013; Herman *et al.*, 2014). For example, a reservoir policy could increase the maximum allowable storage to

augment water supply, but doing so increases flooding risk. Tradeoffs can inhibit decision-making, especially when decision-makers disagree on how best to prioritize conflicting goals (Wheeler *et al.*, 2018).

The benefits provided by the system also depend on factors beyond the control of decision-makers, such as future climate, population, and trade patterns (Alexander, 2018; IPCC, 2022; Yarlagadda *et al.*, 2023). In the context of long-term planning (e.g., 40 years into the future), the future state of these factors is unknown. To quantify the impacts of uncertain factors on performance, States of the World (SOW) can also be tested in the model, where a SOW is defined by values for the uncertain factors. Due to the complexities of climate and socio-economic changes, many plausible SOW may exist. Further, the likelihood of any SOW is often unknown or contended, meaning it is unclear which SOW to assume when measuring the performance of policies. Such conditions are known as *deep uncertainty* (Knight, 1921; Kasprzyk *et al.*, 2013).

There are emerging methods for modelling and data analysis to address these challenges, called Decision Making Under Deep Uncertainty (DMDU) (Herman *et al.*, 2015; Kwakkel and Haasnoot, 2019). Although there exist multiple DMDU frameworks, they have the following in common: testing policy performance in numerous SOW, prioritizing robust policies, and discovering vulnerabilities. To evaluate impacts of uncertain factors on system performance, policies are tested in hundreds to thousands of plausible SOW (McPhail *et al.*, 2018). The SOW are created using a statistical design of experiments to broadly sample values for the uncertain factors (Bryant and Lempert, 2010). Because the likelihood of any SOW is contended, policies are chosen based on their performance across the SOW ensemble (i.e., robustness), which is measured with robustness metrics (McPhail *et al.*, 2021). Then, vulnerability analysis uses factor mapping to discover the conditions (uncertain factors and their values) that lead to decision-relevant outcomes, such as the inability of a policy to achieve historically-expected benefits (Bryant and Lempert, 2010; Hadjimichael, Quinn, *et al.*, 2020). These conditions are communicated to decision-makers as scenarios, which can be used in a monitoring and adaptation system (Haasnoot *et al.*, 2013) or motivate

the search for more robust policies (Watson and Kasprzyk, 2017). These DMDU techniques are appropriate when many plausible SOW exist and their probabilities are unknown; case studies where many policy options exist; and when policy performance is difficult to anticipate without testing it in a model (Marchau *et al.*, 2019, chap. 1.4). This is mostly used for long planning horizons (i.e., multiple decades), but systems that are highly complex with multiple extenuating factors could benefit from DMDU techniques at shorter timescales.

In DMDU, input from stakeholders and decision-makers informs multiple methodological decisions and, ultimately, policy recommendations (Figure 1-1, a). For example, risk-tolerances inform the choice of robustness metric , and choice of robustness metric determines which policies are most robust (Herman *et al.*, 2014; McPhail *et al.*, 2018). These choices can lead to unforeseen and undesirable outcomes, such as a policy favoring one stakeholder at the expense of another (Herman *et al.*, 2014; Alexander, 2018) or scenarios that are uninterpretable for decision-making (Rudin, 2019). Due to complex relationships between policies, SOW, performance, and robustness, it is difficult to anticipate these consequences *a priori* (Miller, 1956; LeCompte, 1999; Quinn *et al.*, 2018; Smith, Kasprzyk and Rajagopalan, 2019). Therefore, it is important for stakeholders to explore the policy recommendations that result from their input and the subsequent methods chosen by analysts. Then, stakeholders can update their preferences after exploring the results, a form of *a posteriori* decision support (Kwakkel and Haasnoot,

2019). Updated preferences may motivate additional analysis prior to choosing a final policy. In this thesis, this iterative loop of feedback and updated analysis is called participatory DMDU.

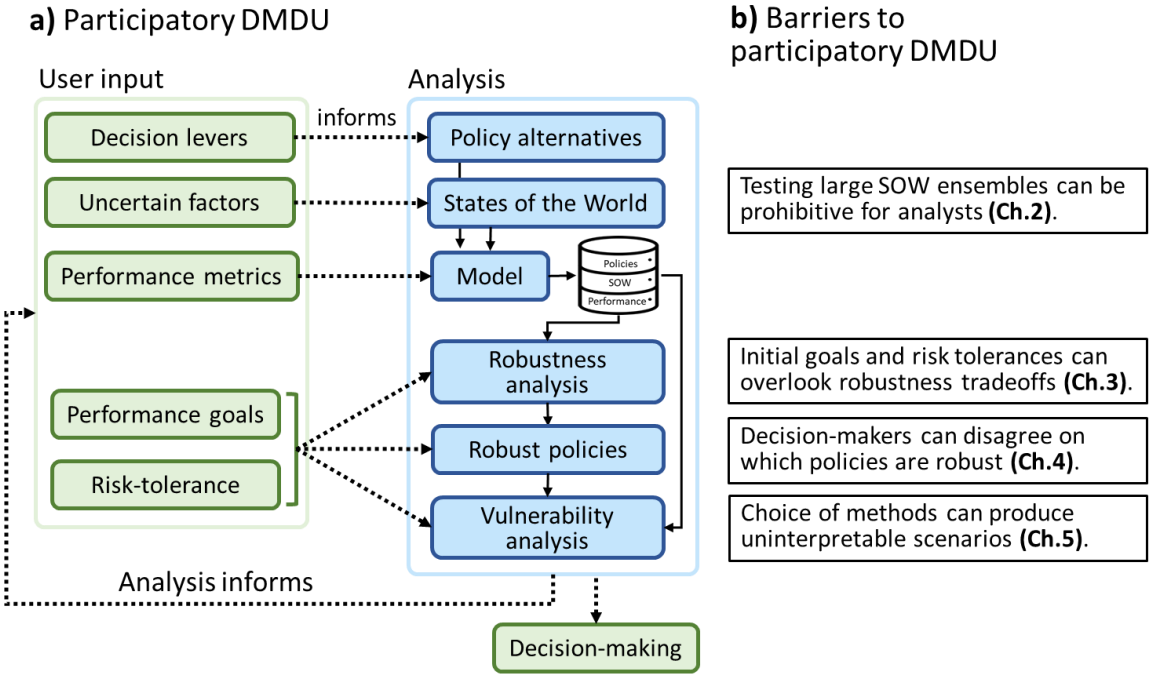


Figure 1-1. a) Barriers to participatory DMDU. b) Novel methods and tools to address the barriers. SOW stands for States of the World.

The goal of this thesis is to address several barriers to participatory DMDU (Figure 1-1, b). First, simulating large SOW ensembles can require significant computing resources, which can be challenging for practitioners with time and budget constraints or when stakeholder feedback warrants additional modelling. Second, choosing robustness metrics is non-trivial because policies that are robust with respect to one metric may be non-robust with respect to other metrics, i.e., policies exhibit robustness tradeoffs. In the presence of tradeoffs, decision-makers may disagree on which policies are most robust, requiring negotiation and compromise to choose policies (the third barrier). Fourth, there is a plethora of machine learning methods to perform a vulnerability analysis. However, the choice of method has consequential impacts on the interpretability of scenarios, and therefore the usefulness of the scenarios for decision-making. Currently, there is limited guidance for choosing methods that are interpretable for the decision-making context.

This thesis contributes novel algorithms and interactive tools to address these barriers. To reduce computing requirements, Chapter 2 contributes a comprehensive framework that evaluates the number of SOW required for accurate robustness ranking compared to a larger SOW ensemble. We show that, depending on the robustness metric, fewer SOW can achieve similar robustness ranking compared to more SOW. Chapter 3 contributes a framework for *a posteriori* robustness analysis, which enables stakeholders to refine their performance goals and risk-tolerances after seeing tradeoffs. We demonstrate our framework with a web tool that provides interactive visualizations, an interface to update choice of metrics, and rapid recalculation of robustness based on updated input. Chapter 4 uses the Self-Organizing Map (SOM), a machine learning method, to create a negotiation framework that helps decision-makers identify compromise policies. The SOM places policies intro groups within which performance is similar, organizes the groups according to predominant tradeoffs, and helps identify policies that strike a balance between competing interests. Chapter 5 contributes a comprehensive review of vulnerability analysis methods and identifies best practices for choosing methods that are interpretable for the decision-making context.

1.2 Background

1.2.1 chosen DMDU framework: Many Objective Robust Decision Making

The specific DMDU framework used in this thesis is Many Objective Robust Decision Making (MORDM) (Kasprzyk *et al.*, 2013). MORDM expands on other DMDU frameworks by generating policies using multi-objective optimization (Hadka and Reed, 2013; Maier *et al.*, 2019). This process couples a multi-objective optimization algorithm with a simulation model, in a loop. The optimization algorithm suggests a policy (i.e., values for the decision levers), the model evaluates policy performance with respect to performance objectives, then the optimization algorithms uses performance objective values to suggest a new policy. Over thousands of iterations, this loop 'evolves' a large set (e.g., hundreds) of high-performing policies that are non-dominated (Quinn *et al.*, 2018; Wheeler *et al.*, 2018). Policy *a* dominates

policy *b* if policy *a* is equal or better in each objective and better in at least one compared to policy *b*. In the resulting set, no policy dominates any other policy – they are non-dominated. In effect, the policies demonstrate tradeoffs (Kasprzyk *et al.*, 2013; Raseman *et al.*, 2020), meaning a policy that improves performance in one objective has worse performance in other objectives. In this step, the simulation model uses a baseline set of SOW, not encompassing all plausible futures, so it is possible that the performance values do not capture the full range of possible performance or important tradeoffs under conditions of deep uncertainty (Herman *et al.*, 2014; Alexander, 2018).

The policies are then reevaluated in a large SOW ensemble to test performance under diverse and challenging SOW beyond those in the optimization step (Herman *et al.*, 2015; McPhail *et al.*, 2018). Here, the analyst defines the plausible upper and lower limits of the uncertain factors then uses a statistical design of experiments to globally sample them. Each sample is a SOW. The policies are then reevaluated in the SOW ensemble.

Robustness analysis quantifies how well policies perform across the SOW ensemble using robustness metrics (McPhail *et al.*, 2018). Robustness metrics involve two decisions: the choice of performance objective, and the method for aggregating performance objective values across the SOW ensemble (i.e., SOW aggregation) (Zatarain Salazar, Castelletti and Giuliani, 2022). Common SOW aggregation methods include calculating the average performance, worst-case performance, or fraction of SOW where a performance threshold is satisfied. The SOW aggregation method reflects a stakeholder's tolerance for uncertainty-related risk (McPhail *et al.*, 2021). Policies can be ranked from most to least robust, helping decision-makers prioritize a small number (e.g., four) of robust policies (Alexander, 2018).

Vulnerability analysis then uses factor mapping to discover scenarios causing 'decision-relevant' performance outcomes (Bryant and Lempert, 2010; Hadjimichael, Quinn, *et al.*, 2020). Input from stakeholders and decision-makers helps define these decision-relevant outcomes, such as a reservoir policy failing to meet 100% of demands. Factor mapping then identifies the subset of uncertain factors

and their values that are the strongest predictors of that outcome. For example, factor mapping could reveal that average precipitation is the strongest determinant of reservoir levels, and that unacceptable levels are expected if precipitation is less than 85% of the historical average (Groves *et al.*, 2013; Reis and Shortridge, 2020). The uncertain factors and their values are concise descriptions of consequential SOW, which can be communicated to decision-makers as scenarios (Steinmann, Auping and Kwakkel, 2020). Because these scenarios are discovered to have decision-relevant outcomes, rather than assumed as consequential, they help avoid perceptions of subjectivity or bias in policy deliberations (Lempert *et al.*, 2006; Bryant and Lempert, 2010).

1.2.2 Participatory decision support

As climate and population stressors deteriorate benefits provided to stakeholders, there is increasing demand for robust and fair policies (UN General Assembly, 2015; IPCC, 2022; Reclamation, 2023e). Participatory decision support engages stakeholders and decision-makers throughout the analysis so their perspectives on robustness and fairness are reflected in decision-making (Smith, Kasprzyk and Dilling, 2017; Kasprzyk *et al.*, 2018; Moallemi *et al.*, 2021). Participation in DMDU-based decision support is particularly important because of the large number of methodological decisions during the analysis and path dependency between those decisions and policy recommendations (Lahtinen, Guillaume and Hämäläinen, 2017; McPhail *et al.*, 2020; Quinn *et al.*, 2020; Reis and Shortridge, 2020). Due to complex relationships between policies, SOW, and robustness, initial input can lead to unanticipated and undesirable policy recommendations. Ideally, stakeholders and decision-makers iteratively update their preferences after seeing the results of analysis (Kollat and Reed, 2007; Woodruff, Reed and Simpson, 2013), a form of *a posteriori* decision support.

Specifically, this thesis addresses the participation of three groups in DMDU: practitioners, stakeholders, and decision-makers. Practitioners include engineers and scientists performing DMDU analyses (Colorado Springs Utilities, 2017; Smith *et al.*, 2022). Stakeholders include those whom policy

decisions impact, such as agriculture districts and environmental non-profits (Reclamation, 2023e). Decision-makers include policy-makers and managers who advise the methodological decisions made by analysts. The delineation between these three groups is flexible, meaning an individual may belong to more than one group. For example, practitioners and decision-makers are part of organizations that also have specific interests impacted by policy decisions, so they can also be considered stakeholders (Stanton and Roelich, 2021). Likewise, practitioners make methodological decisions (i.e., how best to represent the system with a model, methods for creating a SOW ensemble) which can impact policy recommendations (i.e., which policy is most robust) (Quinn *et al.*, 2020; Reis and Shortridge, 2020). In this sense, practitioners are also decision-makers.

Despite the flexibility of the group definition, this thesis frames its contributions specifically to benefit different groups. Chapter 2 seeks to reduce the computing requirements faced by analysts performing DMDU; Chapter 3 contributes a tool to help stakeholders identify robustness metrics and policies that reflect their interests; Chapter 4 contributes a framework to help decision-makers negotiate to compromise policies; Chapter 5 contributes a review to help analysts choose vulnerability analysis methods that are interpretable for different decision-making contexts.

1.3 Overview of work

This thesis addresses four challenges to participatory DMDU, which correspond to Chapters 2-5.

1.3.1 Subsampling and space-filling metrics for computational efficiency (Chapter 2)

The first challenge is the computational requirements for DMDU. This challenge is particularly relevant for studies using climate and population projection data for the uncertain factors, which is common in government-level planning (Reclamation, 2012a; River Management Joint Operating Committee, 2020; Smith *et al.*, 2022). The use of projections for multiple uncertain factors requires a strategy for combining them into SOW, which is commonly done by making every possible combination of the projections – a full-factorial design (Alexander, 2018; Jafino and Kwakkel, 2021). In full-factorial

designs, the number of SOW grows exponentially as either the number of uncertain factors or the number of projections per factor increase (Choi *et al.*, 2021). This method can produce large SOW ensembles, and testing them all in the model can exceed the computing resources and deadlines faced by practitioners. In the absence of projections for the uncertain factors, space-filling methods (i.e., Latin Hypercube Sampling) are used, in part because the analyst can choose a feasible number of SOW (Herman *et al.*, 2015; Joseph, 2016). However, these methods create new SOW, meaning predetermined projections cannot be used. Ideally, SOW ensembles could be created from existing projections and practitioners could choose the number of SOW.

Another challenge is determining the number of SOW to simulate in the model. Given computational requirements and time constraints, is it appropriate to perform a robustness analysis with fewer SOW? For fewer SOW to be appropriate, the ranking of policies by robustness would need to be similar compared to if more SOW were used, since these rankings can inform which policies are preferred by decision-makers. Other studies have concluded that the ranking of policies can change depending on the probability distributions, upper and lower bounds, and pairwise correlations of the uncertain factors (McPhail *et al.*, 2020; Quinn *et al.*, 2020; Reis and Shortridge, 2020, 2021). However, it remains unknown if fewer SOW can achieve the same policy rankings as a larger SOW ensemble.

This thesis introduces a comprehensive sampling framework to improve the computational efficiency of DMDU analyses. First, subsampling methods select a subset of SOW from existing data such that the subset maximally covers plausible values of the uncertain factors (Minasny and McBratney, 2006). Then, the sensitivity of policy rankings to ensemble size is tested by subsampling smaller SOW ensembles from an existing, larger ensemble. The robustness rankings are recalculated using each of the smaller ensembles, and their rank accuracy is measured relative to the larger ensemble (McPhail *et al.*, 2020). Whereas existing strategies for testing sensitivity of DMDU results require simulations for multiple SOW ensembles (McPhail *et al.*, 2020; Quinn *et al.*, 2020; Reis and Shortridge, 2020), the approach in this thesis

reduces computing requirements by recycling existing SOW and model runs. Lastly, the sampling framework creates a statistical relationship between model-free metrics of ensemble quality and rank accuracy. So-called space-filling metrics use distance calculations in the uncertainty space, and thus are model-free (Dupuy, Helbert and Franco, 2015). This relationship can determine if a SOW ensemble of a given size will produce similar robustness rankings as a larger SOW ensemble before testing it in the simulation model.

1.3.2 *a posteriori* selection of robustness metrics (Chapter 3)

The second challenge is choosing robustness metrics. A body of research has shown that the policy deemed most robust often varies depending on which robustness metric is used (Herman *et al.*, 2015; McPhail *et al.*, 2020; Reis and Shortridge, 2020). McPhail et al (2021) addressed the choice of robustness metrics using a questionnaire to solicit the stakeholder's understanding of the problem (e.g., performance thresholds) and preferences (e.g., risk tolerance), and then algorithmically recommend robustness metrics based on their statistical properties. Such mapping of initial preferences to robustness metrics is a form of *a priori* decision making since it is done before performance outcomes are investigated (Kasprzyk *et al.*, 2013; Kwakkel and Haasnoot, 2019). Complex human-environmental systems, though, can result in incomplete and inaccurate understanding of how decisions (such as choice of robustness metrics) can lead to undesirable performance outcomes (Zeleny, 1989; Roy, 1990; Kasprzyk *et al.*, 2012; Woodruff, Reed and Simpson, 2013; Herman *et al.*, 2014). In other words, basing policy decisions on *a priori* preferences can result in in unexpected tradeoffs between performance objectives and exhibit alarmingly poor robustness under alternative metrics.

Alternatively, *a posteriori* decision support helps stakeholders establish their preferences *after* exploring what performance outcomes are possible (Kollat and Reed, 2007; Kasprzyk *et al.*, 2013; Woodruff, Reed and Simpson, 2013). These methods use interactive visualization techniques that allow the stakeholder to explore performance tradeoffs, learn the relationships between policies and

performance, and iteratively refine their preferences. *A posteriori* methods are frequently used to explore tradeoffs between multiple performance objectives, but have seen limited applications for robustness analysis (e.g., Herman et al. (2014, 2015)).

This thesis contributes a framework for *a posteriori* robustness analysis. For each performance objective, a broad selection of robustness metrics is calculated to reflect varying degrees of risk-tolerance and different methods to summarize performance over the SOW ensemble. Then, the framework provides stakeholders with background information and training to interpret what the metrics mean, rather than propose what metrics are most appropriate. Interactive visualizations help explore tradeoffs between robustness metrics and objectives, discover relationships between policies and robustness, iteratively refine which metrics and objectives are important, and remove non-robust policies. To do so, our framework introduces several novel tools for robustness analysis, namely 'on-the-fly' non-dominated robustness sorting and decision tracking. Our dynamic and interactive framework requires an integrated platform, which we create via a web application.

1.3.3 Robustness-informed negotiation and compromise (Chapter 4)

The third challenge is decision-makers is choosing policies when they hold conflicting performance priorities. DMDU-based decision support requires understanding of how alternative policies lead to different performance and robustness tradeoffs. However, having this understanding is non-trivial because the policy set often consists of hundreds of policies characterized by complex interactions between decision variables, objectives, and robustness metrics (Miller, 1956; LeCompte, 1999; Quinn *et al.*, 2018; Smith, Kasprzyk and Dilling, 2019). Moreover, decision-makers may struggle to use all this data to overcome foundational disagreements, such as different weighing of performance objectives (Smith, Kasprzyk and Dilling, 2019) or tolerance for uncertainty-related risk (McPhail *et al.*, 2018; Hadjimichael, Quinn, *et al.*, 2020; McPhail *et al.*, 2021). These challenges demonstrate the need for a framework that

synthesizes the relationships between policies, tradeoffs, and robustness while illuminating policies that strike a compromise between competing goals.

This thesis addresses these challenges by coupling DMDU with the Self-Organizing Map (SOM), a machine learning method (Kohonen, 1982). We use the SOM to place similarly performing policies into groups (clusters), then arrange policy groups on a two-dimensional, tradeoff-based coordinate system, i.e., a tradeoff map. Different DMDU 'layers' can be plotted on this map using the policy clusters, such as performance objectives and decision levers. Doing so reduces the number of policies and tradeoffs to consider by decision-makers while preserving the most pertinent information. We extend previous applications of the SOM (Obayashi and Sasaki, 2003; Zhang *et al.*, 2018) to add robustness layers to the map. Specifically, we add one robustness layer per decision-maker to show their preferred robustness metric. Although each decision-maker can have a unique robustness layer, they share a common coordinate system. We demonstrate how this combination of individual robustness preferences and a shared coordinate system can facilitate a process of negotiation and compromise.

1.3.4 Taxonomy of purposes, methods, and recommendations for vulnerability analysis (Chapter 5)

The fourth challenge is choosing methods for vulnerability analysis that are interpretable for the decision-making context. Early applications of vulnerability analysis used a binary structure of 'decision-relevant' outcomes, e.g., a reservoir operation policy is acceptable if it meets 100% of demands, and unacceptable otherwise. Recently, novel methods have demonstrated new structures for specifying decision-relevant performance outcomes. These include multi-class structures, introduced to address performance inequalities between stakeholders (Jafino and Kwakkel, 2021) and non-stationarity of performance outcomes (Jafino and Kwakkel, 2021). Moreover, recent studies have departed from PRIM for factor mapping, opting instead for more flexible methods such as logistic regression (Hadjimichael, Quinn, *et al.*, 2020) and boosted trees (Trindade, Reed and Characklis, 2019). More flexible methods can address non-linear relationships between policies, SOW, and performance outcomes. The benefit of this

body of literature is exposing DMDU to advanced tools that can address technical challenges. However, a potential limitation is that the resulting scenarios from more complex algorithms may be less interpretable for decision-making (Rudin *et al.*, 2022). Previous reviews have covered the difference between factor mapping and factor ranking for vulnerability analysis (Herman *et al.*, 2015; Kwakkel and Haasnoot, 2019), but have not given a systematic comparison of performance outcome structures, alternative factor mapping methods, or best practices for choosing interpretable methods.

This thesis contributes a review of vulnerability analysis methods and best practices for creating interpretable scenarios. Our review identifies three performance outcome structures – binary, multi-class, and continuous – and demonstrates that different structures can be better suited for different decision-making contexts. We use these three structures to create a taxonomy of factor mapping methods. A helpful framing taken in this review is that factor mapping is a classification and regression exercise, a type of machine learning (James *et al.*, 2013; Rudin *et al.*, 2022). Therefore, we apply the machine learning concepts of interpretability and flexibility to systematically compare factor mapping algorithms. Our review shows that interpretability depends both on the flexibility of the chosen methods and on the specific decision-making purpose. We identify five common purposes for vulnerability analysis and discuss the interpretability of different performance outcome structures and factor mapping methods for those purposes. Lastly, we demonstrate how comparing factor mapping algorithms with testing accuracy, rather than training accuracy, can help improve the interpretability of scenarios for decision-making.

1.4 Case study: reservoir operation policy in the Colorado River Basin

To demonstrate the novel frameworks and interactive tools, this thesis uses a single case study: reservoir operation policies in the Colorado River Basin (CRB).

The CRB is managed with a system of reservoirs to provide water for nearly 40 million people across seven U.S. states, Northwest Mexico, and 30 tribal nations (Reclamation, 2012a, 2018a). Lake Powell and Lake Mead are the largest reservoirs in the system, with storage capacity of four to five times

the historic annual inflow. Annual releases from these reservoirs are chosen to meet delivery-related objectives – e.g., current demands by irrigators and water utilities – and storage-related objectives – e.g., maintaining necessary levels for hydropower and storing water for drought years (Alexander, 2018). These objectives are often-times characterized by tradeoffs, especially during extended periods of low reservoir inflows.

Since 2000, extended periods of low inflow have depleted storage in Lakes Mead and Powell to historic levels, mobilizing federally-mandated delivery reductions to downstream users (DOI, 2022). In an attempt to protect reservoir storage, a delivery reduction policy was established in 2007 (Figure 1-2,a) (Reclamation, 2007). This policy defines pool elevations (y-axis) and corresponding release reductions (shown with colors) for Lake Mead. The policy also determines how releases from Lake Powell, the upstream reservoir, are used to balance storage with Lake Mead. As storage continued to decline, however, additional delivery reductions were added in 2017 and 2019, resulting in a combined policy (Figure 1-2,a) (International Boundary and Water Commission, 2012, 2017; Buschatzke *et al.*, 2019). Storage levels continued to decline, leading to shortages for downstream users in 2022-2024, pursuant the policy (Reclamation, 2023d). The current policy expires in 2026, thereafter a new policy will take effect. Under low inflow conditions, the new policy will determine the relative priority of storage and delivery objectives.

The Bureau of Reclamation, the organization leading post-2026 negotiations, is using DMDU to explore tradeoffs and identify policies robust to future climate (Smith *et al.*, 2022; Reclamation, 2023c). Reclamation seeks meaningful participation from stakeholders throughout the DMDU analysis, requiring mass-communication of modelling results and efficient systems for stakeholders to provide feedback. However, deadlines for this participatory DMDU process are constrained by the 2026 expiration of existing policies.

Inspired by post-2026 negotiations, this thesis uses the CRB as a testbed to develop novel methods for participatory DMDU. Chapters 2-4 use a set of 463 Lake Mead policies provided by Reclamation. These policies were created with multi-objective optimization, as described in those chapters and in Alexander (2018). The 463 policies implement diverse delivery reductions beginning at different reservoir levels, demonstrating different prioritizations of storage vs. delivery objectives. Two examples are given Figure 1-2, b. The policy on the left uses small delivery reductions and favors delivery objectives, whereas the policy on the right uses large reductions and favors storage.

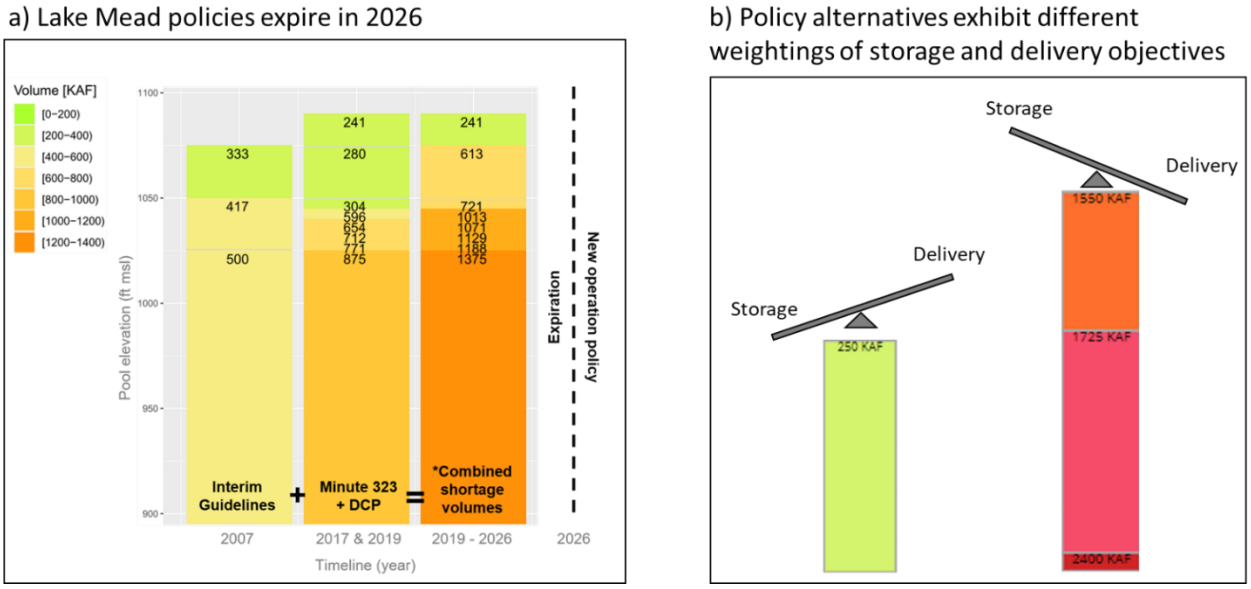


Figure 1-2: a) current reservoir operation policies in the CRB expire in 2026. b) two examples of Lake Mead policies used in this thesis, demonstrating tradeoffs between storage and delivery.

In this thesis, the policies are tested in 500 SOW to evaluate performance in plausible reservoir inflow, demand, and initial storage conditions. The resulting database of model runs is used to demonstrate the novel frameworks and tools in Chapters 2-4 (see Table *1-1*). Chapter 5 then uses the CRB as an illustration of purposes and methods for vulnerability analysis.

Chapter	Summary of Colorado River Basin case study
2: Subsampling and Space-filling Metrics to Test Ensemble Size for Robustness Analysis	• 463 Lake Mead policies tested in 500 SOW • Subsample 50 to 450 SOW from all 500 SOW to test sensitivity of policy rank to number of SOW • Test six robustness metrics, the combination of two SOW aggregation strategies and three performance objectives • Find that fewer than 500 SOW can accurately rank policies, depending on robustness metric
3: Interactive, Multi-Metric Robustness Tradeoffs in the Colorado River Basin	• Contribute a robustness analysis web tool to explore tradeoffs for 463 Lake Mead policies • Policies tested in 500 SOW • Tool supports more than 50 robustness metrics • Tool available online: www.nabocrb.shinyapps.io/CRB-Robustness-App-JWRPM/ • Use tool to identify 7 robust Lake Mead policies
4: Mapping Policies to Synthesize Optimization and Robustness Results for Decision-Maker Compromise	• Self-Organizing Map trained on performance objective values of 463 Lake Mead policies • Create a map of 15 Lake Mead policy clusters organized by storage and shortage tradeoffs • Use robustness maps to identify compromise policies between two hypothetical decision-makers
5: Taxonomy of Purposes, Methods, and Recommendations for Vulnerability Analysis	• Use CRB examples (both from the literature and hypothetical examples) to explain methods and purposes for vulnerability analysis • Perform a simplified, numerical simulation of reservoir storage to demonstrate accuracy and interpretability of two factor mapping methods (logistic regression and random forest)

Table 1-1. Summary of Colorado River Basin case studies in Chapters 2-5

1.5 Research outcomes and organization

Chapter 2 is the journal article "Subsampling and Space-filling Metrics to Test Ensemble Size for Robustness Analysis with a Demonstration in the Colorado River Basin." It is under review in *Environmental Modelling & Software*, and it is co-authored by Drs. Joseph Kasprzyk, Edith Zagona, and Balaji Rajagopalan. This paper demonstrates that, by subsampling SOW with space-filling designs, smaller SOW ensembles can accurately rank policies by robustness compared to larger ensembles. In our CRB case study, for example, a fraction of SOW can identify the ten most robust Lake Mead policies. However, the number of SOW depends on the robustness metric. The study also shows that space-filling metrics are skillful predictors of rank accuracy, creating a simulation-free method to test if a smaller SOW ensemble will yield similar robustness rankings compared to a larger SOW ensemble. This chapter contributes to the participation of practitioners in DMDU by establishing methods that can reduce computing requirements.

Chapter 3 is the journal article “Interactive, Multi-metric Robustness Tradeoffs in the Colorado River Basin.” It is accepted for publication in the *Journal of Water Resources Planning and Management*. The article is co-authored by Drs. Joseph Kasprzyk, Edith Zagona, and Rebecca Smith. The article implements our *a posteriori* framework for robustness analysis, and it is demonstrated with a robustness analysis of Lake Mead policies. We showed that robustness metrics chosen *a priori* for the CRB can lead to poor robustness with respect to worst-case shortages, as discovered with *a posteriori* exploration. We demonstrated how interactive visualizations and a graphic user interface empowers discovery of tradeoffs, efficient stakeholder feedback, and rapid recalculation of robustness. This research was in collaboration with Reclamation, contributing to an upcoming interactive tool that will support post-2026 negotiations.

Chapter 4 is the journal article “post-MORDM: mapping policies to synthesize optimization and robustness results for decision-maker compromise.” It was published in *Environmental Modelling & Software* in 2022, coauthored by Drs. Joseph Kasprzyk and Edith Zagona. The article demonstrates our negotiation framework by organizing 463 Lake Mead policies into 15 representative groups, and synthesizes the two predominant tradeoffs when choosing between policies. Those tradeoffs are 1) storage vs. delivery objectives, and 2) magnitude vs. duration of shortages. Using this tradeoff map, we create robustness layers for two hypothetical decision-makers, one who favors storage and one who favors deliveries. We use their individual robustness layers and the tradeoff coordinates to identify compromise Lake Mead policies.

Chapter 5 is the journal article “Taxonomy of purposes, methods, and recommendations for vulnerability analysis”. The article is in preparation for submission to *Earth’s Future*, co-authored by Drs. Joseph Kasprzyk and Edith Zagona. The article explains the three performance outcome structures (binary, multiclass, and continuous) and reviews three methods for defining multi-class performance outputs. The article compares four factor mapping algorithms for binary performance structures based on

interpretability and flexibility. We compare two algorithms for multi-class performance structures based on overlapping versus mutually exclusive scenarios, explaining their suitability for finding compromise policies versus comparing policies. We explain how continuous factor mapping can empower stakeholders to define performance goals after seeing what performance outcomes are possible. Lastly, we demonstrate how testing accuracy can reveal when a more interpretable scenario is just as accurate at predicting performance outcomes compared to a less interpretable scenario.

Chapter 6 provides a summary, describes where this research has been disseminated, explains ongoing work, and concludes with future research opportunities.

2 Subsampling and Space-filling Metrics to Test Ensemble Size for Robustness Analysis

2.1 Introduction

Decision-makers are faced with irreducible uncertainties such as climate change (IPCC, 2021), population growth (Gold *et al.*, 2019a), global trade patterns (Yarlagadda *et al.*, 2023), and energy market transitions (Steinmann, Auping and Kwakkel, 2020). Scenarios can represent plausible realizations of how the uncertain factors may unfold in the future (Bryant and Lempert, 2010; Kasprzyk *et al.*, 2013; Smith *et al.*, 2022). However, these severe uncertainties can be challenging to represent with a small number of scenarios, since decision-makers do not know or disagree on their probabilities. This condition is referred to as deep uncertainty (Knight, 1921; Lempert, Popper and Bankes, 2003), and analytical methods termed Decision Making Under Deep Uncertainty (DMDU) have been used to combat this (Lempert *et al.*, 2006; Kwakkel and Haasnoot, 2019). Instead of using a small number of scenarios as was traditionally done, a large ensemble of scenarios is used to broadly sample the uncertainty space (Lempert *et al.*, 2006; Bryant and Lempert, 2010). This is especially appropriate due to the multi-dimensional nature of the uncertain factors and their wide range of possible values (Chapman *et al.*, 1994; Jones, Schonlau and Welch, 1998; Loeppky, Sacks and Welch, 2009; Levy and Steinberg, 2010). The resulting ensemble of scenarios creates challenging conditions in which to test policy alternatives.

These large scenario ensembles are used to test policy alternatives in a robustness analysis. Because the probability of any scenario occurring is unknown, policies that perform well in many scenarios – robust policies – are preferred to policies that perform well in one or a few (Lempert *et al.*, 2006; Kasprzyk *et al.*, 2013). Robustness is quantified with robustness metrics, which are statistics that summarize each policy's performance across the scenarios (McPhail *et al.*, 2018). Decision-makers can prioritize policies using robustness (i.e., they are likely to prefer more robust policies to less robust policies).

In a robustness analysis, a large number of scenarios and policies can result in impractical computational overhead. These analyses may test hundreds of policies and hundreds to thousands of scenarios (Quinn *et al.*, 2018; Gold *et al.*, 2019b; Hadjimichael, Quinn, *et al.*, 2020), which can require weeks to months of computing time (Alexander, 2018) and/or supercomputing (Trindade, Reed and Characklis, 2019). This challenge is further exacerbated by models with exceptionally long run times such as global climate-economy models (Nikas, Doukas and Papandreou, 2019; Yarlagadda *et al.*, 2023) and military models (Hill and Miller, 2017), which can require hours to months per simulation. Large computational requirements can be problematic for research studies, but it can be especially challenging for government-level agencies, who may need to hire external consultants to meet deadlines and financial constraints (Means *et al.*, 2010; Reclamation, 2012a; Groves and Bloom, 2013; Molina-Perez *et al.*, 2019).

The challenge of planning under deep uncertainty amidst time and computing constraints is epitomized by the Colorado River Basin (CRB). During the 21st century, streamflow has declined to roughly 72% of the historical average (Lukas and Payton, 2020), which, combined with over-allocation, has steadily depleted the system's largest reservoirs, Lake Mead and Lake Powell (Rosenberg, 2022; Wheeler *et al.*, 2022). Although most climate change projections agree that the future will be hotter with less water supply, they exhibit vast uncertainty in the magnitude and direction of change (Lukas and Payton, 2020). There also exists great uncertainty in future water consumption, since various water users, such as Upper Basin states and Tribal Nations, could utilize more of their water rights in the future (Upper Colorado River Commission, 2016; Reclamation, 2018a, 2023e).

Amidst this uncertainty, the current operating policies that govern releases from Lakes Mead and Powell expire in 2026, and a federal-level renegotiation has begun (Reclamation, 2023b). Recognizing that the current policies have not protected system reservoirs from the risk of reaching power or dead pool (the required storage levels to produce hydropower and make releases), the government is using robustness analyses to evaluate alternative operating policies in streamflow and demand scenarios

(Alexander, 2018; Smith *et al.*, 2022). Since the CRB supports copious stakeholders with often-times conflicting goals (Bonham, J. Kasprzyk and Zagona, 2022a; Reclamation, 2023e), it is critical that these robustness analyses are flexible, meaning they can be reiterated as new policy alternatives are proposed. However, with only a few years remaining until 2026, repeating these robustness analyses with current large scenario ensembles may be impractical. This challenge has been demonstrated in a previous robustness analysis of the CRB, which required two months of continuous computing (Alexander, 2018).

Given this computational burden and time constraint, is it appropriate to perform these robustness analyses with fewer scenarios? This paper contributes a comprehensive framework to explore whether scenario ensembles of different sizes can provide a stable rank-ordering of the policies from most to least robust, following prior research that has used this ranking approach (Herman *et al.*, 2014; Alexander, 2018; McPhail *et al.*, 2018, 2020, 2021). For fewer scenarios to be appropriate, the robustness rankings would need to be similar compared to if more scenarios were used, since these rankings can inform which policies are preferred by decision-makers. Other studies have concluded that the ranking of policies can change depending on the probability distributions, upper and lower bounds, and pairwise correlations of the uncertain factors (McPhail *et al.*, 2020; Quinn *et al.*, 2020; Reis and Shortridge, 2020, 2021). However, it remains unknown if fewer scenarios can achieve the same policy rankings as a larger scenario ensemble.

A major challenge to testing the sensitivity of robustness rankings to the scenario ensemble used is that it has required the repeating of computer simulations for multiple scenario ensembles (McPhail *et al.*, 2020; Quinn *et al.*, 2020; Reis and Shortridge, 2020, 2021). Since the simulations required for one scenario ensemble can be computationally intensive, multiple ensembles of such simulations can be intractable.

To overcome this challenge, our framework performs computer simulations with a single scenario ensemble. Then, statistical methods are used to subsample the resulting database, creating smaller

ensembles of varying sizes. For each subsampled ensemble, the objective values corresponding to the sampled scenarios are used to recalculate robustness and re-rank policies. The framework demonstration in the CRB uses generalizable water resources objectives (reliability, resiliency, and vulnerability) and explores how ranking accuracy changes as a function of objective and robustness metric type. To determine the number of scenarios needed, the subsamples' policy rankings are compared to the rankings that result from all scenarios.

Furthermore, our framework utilizes model-free metrics of scenario ensemble quality to indicate if a smaller ensemble will yield similar policy rankings compared to a larger ensemble. So-called 'space-filling' metrics measure the quality of scenario ensembles with distance calculations in the uncertainty space (Damblin, Couplet and Iooss, 2013; Dupuy, Helbert and Franco, 2015). They require no information about model outputs (i.e., objectives or robustness metrics) and thus are model-free. To develop this method, we calculate space-filling metrics for each ensemble, then build statistical models that predict how similarly a smaller ensemble ranks policies compared to a larger ensemble as a function of its space-filling properties. Multiple space-filling metrics are tested to determine which is the most accurate indicator of rank similarity, then we demonstrate how these metrics can indicate the minimum number of scenarios required for a robustness analysis, potentially reducing computational overhead.

The rest of this paper is organized as follows. Section 2.2 explains our framework, using example data for illustration. Section 2.3 gives the details of the case study of reservoir operation policies in the CRB. Results are in Section 2.4, followed by a discussion and conclusion in Sections 2.5 and 2.6.

2.2 Methods

An overview of the framework is provided in Figure 2-1. In step 1, an, an existing ensemble of scenarios and set of policies are input to a simulation model, the outputs of which are used to calculate robustness and rank policies from most to least robust. In step 2, smaller scenarios ensembles are created by subsampling from the ensemble of all scenarios used in step 1. The objective values corresponding to

the sampled scenarios are then used to recalculate robustness and rank policies. The policy rankings from each subsampled ensemble are compared to that of all scenarios via a rank correlation statistic. Three space-filling metrics are calculated for each scenario ensemble. Lastly, in step 3, statistical models are built to predict rank correlation as a function of the space-filling metrics. Proportion of variance explained and prediction intervals are reported to determine the most skillful space-filling metric. In the following subsections, we use example data and calculations to describe each step. The specific implementation for the case study is given in Section 2.3.

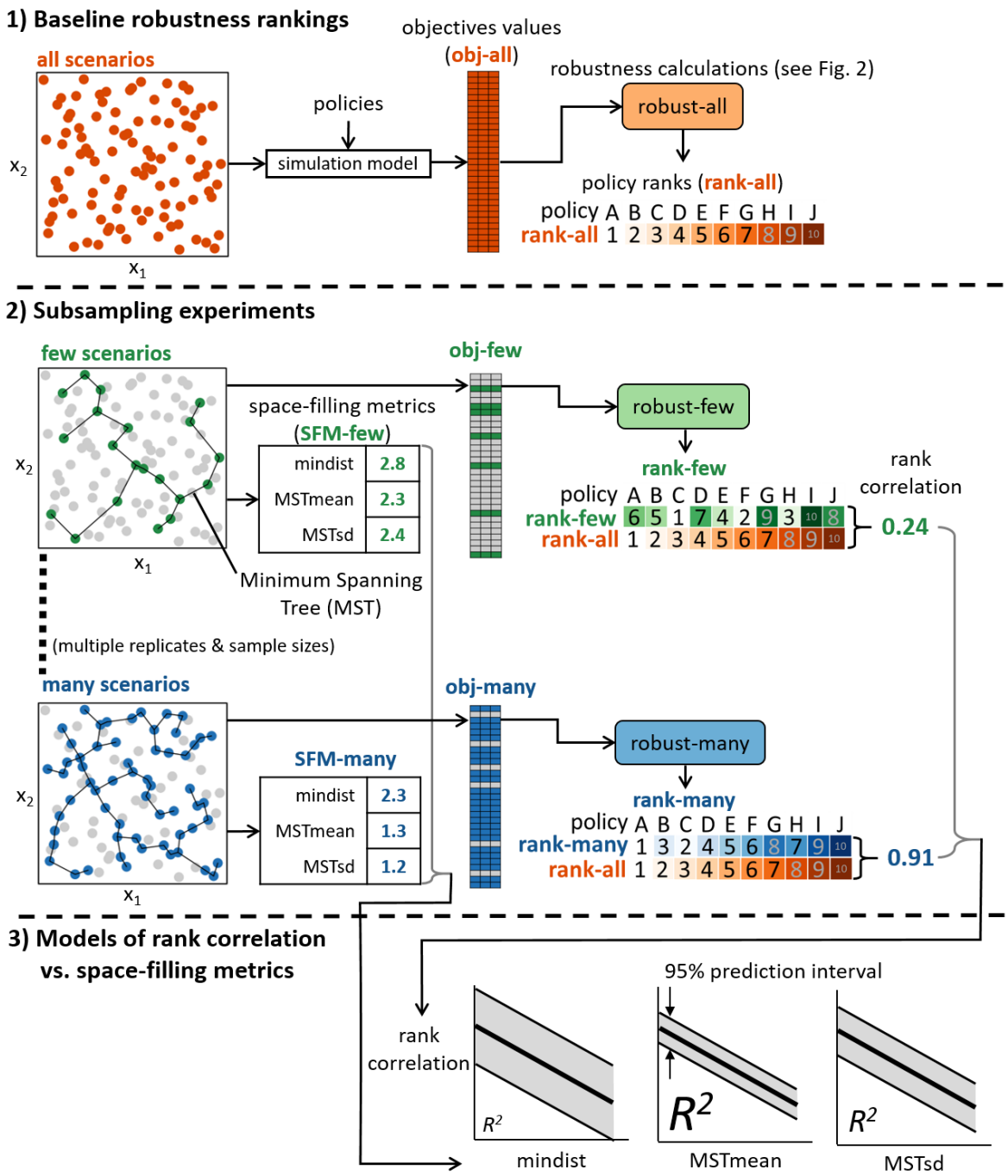


Figure 2-1: Subsampling framework overerview. In step 1, an existing ensemble of scenarios is input to a simulation model to evaluate the performance objectives of a set of a policies. Robustness is calculated for each policy, then the policies are ranked from most to least robust. The ranking of policies from step 1 is used as the baseline for comparision. In step 2, subsampling experiments are performed to create ensembles with fewer scenarios. The objective values are subsetted based on which scenarios are sampled, and are then used to recalculate robustness and rank policies. The policy rankings from each subsampled ensemble are compared to the rankings from all scenarios using a correlation statistic. Space-filling statistics are calculated for each scenario ensemble. In step 3, statistical models of rank correlation as functions of the space-filling metrics are computed, and their accuracy are compared using R^2 and prediction intervals.

2.2.1 Baseline Robustness rankings

2.2.1.1 Model simulations

The purpose of step 1 (Figure 2-1.1) is to establish the robustness rankings of the policy alternatives relative to the baseline ensemble – a large set of 'all scenarios' (orange in Figure 2-1.1). Our framework assumes the baseline ensemble is given (e.g., established in previous work), as the purpose of this framework is to explore how many scenarios are needed to achieve similar robustness rankings. The ensemble of all scenarios is input to a simulation model to evaluate a set of policies (Figure 2-1.2). A policy is defined by a set of decision variable values, which are model inputs that establish how the system will be designed or governed (Lempert, Popper and Bankes, 2003). The policies can be created manually through expert knowledge or generated from optimization (Herman *et al.*, 2015).

2.2.1.2 Performance objectives

The output of the model simulations is a database of performance objective values, which are quantitative measures of system performance. In time-series models, they aggregate the status of a state variable over time, relative to a single policy in a single scenario (Zatarain Salazar, Castelletti and Giuliani, 2022). Although the framework can be implemented with any performance objectives, we demonstrate with three well-known objectives: reliability, resiliency, and vulnerability (Hashimoto, Stedinger and Loucks, 1982). As shown in Figure 2-2a, reliability is the cumulative percentage of time a state variable fails a performance target (shown with green), resiliency is the maximum time required to recover from a failure (orange), and vulnerability is the magnitude of the maximum failure (purple). The example given in Figure 2-2a shows all three objectives calculated for one state variable; however, performance objectives are often calculated for multiple state variables representing alternative pieces of infrastructure, sectors, and/or stakeholders (Kasprzyk *et al.*, 2013; Smith, Kasprzyk and Basdekas, 2018), as illustrated by the case study in Section 2.3.

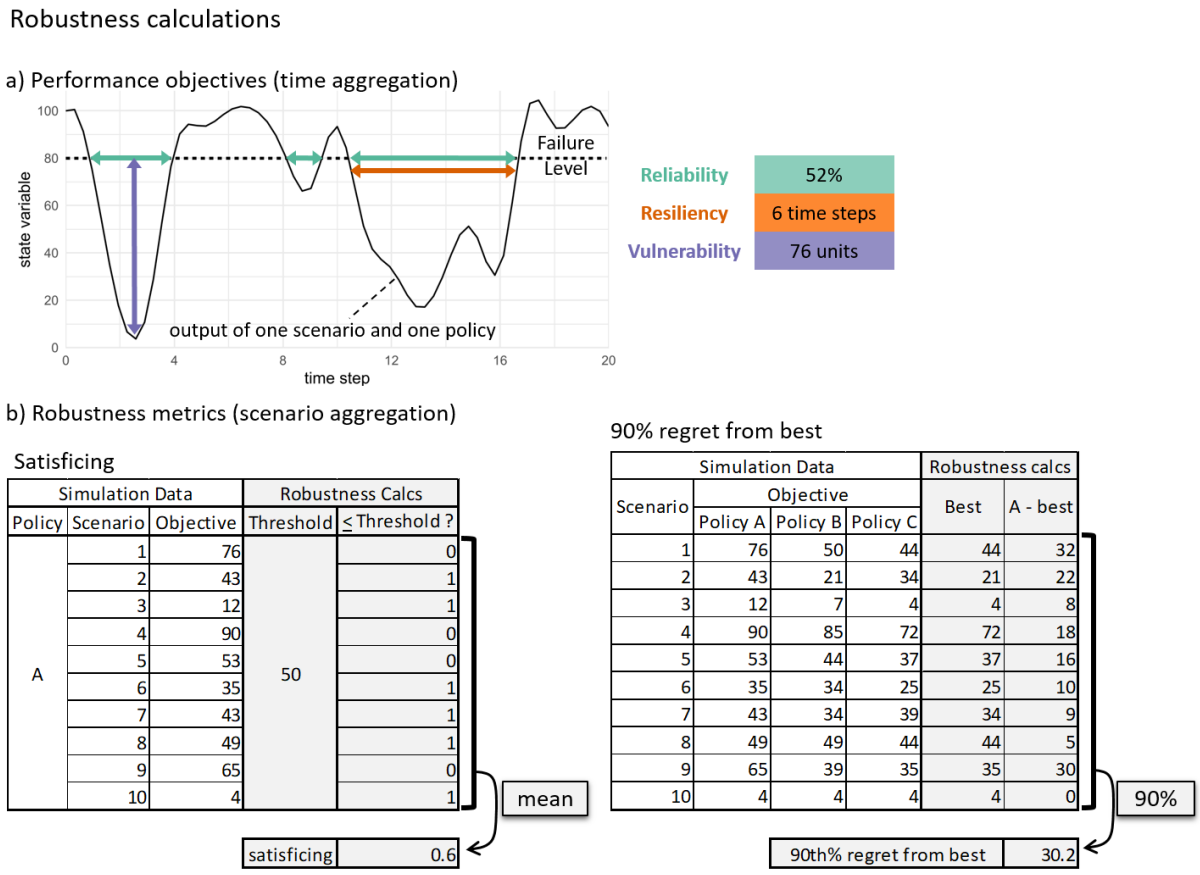


Figure 2-2: The performance objectives and robustness types used for robustness calculations. a) Three types of performance objectives are calculated. Reliability (green) is the percent of time a failure level is violated. Resiliency (orange) is the maximum duration below the failure level. Vulnerability (purple) is the maximum failure. b) For each objective, we calculate two types of robustness. Satisficing (left) is the fraction of scenarios where perfomance threshold is achieved. 90% regret from best measures a policy's deviation from the best performing policy in each scenario. The 90th percentile of the deviations is taken to summarize performance across the scenarios.

2.2.1.3 Robustness calculations

Robustness analysis investigates the range of performance objective values of a policy set when tested in many scenarios; e.g., how much worse is a policy's performance in extreme scenarios, relative to a baseline? Robustness metrics are specific calculations that summarize a policy's performance across the scenario ensemble (McPhail *et al.*, 2018; Zatarain Salazar, Castelletti and Giuliani, 2022). Due to their requirement to summarize complex multivariate data, different robustness metrics exist that could yield different rankings of policy alternatives (Herman *et al.*, 2014; McPhail *et al.*, 2018).

We demonstrate the framework with two types of robustness metrics: satisficing and 90% regret from best (Figure 2-2b). Satisficing (left) is the fraction of scenarios in which a performance objective value meets a performance threshold. Figure 2-2b, left, shows an example where policy A is less than or equal to the performance threshold of 50 in 6 of 10 scenarios (a satisficing score of 0.6). The threshold is specific to each performance objective but constant for all policies and scenarios. 90% regret from best measures the magnitude by which a policy's performance deviates from the best performing policy in each scenario. Figure 2-2b, right, gives an example calculation for Policy A. The column labelled 'Best' records the best performance between policies A, B, and C (assuming 0 is the ideal value). Then, deviation is calculated as the performance of policy A minus the best performance (the 'A-best' column). A value of 0 means policy A is the best performing policy in that scenario (e.g., scenario 10), and larger values indicate another alternative performs substantially better. To summarize regret across the scenarios, the 90th percentile is taken. We use these two robustness types because they implement different techniques for summarizing performance across scenarios (McPhail *et al.*, 2018) and because they are common in robustness analysis case studies (Herman *et al.*, 2015; Reis and Shortridge, 2020, 2021).

Robustness is calculated using each combination of the two robustness types and three performance objectives, resulting in six total robustness metrics. Hereafter, we refer to the robustness metrics using the convention *objective.robustness-type* , e.g., *reliability.satisficing*. We will abbreviate 90% regret from best as *regret*, e.g., *vulnerability.regret*.

2.2.1.4 Robustness ranking

For each robustness metric, the policies are ranked from most to least robust, where rank one is most robust. Our framework assumes the policy rankings that result from all scenarios (e.g., 'rank-all' in Figure 2-1.1) is the most accurate because it samples the uncertainty space more densely than the subsamples.

2.2.2 Subsampling experiments

2.2.2.1 Subsampling scenarios with conditioned Latin Hypercube Sampling

To explore the extent to which robustness rankings change when fewer scenarios are used, Step 2 reevaluates the robustness metrics and policy rankings using subsamples of the scenarios (Figure 2-1.2). Subsampling is performed via observational sampling, which creates smaller ensembles by selecting subsets of all scenarios (Kennard and Stone, 1969; Brus, 2019; Wadoux, Brus and Heuvelink, 2019). For example, Figure 2-1.2 shows two subsampled ensembles with 'few scenarios' (green) and 'many scenarios' (blue).

The observational sampling method used in this study is conditioned Latin Hypercube Sampling (cLHS). cLHS uses an optimization procedure to select a subsample of all scenarios that form a Latin Hypercube in the uncertainty space (e.g., x_1 and x_2 in Figure 2-1) (Minasny and McBratney, 2006, 2010). To determine which scenarios are selected, cLHS stratifies the empirical cumulative distribution functions (eCDFs) of the uncertain factors into *n* equal strata, where *n* is the number of scenarios. For example, if *n* = 2, then each eCDF would be divided into 2 strata that go from 0 to 0.5 and 0.5 to 1.0. In the first iteration of the optimization procedure, *n* scenarios are randomly selected from all scenarios. Next, the objective function is calculated, which is to minimize the number of strata occupied by more than one scenario – i.e., minimize the number of scenarios that violate the definition of a Latin Hypercube. In each subsequent iteration, the objective function is improved by one of two ways: randomly replacing one of the selected scenarios, or randomly replacing scenarios occupying the most overpopulated strata. Because of this randomness, each implementation may yield a different set of scenarios, even for a constant sample size. Therefore, it is recommended to create multiple scenario ensembles (i.e., replicates) per sample size (Worsham *et al.*, 2012; Ma *et al.*, 2020; Wadoux and Brus, 2021).

Our framework uses cLHS because it creates scenario ensembles with similar ranges and probability distributions of the uncertain factors compared to the set of all scenarios (due to the

aforementioned stratification procedure being performed on eCDFs). Maintaining similar probability distributions and uncertainty ranges is important because other studies have shown that the ranking of policies can change if uncertainty ranges or probability distributions are altered (McPhail *et al.*, 2020; Quinn *et al.*, 2020; Reis and Shortridge, 2020). The purpose of our framework is to test the impact of the number of scenarios on policy ranking, so we use cLHS to minimize discrepancies in the range or probability distributions between scenario ensembles.

2.2.2.2 Reevaluation of robustness metrics and policy rankings

The database of objective values from step 1 is then subsetted to contain only the values corresponding to the scenarios selected by cLHS. This process is illustrated by 'obj-few' and 'obj-many' in Figure 2-1.2. The subsetted objective values are then used to reevaluate robustness (e.g., 'robust-few' and 'robust-many') and policy rankings (e.g., 'rank-few' and 'rank-many').

The policy rankings that result from subsampled scenario ensembles may be more or less similar to that of all scenarios depending on the number of scenarios and which scenarios are included (e.g., the cLHS replicate). For example, the 'few scenarios' ensemble (green) has large disagreements with all scenarios (orange) in Figure 2-1.2 – policy A is ranked 6 instead of 1, policy B is ranked 5 instead of 2, etc. In contrast, the 'many scenarios' ensemble (blue) results in policy rankings similar to all scenarios with only minor discrepancies (e.g., policy B is ranked 3 instead of 2).

2.2.2.3 Measuring rank similarity with Kendall's tau-b correlation

To measure the similarity between the subsamples' rankings and all scenarios, our framework uses Kendall's tau-b correlation (labelled 'rank correlation' Figure 2-1.2). This metric was proposed by McPhail et al. (2020) to compare the ranking of policies between multiple scenario ensembles. Kendall's tau is calculated with equation 1.

Equation 1: $$\frac{\#concordant\ pairs - \#\ discordant\ pairs}{\#\ total\ pairs}$$

Consider policies A and B, ranked using all scenarios and again with subsampled ensemble J. If the conclusion about which policy is more robust does not change when using all scenarios compared to ensemble J, then A and B are concordant (Kendall, 1938; McPhail *et al.*, 2020). If the conclusion is reversed between the scenario ensembles, then A and B are discordant. If policies A and B are tied, they are neither concordant nor discordant. This comparison is done for all pairs of policies to find the total number of concordant and discordant pairs. A correlation of 1 means all policies are ranked identically between the two ensembles, -1 means they are ranked in exact opposite order, and 0 means there is no correlation between the rankings. Note that equation 1 does not account for policies that are tied, meaning that values of 1 and -1 are not possible in the presence of ties (even if the same policies are tied using both ensembles). To account for this, Kendall's tau-b makes an adjustment to equation 1 such that the range is restored to [-1,1] in the presence of ties (Kendall, 1945; McPhail *et al.*, 2020). Following McPhail et al. (2020), we use tau-b in this research. As an example, the ensemble of few scenarios has a correlation of 0.24 with all scenarios, while the ensemble of many scenarios has a correlation of 0.91.

2.2.2.4 Space-filling metrics

The quality of a scenario ensemble can be measured by how well the scenarios cover the uncertainty space. For each ensemble, our framework calculates three model-free metrics of ensemble quality: mindist, MSTmean, and MSTsd (Figure 2-1.2). mindist is the minimum Euclidean distance between any two scenarios (Damblin et al., 2013; Dupuy et al., 2015). MSTmean and MSTsd are calculated from the Minimum Spanning Tree (MST) (Damblin et al., 2013; Dupuy et al., 2015; Franco et al., 2009). An MST connects all scenarios with lines whose summed length is minimal. Figure 2-1.2 shows two example scenario ensembles ('few scenarios' and 'many scenarios') with their respective MSTs. To evaluate space-filling properties, the mean and standard deviation of the edge lengths are taken, which we call MSTmean and MSTsd, respectively. The space-filling values shown in the tables in Figure 2-1.2 demonstrate that each space-filling metric decreases with sample size – i.e., for ensembles with more scenarios, the

scenarios are closer together. The magnitude of these metrics also depends on the number of uncertain factors, which will depend on the case study. Therefore, it is more meaningful to scale the space-filling metrics relative to the set of all scenarios. For example, a mindist of 2.8 for few scenarios means the mindist is 2.8 times greater than the value for all scenarios.

The advantage of the space-filling metrics is that they quantify how well the scenarios broadly cover the uncertainty space without using model-derived outputs such as performance objectives or robustness metrics (Joseph, 2016). If these metrics are strong indicators of rank correlation, they could inform the number of scenarios needed without the computational expense of repeating computer simulations for multiple ensembles.

2.2.3 Linear models of rank correlation and space-filling metrics

Next, linear regression models are built to predict rank correlation as a function of space-filling metrics (Figure 2-1.3). The models are trained on the rank correlation and space-filling metric values calculated from the subsampling experiments in step 2. One model is built for each combination of robustness metric and space-filling metric (18 total) according to Equation 2.

Equation 2: $$\overline{rank\ correlation_k} = B_{j,k} * SFM_j$$

$\overline{rank\ correlation_k}$ is the expected (mean) value of Kendall's tau-b for robustness metric *k*; SFM_j refers to mindist, MSTmean, or MSTsd; and $B_{j,k}$ is the slope for robustness metric *k* and space filling metric *j*. The most skillful space-filling metric is selected via R^2, the proportion of variance in actual (observed) rank correlation captured by the models. For the selected metric, the 95% prediction intervals are also reported as an additional quantitative and visual diagnostic of model skill. The 95% prediction interval is the deviation from the model within which 95% of actual rank correlation values are expected to fall (James *et al.*, 2013, chap. 3).

2.3 Case study: shortage policies in the Colorado River Basin

In the case study, we consider shortage operations of Lake Mead in the CRB. In this system, snowmelt from the Rocky Mountains is stored primarily in two reservoirs, Lake Mead and Lake Powell. These reservoirs are the two largest in the US by volume, capable of holding 4-5 years of historical average streamflow (Reclamation, 2012a). Annual releases are set by the federal government with the goals of balancing reservoir storage, generating hydropower, and delivering water to the Southwest and northern Mexico (Alexander, 2018; Bonham, J. Kasprzyk and Zagona, 2022a; Smith *et al.*, 2022).

During the so-called 'millennium drought' (2000-present), storage at Lakes Mead and Powell has declined to critical levels, threatening hydropower production and deliveries (Gangopadhyay *et al.*, 2022; Salehabadi *et al.*, 2022; Wheeler *et al.*, 2022). Under these low storage conditions, releases from Lake Mead have been reduced according to operating policies established under federal law (Reclamation, 2007), interstate agreements (Buschatzke *et al.*, 2019), and an international treaty (International Boundary and Water Commission, 2012, 2017). These reductions result in shortages for downstream users in an attempt to protect reservoir levels.

As the drought has continued, these policies have not curtailed water usage enough to protect the reservoirs from the threat of dropping below minimum power pool or dead pool, at which point no releases could be made. Moreover, the current operating policies expire in 2026, and a large national process to renegotiate when shortages occur and how they are distributed across users in the Southwest has begun (Reclamation, 2023b, 2023e). Thus, the government is using computer simulations to test new shortage operations and their robustness to climate change and increasing population (Smith *et al.*, 2022; Reclamation, 2023e). This case study explores alternative shortage policies for Lake Mead and evaluates their robustness to plausible but uncertain future streamflow and demand scenarios. Because similar analyses are underway as part of the policy renegotiation, and it is likely these analyses will be repeated

as new policy alternatives are proposed by stakeholders (Reclamation, 2023e), we use the framework proposed in section 2.2 to explore how many scenarios are required.

We assume that releases from Lake Powell, which is upstream of Lake Mead, are determined to balance storage with Lake Mead pursuant regulatory requirements (Reclamation, 2007).

2.3.1 Baseline robustness rankings

2.3.1.1 Ensemble of 'all scenarios'

For our ensemble of all scenarios, we use 500 scenarios of streamflow, demand, and initial storage. This scenario set was used in two previous studies in the Colorado River Basin to demonstrate a policy negotiation tool (Bonham, J. Kasprzyk and Zagona, 2022a) and robustness-tradeoff framework (Bonham, Joseph Kasprzyk, and Edith Zagona, 2023). Streamflow values were obtained from existing datasets commonly used in the Colorado River Basin for long-term planning studies (Reclamation, 2012a; Groves *et al.*, 2013; Alexander, 2018), which were derived from the historical record, paleo-reconstructions, CMIP-3 based projections, and statistical resampling (Reclamation, 2012a). Demand and initial storage conditions (i.e., Lake Mead and Lake Powell pool elevations) were sampled via a Latin Hypercube design, where the limits of each factor were informed from demand and pool elevation projections (Upper Colorado River Commission, 2016; Reclamation, 2020). For further details, the reader is referred to Bonham et al. (2022a). In this study, we consider the ranking of alternatives that results from all 500 scenarios as most accurate, and we investigate the accuracies of rankings that result from fewer scenarios.

2.3.1.2 Lake Mead policy alternatives

The policy alternatives provide different values of annual releases from Lake Mead under low storage conditions. The 463 policies were created in previous research using multi-objective simulation-optimization as described in Alexander (2018). Each policy is defined by a vector of pool elevations (i.e. storage levels) at which delivery reductions are enacted and a vector of corresponding volumes by which

the annual release is reduced. Further details of the problem formulation, an exploration of performance and robustness tradeoffs, and visualizations of the policies are given in Bonham et al. (2022).

The policies implement diverse shortage operations, demonstrating different strategies for prioritizing reservoir storage vs. deliveries. These storage and delivery outcomes are quantified in objectives and robustness metrics. Through investigating how different scenario ensembles yield different robustness rankings, this study explores the appropriateness of using smaller numbers of scenarios.

2.3.1.3 Simulation model

To evaluate the performance of Lake Mead policies, we use the Colorado River Simulation System (CRSS), the model historically used by the government for long-term planning studies in the Colorado River Basin (Reclamation, 2012a; Groves *et al.*, 2013; Bloom, 2014; Alexander, 2018; Smith *et al.*, 2022). CRSS is built in RiverWare (Zagona *et al.*, 2001), a hydro-policy modeling software that takes decision variables, streamflow, and demand schedules as inputs, evaluates reservoir storage and deliveries over time via mass-balance calculations subject to regulatory constraints, and outputs multiple user-defined performance objectives. CRSS is run with a monthly time-step, and we implement a 44-year planning horizon in this study as was done in Alexander (2018) and Bonham et al. (2022a). For more information on CRSS, the reader is referred to (Reclamation, 2022).

2.3.1.4 Performance objectives

CRSS is used to evaluate reliability, resiliency, and vulnerability performance objectives (Table 2-1). We consider two state variables: Lake Mead pool elevation (reservoir storage) and delivery shortages. As illustrated in Figure 2-2a, reliability is calculated as the percentage of months that Lake Mead pool elevation drops below a threshold critical for hydropower and water deliveries (1000 feet above mean sea level). Resiliency and vulnerability are calculated on the shortage state variable. In this case, failure occurs whenever shortage occurs (i.e., a failure threshold of 0). Resiliency is the maximum number of consecutive years with shortage, and vulnerability is the maximum cumulative shortage in any

one year. Note that these objectives are three of eight total objectives in the optimization problem used in Alexander (2018) and Bonham et al. (2022a). We chose these three objectives to illustrate how robustness rankings can vary for different types of performance objectives.

Type	Description	Satisficing requirement
reliability	% months that Lake Mead pool elevation < 1000 ft msl	$\leq$ 10 %
resiliency	maximum duration of consecutive years with shortage	$\leq$ 10 years
vulnerability	maximum annual shortage in thousand acre feet (KAF)	$\leq$ 1375 KAF

Table 2-1*:* The specific implementation of the performance objectives in the Colorado River Basin case study. The performance requirements used in the calculation of satisficing are also reported.

2.3.1.5 Robustness metric parameters

In order to calculate robustness metrics, an analyst must choose objectives and additional performance parameters. Specifically, satisficing metrics require performance thresholds to delineate acceptable from unacceptable performance. For our case study, we require reliability $\leq$ 10% (Alexander, 2018; Bonham, J. Kasprzyk and Zagona, 2022a) and resiliency $\leq$ 10 years (Bonham, J. Kasprzyk and Zagona, 2022a), as these thresholds have been recommended in previous studies. For vulnerability, we require maximum shortages[1] $\leq$ 1375 thousand acre feet (KAF). These requirements are also listed in Table 1. 90% regret from best is calculated as described in Figure 2-2b.

2.3.2 Subsampling experiments

2.3.2.1 cLHS implementation

To create smaller scenarios ensembles, the ensemble of 500 scenarios is subsampled using the 'clhs' package (Roudier *et al.*, 2021) in R (R Core Team, 2023). The inputs to cLHS are the uncertain factors

[1] Summing the current Lake Mead shortage guidelines (Reclamation, 2007; International Boundary and Water Commission, 2012, 2017; Buschatzke *et al.*, 2019), 1375 KAF is the maximum shortage. Given that this shortage volume was approved, we assume that shortages at or below this level are acceptable, whereas larger shortages are not.

describing the scenarios: demand, initial Lake Mead pool elevation, initial Lake Powell pool elevation, and 4 features describing the streamflow traces. The streamflow features are the average annual flow of each trace's driest 20-year period, wettest 20-year period, driest 2-year period, and wettest 2-yr period. These features were selected to represent long term wet-dry conditions and short-term wet-dry conditions. See Appendix A.1 for details.

We create scenario ensembles of varying sizes with multiple replicates each. 50 to 450 scenarios are tested at an interval of 50 scenarios (a total of 9 sample sizes). 30 replicates are performed per sample size, resulting in 270 scenario ensembles (9 sample sizes x 30 replicates). For each subsampled ensemble, the performance objective values that resulted from all 500 scenarios are subsetted to correspond with the scenarios included in the subsample. Robustness and policy rankings are then reevaluated for each subsampled scenario ensemble.

2.3.2.2 Our definition of 'accurate ranking' using rank correlation

The policy rankings are compared to that of all 500 scenarios using Kendall's tau-b correlation, calculated using the Kendall package (McLeod, 2022) in R. To determine how many scenarios are needed for 'accurate ranking', we defined 'accurate ranking' as rank correlation of 0.975 or greater. We decided on this threshold based on visual inspection of scatter plots of subsample rankings vs all scenarios rankings (Quinn *et al.*, 2020) for varying values of correlation (Appendix A.2). These plots show the number of positions by which a policy is misranked relative to all scenarios. For some robustness metrics, scenario ensembles with correlation of 0.95 can produce rankings where many policies are tied (i.e., same robustness score), whereas fewer of those policies are tied when using all scenarios. For a correlation of 0.975 and greater, these erroneous ties are resolved and very similar ranking between the subsamples and all scenarios are observed, especially among the top 10 most robust policies. For example, Appendix A.2 shows that scenario ensembles with 0.975 correlation identify the same 10 policies as all scenarios as most robust, and, within the top 10, policies are ranked correctly or are misranked by only 1-3 positions.

We consider that scenario ensembles meeting the 0.975 target would accurately communicate to decision-makers the policies that are most robust but note that other studies could use a more lenient or strict threshold.

2.3.2.3 Calculation of space-filling metrics

The three space-filling metrics are calculated with the DiceDesign package (Dupuy, Helbert and Franco, 2015). Before calculating, the uncertain factors are scaled 0-1 to account for different units and magnitudes. Mindist is calculated using the function of the same name, and MSTmean and MSTsd are calculated using the mstCriteria function. The MST plots in Figure 2-1 were also created using the mstCritieria function with only slight modifications to allow for our chosen color palette. After these calculations, each scenario ensemble has three values for the space-filling measures and corresponding rank correlation values for six robustness metrics.

2.3.3 Calculation of linear models

Next, linear models are built to predict rank correlation as a function of the space-filling metrics. The models are fit with least-squares regression implemented in R using the lm function of the stats package (R Core Team, 2023). R^2 and 95% prediction intervals are calculated using the summary and predict functions, respectively. To check the assumptions of linear models – namely that residuals are homoscedastic and normally distributed with a mean of zero (James *et al.*, 2013, chap. 3) – we provide scatterplots of residuals vs predicted correlation in Appendix A.3.

2.4 Results

2.4.1 How many scenarios are needed for accurate robustness ranking?

To show how many scenarios are needed to accurately rank Lake Mead policies relative to all scenarios, Figure 2-3 shows boxplots of rank correlation vs. sample size. Results are shown as boxplots because each sample size has 30 replicates, as described in Section 2.2.1. Following standard definitions, the horizontal lines in the boxplots show the 1st quartile, median, and third quartile. The upper whiskers

show the maximum rank correlation less than the third quartile plus 1.5 times the interquartile range, while the lower whisker shows the minimum value greater than the first quartile minus 1.5 times the interquartile range. Points show outliers located beyond the whiskers (Wickham, Chang, *et al.*, 2023). The boxplot color indicates reliability (green), resiliency (orange), and vulnerability (purple). The top plot shows the results for the satisficing robustness type, and the bottom plot shows 90% regret from best. Lastly, the bold horizontal line marks our target correlation of 0.975. As discussed in Section 2.3.2.2, scenario ensembles meeting this threshold yield very similar rankings relative to all scenarios, especially among the top 10 most robust policies. In other words, scenario ensembles that meet this threshold would correctly communicate to decision-makers which Lake Mead policies are most robust to streamflow and demand uncertainties.

Fewer scenarios are required for satisficing than regret from best robustness types to achieve accurate ranking relative to all scenarios (Figure 2-3). For satisficing robustness metrics (Figure 2-3, top), ranking is accurate for as few as 50 to 300 scenarios, depending on the objective. *vulnerability.satisficing* requires 50 scenarios (except one outlier), *reliability.satisficing* requires 250 (except one outlier), and *resiliency.satisficing requires* 300. In comparison, 90% regret from best (Figure 2-3, bottom) requires 400 to 500 (all) scenarios for accurate ranking. For *reliability.regret* and *resiliency.regret*, most scenario ensembles with 350 scenarios yield accurate ranking – only outliers and the lower whiskers do not – while all ensembles with 400 or more scenarios obtain accurate ranking. For *vulnerability.regret*, none of the sample sizes yield rank correlations consistently greater than 0.975. Even with 450 scenarios, only the third quartile reaches the 0.975 threshold, meaning only about 25% of the replicates (7 to 8 out of 30 scenario ensembles) meet the threshold. Furthermore, there is a greater range of possible rank

correlation values for *vulnerability.regret* compared to all other metrics, as indicated by the height of the purple boxplot in Figure 2-3, bottom.

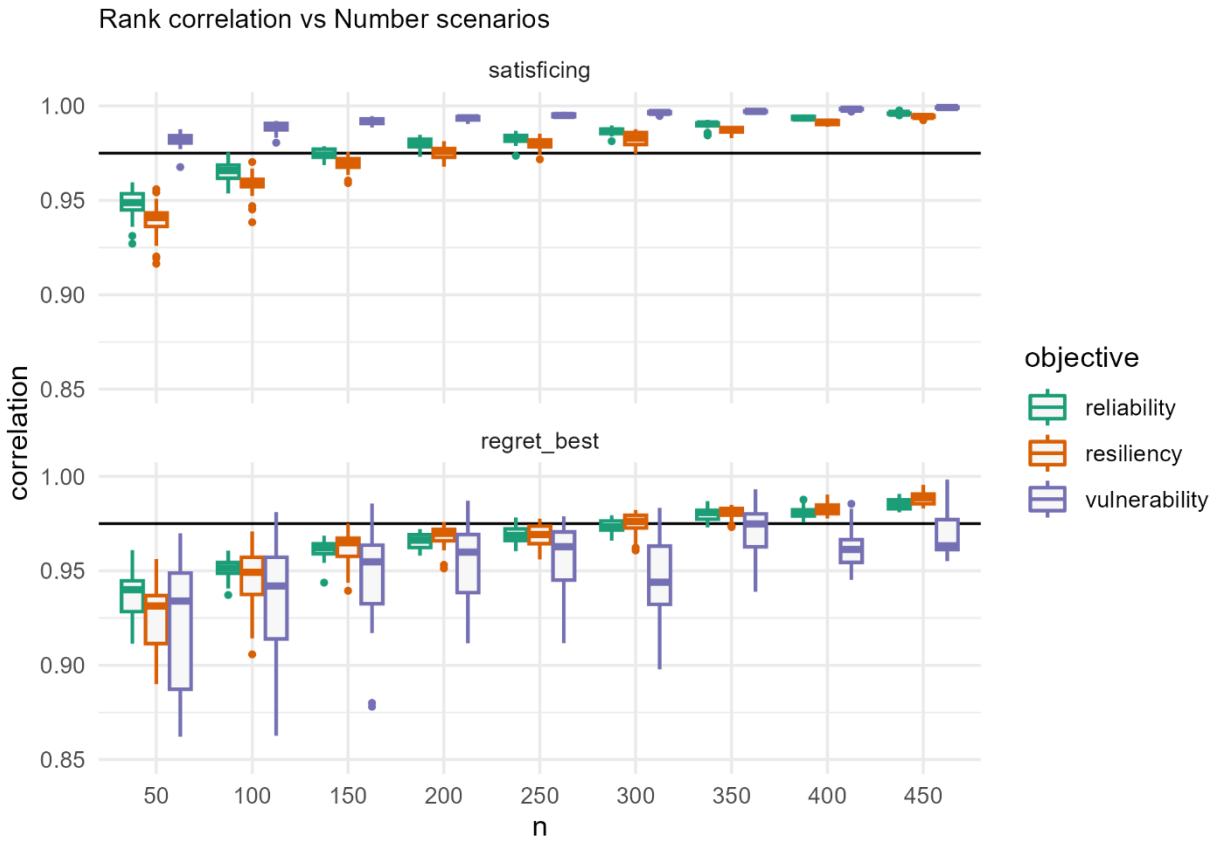


Figure 2-3: Boxplots of rank correlation vs number of scenarios (n) for each objective and robustness type.The top plot shows the results for the satisficing robustness metrics, and the bottom plot shows 90% regret from best. The boxplot color indicates the performance objective. We define rank correlation above the bold horizontal line (correlation = 0.975) as accurate ranking.

2.4.2 Are space-filling properties skillful indicators of rank accuracy?

The three space-filling metrics were calculated for each scenario ensemble, then linear models were fit between them and rank correlation to evaluate if space-filling metrics can accurately determine the number of scenarios required for a robustness analysis. In this section, we discuss the results for MSTmean only because the MSTmean models achieved larger R^2 than MSTsd or mindist for every robustness metric (Appendix A.4). The MSTmean models are shown in Figure 2-4. Recall from Section 2.2.2.3 that one model is built per robustness metric – six total. To measure how accurately MSTmean

predicts rank correlation, R^2 and prediction intervals for each robustness metric are reported in Figure 2-4, left. To compare each model's slope, which shows the rate at which rank correlation improves as MSTmean decreases, all six models are shown in one plot in Figure 2-4, right. Solid lines are satisficing, dotted lines are regret from best, and color indicates the performance objective. Note that the results in Figure 2-4 do not extrapolate to MSTmean values smaller than one. Such values would represent ensembles with more than 500 scenarios, and the linear regression models could predict correlation greater than 1, which is not meaningful.

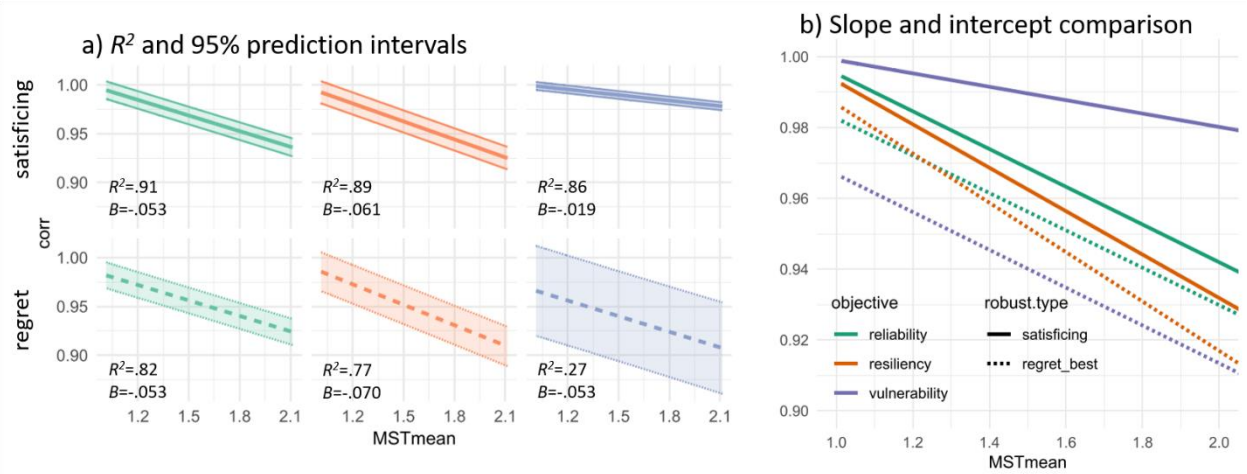


Figure 2-4: Models of rank correlation (y axis) as a function of MSTmean (x axis). a) R^2, 95% prediction intervals, and slopes for each robustness metric. The top row shows satisficing metrics, and the bottom row shows 90% regret from best. The colors are reliability (green), resiliency (orange), and vulnerability (purple). b) All six models are summarized in one plot to compare slopes and intercepts. Solid lines are satisficing, dashed lines are 90% regret from best, and color indicates the performance objective.

The model accuracy differs depending on both the objective type and the robustness metric type. R^2 and prediction intervals show that MSTmean models more accurately predict rank correlation for reliability compared to resiliency or vulnerability objectives and more accurately for satisficing compared to regret from best robustness types (Figure 2-4, left). R^2 varies from 0.27 to 0.91, with all but the least accurate model (*vulnerability.regret*) obtaining R^2 of 0.77 and greater. R^2 decreases from reliability to vulnerability (left to right) and from satisficing to regret from best (top to bottom). The prediction intervals demonstrate similar conclusions about model accuracy. Prediction intervals are smallest for reliability, larger for resiliency, and largest for vulnerability, shown by the height of the prediction intervals increasing

left to right. The exception to this pattern is *vulnerability.regret*, which has the smallest prediction interval. This exception is likely because rank correlation for this robustness metric only ranges from 0.97 to 1.00. Comparing robustness types, the prediction intervals are smaller for satisficing compared to regret from best.

Comparing the slopes of each model (Figure 2-4, right), the values are similar for all but one robustness metric – *vulnerability.satisficing*. This is shown qualitatively by the slopes of the lines in Figure 2-4, right, and the slope values, *B*, which are reported in Figure 2-4, left. The slope is the rate at which correlation is expected to increase per unit decrease in MSTmean. For instance, the slope of -0.053 for reliability satisficing means that rank correlation is expected to increase from about 0.94 to 0.99 if MSTmean is reduced from 2 (MSTmean equal to twice that of all scenarios) to 1 (MSTmean value of all scenarios). Across the models, slopes range from -0.053 to -0.070, except for *vulnerability.satisficing*, which has a slope of -0.019. The models also show that rank correlation is expected to be higher for satisficing compared to regret from best robustness types. This is seen by the solid lines (satisficing) being above the dashed lines (regret from best).

2.5 Discussion

2.5.1 Scenario subsampling can lessen the computational burden of robustness analyses

This research demonstrates that by subsampling scenarios it is possible to lessen the computational burden required for robustness analyses without sacrificing accuracy. Subsampling can be accomplished using observational sampling methods such as Kennard Stone sampling (Kennard and Stone, 1969), Feature Space Coverage Sampling (Brus, 2019; Wadoux, Brus and Heuvelink, 2019), or cLHS (Minasny and McBratney, 2006). These methods select a subset of scenarios that maximally cover the model uncertainty space. Observational sampling methods are an active area of research in the field of digital soil mapping (Schmidt *et al.*, 2014; Wadoux, Brus and Heuvelink, 2019; Ma *et al.*, 2020; Wadoux and Brus, 2021), from which additional methods may arise for subsampling scenarios. In the case study,

we used cLHS because it creates scenario ensembles with similar probability distributions of the uncertain factors. However, our framework can be implemented with other sampling methods to control additional properties of the ensemble, such as correlation between uncertain factors (Minasny and McBratney, 2006; Reis and Shortridge, 2021). Using cLHS, accurate ranking of the policies was obtained using ensembles that were subsampled from a set of 500 scenarios. These policies represent Lake Mead shortage policies that were generated in previous studies on the Colorado River Basin. The implication of this work is that fewer scenarios could be used in similar policy analyses supported by large simulation models, such as the upcoming policy renegotiation in the Colorado River Basin.

2.5.2 Effect of robustness metric on rank accuracy

Ensembles with 50 to 400 scenarios achieved accurate ranking for all robustness metrics except *vulnerability.regret*. Satisficing type robust metrics required fewer scenarios (50 to 300) compared to 90% regret from best (400 scenarios). The exception is *vulnerability.regret*: not even the largest sample size of 450 scenarios was able to meet our accuracy requirement (0.975 correlation or greater) for all replicates. These results are relevant for the CRB case study because, in the ongoing policy renegotiation, states, Tribes, water utilities, irrigation districts, environmental agencies, and copious other stakeholders are likely to propose their own policy alternatives, which will need to be tested for their robustness to uncertain streamflow and demand conditions (Reclamation, 2023e, 2023c). If the computational resources of a planning agency are limited, the results suggest that satisficing robustness metrics can obtain accurate robustness ranking with fewer scenarios. However, the results for both vulnerability-based metrics show that more scenarios and/or additional analysis are warranted, which we discuss in detail in the following subsections.

2.5.2.1 v*ulnerability.satisficing*: suggest iterative reformulation and multi-metric analysis

vulnerability.satisficing required the least number of scenarios for accurate ranking (50). However, many policies achieve the same robustness scores and rankings. Recall that satisficing measures

the fraction of scenarios in which a policy meets a performance threshold. We defined the vulnerability performance threshold as a maximum annual shortage of 1375 KAF or less, which is the maximum shortage under the current policies in the Colorado River Basin. Some of the policy alternatives tested in this study, however, implement a maximum annual shortage of less than 1375 KAF. Since the maximum allowable shortage is a decision variable, these policies meet the performance requirement in every scenario, achieving a satisficing score of one regardless of the scenario ensemble they are tested in. This means many policies obtain the same robustness rank, which may be unhelpful for decision-makers looking to prioritize a few of the most robust policies.

Although this result depends on the specific policies, vulnerability objective, and satisficing threshold, it highlights that robustness metrics must be crafted thoughtfully, iteratively, and considered in conjunction with other metrics. The results suggest that our *vulnerability.satisficing* metric is not helpful for ranking policies, something that can be difficult to know *a priori*. Other studies suggest that the performance threshold could be iteratively refined by exploring how ranking changes for different performance thresholds (Hadjimichael, Quinn, *et al.*, 2020), and/or that policy robustness could be further evaluated by exploring tradeoffs with respect to additional metrics (Woodruff, Reed and Simpson, 2013; Herman *et al.*, 2014; Bonham, Joseph Kasprzyk, and Edith Zagona, 2023). Such exploratory, iterative approaches may help decision-makers identify preferred policies even in the presence of ties and discover important insights that change which policies they prefer.

2.5.2.2 *vulnerability.regret*: ranking highly sensitive to scenario ensemble

vulnerability.regret resulted in highly sensitive policy rankings across the scenario ensembles. This result is demonstrated by the large range of rank correlation values, 0.86 to 0.99, shown in Figure 2-3 and Figure 2-4. This sensitivity may result from combining a vulnerability objective – maximum failure in one time step – with regret from best – which summarizes performance across scenarios by taking the 90th percentile. Because of this sensitivity, even our largest sample size (450 scenarios) failed to achieve

accurate ranking in the majority of scenario ensemble replicates. In our analysis, we kept the ranges and probability distributions between scenario ensembles approximately equal by using cLHS. Nevertheless, rank correlation varied substantially, even among scenario ensembles of the same size. This result is concerning because it suggests the ranking of policies is highly sensitive to exactly which scenarios are included in the scenario ensemble. In future robustness analyses in the Colorado River Basin, we caution against using this metric or similar in isolation as the basis for ranking polices. Again, the metric could be redefined and/or considered in conjunction with additional metrics. Other studies with similar metrics may also find that the ranking of alternatives is highly sensitive to the scenario ensemble used.

2.5.3 Space-filling metrics as indicators of rank accuracy

The results show a strong linear relationship between MSTmean and rank correlation. Although mindist and MSTsd also achieved strong R^2 values, MSTmean was superior for every robustness metric. Except for *vulnerability.regret*, MSTmean was able to explain between 77% and 91% of the variance in rank correlation. This is an interesting result because the linear models show that rank correlation increases at a similar rate for each robustness metric, between 0.05 and 0.07 per unit decrease in MSTmean (again, *vulnerability.regret* being the exception). If this relationship is generalizable, it could be used as a model-free method for planning agencies in the Colorado River Basin to determine the number of scenarios required for a robustness analysis. From Figure 2-4, an analyst could determine the required MSTmean to achieve a target rank correlation (e.g., 0.975). Then, they would compute MSTmean for a candidate ensemble of scenarios – if it is larger than the value pulled from Figure 2-4, then more scenarios would be needed.

There are several limitations to the MSTmean models. First, it is not known if these models are accurate for other case studies or if the linear relationship holds for a larger range of sample sizes. We tested 50 to 450 scenarios sampled from 500, but would these relationships hold if, for example, 100 scenarios were sampled from one million scenarios? Moreover, our results are based on subsampling

from a larger set of scenarios, not creating new scenario ensembles as done, for example, with Latin Hypercube Sampling. However, other studies suggest similar results could be expected so long as the ranges, correlations, and probability distributions are consistent in each scenario ensemble (McPhail *et al.*, 2020; Reis and Shortridge, 2020, 2021). Nevertheless, our simple linear regression models with MSTmean achieved admirable accuracy, demonstrating the potential usefulness of model-free measures of scenario ensemble quality to indicate if a given ensemble will yield accurate robustness rankings. Future studies could implement our framework with alternative statistical models, such as non-linear regression or regression trees.

2.6 Conclusion

Planning agencies often acknowledge that their systems are faced with climatic and/or socioeconomic uncertainty that cannot be reduced nor easily characterized (Knight, 1921; Kasprzyk *et al.*, 2013; Kwakkel and Haasnoot, 2019). To combat this uncertainty, they desire to capitalize on exploratory modeling methods such as multi-objective optimization, robustness, and vulnerability analyses (Brekke *et al.*, 2011; Molina-Perez *et al.*, 2019; Smith, Kasprzyk and Dilling, 2019; Smith *et al.*, 2022). However, such analyses are resource intensive, requiring substantial investment in technical skills, time, computational resources, and/or external consultation (Means *et al.*, 2010). This burden is compounded by the accelerating rate at which such methods are developed (Kasprzyk and Garcia, 2023) and by the recent trend of repeating these analyses for several scenario ensembles – as done in this current study, McPhail et al. (2020), Quinn et al. (2020), and Reis and Shortridge (2020, 2021).

In this research, we present a framework for testing the number of scenarios needed to achieve similar robustness ranking of policies compared to a larger scenario ensemble. Our framework uses a large ensemble of scenarios to evaluate the performance objective values of a set of policies in a simulation model, then ranks policies from most to least robust. Then, an observational sampling technique is used to create smaller scenario ensembles by subsampling from the larger ensemble. The performance

objective values corresponding to the sampled scenarios are then used to reevaluate the ranking of policies. By using this subsampling approach, our framework can determine the minimum number of scenarios required for robustness rankings to be sufficiently similar to the larger scenario ensemble without needing to perform additional model runs.

We demonstrated our framework with a case study of water shortage policies in the Colorado River Basin. We used two common robustness types (satisficing and 90th percentile regret from best) and three common performance objectives (reliability, resiliency, and vulnerability) to demonstrate that our framework can be easily adopted for other case studies. The results show that 50 to 400 scenarios are needed to accurately rank Lake Mead policies compared to a larger ensemble of 500 scenarios, depending on the robustness metric. The exception to this finding is the *vulnerability.regret* metric – only a small number of scenario ensembles resulted in accurate policy ranks with this metric, and the ranking was highly variable depending on the ensemble used, even comparing ensembles with the same number of scenarios.

Next, we investigated model-free methods to determine if a scenario ensemble will yield accurate ranking. Because robustness analyses can be resource intensive, it would be beneficial if there existed a method to test if a given scenario ensemble will yield accurate results without performing additional computer simulations. Space-filling metrics are potentially useful because they measure the quality of scenario ensembles based on distance calculations in the model-input space, meaning no computer simulations are required to evaluate them. Using our case study, we built linear models between three space-filling metrics and rank accuracy. The results show that the MSTmean metric is a particularly skillful indicator of rank accuracy, and that the rate at which rank accuracy improved as a function of MSTmean was similar for most robustness metrics. This research suggests that model-free measures of scenario ensemble quality may be a practical and skillful tool for determining how many scenarios are needed for robustness analyses.

We encourage future research to expand on our efforts, analyzing existing sets of scenario ensembles and performance objective outputs demonstrated in other studies (Quinn *et al.*, 2018; Gold *et al.*, 2019b; Hadjimichael, Quinn, *et al.*, 2020; Jafino and Kwakkel, 2021). Our case study used common robustness types and performance objectives, yet the numbers of scenarios required for each robustness metric and the regression models are likely to change for case studies with different uncertain factors, policy alternatives, simulation models, performance objectives, and robustness types. Because our framework subsamples existing model simulations, other studies could test the generalizability of our results and build new statistical models for their case study with relatively small computing time. For example, the analysis for our case study required 11 minutes of computing time on a laptop computer (tested on a Dell Latitude 5501). Future studies could also test additional sampling methods, robustness metrics, and different numbers of scenarios; test the number of scenarios needed for vulnerability analysis (e.g., (Kwakkel, 2019; Steinmann, Auping and Kwakkel, 2020)); and investigate other model-free methods to decrease the computing costs of DMDU analyses.

2.7 Software and data availability

All code and data to reproduce this study are available on GitHub: https://github.com/nabocrb/Scenario-subsampling-framework.

2.8 Acknowledgements

We would like to thank the Bureau of Reclamation's Research and Modeling Team for providing the Lake Mead objectives and decision variable data. We also thank the anonymous reviewers who contributed to the presentation of this article. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2040434.

Author Contributions:

- Nathan Bonham: conceptualization, analysis, coding, visualizations, writing-original draft
- Joseph Kasprzyk: conceptualization, writing-review and editing, supervision

- Edith Zagona: conceptualization, writing-review and editing, supervision

- Balaji Rajagopalan: analysis-review of linear models, writing-review and editing

3 Interactive, Multi-metric Robustness Tradeoffs in the Colorado River Basin

3.1 Introduction

The Colorado River Basin (CRB) is the preeminent source of water in the southwestern United States, serving roughly 40 million people and sustaining an annual economic impact of $1.4 trillion (Reclamation, 2012a; James *et al.*, 2014). Since 2000, historic drought conditions amidst relatively stable consumptive use has depleted Lake Mead and Lake Powell to roughly 25% of their capacity (Reclamation, 2023a). Full, these reservoirs hold roughly four times the historical annual streamflow. But, the current, low levels pose risks to water supply reliability, hydropower production, environmental health, recreational benefits, and more (Reclamation, 2007). To reduce these risks, the Bureau of Reclamation (hereafter, Reclamation) has established interim shortage policies whereby releases from the reservoirs are cut depending on Lake Mead's pool elevation (Reclamation, 2007). The current policy expires December 31st, 2025, and Reclamation is tasked with creating a new policy that balances the various benefits provided by CRB water.

Post-2026 planning typifies complex water resources management that is confronted with *deep uncertainty*. Deep uncertainty exists when decision-makers do not know or disagree on the probability distribution of the uncertain factors that drive system performance (Knight, 1921; Lempert, Popper and Bankes, 2003; Kwakkel and Haasnoot, 2019). Water resources systems commonly have deep uncertainties in future streamflow conditions and water demand (Kasprzyk *et al.*, 2013; Herman *et al.*, 2015; Smith, Kasprzyk and Basdekas, 2018). In the CRB, for instance, the 21st century drought (Salehabadi *et al.*, 2022), paleo-reconstructions (Gangopadhyay *et al.*, 2022; Salehabadi *et al.*, 2022), and climate-change projections (Lukas and Payton, 2020, chap. 11) suggest future streamflow may be less than 20th century observations. However, future projections vary widely, and the CRB exhibits large interannual variability, so periods of high streamflow are also possible (Lukas and Payton, 2020, chap. 11). Other uncertain

factors, such as demand upstream of Lake Powell, further obfuscate the CRB's future supply-demand balance (Upper Colorado River Commission, 2016).

Planning under deep uncertainty is further complicated by the presence of many stakeholders who often disagree on how to judge the performance of policy alternatives. These disagreements include the relative importance of conflicting performance goals (i.e., objectives) (Smith, Kasprzyk and Dilling, 2019). Moreover, given that the drivers of system performance are deeply uncertain, stakeholders may disagree on the appropriate methods to measure performance outcomes and the degree to which policy decisions should hedge against uncertainty-related risk (Hadjimichael, Quinn, *et al.*, 2020; McPhail *et al.*, 2021).

Several frameworks have been proposed to help identify policies that are *robust* to deep uncertainty (Ben-Haim, 2004; Lempert *et al.*, 2006; Brown *et al.*, 2012; Haasnoot *et al.*, 2013). In this research, we implement the Many Objective Robust Decision Making (MORDM) framework (Kasprzyk *et al.*, 2013). However, a common element across these frameworks is the explicit consideration of deep uncertainty by 'stress-testing' policy alternatives under many diverse future scenarios. These frameworks seek to identify robust policies, meaning they perform well across the scenarios, as quantified with robustness metrics. There exist many robustness metrics, which use various statistical transformations to summarize performance outcomes across the scenarios (McPhail *et al.*, 2018). Ideally, robustness metrics should allow for rank-ordering to choose policies; in other words, a policy that exhibits the best performance with respect to a robustness metric is deemed "most robust" (Herman *et al.*, 2014; Alexander, 2018; McPhail *et al.*, 2020).

A body of research has shown that the policy deemed most robust often varies depending on which robustness metric is used (Herman *et al.*, 2015; McPhail *et al.*, 2020; Reis and Shortridge, 2020). McPhail et al (2021) addressed the choice of robustness metrics using a questionnaire to solicit the stakeholder's understanding of the problem (e.g., performance requirements) and decision preferences

(e.g., risk tolerance), and then algorithmically recommend robustness metrics based on their statistical properties (McPhail *et al.*, 2021). Such mapping of stakeholder preferences to robustness metrics is a form of *a priori* decision making since it is done before alternatives' performance outcomes are investigated (Kasprzyk *et al.*, 2013; Kwakkel and Haasnoot, 2019). Complex human-environmental systems, though, can result in incomplete and inaccurate understanding of how decisions (such as choice of robustness metrics, objectives, and/or policies) can lead to undesirable performance/robustness outcomes (Zeleny, 1989; Roy, 1990; Kasprzyk *et al.*, 2012; Woodruff, Reed and Simpson, 2013; Herman *et al.*, 2014). In other words, basing policy decisions on *a priori* preferences can leave critical tradeoff information undiscovered, thus depriving stakeholders of valuable insights that could change what robustness metrics and, ultimately, policies they choose.

Alternatively, *a posteriori* decision support helps stakeholders choose policies *after* exploring what performance outcomes are possible (Kollat and Reed, 2007; Kasprzyk *et al.*, 2013; Woodruff, Reed and Simpson, 2013). These methods use interactive visualization techniques that allow the stakeholder to explore performance tradeoffs, learn the relationships between decisions and outcomes, and iteratively refine their preferences. *A posteriori* methods are frequently used to explore tradeoffs between multiple objectives, but have seen limited applications for robustness analysis (e.g., Herman et al. (2014, 2015)).

This research demonstrates a real-world robustness tradeoff analysis for the Colorado River using *a posteriori* decision support. For each performance objective, a broad selection of robustness metrics is calculated to reflect varying degrees of risk-tolerance and different methods to summarize performance over future scenarios. Then, we provide stakeholders with background information and training to interpret what the metrics mean, rather than propose what metrics are most appropriate. We provide interactive visualizations for them to explore tradeoffs between robustness metrics and objectives, discover relationships between decisions and outcomes, iteratively refine which metrics and objectives are important, and remove non-robust policies. Doing so, we introduce several novel tools for *a posteriori*

decision support, namely 'on-the-fly' non-dominated robustness sorting and global linking across multiple web pages of robustness metric visualizations. Our dynamic and interactive framework requires an integrated platform, which we create via a web application (app). The code is open source, and although CRB data are included in the app, users can adapt this for their own purposes by changing the app's underlying database. We hypothesize that our framework can reveal salient performance tradeoffs that refine which objectives and robustness metrics stakeholders include in their analysis, avoiding undesirable and unexpected performance outcomes.

We test this hypothesis using a case study of Lake Mead shortage policies. Previous research in the CRB has used MORDM to generate Lake Mead shortage policies and identify policies that are robust with respect to a single type of robustness metric (Alexander, 2018; Bonham, J. Kasprzyk and Zagona, 2022a; Smith *et al.*, 2022). This paper will expand on these efforts to demonstrate our novel *a posteriori* framework.

3.2 Methods

3.2.1 Many Objective Robust Decision Making

MORDM consists of four steps (Kasprzyk *et al.*, 2013). First, the decision problem is framed in terms of uncertain factors, decision variables, a simulation model, and objectives (Lempert, Popper and Bankes, 2003; Lempert *et al.*, 2006). Uncertain factors are exogenous factors outside the control of decision makers, such as precipitation or streamflow. Decision variables (DVs) describe the management options decision makers can control to achieve their desired system goals. A policy is defined by a set of values for the DVs. The simulation model is used to evaluate multiple performance objectives, which measure the performance outcomes of a policy given specified values for the uncertain factors.

After formulating the problem, tens to hundreds of policies are generated using multi-objective simulation-optimization (Hadka and Reed, 2013; Maier *et al.*, 2019). In this step, an optimization algorithm, is coupled with the simulation model in a loop. The optimization algorithm suggests a new

policy, then the simulation model evaluates the performance objectives and returns that information to the optimizer that uses it to improve the policy for the next iteration. At this step in MORDM, the model is forced with a narrow set of assumptions about uncertain factors, often using historical values (Kasprzyk *et al.*, 2013; Alexander, 2018). This loop occurs for thousands of iterations to 'evolve' better performing policies using techniques inspired by nature (e.g., survival of the fittest, genetic crossover, mutations). The output is a set of non-dominated policies. A policy is non-dominated if, when compared to any other policy, it is better in at least one objective. The resulting policies exhibit tradeoffs, where improving performance in one objective necessitates inferior performance in one or more other objectives.

After generating the set of non-dominated policies, deep uncertainty is explicitly considered in robustness analysis (Herman *et al.*, 2015; McPhail *et al.*, 2018). Each policy is stress-tested in several hundred States of the World (SOW), where each SOW is a plausible future realization of the uncertain factors. The range of values for the uncertain factors is greatly expanded compared to that used in the optimization step, often encompassing the range of values observed in paleo-reconstructions and future climate projections (Alexander, 2018; Quinn *et al.*, 2020; Reis and Shortridge, 2020). A policy is robust if it performs well across the SOW for specified objectives. Robustness is quantified with one or more robustness metrics, which are statistics that summarize a policy's performance across the SOW. Stakeholders use the values of the robustness metrics to choose one or more policies of interest, often-times via rank ordering (Alexander, 2018; McPhail *et al.*, 2020).

In the last step of MORDM, one or a small number of robust policies are subject to vulnerability analysis (Bryant and Lempert, 2010). Vulnerability occurs when a policy fails stakeholder-defined performance thresholds. Vulnerability analysis uses statistical learning to discover the uncertain factors that are the strongest predictors of vulnerable outcomes and the corresponding values under which vulnerability occurs (Jafino and Kwakkel, 2021). The current study is concerned with how stakeholders choose robustness metrics and identify robust policies; we do not perform vulnerability analysis. Instead,

the framework presented in this research helps identify a small number of robust policies that would then be subject to vulnerability analysis.

3.2.2 Learning decision preferences by exploring robustness tradeoffs

There exist many robustness metrics (McPhail *et al.*, 2018, 2020), so a major challenge to finding robust policies is determining which metric(s) to use. Metrics include, for example, worst-case performance in the SOW ensemble (maximin), regret from best possible performance (regret from best), and fraction of SOW for which performance thresholds are satisfied (satisficing) (Herman *et al.*, 2015; McPhail *et al.*, 2018). Importantly, the choice of robustness metric is non-trivial because the ranking of policies can be sensitive to robustness metric selection (Herman *et al.*, 2014; McPhail *et al.*, 2018; Reis and Shortridge, 2020).

To help stakeholders choose robustness metrics, McPhail et al. (2021) contributed a guidance framework that recommends robustness metrics based on the stakeholder's response to a series of questions. Some of the questions include, for example: Does a "meaningful threshold or level of performance exist?", and is it "most important to minimize magnitude of failure, or maximize the number of scenarios [i.e., SOW] with acceptable performance?" Overall, the questions solicit the stakeholder's understanding of the decision problem (e.g., performance thresholds) and their decision preferences (e.g., risk tolerance). Then, the framework recommends one or more robustness metrics. For instance, if a stakeholder has established performance thresholds and wants to maximize the number of SOW with acceptable performance, then the framework suggests the satisficing metric. Alternatively, if a stakeholder is highly risk averse, the framework could suggest the maximin metric. In the McPhail et al. (2021) framework, the stakeholder's initial interpretation of the decision problem and decision preferences drive the selection of robustness metrics and policies.

In contrast, others have demonstrated that, in complex environmental systems, decisions made on the basis of *a priori* preferences can result in undesirable tradeoffs and poor robustness in alternative

metrics (Zeleny, 1989; Roy, 1990; Kasprzyk *et al.*, 2012; Woodruff, Reed and Simpson, 2013). Instead, these studies advocate for *a posteriori* decision support, wherein the stakeholder's understanding of the problem and their decision preferences are iteratively refined by exploring tradeoffs between multiple metrics (Kwakkel and Haasnoot, 2019). For instance, in a workshop with nine water managers, Smith et al. (2019) observed that seven managers changed their selection of policies when given tradeoff information for four objectives compared to two. Similarly, Herman et al. (2015) showed that the most robust alternative according to a regret metric was one of the least robust according to a satisficing metric. These studies underscore that stakeholder preferences are contextual, meaning they evolve as stakeholders discover what performance tradeoffs exist (Brill *et al.*, 1990; Woodruff, Reed and Simpson, 2013). Indeed, the McPhail et al. (2021) framework acknowledges that such tradeoffs may exist (see also McPhail et al. (2018, 2020)). However, the choice of metric(s) is still determined from the stakeholder's responses to the questionnaire.

To help stakeholders discover pertinent tradeoffs, this research contributes an *a posteriori* framework for robustness analysis, as shown in Figure 3-1. First, we test policies in many SOW and calculate a broad selection of robustness metrics. The metrics reflect varying degrees of risk-avoidance and use different methods to summarize performance over the SOW ensemble. We then provide training on the robustness metrics and the case-study's performance objectives. The training material provides foundational knowledge and reference material that empowers stakeholders to iteratively explore tradeoffs, refine preferences, and choose policies using a dynamic web application, the details of which are given below.

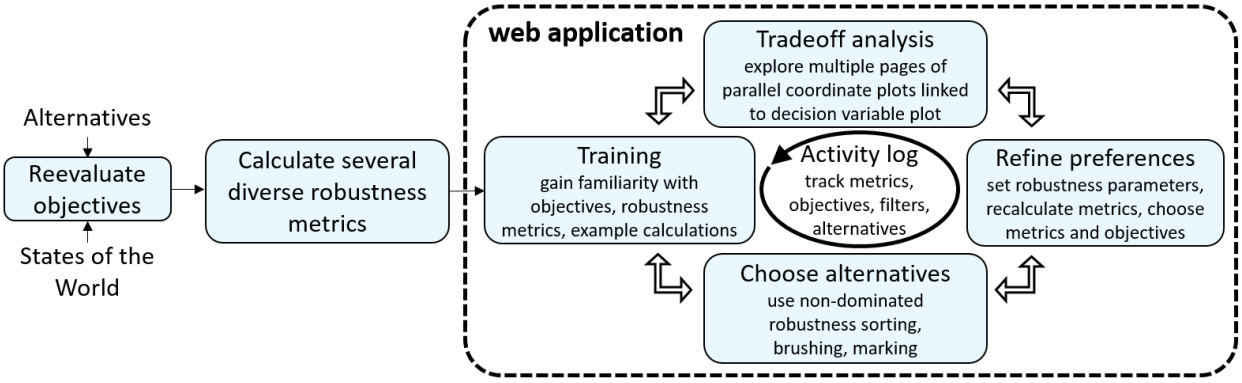


Figure 3-1. Overview of the *a posteriori* robustness framework proposed in this research.

3.2.3 Linking interactive plots of robustness metrics and decision variables

Parallel coordinate (PC) plots are a proven tool for exploring tradeoff information (Inselberg, 2009; Herman *et al.*, 2014; Smith, Kasprzyk and Basdekas, 2018). In a PC plot, objectives are plotted as a series of parallel axes, and policies are shown as individual traces crossing the axes at their respective performance in each objective. Tradeoffs are shown wherever traces cross between two adjacent axes. PC plots are interactive with features like 'brushing' (highlighting policies that meet performance goals), reordering of axes (to explore tradeoffs between different pairwise-combinations of objectives), and 'marking' (highlighting policies by selecting them in a data table) (Raseman, Jacobson and Kasprzyk, 2019). These features enable stakeholders to rapidly select and remove policies. Notably, PC plots were used in the aforementioned workshop of water managers, who effectively used them to interpret tradeoffs between two to five objectives and choose their preferred policies (Smith, Kasprzyk and Dilling, 2019).

However, there have been limited applications of PC plots to compare across several different types of robustness metrics. In a study of four interconnected water utilities, Herman et al. (2014) calculated the satisficing metric for each utility's respective performance thresholds (resulting in four metrics), then used a PC plot to demonstrate tradeoffs between them. In a follow-up study, Herman et al. (2015) expanded their tradeoff analysis to consider two versions each of satisficing and regret-based

robustness metrics (also a total of four metrics). Building on these studies, this research uses PC plots to explore tradeoffs between over 50 robustness metrics, the result of calculating several types of robustness metrics for every performance objective. A PC plot with such large number of axes would be difficult to interpret due to visual clutter (Raseman, Jacobson and Kasprzyk, 2019). Therefore, we organize the robustness metrics using multiple *linked* PC plots, meaning a stakeholder can select/remove policies using one PC plot (corresponding to one type of robustness metric) and their selection is automatically shown on the other PC plots. By greatly expanding the number of robustness metrics and objectives, we enable the stakeholder to explore what robustness/performance outcomes are possible and refine their decision preferences.

Our framework also capitalizes on previous work that has linked PC plots to plots of DV values (Kollat and Reed, 2007; Raseman, Jacobson and Kasprzyk, 2019; Smith, Kasprzyk and Dilling, 2019). In these, as stakeholders use PC plots to select policies according to performance outcomes, the DV plot is updated to show the corresponding DV values. DV plots can use case-study specific figures to illustrate the DV values, such as reservoir operation diagrams (Alexander, 2018; Bonham, J. Kasprzyk and Zagona, 2022a), or can use generic, high-dimensional figures like PC plots (Smith, Kasprzyk and Dilling, 2019). Building on these previous studies, our framework links each PC plot of robustness metrics/objectives to a DV plot, helping the stakeholder see what decisions led to specific robustness outcomes.

3.2.4 A web application for dynamic decision support

Central to our framework is iterative and rapid exploration. So, in addition to interactive visualizations, stakeholders dynamically redefine robustness parameters, such as performance thresholds and risk tolerances. After exploring outcomes of their choices, stakeholders can add their preferred metrics and objectives to a custom PC plot. To identify policies that perform well with respect to their selection, we introduce ‘on-the-fly’ non-dominated robustness sorting, which extends the traditional use of non-dominance for optimization problems to user-selected robustness metrics (demonstrated below).

Our framework allows for many-iteration and unique robustness analyses. To make each analysis reproducible and easily communicated, our framework records which robustness metrics, objectives, and filters are used with an activity log, an example of visual analytics *provenance* (Ragan *et al.*, 2016; Chakhchoukh, Boukhelifa and Bezerianos, 2022).

To accommodate these decision support methods, this research contributes an interactive web app with functionality that differs from existing MORDM-related software. The Exploratory Modelling Workbench (Kwakkel, 2017), Rhodium (Hadjimichael, Gold, *et al.*, 2020), and OpenMORDM (Hadka, 2015) provide functions for established MORDM tasks such as optimization, creating SOW, robustness simulations, and vulnerability analysis. In contrast, the focus of our app is interactive visualizations and filtering methods necessary for our robustness framework. The RAPID package calculates robustness metrics using the guidance given in McPhail et al. (2021), as described earlier. In contrast, our app demonstrates an alternative robustness framework. Lastly, Parasol is a library to create web apps of linked PC plots (Raseman, Jacobson and Kasprzyk, 2019). Our software is different first because it is an app that demonstrates our novel robustness analysis, not a library to create apps. As such, our app performs robustness calculations and non-dominated sorting, features beyond the scope of Parasol. Further, to our knowledge, our app is the first demonstration of simultaneously linking multiple pages of PC and DV plots.

3.3 Case study: shortage operations in the Colorado River Basin

In this section, we discuss the CRB decision problem used to demonstrate our framework. First, we explain how we used simulation-optimization to generate Lake Mead shortage policies. Then, we describe the SOW ensemble we used to evaluate those policies under various streamflow and demand conditions. From those simulations, we calculate eight types of robustness metrics. Finally, we demonstrate how our robustness framework discovers insightful tradeoffs that iteratively refine the robustness metrics, objectives, and policies of interest.

3.3.1 Multi-objective optimization of Lake Mead policies

This case study considers shortage operation policies for Lake Mead. In 2007, Reclamation established interim guidelines that define pool elevations and corresponding volumes by which deliveries to the Lower Basin (LB) would be reduced during times of low reservoirs levels. The guidelines also determined how Lake Powell, the primary reservoir for the Upper Basin (UB), would be operated in coordination with Lake Mead. These guidelines are intended to balance storage related objectives, such as protecting hydropower-related pool elevations, and delivery objectives, such as meeting LB demand. After 2007, however, the drought continued, and storage in both reservoirs has continued to decline. So, additional delivery reductions were added to the guidelines via a US-Mexico agreement (International Boundary and Water Commission, 2012, 2017) and a LB drought contingency plan (Buschatzke *et al.*, 2019). Together, these policies result in a cumulative delivery reduction volume at Lake Mead. Hereafter, we refer to these delivery reductions as a 'shortage policy'. The current policy expires December 31st, 2025, thereafter a new policy will take effect.

In this research, we created alternative shortage policies using multi-objective simulation-optimization. Consistent with the current policy, each policy is defined by a vector of DV values for the pool elevations at which shortages begin and corresponding shortage volumes. To demonstrate, Figure 3-2. shows three hypothetical policies, demonstrating how the generated policies can vary in the number of shortage tiers, shortage volumes (T1V-T6V), and pool elevations (T1e-T6e). For the optimization, we used the Borg evolutionary algorithm (Hadka and Reed, 2013) coupled with the Colorado River Simulation System (CRSS). CRSS is a hydro-policy model built in RiverWare (Zagona *et al.*, 2001) that Reclamation uses for long-term planning. We used CRSS to evaluate eight objectives that describe performance for the UB and LB in terms of water storage and deliveries (Table *3-1*). For further details, we refer the reader to Alexander (2018). The result of the optimization is 463 shortage policies.

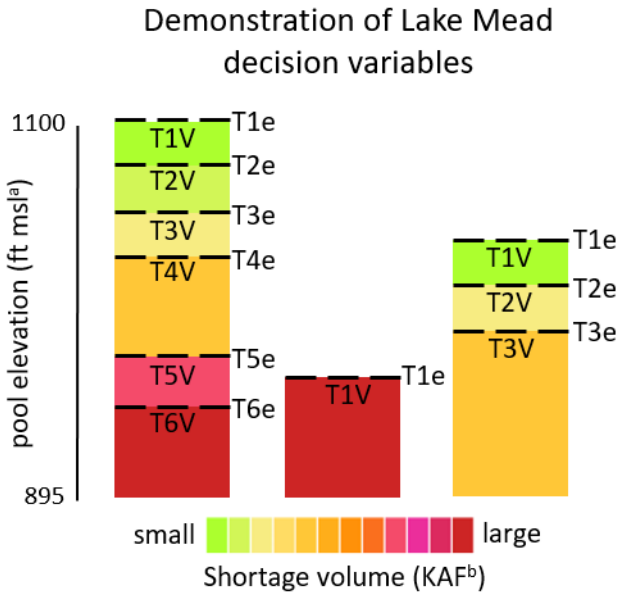


Figure 3-2. Example Lake Mead shortage policies to illustrate the decision variables used in optimization. [a]feet above mean sea level, [b]thousand acre-feet.

Objective	Units	Description
Upper Basin		
LF.Deficit	%	Minimize % of time that annual 10 year compact volume falls below 75 MAF[a] at Lees Ferry
P.WYR	MAF[a]	Minimize cumulative average annual Water Year release from Lake Powell
P3490	%	Minimize % of time that monthly Lake Powell Pool Elevation is less than 3,490'
Lower Basin		
M1000	%	Minimize % of time that monthly Lake Mead Pool Elevation is < 1,000'
LB.Avg	KAF[b]	Minimize the cumulative average annual Lower Basin total shortage volume
LB.Freq	%	Minimize the frequency (% of time) that the system is in an annual shortage operation
LB.Max	KAF[b]	Minimize the maximum annual Lower Basin policy shortage volume
LB.Dur	Years	Minimize the maximum duration of consecutive years in shortage operation

[a]million acre-feet, [b]thousand acre-feet

Table 3-1: Optimization objectives

3.3.2 Future scenarios of streamflow, demand, and initial reservoir storage

To calculate robustness metrics, we first reevaluate each shortage policy in many future SOW. For this analysis, we used conditioned Latin Hypercube Sampling (Minasny and McBratney, 2006) to select a subset of 500 SOW from a larger set, the details of which are given in Bonham et al. (2022a). The uncertain factors are streamflow, demand, and initial storage. For streamflow uncertainty, we sample annual natural flow at Lees Ferry, Arizona from a combination of historical observations, paleo-reconstructions, and CMIP-3 based climate change projections (Reclamation, 2012a). Each SOW assumes a value for demand in the UB ranging from 4.2 to 6.0 million acre-feet (MAF), which accounts for both curtailments and growth (Upper Colorado River Commission, 2016). Lastly, each SOW assumes initial pool elevations at Lake Mead, ranging from 1000 to 1185 feet above mean sea level (ft msl) (16 to 76% capacity), and Lake Powell, ranging from 3450 to 3675 ft msl (18 to 85% capacity) (Reclamation, 2011, 2020; Root and Jones, 2022).

3.3.3 Robustness metrics

Our case study includes eight types of robustness metrics as summarized in Figure 3-3. We selected these metrics because they use various methods to summarize performance over the SOW ensemble and reflect different degrees of risk-tolerance demonstrated in the literature (Kasprzyk *et al.*,

2013; Herman *et al.*, 2014, 2015; McPhail *et al.*, 2018, 2021). The left half of Figure 3-3, *Description*, provides conceptual definitions and guidance on interpreting robustness values. The right half, *Calculation*, describes how the metrics are calculated using a taxonomy adapted from McPhail et al. (2018). Metrics highlighted with an asterisk require stakeholder-defined parameters (e.g., performance thresholds or percentiles), which can be iteratively defined in app. Figure 3-3 is taken directly from the CRB robustness app's *For Reference* page, which also provides example calculations. Additional details are given in the 'For Reference page' section below. We use this broad selection of robustness metrics and adjustable parameters to facilitate extensive tradeoff exploration.

Description				Calculation			
Category	Name	Definition	Interpretation	Transformation of objective	SOW used	Normalization factor	Summary statistic
threshold	*Satisficing	The fraction of SOW where a given policy satisfies user-defined performance thresholds.	1 indicates the performance thresholds have been satisfied in every SOW, whereas 0 indicates the thresholds are violated in every SOW.	Satisfactory (1) or Not Satisfactory (0)	All	None	mean
threshold & regret	*Satisficing deviation	The average percent by which a policy satisfies a user-defined performance threshold.	Negative percentages indicate performance better than the threshold, and positive indicates performance worse than the threshold.	Deviation from satisficing threshold	All	Threshold	mean
regret	Regret from best	The average deviation of a solution's performance in each SOW from the best performance obtained by any policy in each SOW.	0 percent indicates the policy is the best performing policy in every SOW, and greater positive values indicate larger average regret compared to the best performing policies.	Deviation from best performance by any policy in each SOW	All	None	mean
	Percent deviation from optimization	The percent by which the 90th percentile performance of a solution deviates from its performance during optimization.	Positive percentages indicate the policy performs worse during robustness simulations compared to optimization simulations, whereas negative values means the policy performs better.	Deviation from optimization performance	90th percentile	Optimization performance	None
No objective transform	Mean (Laplace's Principle of Insufficient Reason)	The performance averaged over the SOW ensemble.	The units and interpretation of each performance objective are maintained (e.g. smaller values are desired for minimization objectives)	None	All	None	mean
	*Hurwic'z optimism-pessimism	The weighted average of the best and worst case performance.	The units and interpretation of each performance objective are maintained (e.g. smaller values are desired for minimization objectives)	None	Best and worst-cases	None	weighted mean
	Mean-variance	The average performance multiplied by the standard deviation (in the case of minimizatin objectives).	The units are that of the objective squared. Small values are desirable, indicating small (good) performance that is consistent (low standard deviaiton) across the SOW.	None	All	None	mean x standard deviation
	*Maximin	The worst performance obtained by a solution in the SOW ensemble. Alternatively, other percentiles can be used (95th or 90th percentile performance, for example).	The units and interpretation of each performance objective are maintained (e.g. smaller values are desired for minimization objectives)	None	Worst-case (or user-defined percentile)	None	None

Example calculations

Summary table | Satisficing | Satisficing deviation | Regret from best | % deviation from optimization | Mean | Hur

Figure 3-3. Screenshot from the CRB robustness app summarizing the included robustness metrics with example calculations.

3.3.4 Example robustness analysis

Our demonstration of the CRB robustness app occurs in two phases. The first phase uses a single type of robustness metric, satisficing, following a previous study (Alexander, 2018). For a policy to be robust in this phase, it must obey M1000 < 10%, P3490 < 5%, and LB.Avg < 600 thousand acre-feet (KAF). These thresholds protect critical reservoir levels while maintaining small average shortages. See Table *3-1* for additional details. We calculate the satisficing metric for each performance threshold (resulting in three metrics), instead of aggregating the performance thresholds into one metric as done in Alexander (2018). Then, we explore tradeoffs between the three metrics and use non-dominated sorting to select the best performing policies. We use this phase of the analysis to establish a baseline set of policies that are robust with respect to predetermined performance thresholds and a single robustness metric type, but that might not be robust with respect to additional metrics available in the app. In the demonstration below, we call this phase 'non-dominated sorting with existing performance thresholds'.

In the second phase, we take the remaining policies from the first-phase and explore tradeoffs with additional objectives and robustness metrics. We use multiple pages of linked PC plots to refine our robustness preferences and reduce the number of remaining policies based on the tradeoffs we discover. We call this phase 'refining robustness preferences'. The analysis demonstrates the extent to which adding different robustness metrics will lead to choosing different policies.

3.4 Results

3.4.1 Accessing the app

To use our app, the stakeholder simply clicks on the following url: https://nabocrb.shinyapps.io/CRB-Robustness-App-JWRPM/. The only software requirements are a web browser and internet connection. The app is built with the R language (R Core Team, 2022) and depends on several open source packages for the graphic user interface (Chang *et al.*, 2022; Sievert *et al.*, 2023), interactive visualizations (Sievert *et al.*, 2022; Wickham, Chang, *et al.*, 2023), and data management

(Wickham, François, *et al.*, 2023). We encourage the interested reader to open the tool and follow along with this demonstration. Please note the app will disconnect from the server after 40 minutes of inactivity.

It will be useful to define several terms and provide a brief overview of the graphic user interface. A screenshot of the app is shown in Figure 3-4. We use the term page to refer to the options provided in the blue ribbon at the top of Figure 3-4 (e.g., *For reference* page). The *For reference* and *Optimization objectives* pages include subpages which are accessed with tabs (e.g., the *Mean* tab on the *Robustness metrics* page). Lastly, we use the term buttons to refer to actions taken by the stakeholder in the left sidebar (e.g., *Download* button).

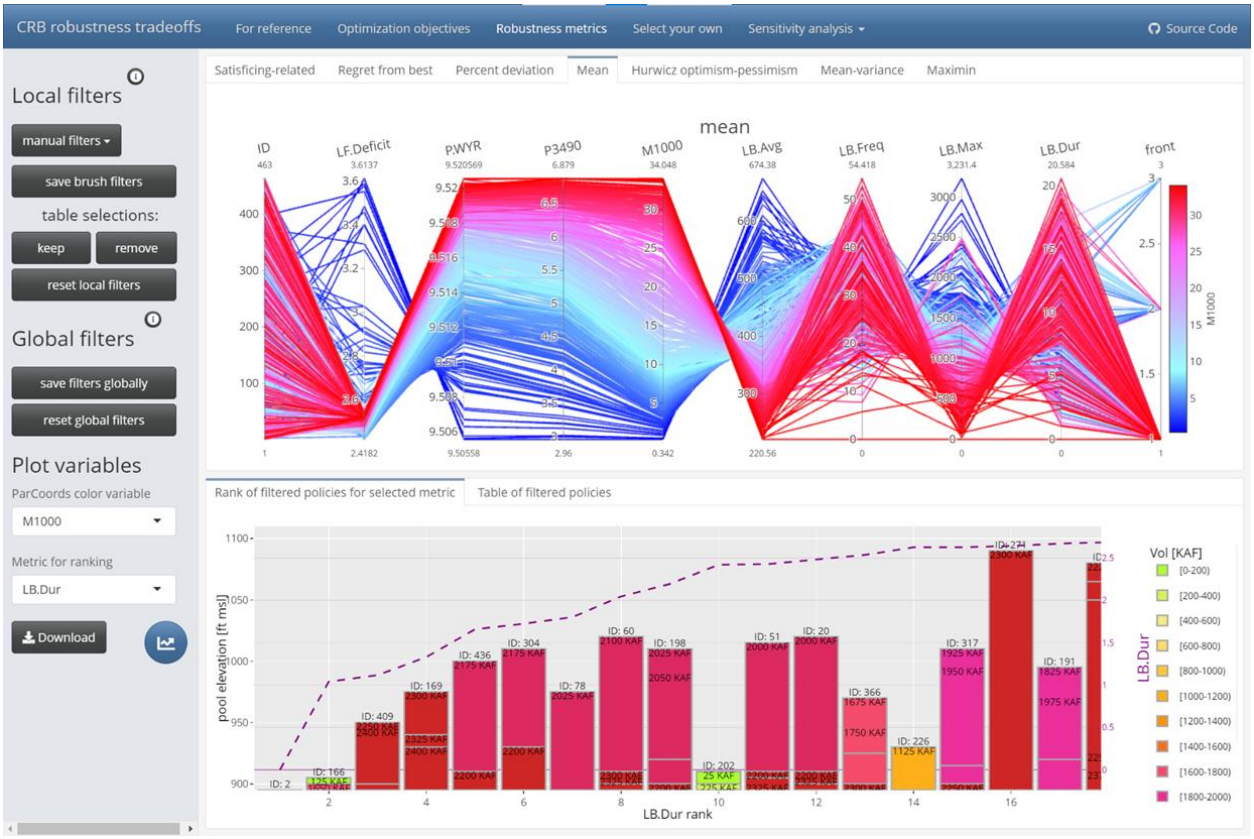


Figure 3-4. A screenshot of the CRB robustness app to demonstrate the user interface.

3.4.2 Parallel coordinate plots and operation diagrams

The *Optimization objectives*, *Robustness metrics*, and *Select your own* pages use the same layout with PC plots and DV diagrams. In the PC plot (Figure 3-4, top), the left-most vertical axis shows the unique ID for each policy, and the other axes correspond to the performance objectives in Table *3-1*. Depending on which page of the app the user is on, the values shown on the axes correspond to either the performance during optimization (*Optimization objectives* page) or a robustness metric (*Robustness metrics* page). The right-most axis, labelled 'front', shows each policy's non-dominated front with respect to the selected robustness metric. A demonstration is given below in the section 'Phase 1: non-dominated sorting'. The axes are oriented such that the best performance is always downward (i.e., the best policy would be shown with a straight line across the bottom). The PC plots allow for interactive brushing and reordering of axes. The color shows the value of a stakeholder-selected metric or DV (selected with the *ParCoords color variable* button). For instance, blue traces in Figure 3-4, top, indicate policies that are robust with respect to the M1000 objective and mean robustness metric, whereas red traces indicate poor robustness.

The PC plot is linked to a plot of Lake Mead policies (Figure 3-4, bottom). The y-axis shows the pool elevation at which LB shortages begin (T1e-T6e in Figure 3-2), and the color shows the shortage magnitude (T1V-T6V in Figure 3-2). Policies are ranked left to right according to a stakeholder-selected objective or DV. For instance, the policies in Figure 3-4 are ranked according to the maximum duration the LB experiences shortage conditions (the LB.Dur objective, Table *3-1*). The top 20 policies are shown by default, and the stakeholder can zoom or scroll to view more. The second y-axis (on the right) and the dashed purple line show the magnitude of the selected objective. This axis can be rescaled by clicking and dragging along it to improve readability, which was done to produce Figure 3-4. The magnitude is helpful for illustrating how similar/dissimilar performance is between ranks.

3.4.3 For reference page: foundation for exploration

Upon opening the app, the stakeholder reviews the *For reference* page to establish familiarity with the objectives and robustness metrics. On this page, the *experimental design* tab describes the DVs and objectives and how we used CRSS to reevaluate policies in 500 SOW. The *robustness metrics* tab opens a pdf that provides background on the definition of robustness and how robustness metrics are calculated. The final tab, e*xample calculations*, opens a Google Sheet that contains a summary table of the robustness metrics and example calculations as shown in Figure 3-3. After reviewing this material, stakeholders explore tradeoff information to learn what objectives and robustness metrics are of interest.

3.4.4 Phase 1: non-dominated sorting with existing performance thresholds

In this demonstration, we create custom robustness metrics based on the performance thresholds described earlier (from Alexander 2018). To insert our performance requirements, we navigate to the *Robustness metrics* page and the *satisficing-related* tab, and then select the *Satisficing calcs* dropdown button. Under *Select objectives(s)*, we select M1000, P3490, and LB.Avg. Then, we use the slider buttons to set the performance thresholds of 10%, 5%, and 600 KAF, respectively. After pressing the *Calculate* button, the PC plot is updated to show the satisficing and satisficing deviation metrics calculated with our user-defined performance thresholds.

Next, we use on-the-fly non-dominated sorting to select policies that perform well with respect to our custom satisficing metrics. We describe the process below, and the result is shown in Figure 3-5. First, we navigate to the *Select your own* page and click the *select metrics* button. We select our satisficing metrics by choosing *satisficing* under the *metric 1 type* button, then selecting sat.M1000 with the *metric 1:* button. We repeat this process to select sat.P3490 and sat.LB.Avg under the auto-created *metric 2:* and *metric 3:* buttons. After clicking *apply selected metrics*, our metrics are added to the PC plot. The satisficing metrics are labelled with the prefix ‘satisficing.sat’ (e.g., satisficing.sat.M1000). A value of one means the performance threshold is satisfied in all SOW, and zero indicates the requirement is never satisfied (see

Figure 3-5). Next, we select the *Calculate fronts* button, which adds an axis to the far right labeled 'front'. A policy belongs to the first front, if it is non-dominated with respect to the three satisficing metrics. To select the non-dominated policies, we use a brush filter by clicking and dragging over front=1 (see the heavy pink line at 1 on the front axis in Figure 3-5). Note that sat.P3490 is always greater than 0.7, but sat.M1000 and sat.LB.Avg are as low as 0.31 and 0.51, respectively. We choose to balance the three satisficing metrics, so we use brushing to select policies with at least 0.7 for both sat.M1000 and sat.LB.Avg. We used brush filters in this example, but the stakeholder can manually type thresholds using the *manual filters* button in the left sidebar (Figure 3-4). After selecting the *save brush filters* button, 14 policies remain.

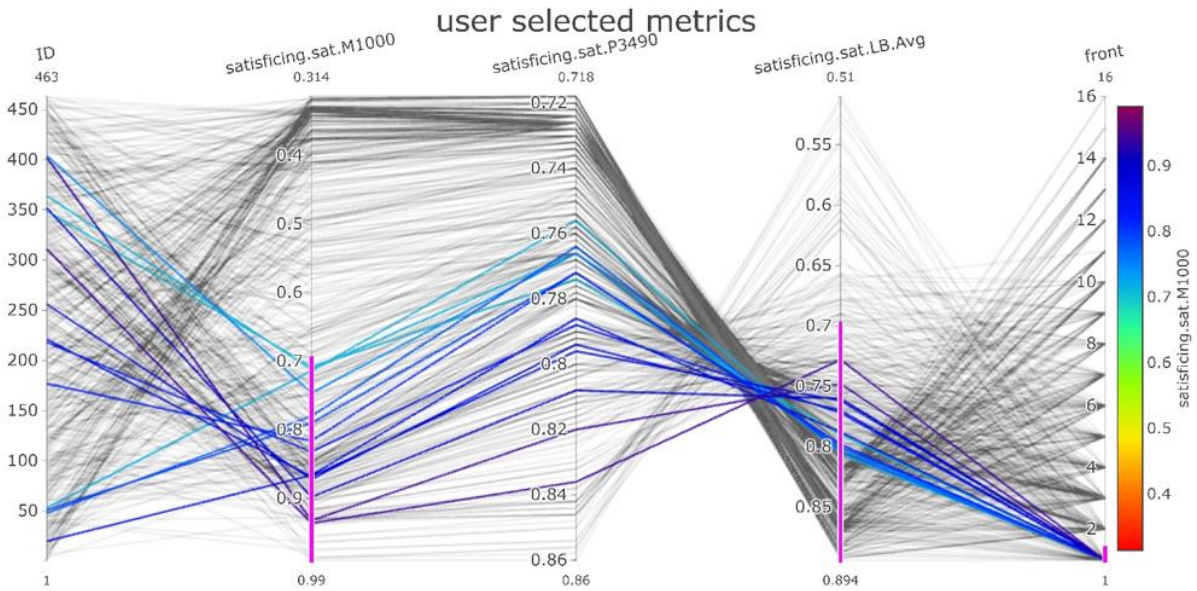


Figure 3-5. Parallel coordinate plot showing user-created robustness metrics and on-the-fly non-dominated sorting.

3.4.5 Phase 2: refining robustness preferences

Next, we explore tradeoffs of the remaining policies with additional robustness metrics using global linking. While on the *Select your own* page, we select the *save filters globally* button, which means only the remaining policies will be shown on other pages and tabs. For example, we navigate to the

Robustness metrics page and *maximin* tab, which shows the worst performance across all SOW for the specified objective (see description Figure 3-3). The resulting PC plot is also shown in Figure 3-6. This metric reflects a high level of risk aversion (McPhail *et al.*, 2018, 2021), which is a common preference of water providers (Smith, Kasprzyk and Dilling, 2017). Only the 14 remaining policies are shown, but, importantly, the range of each axis shows the possible values across all 463 policies. Five policies result in severe annual shortages (the LB.Max axis), annotated with the black-dotted oval in Figure 3-6. There remain other policies that reduce maximum annual shortages by 1000 KAF or more (seen by the gap between the policies within the oval and those below 2500 KAF). To remove the severe-shortage policies, we use a brush filter to highlight the policies below 2500 KAF on the LB.Max axis and select *save brush filters*. During this step, we added an additional robustness metric to our initial selection because of the tradeoffs we discovered, leaving nine policies remaining.

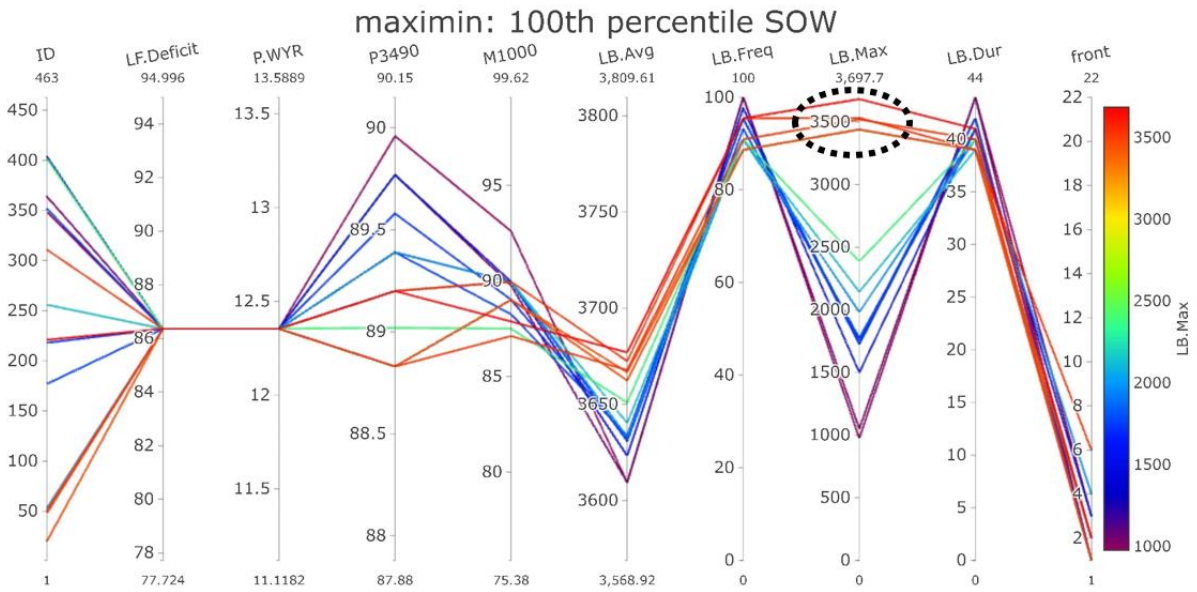


Figure 3-6. Parallel coordinate plot demonstrating global linking for *a posteriori* exploration of additional robustness metrics.

3.4.6 Choosing policies with interactive data-tables

Next, we select a small number of interesting policies, which could become the focus of deliberation or interrogated further in a vulnerability analysis. One way to accomplish this, besides with brush or manual filters, is via an interactive data table. To do so, select *save filters globally*, navigate to the *Regret from best* tab, and select the *Table of filtered policies* tab. The result is shown in **Error! R eference source not found.**. This metric describes the deviation of a policy's performance from the best performing policy, averaged across the SOW (see Figure 3-3). Since public agencies can be criticized when, in hindsight, the decision made could have yielded better results, this metric may be of interest to water resources agencies (McPhail *et al.*, 2021). Indicated by crossing lines, **Error! Reference source not found.**, t op, shows that reservoir storage (M1000 and P3490) trades off with average shortage (LB.Avg), and that average shortage trades off with frequency of shortage (LB.Freq). As shown in **Error! Reference source n ot found.**, bottom, we use the interactive data table to select (by clicking on) one policy that prioritizes reservoir storage (ID 402), one with low average shortage (ID 364), and two that balance the tradeoffs (IDs 256 and 177). The resulting shortage policies (**Error! Reference source not found.**) can be viewed by s electing the *keep* button under *table selections:* in the left sidebar, then navigating back to *Rank of filtered policies for selected metric* tab. In **Error! Reference source not found.**, the policies are ranked ordered

according to the P3490 objective and Regret from best metric, which can be changed with the *Metric for ranking* button in the left sidebar.

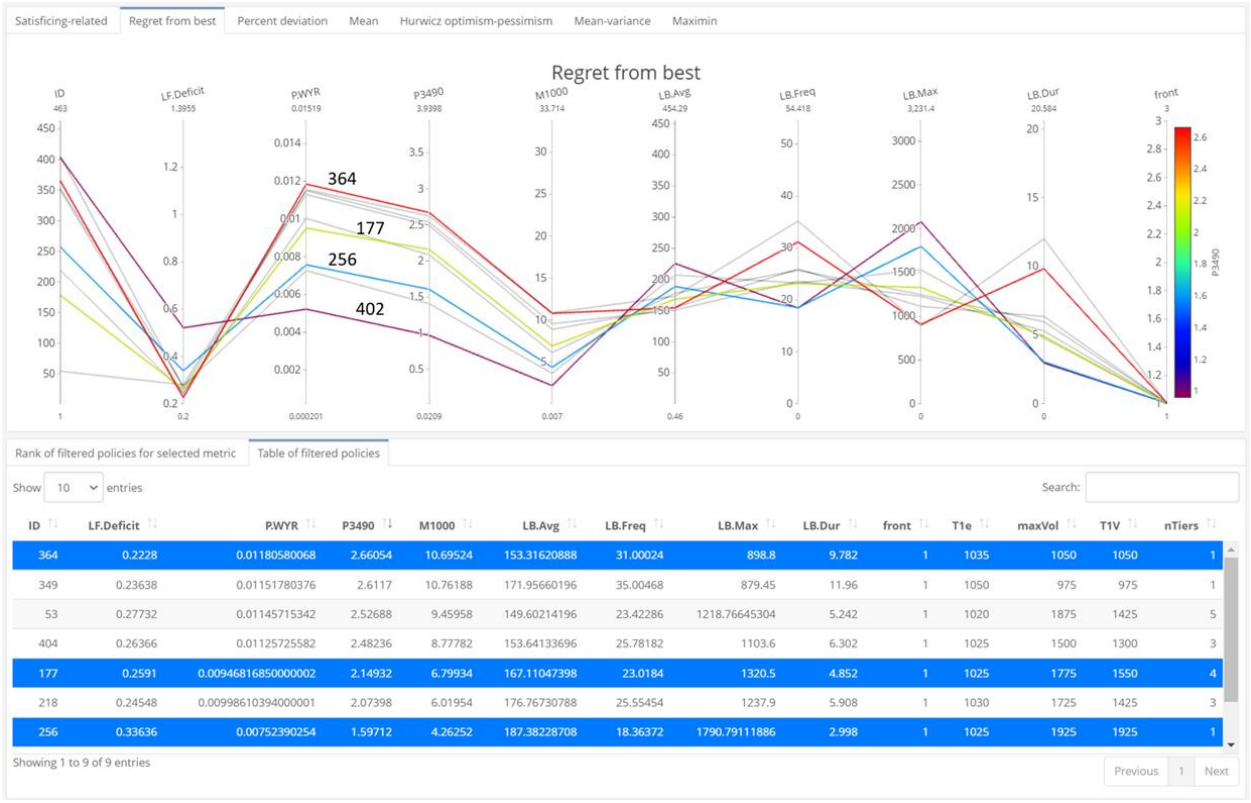


Figure 3-7. The shortage operation diagrams of the chosen policies.

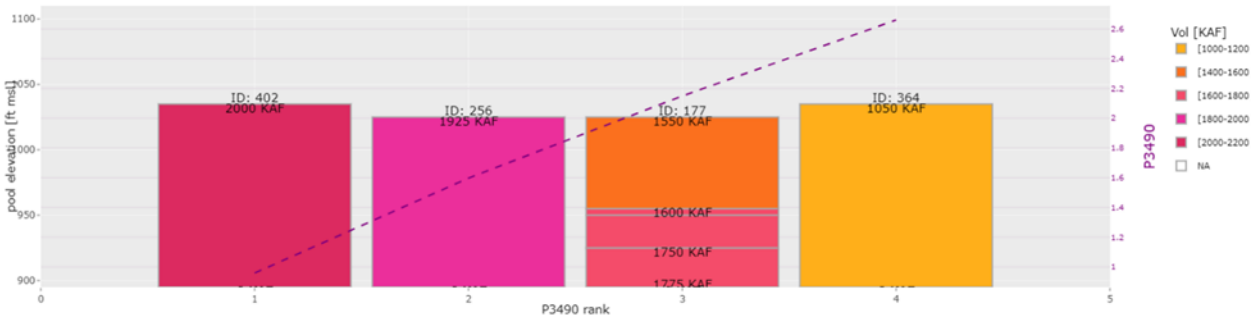


Figure 3-8. Screenshot showing an interactive data table to choose four policies with interesting tradeoffs.

3.4.7 Decision provenance for reproducibility and communication

Since our framework allows for unique, dynamic, and multi-step analyses, it is important that the results are reproducible. So, the app uses an activity log to track the objectives/robustness metrics that

filters are applied to and the corresponding filter criteria. The activity log is downloaded as a spreadsheet using the *Download* button at the bottom-left of Figure 3-2. A screenshot of the activity log for this demonstration is in Appendix A Figure 3-9.

It is also important that the results are easily communicated between stakeholders. As such, all PC plots and DV plots are downloadable, and several download parameters can be adjusted to fit the needs of reports, presentations, etc. Plot settings are accessed with the blue *plot options* button (to the right of the *Download* button in Figure 3-4). The PC plots are highly adjustable, with settings for color palette, title size, label size and angles, image size, and file type. For example, Figures 3-5, 3-6, and 3-8 were downloaded directly from the app.

3.5 Discussion and conclusion

This research demonstrates a robustness tradeoff analysis of Lake Mead shortage policies using an interactive web app. In our analysis, we show how *a posteriori* robustness exploration can reveal significant tradeoffs that refine stakeholders' robustness definitions and rapidly remove non-robust policies. We tested if existing performance thresholds and the satisficing metric were sufficient to identify robust shortage policies. Using non-dominated robustness sorting and interactive PC plots, we identified policies that are robust with respect to the three satisficing metrics. Because performance thresholds were already established in previous research (Alexander, 2018), existing robustness frameworks suggest that it would be appropriate to choose policies based on the satisficing metric alone (McPhail *et al.*, 2018, 2021). However, using our framework, we discovered that several remaining policies resulted in very severe LB shortages (the maximin metric), roughly 1 MAF larger than the other remaining policies. Had we chosen policies based on our initial priorities (the three satisficing metrics), this knowledge would have gone undiscovered. Instead, our *a posteriori* framework challenged and refined our initial preferences, and the policies resulting in severe LB shortages were removed. Finally, we demonstrated how other tools

in our framework, such as data tables, marking, and activity logs, help stakeholders choose a small number of policies and communicate their decision to others.

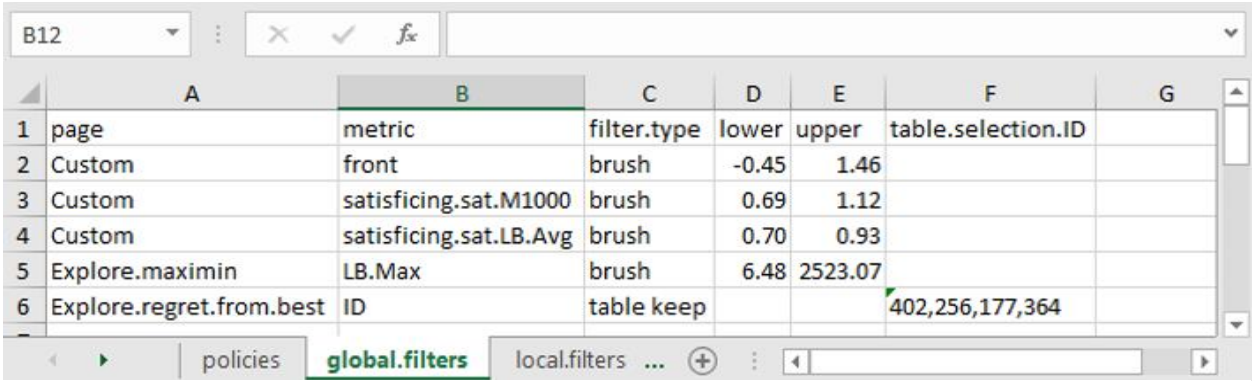
We see several opportunities for future research. First, we hope that future studies apply our framework to new decision problems. Although the app as presented in this article uses CRB data, it can be modified for other studies by changing the underlying database of policies, performance objectives, and robustness metrics. We refer the interested reader to our GitHub repository (Bonham, 2023), which includes the source code and instructions to download and run the app locally. Building on our robustness framework, future research could investigate the effectiveness of interactive web tools for other robust decision making techniques, such as vulnerability analysis (Bryant and Lempert, 2010; Hadjimichael, Quinn, *et al.*, 2020), adaptation pathways (Haasnoot *et al.*, 2013), and negotiation (Gold *et al.*, 2019b; Bonham, J. Kasprzyk and Zagona, 2022a). Our framework uses an activity log to make the robustness analysis reproducible and communicable, an example of provenance. Future research could build on this effort with recent advances in provenance methods that record what insights were learned during an analysis, the rationale behind decisions, and integrate the provenance information into interactive visualizations (Ragan *et al.*, 2016; Chakhchoukh, Boukhelifa and Bezerianos, 2022). Such methods could improve the efficacy of *a posteriori* methods for decision support. Given the rise in interactive web apps for education (Peñuela, Hutton and Pianosi, 2021) and decision support (Raseman, Jacobson and Kasprzyk, 2019), future research could use workshops, surveys, and retrospective studies to test their efficacy and guide future research questions (Smith, Kasprzyk and Dilling, 2017; Pianosi, Sarrazin and Wagener, 2020).

Through demonstrating our *a posteriori* framework, our research opens up further opportunities for collaborative MORDM analyses. For example, we used the CRB robustness app in a participatory workshop where Reclamation explored robustness metrics, applied the interactive filtering tools, and tested the various mechanisms for customizing robustness metrics and PC plots. Building on our collaboration, Reclamation is adopting and expanding the CRB robustness app for use in post-2026

planning (Smith *et al.*, 2022; Reclamation, 2023c). Reclamation will use the expanded app to communicate policy performance, robustness, and vulnerability to solicit preferences from a diverse group of stakeholders including water utilities, state agencies, irrigation districts, environmental agencies, Tribal leadership, etc. In parallel with the app development, Reclamation is holding training sessions to ensure stakeholders can meaningfully engage with the tools (Reclamation, 2023c). We believe our app and Reclamation's ongoing development are a significant milestone in the adoption of robust decision making techniques in collaborative, international water resources management. We encourage future studies to develop decision support frameworks and tools in collaboration with end users to improve their efficacy, increase likelihood of organizational uptake, and expedite their real-world application (Stanton and Roelich, 2021).

3.6 Appendix A: Activity log of example robustness analysis

The activity log for the example analysis above is shown in Figure 3-9. The activity log tracks which pages, robustness metrics/objectives, and filters were used to arrive at the chosen policies.



	A	B	C	D	E	F	G
1	page	metric	filter.type	lower	upper	table.selection.ID	
2	Custom	front	brush	-0.45	1.46		
3	Custom	satisficing.sat.M1000	brush	0.69	1.12		
4	Custom	satisficing.sat.LB.Avg	brush	0.70	0.93		
5	Explore.maximin	LB.Max	brush	6.48	2523.07		
6	Explore.regret.from.best	ID	table keep			402,256,177,364	

Figure 3-9. Activity log of the robustness analysis performed above.

3.7 Data availability statement

Some or all data, models, or code generated or used during the study are available in a repository or online in accordance with funder data retention policies. It can be accessed at the corresponding author's GitHub repository: https://github.com/nabocrb/CRB-robustness-app---JWRPM (Bonham 2023).

3.8 Acknowledgements

We would like to thank Reclamation for providing the Lake Mead policies and input on the app's design. We would also like to thank anonymous reviewers who helped improve the clarity of this article. This research is based on work supported by Reclamation's Upper and Lower Colorado regions under project R19AC00053 and the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2040434.

4 Mapping policies to synthesize optimization and robustness results for decision-maker compromise

4.1 Introduction

Decision making for coupled human-environmental systems is a paramount challenge of the 21st century. Decision-makers (DMs) need to identify policy actions that are simultaneously equitable, balance competing objectives, and are robust to future uncertainty (UN General Assembly, 2015; Committee to Advise the U.S. Global Change Research Program *et al.*, 2021; IPCC, 2021). Simulation models are often used to support DMs by quantifying their system's key performance outcomes, and elucidating how performance outcomes relate to policy decisions and exogenous driving forces of system behavior. However, analysts and DMs often disagree on the relationships between the driving forces and performance outcomes of their systems. Further, the probability distributions of driving forces are unknown, and/or disagreement exists on how to weigh performance outcomes of alternative decision actions. Such decision problems are described as deeply uncertain (Lempert, Popper and Bankes, 2003; Kwakkel and Haasnoot, 2019). When facing deep uncertainty, implementing a policy can require negotiation between DMs, but reaching a compromise can be difficult because of foundational disagreements such as divergent framings of the problem (Wheeler *et al.*, 2018; Lempert and Turner, 2020), different prioritization of performance outcomes (Smith, Kasprzyk and Dilling, 2019), or different tolerances of uncertainty-related risk (McPhail *et al.*, 2018, 2021).

Many-Objective Robust Decision Making (MORDM) is a simulation-based decision support framework that helps DMs identify promising policy actions when faced with deep uncertainty (Kasprzyk *et al.*, 2013). MORDM has been shown to be effective for various human-environmental systems, with applications such as municipal water supply portfolios (Herman *et al.*, 2014, 2015; Gold *et al.*, 2019a), irrigation and groundwater sustainability (Li and Kinzelbach, 2020), and reservoir operation policies (Alexander, 2018; Quinn *et al.*, 2018). MORDM produces three types of decision-relevant information for DMs to consider: decision variable values, objective values, and robustness values (Figure 4-1a). In panel

i, a simulation model of the system is coupled with an optimization algorithm to generate a set of many policy alternatives. Each policy is defined by a vector of decision variable values (yellow box), and each policy has corresponding data on its performance objective values (green box). MORDM then 'stress-tests' each policy in a robustness analysis (panel ii). At this stage, uncertainty regarding driving forces of the system is explicitly considered by simulating each policy in many plausible future States of the World, which densely sample the range of plausible future scenarios. Robustness metrics are used to quantify how well a policy performs in performance objectives across all the SOW. This produces the third type of decision-relevant data (blue box).

Several challenges hinder DMs from utilizing the decision-relevant information produced from MORDM to select one or a small subset of policies for potential implementation. First, interpreting the cause-effect relationships between DMs' actions (i.e. DV values) and performance/robustness tradeoffs is non-trivial because the policy set often consists of hundreds of policies characterized by complex interactions between decision variables, objectives, and robustness metrics (Miller, 1956; LeCompte, 1999; Saaty and Ozdemir, 2003; Herman *et al.*, 2014; Alexander, 2018; Quinn *et al.*, 2018; Wheeler *et al.*, 2018; Smith, Kasprzyk and Dilling, 2019). Moreover, in environmental systems that provide services to diverse stakeholders, DMs may struggle to use all this data to overcome foundational disagreements, such as different framings of the decision problem (Wheeler *et al.*, 2018; Lempert and Turner, 2020), different weighing of performance objectives (Smith, Kasprzyk and Dilling, 2019), or different risk tolerances towards uncertain future conditions (McPhail *et al.*, 2018, 2021; Hadjimichael, Quinn, *et al.*, 2020).

Multiple studies in the engineering design domain have highlighted the Self-Organizing Map (SOM) as a promising machine learning algorithm to alleviate the cognitive burden faced by DMs (Obayashi and Sasaki, 2003; Koishi and Shida, 2006; Li, Liao and Coit, 2009; Mosnier, Gillot and Ichchou, 2013; Zhang *et al.*, 2018). In this research, we uniquely couple the SOM with MORDM, introducing an alternative paradigm for how the relationships between decision variables, objectives, and robustness are

interpreted. As an analogy, consider Figure 4-1b. Panel i shows four geospatial map layers of Lake Mead, located just east of Las Vegas, Nevada. Each layer shows a different type of data (satellite imagery, hydrologic drainage network, administrative boundaries, and major road network), but, importantly, the data is organized by latitude and longitude. In other words, every coordinate pair has multiple layers of corresponding data types. Moreover, the geographic arrangement of map layers helps explain the relationships between data layers. For example, as you move westward from Lake Mead (leftward on the longitude axis), the number of administrative units and major roads increases, which is because you are moving away from Lake Mead and into downtown Las Vegas. The SOM creates analogous means for interpreting the MORDM data as multiple data layers organized by a two-dimensional coordinate system. The SOM learns the many-dimensional patterns of the policy set, groups and arranges policies into a two-dimensional coordinate system, and establishes a map-like visualization for elucidating the relationships between decision variables, objectives, and robustness. In this paper, we refer to performance objective values, decision variable values, and robustness values as ‘layers’ in accordance with this paradigm.

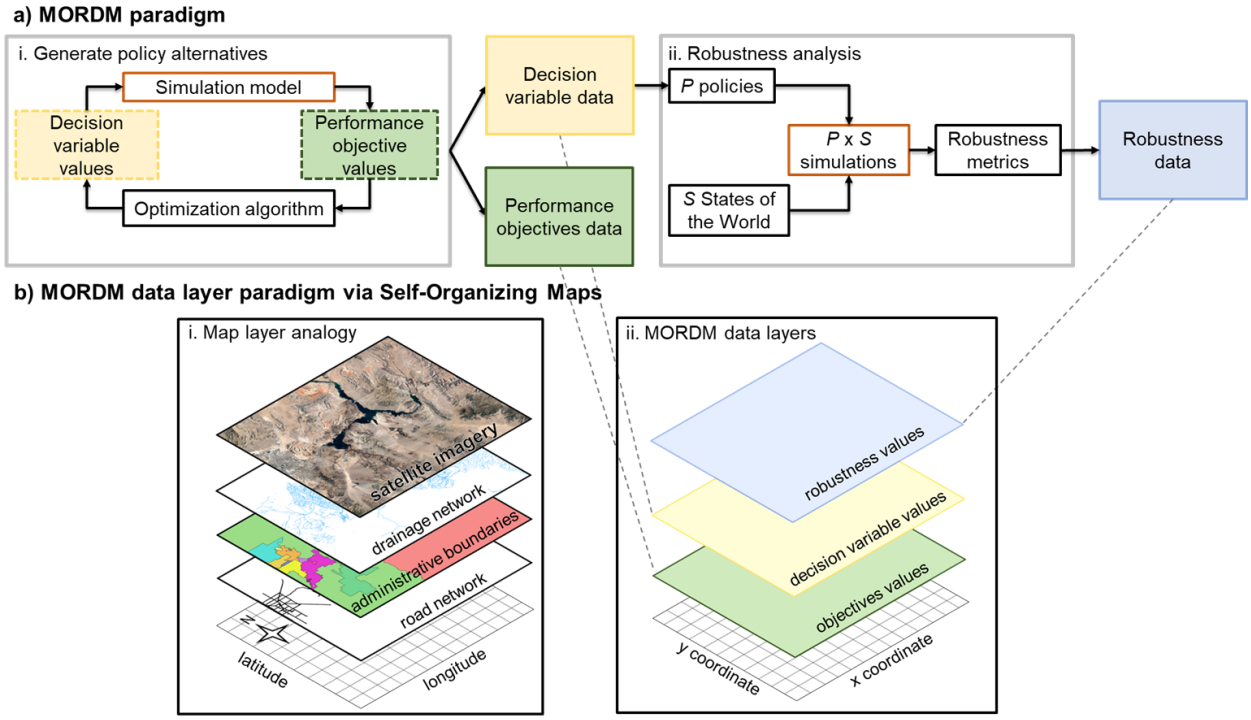


Figure 4-1: An overview of the MORDM data layer paradigm presented in this paper. **a)** The MORDM framework outputs decision variable data and performance objectives data from simulation-assisted optimization, which generates many policy alternatives (panel i). Then, robustness analysis is performed, which is where all P policies are simulated in S States of the World, which are plausible future scenarios used to test the performance of policies in various realizations of exogenous uncertainty. Robustness metrics quantify how well a policy performs in a given objective across the States of the World, which results in the third decision-relevant data type, robustness data (panel ii). **b)** This research uses the Self-Organizing Map (SOM) as a MORDM post-processing tool to help DMs visualize and navigate large sets of policies while considering objective values, decision variable values, and robustness values in their decision processes, creating an alternative paradigm where each data type is considered its own layer, and the layers are arranged according to a map-like coordinate system. We use panel i to explain this paradigm with an analogy to the geospatial sciences. Geospatial data is organized according to a latitude and longitude coordinate pair and by layers, where each layer is a collection of data of the same type. For example, satellite imagery, drainage network information, administrative boundaries, and road network information are each a different map layer, and the data within each layer are arranged spatially according to latitude and longitude. This research uses the SOM to organize, visualize, and navigate MORDM-derived data as MORDM data layers (panel ii). We use the SOM to discover the topological patterns exhibited by objective values, decision variable values, and robustness values, then arrange each layer according to a two-dimensional, map-like coordinate system.

Our review of the SOM literature has shown two critical research gaps as it pertains to MORDM and decision support applications (Obayashi and Sasaki, 2003; Koishi and Shida, 2006; Li, Liao and Coit, 2009; Mosnier, Gillot and Ichchou, 2013; Zhang *et al.*, 2018). First, SOM applications have not considered robustness, a critical component of MORDM, instead being limited to the decision variable and

performance objective layers. Second, SOM applications have been implemented in the context of a single analyst, design team, or organization gaining important system understanding or selecting an engineering design. Such applications are fundamentally different than the policy or management decisions in an environmental system, where the problem is often characterized by negotiation between multiple local, state, or federal governments, NGOs, landowners, and various other interest groups (Reclamation, 2007, 2012a; Wheeler *et al.*, 2018; Molina-Perez *et al.*, 2019). To our knowledge, the SOM has not been implemented in such a negotiation context.

Building on previous SOM applications, this paper introduces post-MORDM, a framework that assists DMs and analysts interpret, visualize, and negotiate large sets of policies. Post-MORDM augments MORDM by using the SOM to a) elucidate the relationships between decision variables, objectives, and robustness, b) reduce the number of alternatives DMs need to consider, and c) establish a visual, structured platform whereby DMs with divergent performance priorities and risk tolerances are assisted in a process of negotiation towards compromise policies. Post-MORDM contributes to the SOM literature by expanding layer visualization to robustness, and demonstrating how the construction and utilization of the SOM can be implemented in negotiation contexts for human-environmental systems.

The remainder of the paper is organized as follows: Section 4.2 provides background on MORDM, then motivates our use of the SOM with a review of machine learning algorithms that have been used to post-process MORDM data. Section 4.3 describes the SOM algorithm and its fundamental benefits before outlining the post-MORDM framework. Section 4.4 demonstrates post-MORDM in a case study of reservoir operation policies in the Colorado River Basin, USA. The discussion and conclusion follow in Sections 4.5-4.6.

4.2 Background and motivation

4.2.1 Many Objective Robust Decision Making (MORDM)

Many-Objective Robust Decision-Making (MORDM) is a simulation-assisted decision-support framework that helps DMs identify promising policies when faced with deep uncertainty (Kasprzyk *et al.*, 2013). The framework consists of four steps – problem formulation, policy generation, robustness analysis, and scenario discovery. This section describes the steps and how they result in three MORDM data layers – decision variable values, objective values, and robustness values. MORDM is broadly applicable to policy analysis and environmental problems. For clarity, the examples we provide in this section focus on a specific familiar human-environmental system, namely a river system managed by water storage reservoirs.

First, problem formulation defines the scope of the decision problem in terms of decision variables, performance objectives, and driving forces characterized by exogenous uncertainties (Lempert, Popper and Bankes, 2003; Lempert and Collins, 2007; Kasprzyk *et al.*, 2013). Exogenous uncertainties are factors outside the explicit control of the DMs, like hydroclimatic factors (e.g. temperature, precipitation, runoff) and socioeconomic factors (e.g. water demand, irrigation efficiency). Uncertainty is characterized using what-if scenarios called States of the World (SOW). Each SOW is a multivariate sample of the uncertainty factors, and an ensemble of SOW is created to extensively sample each factor's plausible bounds. Decision variables (DVs), $\boldsymbol{x} = x_1, x_2, \ldots, x_L$, represent policy actions where x can be continuous (e.g. volume of water to release from a reservoir at a given time step) or discrete (e.g. augment water supply by either expanding a reservoir, building a desalination plant, or purchasing water rights). Performance objectives, $\boldsymbol{f} = f_1, f_2, \ldots, f_M$, are metrics that quantify how well the system performs, such as reliability of meeting water demands or average hydropower production. For each policy and SOW, a simulation model calculates the values of the performance objectives.

After problem formulation, policy generation is performed using multi-objective simulation-based optimization, commonly using Multi-Objective Evolutionary Algorithms (MOEAs). The MOEA is coupled with the simulation model in a loop where the MOEA generates policies, and the model evaluates performance objectives. In this paper, we use the term 'policies' to refer to MOEA solutions, i.e., vectors of DV values (Giuliani *et al.*, 2014). At this stage in MORDM, a small number of SOW is used to force the simulations and calculate performance objective values, often using SOW that reflect values of uncertain factors observed in the historical record (Kasprzyk *et al.*, 2013; Alexander, 2018). The MOEA creates new policies over thousands of iterations of the simulation-optimization loop, using operators inspired by concepts of evolutionary theory like genetic crossover, random mutations, and 'survival of the fittest' (Hadka and Reed, 2013; Maier *et al.*, 2019). The output of the MOEA is a set of *non-dominated* policies. Policy **a** *dominates* policy **b** if the performance of policy **a** is equal to or better than the performance of **b** in all objectives while being better than the performance of **b** in at least one objective. In the resulting policy set, no policy dominates any other policy, i.e., they are *non-dominated*. In effect, the policies exhibit performance tradeoffs, where improving performance in one objective necessitates inferior performance in one or more other objectives. As a result of the policy generation step, each policy has two corresponding MORDM data layers – objective values and DV values.

DMs use interactive visualizations to explore tradeoffs and interpret the relationships between DV and objective layers. Visualization types include glyph plots (Kollat and Reed, 2007; Kasprzyk *et al.*, 2013) and parallel axis (PA) plots (Inselberg, 2009), which are commonly used because of the relative ease with which greater than three dimensions are visualized. Using PA plots, DMs apply their preferences by changing the order of the axes, filtering policies that meet performance goals, and clustering policies with similar DV and/or objective values (Inselberg, 2009; Raseman, Jacobson and Kasprzyk, 2019). To elucidate the relationships between objective and DV layers, previous studies have 'linked' a PA plot of objectives to a PA plot of DVs (Smith, Kasprzyk and Basdekas, 2018; Raseman, Jacobson and Kasprzyk, 2019; Li and

Kinzelbach, 2020; Raseman *et al.*, 2020). For example, Smith et al. hosted a workshop with Colorado water managers to query how MOEA results could improve decision-making in their respective agencies (2019). Participants identified policies of interest according to performance objective preferences in one plot, and the corresponding DV values were highlighted in another 'linked' plot, demonstrating the types of actions needed to achieve their performance preferences.

The objectives layer is calculated using a small number of SOW. Although policies are non-dominated with respect to the objectives layer, it is possible their performance deteriorates when the assumptions about deeply uncertain factors are incorrect. Therefore, policy generation is followed by *robustness* analysis, where policy alternatives are 'stress-tested' by simulating them in the ensemble of SOW defined in the problem formulation step. Robustness is the degree to which a policy's performance is insensitive to this broad sampling of SOW (Kasprzyk *et al.*, 2013; Herman *et al.*, 2015; McPhail *et al.*, 2018). Performance sensitivity is quantified with robustness metrics, which are statistics that describe how well a policy performs across its distribution of performance for specified objectives in the SOW ensemble. Robustness metric values define the third MORDM data layer.

There exist different robustness metrics, which use varying transformations and statistical calculations across the sampled SOW (McPhail *et al.*, 2018, 2021). These *different robustness metrics reflect differing prioritization of objectives, minimum performance thresholds, and risk tolerances of DMs* (McPhail *et al.*, 2018, 2021; Quinn *et al.*, 2018; Gold *et al.*, 2019a; Hadjimichael, Quinn, *et al.*, 2020). Examples of robustness metrics include the expected value of performance (Laplace's Principle of Insufficient Reason), regret from best possible performance (regret from best), the 'worst-case' performance (maximin), or the fraction of SOW where a DMs expressed performance criteria are achieved (satisficing) (McPhail *et al.*, 2018). The previously described interactive visualization techniques can also be used to explore the tradeoffs between different robustness metrics and their relationships to DVs (Giuliani *et al.*, 2014; Herman *et al.*, 2015; Cohen and Herman, 2021). Notably, in a study of four connected

water utilities, Herman et al. demonstrate how a different policy is the most robust for each utility because they each define robustness based on their respective performance criteria. In other words, there exists "interutility robustness tradeoffs", and settling on a single policy would require the utilities to negotiate and compromise (2014). This example underscores the challenge of selecting a policy in a system characterized by deep uncertainty and multiple DMs, plus the potential benefits of a structured negotiation platform that facilitates negotiation and compromise.

In the last step of MORDM, policies are interrogated further via Scenario Discovery. Scenario Discovery is a type of vulnerability analysis, where vulnerability is defined by the violation of DM-defined performance criteria. Scenario Discovery identifies the uncertainty factors sampled in the SOW ensemble that are the best predictors of when a policy will be vulnerable, and the corresponding values of the factors in which vulnerability occurs. Traditionally, Scenario Discovery is performed on a small subset of policies selected based on the results of policy generation and robustness analysis (Kasprzyk *et al.*, 2013; Giuliani *et al.*, 2014; Quinn *et al.*, 2018). In this paper, we focus on using MORDM data layers – objective, DV, and robustness values – to identify a small subset of policies. These policies would subsequently be input to a separate Scenario Discovery process and/or real-world implementation; the workflow in this paper does not include Scenario Discovery.

4.2.2 Challenges and gaps to MORDM decision support

In the selection of policies using MORDM techniques, DMs will want to consider the DV, objective, and robustness layers. However, several challenges arise when using MORDM data layers to select policies. For instance, the quantity of policy alternatives and the dimensionality of MORDM data layers exceeds average human processing limitations. We use 'dimension' to mean the numbers of DVs, objectives, and robustness metrics in each MORDM data layer. For example, published applications of MORDM have exhibited hundreds of policies (Herman *et al.*, 2014; Zeff *et al.*, 2014; Quinn *et al.*, 2018; Wheeler *et al.*, 2018), less than five to over 100 DVs (Herman *et al.*, 2014; Quinn *et al.*, 2018), three to

eight objectives (Alexander, 2018; Quinn *et al.*, 2018; Wheeler *et al.*, 2018), and one to four or more robustness metrics (Herman *et al.*, 2014). However, studies suggest that fewer than three to a maximum of nine alternatives be ideally examined at one time (Miller, 1956; Brill, Chang and Hopkins, 1982; LeCompte, 1999; Saaty and Ozdemir, 2003). Moreover, in the context of negotiation between DMs, large quantities of information can have negative consequences by increasing egocentric interpretations of what constitutes a fair resolution, thus increasing the time needed to identify a compromise policy (Thompson and Loewenstein, 1992; Tsay and Bazerman, 2009). These studies highlight the cognitive burden DMs face when selecting policies from large, many-dimensional policy sets.

Selecting policies also requires DMs to understand the cause-effect relationships between their actions (i.e., DV values) and the performance/robustness tradeoffs of the system. However, when the policy set is very large and/or the relationships between MORDM data layers is complex, elucidating such relationships is non-trivial. Although interactive visualization methods like PA plots can be effective for linking DV values to objective values for one or a small subset of policies, it remains a challenge to synthesize the relationships across the entire policy set because of the large number of dimensions in each MORDM data layer; complex and non-linear interactions within the system (Smith, Kasprzyk and Basdekas, 2018; Hadjimichael, Gold, *et al.*, 2020); noisy or low-signal DVs, objectives, or robustness metrics (Smith, Kasprzyk and Rajagopalan, 2019); and surprising effects of challenging SOW on robustness values. Indeed, the participants in Smith et al. described that training personnel in their respective organizations to understand MOEA results and communicating them to DMs remains a challenge, and the study concluded that "structured information about the relationships between decision levers [variables] and performance [objectives] would be beneficial for interpreting tradeoffs (2019)." This study demonstrated the difficulty of synthesizing relationships between DV and objective layers; moreover, the addition of the robustness layer, which potentially includes multiple definitions of robustness, would likely exacerbate this challenge.

Another challenge to selecting policies is that DMs and stakeholders involved in environmental decision problems may have foundational disagreements that are not easily overcome by only exploring the objective, DV, and robustness layers. As described in the previous section, DMs may hold conflicting prioritization of objectives, which, combined with different degrees of risk tolerance, can result in different definitions of robustness. Further, DMs may not agree on a single problem formulation, instead using multiple problem formulations that reflect their respective world views (Quinn *et al.*, 2017; Wheeler *et al.*, 2018; Lempert and Turner, 2020). For DMs with foundational disagreements to identify compromise policies, MORDM data layers need to be presented to DMs in such a way that facilitates discussion, negotiation, and compromise.

4.2.3 Clustering and dimension-reduction methods with MORDM

Previous studies have reduced the number of alternatives DMs need to consider and improved the interpretability of MOEA-derived objectives and DV layers via two classes of statistical techniques – namely clustering and dimension reduction. Clustering is a method to group data such that data within a cluster are similar and data in different clusters are more dissimilar (Hastie, Tibshirani and Friedman, 2009, chap. 14.3). Clustering has previously been used to group policies with similar DV and/or objective values (Kansara, Parashar and Xue, 2015; Raseman *et al.*, 2020). In this approach, DMs consider a small number of clusters rather than hundreds of individual policies, which can help reduce the cognitive load faced by DMs to be within the range of practical human processing limitations discussed in the previous section. For example, Raseman et al. group MOEA-derived water treatment plant policies using k-means clustering, summarizing the DV and objective layers with three clusters and a representative policy from each (2020). Although clustering techniques can reduce the number of alternatives DMs consider, it does not elucidate the interactions between objectives or the relationship between objectives and DVs. Further, to our knowledge, clustering applications have not considered robustness.

To elucidate the interactions between objectives and DVs, clustering can be complemented with dimension reduction methods. Dimension reduction is the process of either selecting a subset of the most important features of a data set (feature selection) or creating a low-dimensional representation of the data by linear or non-linear combinations of features (feature extraction) (Khalid, Khalil and Nasreen, 2014; Hira and Gillies, 2015; Ghojogh *et al.*, 2019). In both cases, the goal of dimension reduction is to 'simplify' a dataset meanwhile preserving the information contained within it by removing or combining noisy, redundant, or irrelevant features. In MOEA applications, dimension reduction methods have helped DMs relate DV values to objective outcomes. For example, Smith et al. perform feature selection via multivariate regression trees to identify the DVs that are most influential on objectives, meanwhile using a tree-based visualization to intuitively guide decision-makers towards DV values that achieve their desired objective outcomes (2019). Kansara et al. utilize principal component analysis, a feature extraction method, to reduce 12 objectives into a two-dimensional summary that highlights the most important objectives and the correlation structures between them (2015). Dimension reduction also helps reduce the cognitive burden faced by DMs; instead of interpreting the relationships between many DVs and objectives, the relationships can be synthesized in a low-dimensional summary that captures the most decision-relevant information. These studies demonstrate the benefits of clustering and dimension reduction techniques to enhance decision-support efficacy with DV and objective layers, and they motivate similar techniques to be expanded to the robustness layer.

4.2.4 Motivation for Self-Organizing Maps in post-MORDM

In summary, MORDM-based decision support for environmental systems is characterized by many policies; three multi-dimensional data layers – DVs, objectives, and robustness; and negotiation between DMs to overcome foundational disagreements. Previous studies have demonstrated the utility of clustering and dimension reduction techniques applied to DV and objective layers, revealing two research gaps for this study. First, the robustness layer should also be considered in clustering and dimension

reduction applications. Second, decision-support via MORDM data layers also needs to address the foundational disagreements between DMs described above, providing a structured, visual, and easily-interpretable methodology that encourages the discussion and negotiation needed to identify compromise policies.

To address these gaps, we introduce the post-MORDM framework, which augments MORDM via a novel implementation of the Self-Organizing Maps (SOM). The SOM is a type of artificial neural network wherein the neurons learn the cluster structure and feature space patterns of a multivariate data set (Kohonen, 1982). We apply the SOM to MORDM data layers, providing the benefits of policy clustering and dimension reduction in a single algorithm (Kohonen, 1990; Clark, Sisson and Sharma, 2020). Moreover, the SOM enables DMs to simultaneously visualize DV, objective, and robustness layers with an intuitive, map-like visualization of the relationships within and between each layer.

The post-MORDM framework builds on previous SOM applications by expanding to robustness, a decision-relevant data layer for human-environmental systems. Previous studies in the mechanical engineering domain have trained a SOM on an MOEA objectives layer, then used the resulting SOM to visualize the inverse relationship to the DV layer. Example design problems include aircraft wings (Obayashi and Sasaki, 2003), automotive tires (Koishi and Shida, 2006; Mosnier, Gillot and Ichchou, 2013), and switched reluctance machines (Zhang *et al.*, 2018). In contrast, environmental decision problems often consider objectives, DVs, ***and robustness*** because they are characterized by deeply uncertain hydro-climatic and socioeconomic factors (Herman *et al.*, 2014, 2015; Quinn *et al.*, 2018; Gold *et al.*, 2019a; Li and Kinzelbach, 2020). Therefore, we build on these previous MOEA-SOM studies by expanding the SOM to the robustness layer.

Further, the post-MORDM framework implements the SOM to guide DMs through a process of policy negotiation and compromise. In the published MOEA-SOM studies, the purpose has been to support a single analyst or organization in the choice of a design. Conversely, environmental decision

problems often involve multiple DMs who reflect the interests of local, state, or federal government, environmental NGOs, and various other interest groups (Reclamation, 2007, 2012a; Wheeler *et al.*, 2018; Molina-Perez *et al.*, 2019). The post-MORDM framework uniquely implements the SOM as a discussion and negotiation platform for multiple DMs, using the SOM to navigate to compromise policies.

Moreover, we desire the SOM to reduce the number of alternatives that DMs consider from several hundred policies to a reasonably small number of neurons. In this context, a neuron is a model that represents one or more policies, similar to a cluster. The size of the SOM, i.e. the number of neurons, is determined by the user. Previous SOM applications have used a SOM with several hundred neurons, which is often determined by heuristics that calculate the number of neurons given the sample size of the user's data set (Kohonen, 2001, chap. 3; Obayashi and Sasaki, 2003; Koishi and Shida, 2006; Mosnier, Gillot and Ichchou, 2013; Clark, Sisson and Sharma, 2020). In contrast, the post-MORDM framework uses the smallest SOM possible to adequately represent the cluster structure of MORDM data layers (see Section 4.3.1.1). The result is a SOM that summarizes hundreds of MOEA policies with a relatively small number of neurons to reduce the cognitive burden faced by DMs.

4.3 Self-Organizing Maps and the post-MORDM framework

This section is organized as follows. We define the SOM algorithm in Section 4.3.1, then describe the essential attributes of SOM via an example with DV and objective layers. We describe the post-MORDM framework in Sections 4.3.2 – 4.3.5.

4.3.1 Self-Organizing Maps (SOM)

A SOM is a type of two-dimensional artificial neural network used for feature extraction, clustering, and topologic visualization of multidimensional data sets with many samples or points (Kohonen, 1990, 2001, 2013; Clark, Sisson and Sharma, 2020). A SOM consists of an interconnected grid of neurons, where a neuron is a prototype data point. Each neuron is defined by a vector of values, one value for each feature of the data. Each neuron is parametrized with an integer-based coordinate pair

that identifies the location of the neuron within the grid and the topologic, or neighborhood-based, relationships between neurons (Kohonen, 2001; Hastie, Tibshirani and Friedman, 2009, chap. 14.4). During the training process, the neurons are iteratively updated to better represent the multi-dimensional structure of the input data, meanwhile maintaining their topologic relationships to each other within the grid. After training, the data points are projected onto the two-dimensional grid, resulting in a topology map that visualizes the data's cluster structure and most salient patterns (Clark, Sisson and Sharma, 2020). In this paper, we describe the steps of creating a SOM in two categories: pre-training setup, and the neuron update function.

4.3.1.1 pre-training setup

Before training a SOM, the user must normalize or scale the features of the input data so that differences in their magnitudes and variances do not bias the training process. Next, the length-width ratio of the SOM is calculated such that it represents the shape of the data. To accomplish this, the user sets the length-width ratio equal to the ratio of the first and second eigenvalues of the data's correlation matrix (Kohonen, 2001; Clark, Sisson and Sharma, 2020). In effect, this process allocates proportionally more neurons to the SOM along the direction of the data's feature space with the most variance.

After establishing the length-width ratio, the user chooses the total number of neurons and several hyperparameter values, both of which requires training multiple SOMs and evaluating quality of fit metrics. First, to establish a practical upper and lower limit on the number of neurons to test, the user estimates the number of clusters that best represents the intrinsic cluster structure of the input data. This process can be performed via the k-means clustering 'elbow' method, which requires calculating a cluster quality metric, such as the Silhouette or Davies-Bouldin Index, for 1 to k_{max} k-means clusters. The user then identifies k such that larger numbers of clusters exhibit sharply diminishing marginal improvement in cluster quality (Hastie, Tibshirani and Friedman, 2009, chap. 14.3; Rendón *et al.*, 2011; Clark, Sisson and

Sharma, 2020). The user then tests multiple SOMs of different sizes, ranging from a lower and upper limit centered around k, where the limits are based on the computational and time constraints of the user.

Second, the user sets the hyperparameter values. Hyperparameters include neighborhood radius, neighborhood function, distance function, and edge neuron behavior. We describe these hyperparameters in Appendix B.1. The user can also decide between a rectangular or hexagonal grid structure, but in this study we use a hexagonal topology because they tend to outperform rectangular grids both in terms of visualization and quality of fit (Kohonen, 2001; Clark, Sisson and Sharma, 2020). The user selects the number of neurons and hyperparameter set based on the tradeoff between two fit metrics. Percent of variance explained (PVE) captures the degree to which neuron prototype vectors represent the input data. Topographic error (TE) measures the degree to which the mapping of input data onto the two-dimensional map preserves the many-dimensional data structure (Clark, Sisson and Sharma, 2020; Boelaert *et al.*, 2021). For equations and further descriptions of these metrics, see Appendix B.2. PVE and TE conflict, where PVE improves and TE worsens with an increasing number of neurons. Thus, the user tests multiple SOMs with different map sizes and hyperparameter sets, then makes a selection that balances the metrics.

Once the number of neurons and the hyperparameter values are set, the user initializes the neurons by uniformly aligning them along the plane formed by the first and second principal components of the data's feature space. Principal component one (PC1) is the linear projection of the feature space along which the data varies the most, and the direction along which the longer edge of the SOM is aligned. Principal component two (PC2) is orthogonal (uncorrelated) to PC1, and indicates the second greatest mode by which the data varies (Hastie, Tibshirani and Friedman, 2009, chap. 14.5; James *et al.*, 2013, chap. 10.2). In this paper, all visualizations of the SOM will be oriented such that PC1 and PC2 are aligned with the horizontal and vertical directions, respectively. Because SOM neurons are initialized along PC1 and PC2, the resulting topology map can be 'navigated', interpreting movement along the topology map

according to the contributions of each feature to the PCs. For practical guidance on all pre-training steps described above, see the code included in Appendix B.5.

4.3.1.2 SOM update function

During SOM training, the neurons are iteratively fit to the data points via an update function whereby individual neurons compete to 'win' data points, and neurons within neighborhoods cooperate to win data points. The neighborhood of a neuron is defined to be all neurons within a user-defined neighborhood radius, measured in two-dimensional map space. At every iteration, each data point is assigned to its best matching unit (BMU), which is the neuron closest to the data point, measured in data space, according to a distance function. Neurons compete to be the BMU of each data point, while also cooperating with neurons within their neighborhood, via an update function. The update function awards neurons and their neighbors by moving them closer to the data points. Effectively, the grid of neurons is bent, twisted, and stretched from its original position on the principal component plane to better represent the non-linear patterns of the data (Hastie, Tibshirani and Friedman, 2009, chap. 14.4; Clark, Sisson and Sharma, 2020). We employ the batch version of the SOM update function because it converges faster than the stepwise recursive function and has no random component (Kohonen, 2013). For further information on the batch update function, see Appendix B.1.

After training iterations are complete, the final step in creating a SOM is assigning the data points to their BMU (Clark, Sisson and Sharma, 2020). The assignment of data points to neurons is analogous to k-means clustering, where neurons are akin to cluster centroids (Hastie, Tibshirani and Friedman, 2009, chap. 13.2; James *et al.*, 2013, chap. 10.3; Raseman *et al.*, 2020). Importantly, SOM neurons are arranged based on their similarities because of the neighborhood-based cooperation during training (Clark, Sisson and Sharma, 2020). For practical assistance in creating a SOM, several packages are supported in R (Wehrens and Kruisselbrink, 2019; Boelaert *et al.*, 2021), Python (Smith, 2021; Vettigli, 2021), and MATLAB (*Cluster with Self-Organizing Map Neural Network - MATLAB & Simulink*, no date).

After creating the SOM, the user visualizes it on a topology map, which is created by plotting the data onto the SOM's two-dimensional grid. Within the topology map, neurons close to each other are more similar than neurons far apart. Moreover, the most significant data patterns along the horizontal and vertical dimensions of the topology map can be interpreted via the relative contribution of features to PC1 and PC2 (Clark, Sisson and Sharma, 2020). In the next section, we demonstrate a SOM applied to MOEA-derived DV and objective layers. We use the illustration to describe the benefits of the SOM and to introduce the plot types used in the post-MORDM framework.

4.3.1.3 example SOM on MOEA-created policies

To demonstrate the benefits of the SOM, we provide an example in Figure 4-2. Figure 4-2a shows MOEA-derived objective values, symbolized with a PA plot. Each vertical axis is an objective, f_i (i = 1, 2, …, M, where M is the number of objectives), and each trace corresponds to one policy. Using the SOM, the M-dimensional objective space is summarized with a two-dimensional topology map, which we demonstrate with Figure 4-2b. In the topology map, each neuron is represented by a radar plot, and each axis of the radar plot is an objective. PC1 summarizes the largest variations in objective values and the correlation between objectives. For example, moving from left to right in Figure 4-2b, f_1 and f_3 decrease, and f_2 and f_M increase. From bottom to top, f_2 decreases and f_M increases slightly. The objectives demonstrate greater variance along PC1 compared to PC2, which is determined via their eigenvalues but can also be observed visually. For instance, contrast the neuron on the bottom-left to the neuron on the bottom-right; the blue surface area is markedly smaller. Then, compare the bottom-left neuron to the top-left neuron – the surface area is also smaller, but with relatively less change. Thus, PC1 is allocated four neurons compared to three neurons for PC2 to better capture the larger variation of objective values.

By utilizing principal component-based dimension reduction, SOM summarizes the most important patterns of the objective layer in two dimensions and creates a navigable visualization.

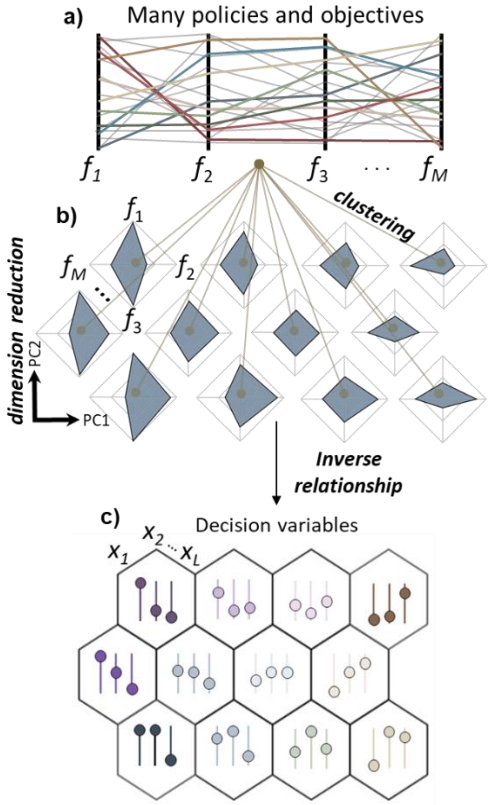


Figure 4-2: An example SOM applied to MOEA data. Subplot (a) symbolizes the objective values of MOEA-derived policies, where each axis is an objective and each trace corresponds to one policy. Subplot (b) shows the trained SOM, where each neuron is visualized as a radar plot and objectives are plotted on each axis. Policies are clustered to a neuron, and the objective layer is dimensionally reduced to principal components one and 2 (PC1 and PC2). Subplot (c) shows the corresponding decision variable values projected onto the trained SOM. Each hexagon is a neuron, and the position of each circle on its axis shows the value of DVs. Relationships between decision variables and objectives are elucidated via comparison of (c) and the PCs of (b).

We use Figure 4-2b to demonstrate the clustering of policies and the intuitive arrangement of clusters provided by SOM. Figure 4-2 symbolizes the clustering process via lines connecting part a to neurons in part b. Effectively, SOM reduces the number of alternatives a DM would consider from the hundreds of MOEA-derived policies to the number of neurons in the SOM. Moreover, neurons close together in the topology map are more similar than those far apart. For instance, compare any two adjacent neurons in Figure 4-2b. The shape of the blue area is more similar than that of neurons on

opposite ends of the map. Effectively, SOM both clusters policies and arranges the clusters based on their similarities.

SOM topology maps are also a powerful tool for synthesizing and visualizing the relationships between various data layers. We demonstrate the visualization of a DV layer in Figure 4-2c. Each hexagon is one neuron, and within each neuron are DVs. The value of the DV is indicated by the position of the circle on its axis. Note that Figure 4-2 b and c are one and the same SOM, meaning the assignment of each policy to a neuron and the location of the neurons are the same, but they visualize two different MORDM data layers. Therefore, any patterns observed in the objective layer can be related to patterns observed in the DV layer. For example, we discussed earlier that f_2 decreases top to bottom in Figure 4-2b. In Figure 4-2c, x_2 decreases from top to bottom. Thus, the conclusion is that decreasing the DV x_2 results in the decrease of objective f_2. In the next section, we describe how the post-MORDM expands on previous SOM applications to also visualize the robustness layer.

4.3.2 the post-MORDM framework

To implement post-MORDM, three data layers are needed: DV values, performance objective values, and robustness metric values. Figure 4-3 demonstrates the workflow of post-MORDM, which we describe in the following subsections.

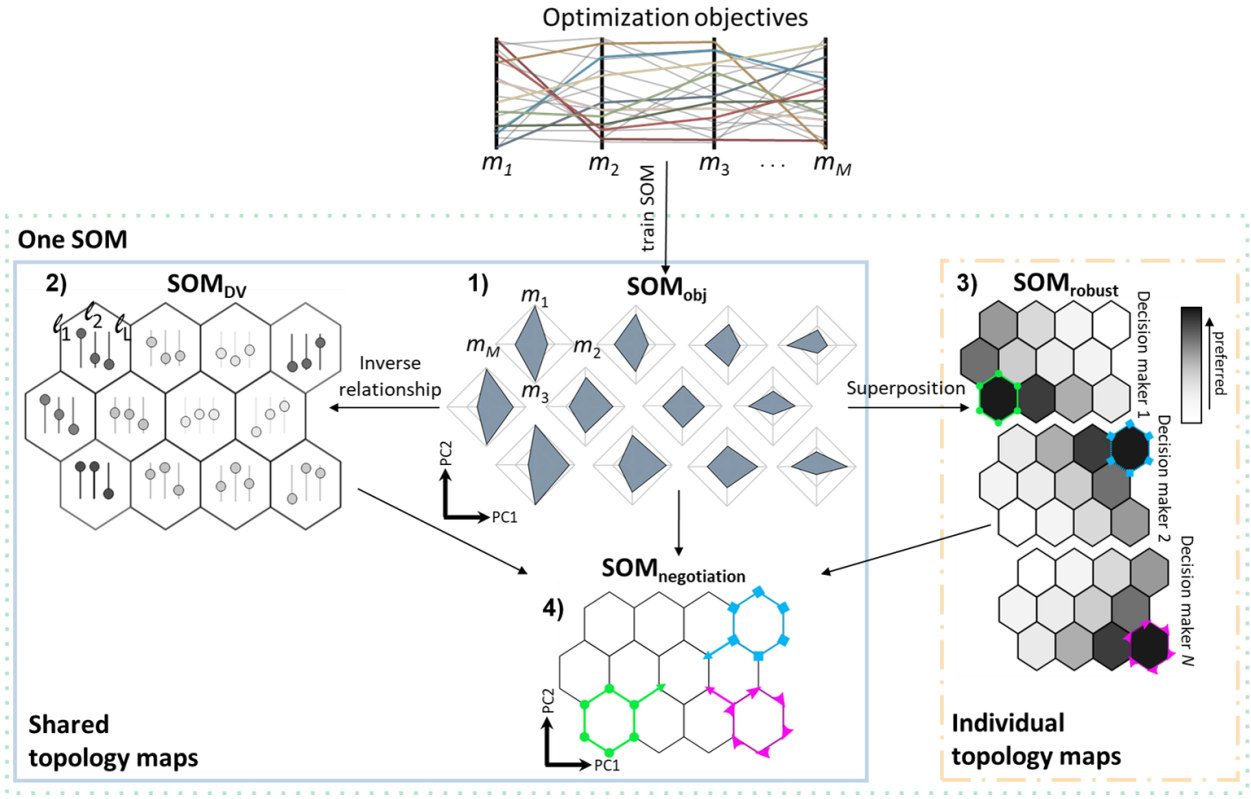


Figure 4-3: The post-MORDM framework for synthesizing the relationships between decision variables (DVs), performance objectives, and robustness, and establishing a negotiation-compromise platform. Top) the post-MORDM framework begins by training a SOM on the objective layer of MOEA-derived policies, establishing the one and only SOM used in the framework. 1) First, the objective layer is visualized on the SOM, called SOM_{obj}, utilizing principal components 1 and 2 (PC1, PC2) to interpret the most prominent patterns. 2) Next, the DV layer is projected onto the SOM, called SOM_{DV}, to investigate the inverse relationship of performance objectives to decision variables. 3) Post-MORDM augments previous SOM applications by then superimposing decision maker-defined robustness metric(s) onto the SOM, collectively called SOM_{robust}. The color of each hexagon shows the robustness value averaged over all policies assigned to the neuron, where darker is preferred. Decision makers identify their neuron(s) of preference, demonstrated by the green-circles, blue-squares, and pink-arrows for decision makers 1 through N. 4) Individual robustness preferences are projected onto the SOM, called $SOM_{negotiation}$, displaying the different robustness preferences on a single platform. Decision makers engage in a negotiation process, navigating from their individually identified neurons towards each other. While navigating $SOM_{negotiation}$, the objective and robustness tradeoffs and changing decision variable values are interpreted via SOM_{obj} and its PCs, SOM_{robust}, and SOM_{DV}. The goal is for decision-makers to identify a neuron, or small subset of neurons, that represent mutually-feasible compromise policies.

4.3.2.1 Train SOM on performance objective layer

The first step of post-MORDM is training a SOM on the objective layer of a set of policies. As described in Section 4.3.1.1, creating the SOM first requires normalizing or scaling each objective, calculating the length-width ratio, determining the number of neurons, setting hyperparameter values, initializing the neurons along PC1 and PC2, then implementing the training algorithm. We recommend testing multiple SOMs with different sizes and hyperparameter values, then selecting the smallest SOM that achieves sufficient PVE and TE. After training, the objective layer is plotted on the SOM. In this paper, we denote the MORDM data layer being visualized with a subscript; for example, SOM_{obj} shows the topology map of the objective layer (Figure 4-3.1). Each neuron in SOM_{obj} is depicted with a radar plot, which we described in Section 4.3.1.3.

4.3.2.2 Analyze inverse relationships to decision variables

Next, the DV values for each neuron are visualized via SOM_{DV}, shown in Figure 4-3.2. This visualization is the result of training SOM on the objective layer, then projecting the DV layer onto the SOM. We use the visual patterns in SOM_{DV} and the principal components of SOM_{obj} to investigate the relationships between the objective and DV layers, as discussed earlier in Section 4.2.3.2 with Figure 4-2 b-c. In Figure 4-3, we summarize each DV with a single value per neuron, which could be the mean or median value. However, alternative visualizations that show the DV values of each policy assigned to the neuron (see Section 4.4.3.2), or a visualization of their distribution (box plots, violin plots, etc.), can also be used.

4.3.2.3 Superposition robustness metrics

After establishing SOM_{obj} and SOM_{DV}, the DMs define their robustness metrics of choice. This includes the type of robustness metric, such as regret or satisficing, any performance objectives and corresponding thresholds, and any considerations of risk tolerances that result in unique definitions of robustness (see section 4.2.1). For a review of robustness metrics and guidance on selecting them, we

refer the reader to McPhail et al. (2018, 2021). We denote each DM and their robustness definition with an index, 1,2, …, *N*. Because the SOM is trained on the objective layer, not robustness layer, DMs can change or modify robustness metrics without needing to retrain the SOM.

We then superposition each unique robustness metric onto a topology map, shown in Figure 4-3.3, resulting in *N* robustness visualizations collectively called SOM_{robust}. The creation of SOM_{robust} is similar to the visualization of DV with SOM_{DV}; the SOM was trained on the objective layer, after which the robustness layer is superpositioned onto it. In Figure 4-3.3, we plot each DM's robustness value on an individual topology map, coloring each hexagon by the robustness value averaged over policies assigned to each neuron. In the SOM literature, topology maps that show only one feature are called component planes (Clark, Sisson and Sharma, 2020). The relationships between MORDM data layers are explored visually and via the PCs of SOM_{obj}. For example, we discussed earlier that, moving from bottom to top, DV x_2 decreases (Figure 4-3.2), resulting in the decrease of objective f_2 (Figure 4-3.1). Now consider SOM_{robust}. Darker neurons represent better robustness values, so when x_2 and f_2 both decrease, this is related to decrease in robustness for DM_1, increase in robustness for DM_2, and decrease in robustness for DM_N.

In a negotiation context, each DM is presented their unique robustness visualization, which they use collectively with SOM_{obj} and SOM_{DV} to identify their preferred neuron(s). In the example illustrated in Figure 4-3.3, DMs 1, 2, and *N* maximize their individual robustness preferences in the lower left, top right, and bottom right neurons, respectively. We have highlighted each DM's preferred neuron with green circles, blue squares, and pink arrows, respectively.

4.3.2.4 Navigate SOM to compromise neurons

To encourage discussion, negotiation, and compromise, we establish a topology map that is shared between the DMs. In Figure 4-3.4, we begin with a colorless topology map, meaning the neurons are not colored by robustness values, but the assignment of policies to neurons and the position of neurons are the same as SOM_{obj}, SOM_{DV}, and SOM_{robust}. Then, we project the DMs' robustness preferences

established in Figure 4-3.3 onto the colorless topology map, establishing a shared negotiation platform called $SOM_{negotiation}$. When the DMs' preferred neurons are located far apart, this indicates conflicting preferences in the weighing of objectives, DVs, and robustness. For example, DM_1 prefers the lower left neuron, which is characterized by policies with high values of x_1 and f_3, but, in contrast, DM_2 prefers the top right neuron, where policies have a small x_1 and f_3. To negotiate and compromise, DMs navigate from their individual preferences towards a neuron between DMs, the tradeoffs of which are interpreted via SOM_{obj} and its PCs, SOM_{DV}, and SOM_{robust}. For instance, the example DMs in Figure 4-3.4 negotiate to a neuron that requires each to compromise a similar amount. For DM_1, this requires a decrease in objectives f_1 and f_3 (left to right along PC1), decrease in f_2, and increase in f_M (moving upward along PC1). Considering DVs, the compromise neuron implements less of DVs x_1 and x_2 and more of x_L.

DMs may negotiate to a single neuron of mutual interest, or several neurons of mutual interest. In either case, the number of policies under consideration is significantly reduced, and the neurons can now be investigated further by analyzing the individual policies within them. In the next section, we provide an example with a case study of reservoir operation policy in the Colorado River Basin.

4.4 Post-MORDM case study: reservoir operation policy in the Colorado River Basin

4.4.1 Motivation

The Colorado River Basin (CRB) supplies municipal water for nearly 40 million people in seven US states (Basin States), 29 federally recognized tribes, and northern Mexico. The CRB is a significant source of hydropower, producing about ten billion kilowatt-hours of electricity annually – enough to power one million US households (Reclamation, 2018b; *Frequently Asked Questions (FAQs) - U.S. Energy Information Administration (EIA)*, 2020, 2021b). Moreover, CRB surface water is the primary source for the Basin States' agriculture sector, which is responsible for 70% of the CRB's consumptive use and losses (Reclamation, 2018a).

The CRB is regulated according to the *Law of the River*, a compilation of compacts, treaties, federal law, and court decisions dating back to 1922 (Reclamation, 2015). Pursuant to the *Law of the River*, the Basin States are divided into the Upper Basin (UB) – Colorado, Wyoming, Utah, and New Mexico – and the Lower Basin (LB) – Arizona, Nevada, and California – divided by a streamflow gauge at Lees Ferry, Arizona. Each basin is allocated 7.5 million-acre feet (MAF) annually for consumptive use, of which the UB is yet to fully utilize. In addition, Mexico is allocated 1.5 MAF, totalling 16.5 MAF basin-wide.

During the 21st century, persistent drought in the CRB has exacerbated the risk of 'temporary or prolonged interruptions in water supplies' (Buschatzke *et al.*, 2019). Average annual streamflow has dwindled to 72% of the historical average (Lukas and Payton, 2020), and, as of November 2021, system reservoirs are filled to only 38% of full capacity (Reclamation, 2021a). Potential consequences of this drought include LB shortages, curtailments of UB consumptive use, and critically low reservoir levels, which can also diminish hydropower production, recreational services, and environmental benefits (Reclamation, 2007).

In an effort to minimize these risks, the US Bureau of Reclamation (Reclamation), the Basin States, and Mexico have recently legislated multiple shortage operation policies for Lake Mead, the largest reservoir in the system. The 2007 Interim Guidelines (Guidelines) defined the pool elevations and corresponding volumes by which deliveries to the LB would be reduced during times of low reservoir levels (i.e. shortage volumes). Further, the Guidelines dictate how Lake Powell, the upstream reservoir from Lake Mead and the second largest in the system, would be operated in coordination with Lake Mead (Reclamation, 2007). Three Lake Mead policy alternatives were considered for the Guidelines: two alternatives prioritized water delivery and storage, respectively, and the third alternative, which was selected, 'incorporates operational elements' from both of the other two alternatives (Reclamation, 2007, p. 8). After 2007, drought persisted and reservoir levels continued to decline. Therefore, CRB stakeholders later augmented shortage volumes established in the Guidelines via Minute 323 between the US and

Mexico and the LB Drought Contingency Plan (DCP). Collectively, the Guidelines, Minute 323, and the LB DCP establish the cumulative shortage operations in effect at the time of writing this paper (International Boundary and Water Commission, 2017; *Colorado River Basin Drought Contingency Plans | Bureau of Reclamation*, 2019). Although these policies differ in terms of whether or not users can 'recover' delivery reductions when reservoir storage increases, all policies functionally decrease the risk of pool elevations at Lakes Mead and Powell declining to critically low levels.

We provide an overview of the Lake Mead shortage operations in Figure 4-4. The projected pool elevation for January 1st of the coming year is shown on the y-axis in feet above mean sea level (msl). The pool elevation determines the volume of water by which downstream deliveries are reduced for the calendar year (i.e. the shortage volume), as indicated by color. The Guidelines, Minute 323, and the LB DCP are ordered chronologically by year of implementation along the x-axis, and the cumulative shortage operation is shown on the right. The policies expire December 31st, 2025, thereafter a new policy will take effect.

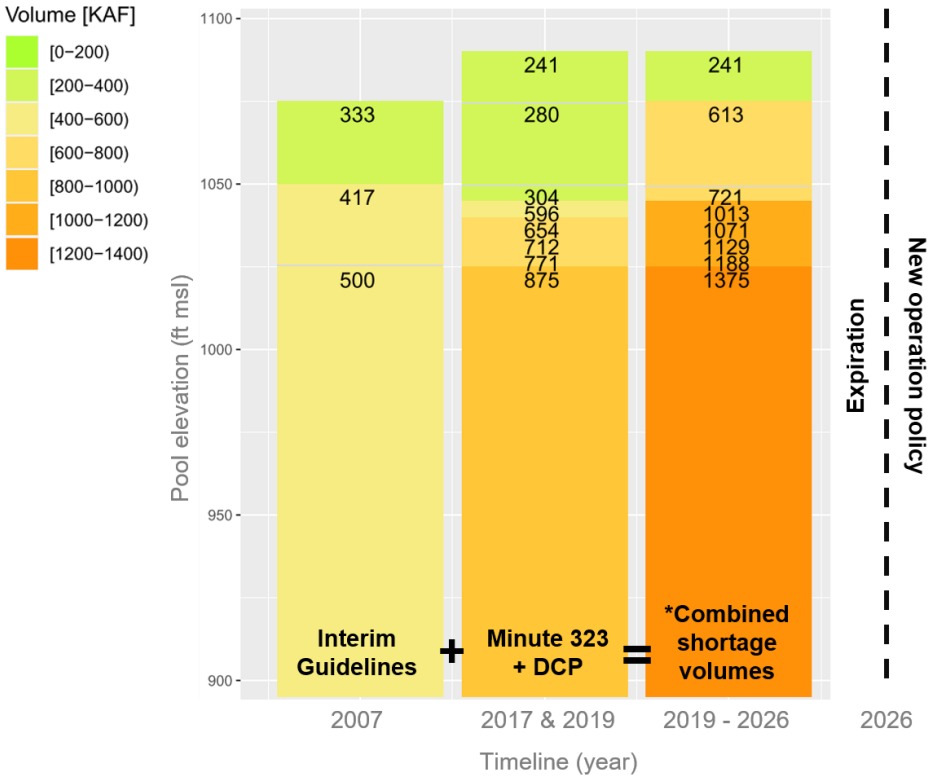


Figure 4-4: The current Lake Mead operation policy will expire at the start of 2026, and decision makers will need to negotiate a new policy. This figure provides an overview of the policies that sum to the current, combined Lake Mead shortage operations, with policies arranged chronologically from left to right. The y-axis shows Lake Mead pool elevation in feet above mean seal level (msl) and color shows the volume of water subtracted from downstream deliveries in thousand-acre feet (KAF), which we refer to as 'shortage volumes'. The 2007 Interim Guidelines was established first, then Minute 323 between the US and Mexico and the Lower Basin Drought Contingency Plan (DCP) took further precautions in an effort to avoid critically low pool elevations, resulting in the combined shortage policy in effect until expiration. *Shortage volumes under the Minute 323 Binational Water Scarcity Contingency Plan are DCP are 'recoverable' during periods of high pool elevations, unlike the Interim Guidelines or Minute 323 delivery reductions; nevertheless, they all contribute to a cumulative shortage volume at the pool elevations shown.

In this case study, we contribute to the negotiation of new Lake Mead shortage operations beginning in 2026. First, we employ MOEA to identify a set of non-dominated Lake Mead shortage operations and quantify objective values. Then, we calculate multiple robustness metrics to reflect the conflicting interests of storage and delivery stakeholders. Finally, we use post-MORDM to demonstrate a process of learning, negotiation, and compromise between two illustrative DMs.

4.4.2 Implementation of MORDM

Our MORDM problem formulation is summarized in Table 4-1. We describe the DVs, simulation model, and objectives when discussing MOEA-optimization. Next, we describe the sources of uncertainty and the methods for creating the SOW ensemble. Then, we define the robustness metrics of two example DMs in the CRB before demonstrating post-MORDM.

Problem formulation	**Description**
Uncertainty	Hydrology, demand, initial reservoir pool elevations
Decision variables (***x***)	14 variables for operation of Lake Mead. Defines pool elevations at which shortage operations occur and the associated shortage volumes (right).
Simulation model	Colorado River Simulation System Hydro-policy model in RiverWare 44 year simulation Monthly timestep
Performance objectives (***f***)	**LF.Deficit**: percentage of months annual 10yr compact volume falls below 75 maf **P.WYR**: average annual water year release from Lake Powell **P3490**: percentage of months Lake Powell pool elevation < 3490 ft msl **M1000**: percentage of months Lake Mead pool elevation < 1000 ft msl **LB.Avg**: average annual Lower Basin total shortage volume **LB.Freq**: percent of years Lower Basin is in shortage conditions **LB.Max**: max annual Lower Basin policy shortage volume **LB.Dur**: max consecutive years Lower Basin is in shortage (All minimization objectives)

Decision variables (***x***)

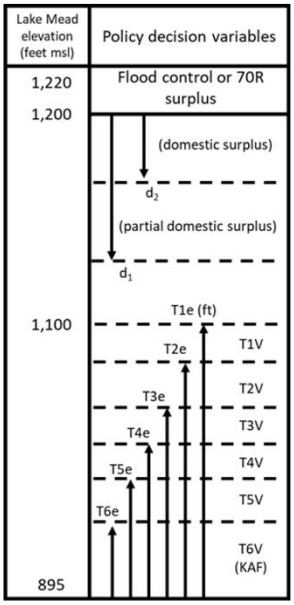


Table 4-1: The Colorado River Basin case-study problem formulation. Decision variables (***x***), the simulation model, and performance objectives (***f***) are described in section 4.4.2.1. Decision variables (right), include 6 pool elevations (T1e-T6e) in feet above mean sea level (msl) with 6 corresponding shortage volumes (T1V-T6V) measured in thousand-acre feet (KAF). d_1 and d_2 determine surplus operations and were included in MOEA optimization, but this case study will discuss only shortage operations. The characterization and sampling of uncertainty are described in Section 4.4.2.2.

4.4.2.1 Policy alternatives

We use a policy set adapted from Alexander (2018) provided by Reclamation. Policies were generated with the Borg-MOEA (Hadka and Reed, 2013), which was coupled with the Colorado River Simulation System (CRSS), a hydro-policy model built in RiverWare that serves as Reclamation's long-term planning model for the CRB (Zagona *et al.*, 2001). The simulation is 44 years long and uses a monthly

timestep, evaluating eight performance objectives. The objectives quantify tradeoffs between UB and LB interests and delivery vs. storage objectives. See Table 4-1 for definitions. Borg seeks to minimize the objectives by adjusting 14 DV (Table 4-1, right). 12 DVs control shortage operations – of which six define the pool elevations where shortage operations begin (T1e – T6e), and six are the corresponding shortage volumes subtracted from LB deliveries (T1V – T6V). The remaining two DVs are the elevations at which surplus operations begin, but this case study will discuss shortage operations only. The result is a set of 463 policy alternatives, the objectives and tradeoffs of which are explored via post-MORDM in section 4.4.3.

4.4.2.2 Robustness analysis

4.4.2.2.1 SOW ensemble generation

Our robustness analysis considers three sources of uncertainty. 1) annual cumulative natural flow above Lees Ferry, Arizona, 2) annual consumptive use in the UB, which is sampled for each simulation but held constant with respect to time, and 3) initial reservoir pool elevations at Lake Mead and Lake Powell. This section describes how we sampled the uncertainty with a 500-member SOW ensemble, the result of which is shown in Appendix B.4.

We considered four hydrology ensembles historically used by Reclamation and thus familiar to CRB DMs (Reclamation, 2007, 2012a, 2018a). 1) The Observed Resampled ensemble is the result of the Index Sequential Method (ISM) applied to the observed 1906-2007 cumulative natural flow record. 2) The Global Climate Model (GCM) ensemble is based on bias corrected and spatially downscaled CMIP3 climate projections of future high, medium, and low emission scenarios run through the Variable Infiltration Capacity (VIC) model. 3) The Paleo Resampled ensemble applies the ISM method to paleo-reconstructions dating 762 to 2005. 4) Lastly, the Paleo Conditioned ensemble uses a non-parametric technique to “blend” the wet/dry sequences from the paleo record with magnitudes from the observed record. In sum, there are 1963 streamflow traces that describe the envelope of hydrologic uncertainty. Consistent with the

philosophy of MORDM, we use the traces to broadly sample the hydrologic uncertainty space as described below. For more information on the ensembles, we refer the reader to the 2012 CRB Supply and Demand Study (Reclamation, 2012a).

Next, we created a 1000 sample Latin Hypercube (LH) of annual UB consumptive use, initial pool elevation at Lake Mead, and initial pool elevation at Lake Powell. Annual UB consumptive use ranges from 4.2 to 6.0 MAF, which considers both curtailments and growth. For comparison, in 2016 the Upper Colorado River Commission estimated consumptive use at 4.33 MAF and forecasted 5.22 MAF in 2060 (2016). Initial reservoir levels consider Lakes Mead and Powell because their combined storage accounts for about 87% of the entire system (Reclamation, 2021a). Sampling ranges were informed from the 10^{th} (low end) and 90^{th} (high end) percentile values from Reclamation's April 2020 five-year projections, rounding the low end down to the nearest 50 feet (Reclamation, 2020). Thus, Powell's initial pool elevation ranges from 3450 to 3675 feet above mean sea level (msl), and Mead ranges from 1000 to 1185 feet msl. The pool elevation projections end December 2026, accounting for the range of possible pool elevations at the expiration of the current operation policy. After creating the LH, we combine every sample of pool elevations and UB consumptive use with every hydrology trace to create a large set of SOW from which to select a subset for robustness simulations.

To reduce computational costs, we sample a subset of 500 SOW using conditioned Latin Hypercube Sampling (cLHS), which is an extension of Latin Hypercube Sampling (LHS) (Minasny and McBratney, 2006). Instead of creating new multivariate samples that form a LH, cLHS employs an optimization algorithm to select existing observations that form a LH in the multivariate feature space while mimicking the distributional properties of the original population (Minasny and McBratney, 2010; Brus, 2019; Roudier, 2020). Practically, cLHS allowed us to use existing hydrology traces (our existing 'observations'), select a subset of SOW with minimal repeats of hydrology traces, and preserve the desired uncertainty ranges.

4.4.2.2.2 Decision-maker robustness metrics

Our robustness analysis uses two illustrative DMs with conflicting preferences, namely *Delivery* and *Storage*. Delivery is of greatest concern to the LB since LB allocations are 100% utilized for irrigation, municipalities, groundwater recharge and all other uses. The Storage DM reflects hydropower interests at Lake Powell and Lake Mead. Moreover, storage is of special concern for shoreline recreational services like boat ramps and marinas (Reclamation, 2012a).

The robustness preferences of Delivery and Storage are both quantified with the satisficing metric. Satisficing is the fraction of SOW where a policy satisfies minimum performance thresholds defined by the DM. Satisficing ranges from 0 (performance thresholds satisfied in zero SOW) to 1 (performance thresholds are satisfied in 100% of SOW).

The performance requirements for Delivery are shown in condition 1:

- Delivery performance requirements: LB.Avg $\leq$ 600 KAF and LB.Dur $\leq$ 10 years

LB.Avg is the annual shortage volume in the LB, averaged over the simulation. LB.Dur is the maximum consecutive years the LB is in shortage conditions (see Table 1). Thus, Delivery's performance thresholds require both acceptable average magnitudes and maximum duration of shortages. The performance requirements for Storage are shown in condition 2:

- Storage performance requirements: M.1000 $\leq$ 10% and P.3490 $\leq$ 5%

M.1000 is the percentage of simulation months where Lake Mead's pool elevation is below 1000 feet msl, and P.3490 is the percentage of simulation months where Lake Powell's pool elevation is below 3490 feet msl. Together, the performance thresholds for Storage require that both Lake Mead and Lake Powell consistently stay above critical pool elevations.

The DMs performance thresholds are used to calculate satisficing for each policy and DM according to the equation

$$Satisficing_{i,p} = \frac{1}{S}\sum_{j=1}^{S} \quad g_{i,j}(x_p)$$

where *S* is the total number of SOW, *j* is a SOW index, ***x*** is the decision variable vector for policy *p*, and $g_{i,j}$ is an indicator function. $g_{i,j}$ = 1 if the performance thresholds of DM *i* (Delivery or Storage) are satisfied in SOW *j*, and $g_{i,j}$ = 0 otherwise.

4.4.3 Implementation of post-MORDM

4.4.3.1 Training the SOM on the objective layer

After normalizing the objectives, we selected the number of neurons and SOM hyperparameter values using a grid search of 1000 LH samples. The tested hyperparameters include neighborhood radius, neighborhood function, distance function, and edge behavior. The number of neurons in length-width directions of the SOM was calculated given the number of total neurons sampled in the LH and the calculated ratio of the first and second eigenvalues of the objective layer. We chose a hexagonal neighborhood topology according to the recommendation of Clark et al. (2020). We evaluated the number of neurons and each hyperparameter set with PVE and TE. SOM training was performed in R using the kohonen package (Wehrens and Buydens, 2007; Wehrens and Kruisselbrink, 2019), and fit metrics were calculated using the aweSOM package (Boelaert *et al.*, 2021; R Core Team, 2022). The selected SOM is 5x3 neurons because larger maps achieved small increases in PVE while smaller ones saw significant decrease in PVE, in both cases attaining similar TE. For additional details on the training process, the number of neurons to use, and the selection of hyperparameter set, including code, see Appendix B.5.

Figure 4-5 shows SOM_{obj}, which reveals the two most important modes by which policies vary with respect to the objective layer. Within the radar plots, each objective is scaled 0 (center) to 1 (outer edge), where 0 is ideal because they are minimization objectives. Every policy is plotted in its assigned neuron, using a transparent blue fill to visualize the number of policies and the degree to which their objective

values are similar. For example, consider neuron 6 (middle row, far left). These policies result in low average and maximum shortages (LB.avg and LB.max) at the expense of frequent, long-duration shortages (LB.Freq and LB.Dur). They also perform poorly in water storage reliability at Lakes Powell and Mead (M.1000 and P.3490). The black borders around each policy overlay each other almost exactly, indicating high similarity among the policies. In contrast, consider neuron 15 (top row, far right). The policies have large average and maximum shortages; however, they perform comparatively well with respect to shortage frequency, shortage duration, and reservoir storage. The borders around each policy are easy to distinguish from one another, especially for LF.Deficit. This indicates notable variation of performance for the policies assigned to neuron 15. For the interested reader, Appendix B.6 reports the average objective value in each neuron (i.e, component planes).

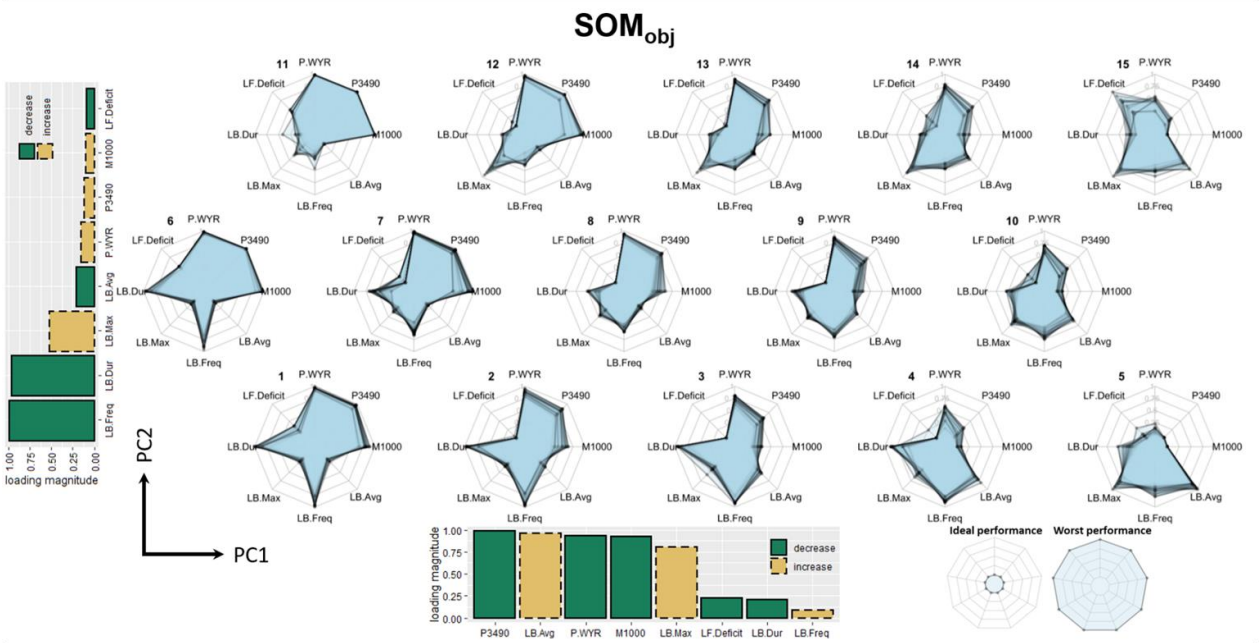


Figure 4-5: Using SOM_{obj} to synthesize the most pertinent patterns in the objective layer and cluster similar policies. Each neuron is visualized by a radar plot, where the axes show performance objectives scaled 0 (best) to 1 (worst). Each policy is plotted with a transparent blue fill circumscribed by a solid black line to visualize the number of policies per neuron and the degree of similarity between them. Neurons near each other in the map exhibit more similar objective values than neurons further apart in the map. SOM_{obj} is aligned with the first and second principal components (PC1 and PC2, respectively), for which the loading scores are given with bar plots. The height of the bar indicates the degree to which the objective contributes to the PC, and the color indicates the direction of change.

To facilitate the interpretation of the PCs, we include the loading scores of each objective, shown in the bar plots. Loading scores quantify the degree to which each objective contributes to the PC, where larger magnitudes mean the objectives contribute more to the change in performance (James *et al.*, 2013, chap. 10.2). Green bars indicate decreasing values from left to right (increase in performance), whereas gold-dashed bars indicate increasing values (decrease in performance). For example, consider PC1, left to right. P.3490, P.WYR, and M.1000 are improving, but LB.Avg and LB.Max are worsening. The other objectives have relatively little to no change in the horizontal direction. Moving from bottom to top along PC2, LB.Freq and LB.Dur are decreasing, while LB.Max is increasing. In other words, the two most significant tradeoffs for the DMs to navigate is first the tradeoff between reservoir storage reliability and LB shortage magnitudes (average and maximum) and, second, the tradeoff between shortage duration/frequency and maximum magnitude.

4.4.3.2 Lake Mead operation policies

After we establish SOM_{obj}, we visualize the inverse relationship to Lake Mead DVs with SOM_{DV}, shown in Figure 4-6. Each vertical bar represents a Lake Mead policy, where the y-axis is water surface elevation, and the colors show the magnitude of shortage. The number of policies in each neuron is shown to the right of the neuron index in parentheses. For neurons with more than 20 policies, 20 policies are plotted at random to conserve space. For simplicity, the policies are ordered randomly along the x-axis, but could be ordered according to a DV, robustness metric, or objective. To facilitate comparisons between neurons and between MORDM data layers, we report the tier 1 elevation (T1e, also indicated by the horizontal dashed line, in feet msl) and volume of the first shortage (T1V, KAF), plus maximum shortage (maxV, KAF), averaged across all policies in a neuron.

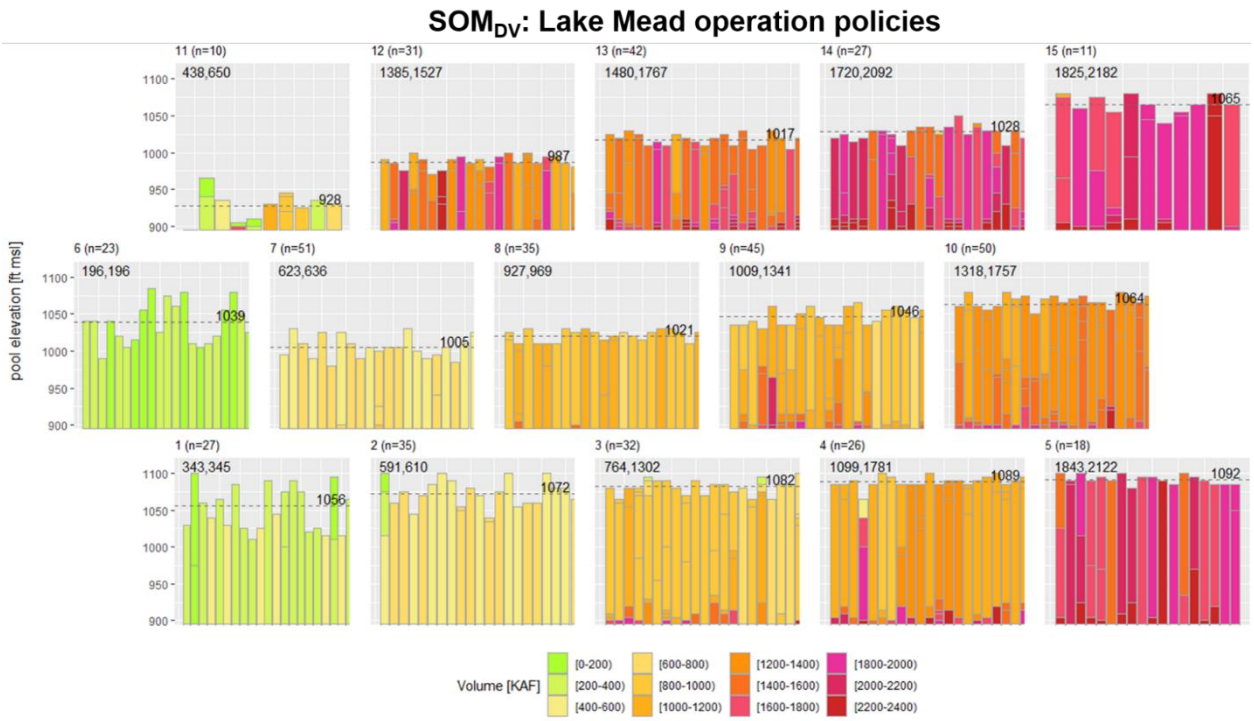


Figure 4-6: Using SOM$_{DV}$ to visualize the inverse relationship of performance objectives to Lake Mead decision variables. Each bar represents a Lake Mead policy, where the y-axis denotes water-surface pool elevation and fill color indicates the corresponding shortage volume. In the top left of each neuron we report the volume of the first shortage (T1V, KAF), plus maximum shortage (maxV, KAF), averaged across all policies in a neuron. T1V and maxV are reported in the format (T1V,maxV). Further, the horizontal dashed line shows the average tier 1 pool elevation (t1e), averaged across all policies in the neuron. Although the SOM was trained on performance objectives, salient patterns in the decision variable layer are discovered. From left to right, shortage volumes increase (green to red gradient) and shortage elevations increase (height of bars increases). SOM$_{DV}$ is compared to SOM$_{obj}$ in Figure 4-4 to elucidate the relationships between decision variable values and objective performance.

Figure 4-6 shows salient patterns in shortage volumes and T1e corresponding to the PCs of SOM$_{obj}$. From left to right, T1V, maxV, and T1e tend to increase. This is shown by the gradient of green-yellow-red and the increasing bar height. Comparing to SOM$_{obj}$ in Figure 4-5, the result is improved reliability of reservoir storage, the tradeoff being increased average and maximum shortages. From bottom to top, T1e tends to decrease, especially to the left side of SOM$_{DV}$. T1V and maxV also tend to increase. The related performance outcome is decreasing frequency and duration of shortages at the expense of increasing maximum shortage. Interestingly, T1V and maxV increase both left to right and bottom to top, but the frequency and duration of shortages experienced by the LB responds almost exclusively in the

vertical direction, as indicated by loading scores of near one for the vertical PC compared to loading scores less than 0.25 for the horizontal PC. Considering that T1e tends to increase left to right but decreases bottom to top, we conclude that reducing the frequency and duration of LB shortages meanwhile achieving reliable reservoir storage requires that larger LB shortages be paired with lower pool elevations. This is the difference between neuron 15 and 5; both implement large shortage volumes – an average T1V volume of 1825 and 1843 KAF, respectively, but policies in neuron 15 achieve less frequency and duration of LB shortages via being more 'patient', waiting until lower T1e (1065 vs 1092 ft msl) to implement the first shortage. This difference in operation philosophy results in shortage conditions occurring in 12.01% less simulation months in neuron 15 compared to 5 (not shown in Figure 4-5, see Appendix B.6).

4.4.3.3 Decision-maker robustness maps

Figure 4-7 a-b shows the individual robustness values of the Delivery and Storage DMs, SOM_{robust}. The color and label of each neuron indicates the average satisficing of the policies assigned to each neuron, where darker colors are preferred. Clear topologic patterns exist, demonstrating the effectiveness of robustness metrics superpositioned on a SOM fit to the objective layer. In this example we present two satisficing robustness metrics; moreover, we demonstrate the effectiveness of this method using TE and PVE for three additional metrics in Appendix B.7 – B.13.

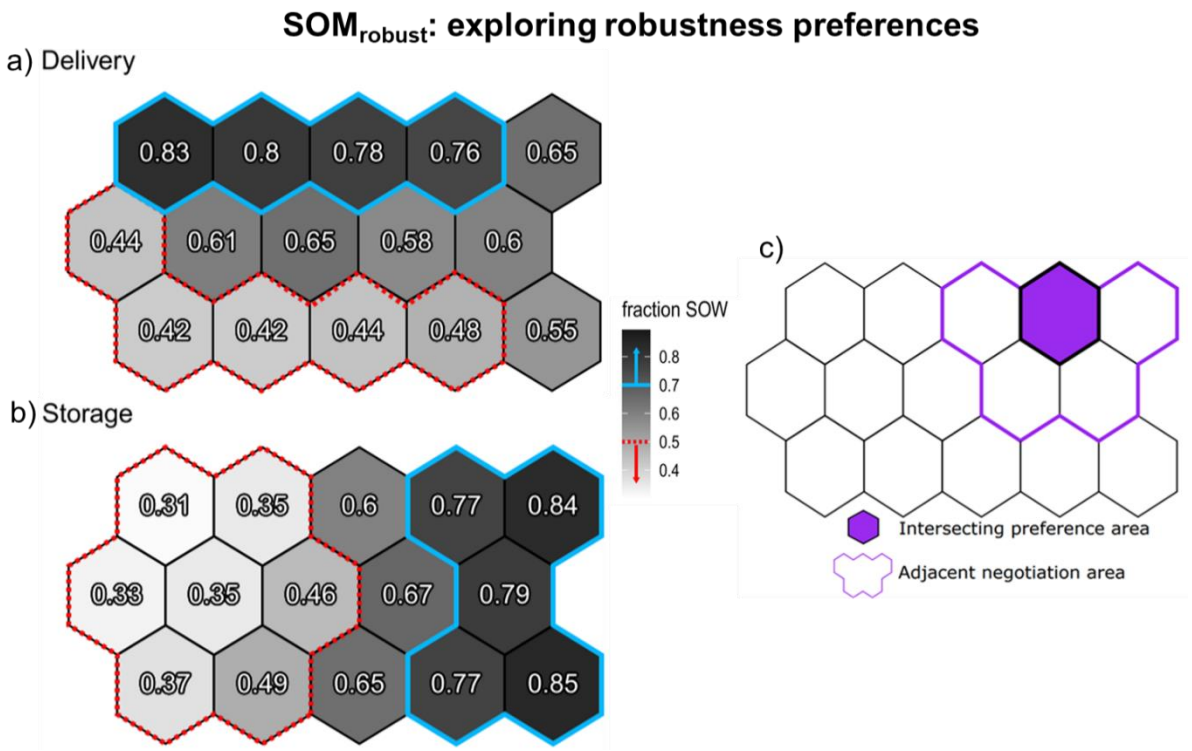


Figure 4-7: Using SOM_{robust} to identify neurons of individual interest and establish a mutual negotiation area. Delivery and Storage satisficing metrics are superimposed on the SOM, where the color indicates the satisficing value averaged over each policy assigned to the neuron (a-b). The decision makers identify their preferred neurons (outlined in blue, defined by performance thresholds satisfied in greater than 70% of SOW) and unacceptable neurons (dashed red, less than 50% of SOW). One neuron is located within the intersection of the decision makers' preferred regions (subplot c, purple fill). Further, five adjacent neurons lie outside both of the decision makers' unacceptable regions, establishing a mutually feasible area to be investigated in further negotiation (purple outline).

For Delivery, neurons in the top left are the most robust, with satisficing decreasing left to right on the top row (decreasing robustness). Comparing to SOM_{DV} in Figure 4-6, the policies that result in the best Delivery robustness do so by two operational strategies. First, consider neuron 11. These policies implement small shortage volumes (maxV of 650 KAF) at low pool elevations (T1e less than 950 feet in all but one policy), draining Lake Mead with minimal storage reliability safeguards. Alternatively, a policy can be robust by delicately balancing shortage volume and elevation (neurons 12-14). These neurons implement moderate to severe T1V (1385 to 1720 KAF) at moderately high elevations (T1e from 987 to 1028 feet msl). Other neurons do not exhibit this balance; for example, consider neurons 1 and 6. These policies implement small T1V shortage volumes (196 to 345 KAF) implemented at moderate to high

elevations (1039 to 1056 feet). These neurons are not robust, likely because the shortage volume is not large enough to raise Lake Mead's pool elevation out of shortage operations, resulting in durations of LB shortage exceeding 10 years. Contrast neuron 15 (top right) to neuron 5 (bottom right). Both implement large T1V shortage volumes (1825 and 1843 KAF, respectively), but neuron 15 policies wait until lower pool elevations to begin shortages (T1e of 1065 vs 1092 feet msl), satisfying Delivery's performance thresholds in 10% more SOW, on average. By exploring the relationships between SOM_{robust} and SOM_{DV}, the Delivery stakeholder can identify policies that achieve this balancing act to satisfy both average shortage and shortage duration performance thresholds.

For Storage, neurons to the far right are the most robust, and satisficing decreases towards the left in each row. Considering SOM_{DV} in Figure 4-6, the policies in neurons 5, 10, and 15 have the most aggressive shortage operations, implementing large T1V and maxV (1318 to 2182 KAF) at high pool elevations (T1e from 1064 to 1092 feet msl). Decreasing robustness is caused by smaller shortage volumes beginning at lower pool elevations.

In a negotiation context, each DM is provided with the topology map showing their unique robustness definition, upon which they identify neurons with preferred and unacceptable performance. In Figure 4-7, we define preferable performance as greater than 0.70 (blue), unacceptable below 0.5 (dashed-red), and for neurons from 0.5-0.7 the DMs have weak preferences. Then, we project the intersection of their preferences onto a blank topology map, shown in Figure 4-7-c. One neuron presides in both DMs preferred areas, shown in solid purple. This is an intuitive neuron for both DMs to investigate further. However, the adjacent neurons lie within either the preferred or weak preference areas of both DMs, thus we consider these neurons a feasible negotiation space.

4.4.3.4 Negotiation navigation

Next, we visualize only the neurons defined to be inside the feasible negotiation space. This enables the DMs to investigate further the DV, objective and robustness layers of mutually feasible

policies, shown in Figure 4-8. Here, we have replaced the robustness topology maps from Figure 4-7 with boxplots, providing information on the spread of robustness in each neuron. Delivery is shown in orange, and Storage is shown in blue. We have provided the boxplots of all neurons in Appendix B.14.

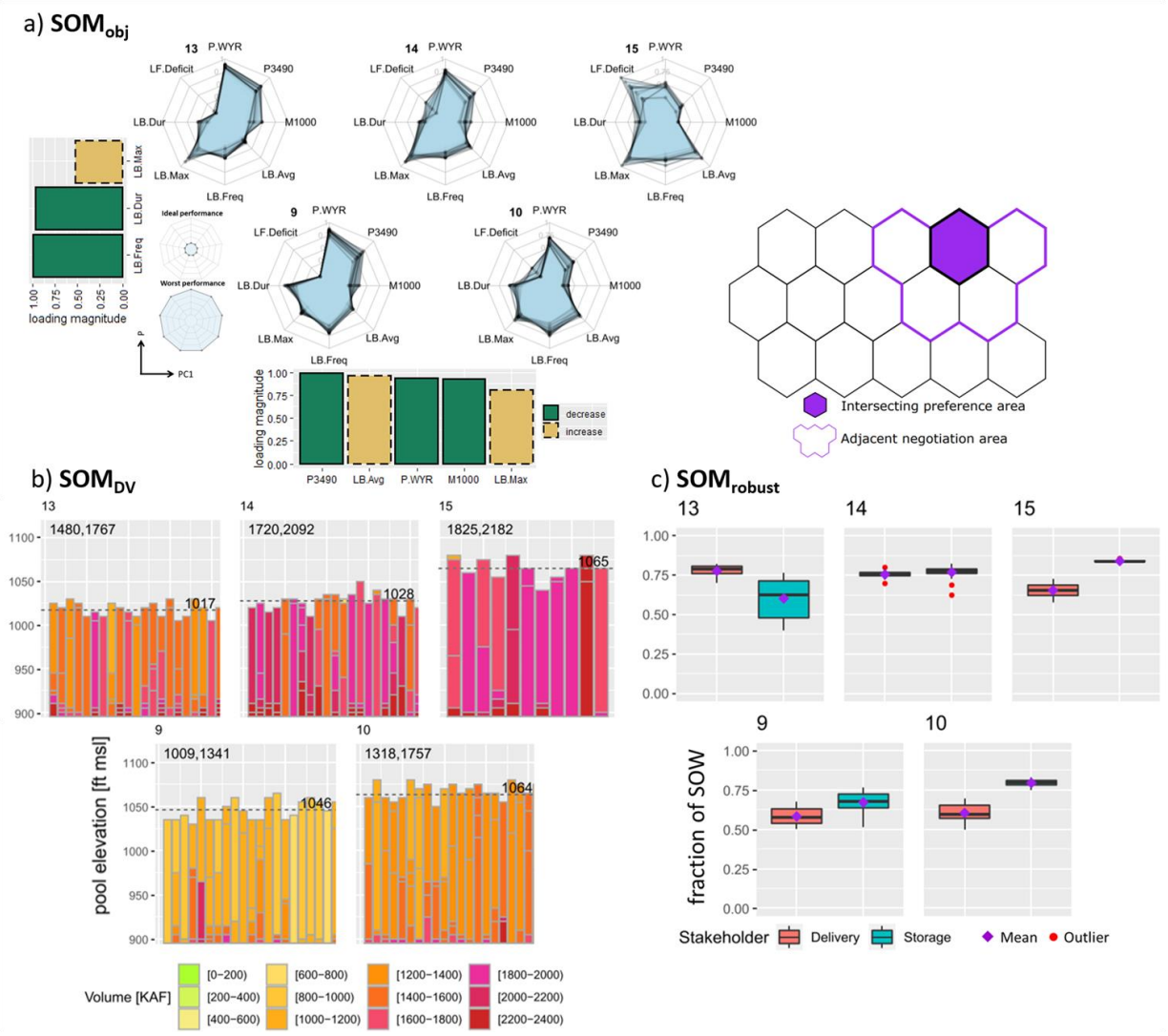


Figure 4-8: Decision makers negotiate Lake Mead policies within the mutually feasible area. Only the five neurons within the feasible negotiation area are plotted, shown in SOM_{obj} (a), SOM_{DV} (b), and SOM_{robust} (c). We truncated the bar plots in SOM_{obj} (a) to include only objectives with loading scores exceeding 0.25. For SOM_{robust} (c), we have replaced the visualization from Figure 4-6 a-b with boxplots to show the distribution of satisficing values. The Delivery and Storage decision makers are shown with orange and blue, respectively. Section 4.3.4 describes how the decision makers could use objectives not considered in their robustness definitions to negotiate for neurons other than 14, the only neuron within each of their preference areas. The effects of negotiation are interpreted via SOM_{obj} and its PCs, SOM_{DV}, and SOM_{robust}. By negotiating the map and selecting a neuron, decision makers greatly reduce the number of policies under consideration and identify the characteristics of decision variable values that result in their mutually agreed upon performance and robustness tradeoffs.

Neuron 14 mutually intersects both DMs preferred areas; however, additional negotiation can be facilitated with Figure 4-8. For example, Delivery may try to negotiate leftward to neuron 13, improving

their satisficing score and avoiding comparative disadvantage to Storage (Blount and Bazerman, 1996; Tsay and Bazerman, 2009). In this neuron, shortage volumes are smaller and implemented at lower pool elevations. Contrarily, Storage may negotiate to neuron 15 to increase robustness, meaning that pool elevations and shortage volumes would increase.

The map-based visualization of multiple MORDM data layers, such as demonstrated in Figure 4-8, can also help DMs overcome cognitive myopia in a negotiation context. Cognitive myopia can occur when the expressed interests of a DM, such as their definition of robustness or weighing of objectives, limits the exploration of mutually feasible policy alternatives (Kasprzyk *et al.*, 2013; Giuliani *et al.*, 2014). For example, when considering only robustness, it does not appear that either DM would individually desire moving downward to neurons 9 or 10 because moving horizontally yields the greatest individual increases in satisficing. However, the consideration of objective and DV layers reveals other reasons these neurons may be of interest. For instance, LB.max is improved by moving downward to neurons 9 and 10. Perhaps the satisficing performance thresholds on LB.Avg and LB.Dur reflect Delivery's highest and most well-defined priorities, but other findings like this could make these neurons interesting. If so, Delivery may negotiate to neuron 10, which implements smaller shortage volumes at higher elevations. Storage may accept neuron 10 because their satisficing value is mostly unaffected. By simultaneously visualizing multiple MORDM data layers via topology maps, DMs are encouraged to explore policy alternatives they might not otherwise, which can help DMs negotiate to a compromise policy.

4.4.4 Robust shortage policies vs existing Lake Mead operations

The combined Lake Mead shortage operation (Figure 4-4) does not resemble the neurons within the feasible negotiation space of Delivery and Storage (Figure 4-7). This section describes how post-MORDM can explore the potential reasons why and the consequences thereof.

Compared to the feasible neurons, the combined operation implements T1e at too high of an elevation with too small of a T1V. Further, maxV of the combined operation is small in comparison to the

feasible neurons, except neuron 9. Notably, the combined operation resembles neuron 9 if the combined operation's shortage volumes of less than 1000 KAF are removed. Without these smaller shortage volumes, T1e, T1V, and maxV of the combined operation are similar to the average values of neuron 9 (1045 feet msl, 1013 KAF, 1375 KAF compared to 1046 feet msl, 1009 KAF, 1341 KAF).

By navigating SOM_{DV}, SOM_{obj}, and SOM_{robust}, we can explore the potential consequences of these relatively high elevation, low volume shortage operations used in the combined operation. In this comparison, we will use neuron 3 to represent the combined operation because of similar T1e, T1V, and maxV. For a detailed comparison of neuron 3 and the combined operation, please see Appendix B.15. Using SOM_{DV} (Figure 4-6), traverse from neuron 9 to neuron 3; T1e increases by 36 feet, T1V decreases by 245 KAF, and maxV decreases by 39 KAF. This change in DV values resembles the difference between neuron 9 and the combined operation (increasing T1e, decreasing T1V, and similar maxV). The result of moving from neuron 9 to neuron 3, as indicated by SOM_{obj} (Figure 4-5), is greater frequency, duration, and average volume of shortages.

The increased frequency, duration, and average volume of shortages explains why the policies within the DMs' feasible negotiation space are dissimilar to the combined operation. Policies that combine high T1e with small T1V fail to satisfy Delivery's robustness criteria, which requires that the maximum duration of shortage not exceed 10 consecutive years and that average shortage volume not exceed 600 KAF. This conclusion is consistent with SOM_{robust}, which shows that these policies satisfy both of Delivery's performance criteria in only 44% of SOW (Figure 4-7).

To summarize, the combined operation implements relatively small shortages at high pool elevations, and our results suggest that this operational strategy can result in high frequency, duration, and average volume of shortages unfavorable to water users. In contrast, policies that begin shortages at lower elevations and larger volumes can reduce the frequency, duration, and average volume of shortage, with the tradeoff being larger maximum shortages. Therefore, this tradeoff should be emphasized during

the renegotiation of Lake Mead shortage operations to solicit feedback from stakeholders. Further, these results highlight that DMs in the CRB and other river systems facing deep uncertainty need to consider which types of water reductions (high frequency vs. high magnitude) lend themselves towards more sustainable agriculture and public acceptance.

4.5 Discussion

In Sections 4.3 and 4.4, we have introduced the post-MORDM framework then demonstrated it with a case study of reservoir operation policy. In the following discussion, we describe several ways that post-MORDM allows for application-specific flexibility, meanwhile highlighting best-practices. Then, we discuss how flexibility in post-MORDM creates ample opportunity for future research to build on the case study presented in this paper.

4.5.1 Flexibility and best practices with post-MORDM

One goal of the post-MORDM framework is reducing the number of alternatives that DMs need to consider. In Section 4.4.3.4, DMs negotiated between five neurons, where each neuron summarizes the key attributes of a group of similar policies. This number of neurons is consistent with the psychology literature that suggests 3 – 9 alternatives be ideally considered at one time (see Section 4.2.2). However, in applications with more than two DMs and/or a large number of feasible neurons, 10 or more neurons may be needed to represent the negotiation space. After identifying one or more compromise neurons, DMs may want to consider the individual policies assigned to them. However, the number of policies assigned to each neuron may exceed 9 (it does in Section 4.4.3.4). In this case, additional steps can be taken to reduce the number of policies if so desired. For example, policies can be filtered by DV, objective, or robustness values. Alternatively, it is possible to reduce the number of policies the SOM is trained on by applying filters before implementing post-MORDM. Doing so could reduce the number of policies assigned to each neuron, result in neurons that are better approximations of the policies, and result in neurons that are more distinct from each other. For instance, the illustrative DMs in our case study

defined nine neurons as unacceptable, which suggests that some policies could have been removed before training the SOM. In other cases, however, defining the criteria by which policies are filtered may be non-trivial and not agreed upon by DMs. Thus, we advocate for *a posteriori* defining of filter criteria, if any, based on exploration of topology maps.

In the post-MORDM framework, we have used robustness values as the data layer which DMs use to identify their policies of interest and the starting point for ensuing negotiation. We encourage this method because robustness metrics take into account critical driving forces of the system that are characterized by deep uncertainty. Alternatively, DMs could express their preferences based on the objective or DV layer, or a combination of layers. We advocate that DMs identify their preferred neurons beginning with the robustness layer; however, the post-MORDM framework allows for such flexibility.

The post-MORDM framework is also flexible with respect to the types of plots used in topology maps. In our case study, we have created topology maps with radar plots (Figure 4-5), reservoir operation diagrams (Figure 4-6), component planes (Figure 4-7 and Appendix B.6), and box plots (Figure 4-8c). Alternative visualizations can be used based on the application or preferences of DMs, and future research can explore alternative methods for visualizing the intra-neuron variance besides the radar plots and box plots used in this case study. Regardless of the plot type, we believe it is critical to maintain the topologic arrangement of neurons produced by the SOM, and to use easily interpretable visualizations that facilitate the exploration of tradeoffs and the relationships between MORDM data layers.

Another fundamental goal of post-MORDM is to provide DMs with a shared negotiation platform, which we accomplish with navigable topology maps. However, we do not prescribe the exact topology map visualization type to be used in negotiation. For instance, in Section 4.4.3.4 we facilitated DM negotiation with a blank topology map, meaning the hexagons in $SOM_{negotiation}$ were colorless except for the outlines that indicated the preferred neurons of each DM. Then, the illustrative DMs used SOM_{DV}, SOM_{robust}, and SOM_{obj} to interpret the results of moving on $SOM_{negotiation}$. This plot type presents

$SOM_{negotiation}$ as a neutral topology map, choosing not to visualize any objective, robustness, or DV values on it. However, we can imagine a circumstance where DMs would benefit from including additional information on $SOM_{negotiation}$. For instance, an objective or robustness metric that represents a shared concern of the DMs could be plotted to further encourage negotiation. In the CRB case study of the Delivery and Storage DMs, perhaps an environmental objective could be used.

We have demonstrated how post-MORDM facilitates negotiation and compromise via topologic visualization of multiple MORDM data layers. We do not define the exact procedure by which negotiation occurs; instead, post-MORDM can be implemented with formal negotiation rules that are application specific and agreed upon by the parties to a negotiation. For example, Gold et al. implement *fallback bargaining* to identify two compromise water portfolios in a case study of four interconnected water utilities (2019b), building on the work of Herman et al. described in Section 4.2.1 (2014). Under fallback bargaining, DMs first rank policies according to their individual preferences. Then, DMs consider the top-ranking policy for each DM; if they do not agree, each DM falls back one level in their ranking of policies. DMs continue to fall back until there exists a policy that is acceptable to every DM, and this policy is selected as a compromise (Brams and Kilgour, 2001; Madani, Shalikarian and Naeeni, 2011). The post-MORDM framework could complement fallback bargaining by facilitating DMs in the ranking of policies. Using post-MORDM, DMs could rank a tractable number of neurons, as opposed to hundreds of policies in the context of MOEA. Further, post-MORDM could help DMs rank policies based on simultaneous consideration of the objective, DV, and robustness layers, as facilitated with topology maps. Lastly, DMs could use SOM_{obj}, SOM_{DV}, and SOM_{robust} to track the tradeoffs resulting from each fallback step in the negotiation. Overall, the post-MORDM framework provides navigable visualizations that promote understanding and negotiation, but it does not prescribe a formal negotiation procedure. Thus, post-MORDM could be used to enhance the efficacy of prescribed negotiation procedures such as fallback bargaining.

Throughout this paper we have emphasized post-MORDM in negotiation contexts. Indeed, we believe this to be a significant contribution to environmental modeling and decision support literature. Alternatively, post-MORDM can be utilized by an individual analyst, design group, or organization because the benefits of post-MORDM (clustering, dimension reduction, and map-based visualization of multiple data layers) can also help individual entities attain greater understanding of their system and make decisions.

4.5.2 Future research opportunities: other data layers

Future research could implement post-MORDM to explore additional decision-relevant data layers. Scenario Discovery, the last step in MORDM, is traditionally performed on a small subset of policies. Alternatively, vulnerability information for each policy, or a representative policy from each neuron, could be calculated then displayed on a topology map. Then, vulnerability topology maps could be visualized alongside SOM_{DV} to analyze the relationships between DVs and vulnerability.

Several recent publications have highlighted another potentially decision-relevant data layer, which is the degree to which robustness values are sensitive to statistical properties of the SOW ensemble. These properties include the number of SOW, the upper and lower bounds of the uncertain factors, their correlation structure, and their probability distributions (Hadjimichael, Quinn, *et al.*, 2020; Reis and Shortridge, 2020). McPhail et al. established a framework to quantify the sensitivity of robustness magnitude and robustness ranking (2020), which could be combined with the post-MORDM framework to identify DV and objective values that correspond to policies whose robustness is the most insensitive to SOW ensemble design.

In this paper, we have demonstrated the utility of the SOM for map-like interpretation of objective, DV, and robustness layers, and we discussed above how post-MORDM could be expanded to vulnerability and robustness sensitivity layers. Furthermore, we believe the foundational methods and goals of post-MORDM could be implemented with layers relevant to decision support frameworks other

than MORDM. Of particular interest, Dynamic Adaptive Policy Pathways (DAPP) is another popular framework for systems faced with deep uncertainty because it frames policy implementation as conditional on future observations of uncertain factors (Haasnoot *et al.*, 2013; Kwakkel and Haasnoot, 2019, p. 359). Like MORDM, DAPP is characterized by several layers of decision-relevant data, where understanding the relationships between them is important to DMs. Depending on how DAPP is implemented, these data layers can include long-term policy decisions (dynamic planning), adaptive policy decisions (short-term contingency planning), signpost variables, signpost triggers, and performance objectives (Haasnoot *et al.*, 2013; Zeff *et al.*, 2016). The benefits of post-MORDM, namely the simultaneous, map-based visualization of related data layers and a negotiation platform, could also enhance the synthesis and communication of other decision-support frameworks to DMs.

4.6 Conclusion

This paper presented the post-MORDM framework, which enhances MORDM-based decision support via a novel implementation of the SOM. Post-MORDM constitutes an alternative paradigm for how policy-relevant data is explored by interpreting MORDM data as multiple layers arranged according to a two-dimensional map system. Post-MORDM expands on previous applications of clustering, dimension reduction, and the SOM to 1) help DMs elucidate the relationships between DVs, objectives, and robustness; 2) reduce the number of alternatives DMs need to consider; and 3) establish a visual, structured platform whereby DMs with foundational disagreements are assisted in a process of negotiation and compromise.

We demonstrated post-MORDM with a case study of reservoir operation policy in the Colorado River Basin, USA. Using a topology map of objective values (SOM_{obj}), our results showed that the primary tradeoff DMs need to navigate is the tradeoff between reservoir storage reliability and average magnitude of water delivery shortages. The second strongest tradeoff is between frequency/duration of shortages and maximum shortage magnitude. We illustrated how topology maps can be used in a process of

negotiation between two illustrative DMs that represent delivery and storage interests in the CRB. Based on individual definitions of robustness, we showed that five neurons of policies could be mutually feasible, and demonstrated how topology maps facilitate negotiation because of their map-like navigation and interpretation. We discuss how the combined Lake Mead shortage operation, which is in effect until 2026, contrasts with the mutually feasible neurons because of a combination of high elevation, low volume shortage tiers. We used topology maps of DV and objective values (SOM_{DV} and SOM_{obj}) to describe how high elevation, low-volume shortage tiers increase the frequency and duration of delivery shortages while reducing the maximum shortage volume. In the renegotiation of Lake Mead's shortage policy, DMs will need to consider which tradeoff the CRB's diverse stakeholders can tolerate - long, persistent water shortages of smaller magnitude, or less frequent, shorter, but harsher shortage magnitudes. Moreover, future research should incorporate DVs for Lake Powell operations into MOEA optimization, further investigating performance and robustness tradeoffs while maximizing the benefits of coordinated operation between Lakes Powell and Mead.

The post-MORDM framework and the case study presented in this paper contribute to one of the grand challenges of the 21st century - identifying policies for human-environmental systems that balance competing objectives and are robust to uncertainty. Addressing the decision-related challenges posed by deep uncertainty requires the integration of research across multiple disciplines. Therefore, the post-MORDM framework is a demonstration of this integration, pulling from research in the domains of machine learning, engineering design, psychology, and water resources management in an effort to build a bridge between decision support systems originating in academia to DMs. Our hope is that the post-MORDM framework will facilitate negotiation and compromise as decision support frameworks like MORDM are increasingly implemented in real-world applications. Moreover, we believe this research offers an alternative paradigm through which tradeoff analyses and negotiations can occur, encouraging

future studies to expand upon our implementation of the SOM while also exploring other innovative approaches.

4.7 Software and data availability

All R code and data to reproduce the case study is available on GitHub: https://github.com/nabocrb/post-MORDM. We have formatted the code to facilitate straightforward application of post-MORDM in other case studies, including code for every step described in Section 4.3 and the variety of topology maps used in this paper.

4.8 Acknowledgements

We would like to thank the Bureau of Reclamation for providing the Lake Mead objectives and decision variable data and the anonymous reviewers who contributed to the clarity and technical thoroughness of this article. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 2040434.

Author contributions:

- Nathan Bonham: conceptualization, analysis, coding, visualizations, writing-original draft
- Joseph Kasprzyk: conceptualization, writing-review and editing, supervision
- Edith Zagona: conceptualization, writing-review and editing, supervision

5 Taxonomy of purposes, methods, and recommendations for vulnerability analysis

5.1 Introduction

Coupled, human-environmental systems are managed with policies to provide benefits to stakeholders. Policies refer to specific rules defined by values of decision levers. In river systems, for example, reservoirs are operated according to policies (Alexander, 2018; Quinn *et al.*, 2018); the decision levers would determine levels of storage that trigger actions such as flood releases, also defined by decision lever values. Due to their complexity, it can be difficult to know the impacts of policies on system benefits. To provide decision support, simulation models estimate system benefits under a given policy.

Policy performance is impacted by uncertain factors that are out of the decision makers' control, such as future climate and population (Lempert, Popper and Bankes, 2003; Kasprzyk *et al.*, 2013). Values for these factors are unknown, so analysts model them as individual states of the world (SOW). Each SOW represents a combination of values of the uncertain factors. Traditional scenario analysis creates a small number of scenarios and estimates how likely each scenario is. However, environmental systems are characterized by *deep uncertainty,* in which decision makers do not know or do not agree on the likelihood of a given SOW (Knight, 1921; Kwakkel and Haasnoot, 2019). Decision Making Under Deep Uncertainty (DMDU) methods create many SOW to fully explore these uncertainties. These multiple SOW necessitate multiple runs of the simulation model to evaluate the performance of a single policy.

In addition to multiple policies and SOW, systems must meet multiple, conflicting goals. River systems, for example, must often provide reliable water supply, produce hydropower, and prevent flooding (Alexander, 2018; Quinn *et al.*, 2018). From the simulation model runs, system reliability and benefits are quantified using multiple performance metrics, such as the percent of time water demands are met, and the percentage of time with sufficient storage to produce hydropower. Tradeoffs exist when increased performance in one metric comes at the detriment of other metrics (Kasprzyk *et al.*, 2013). For example, a reservoir policy could increase releases to downstream users, meanwhile reducing reservoir

storage and hydropower production (Alexander, 2018). In the presence of tradeoffs, decision-makers choose policies that obtain their desired prioritization of conflicting goals.

Vulnerability analysis discovers concise descriptions of policies and SOW that result in 'decision-relevant' performance outcomes (Steinmann, Auping and Kwakkel, 2020). A common example is using vulnerability analysis to discover values for the uncertain factors that bifurcate the performance outcomes into a binary classification: acceptable versus unacceptable performance (Bryant and Lempert, 2010; Hadjimichael, Quinn, *et al.*, 2020). Multiple terms are used in the literature to describe these performance outcomes, including 'interesting' (Bryant and Lempert, 2010; Steinmann, Auping and Kwakkel, 2020), 'policy-relevant' (Jafino and Kwakkel, 2021), and 'consequential' (Hadjimichael, Quinn, *et al.*, 2020).

This review contributes a systematic treatment of how *performance outcome structure* is defined. Specifically, performance outcome structures are binary, multi-class, or continuous. For multi-class structures, new methods define decision-relevant outcomes based on performance inequalities across multiple stakeholders (Jafino and Kwakkel, 2021) and non-stationarity of performance over time (Steinmann, Auping and Kwakkel, 2020). Depending on the decision-making context, however, binary (Dixon, Lempert, LaTourrette and Reville, 2007) and continuous outcome structures are also beneficial (Quinn *et al.*, 2020).

After the SOW ensemble, model runs, and the performance outcome structure are defined, factor mapping is performed. Factor mapping discovers the subset of model inputs (decision levers and uncertain factors) that are the strongest predictors of performance outcomes (Bryant and Lempert, 2010; Herman *et al.*, 2015). Factor mapping also returns the values for the model inputs that lead to a decision-relevant outcome (e.g., which values cause performance to be in the "unacceptable" binary class). For example, factor mapping could reveal that average precipitation is the strongest determinant of reservoir levels, and that unacceptable levels are expected if precipitation is less than 85% of the historical average (Groves *et al.*, 2013; Reis and Shortridge, 2020). These concise descriptions are communicated to decision-makers

as scenarios, which can be helpful for comparing policies (Groves *et al.*, 2013) or identifying helpful modifications to an existing policy (Dixon, Lempert, LaTourrette and Reville, 2007).

Novel methods for factor mapping are increasingly proposed in the literature. The new methods better address multi-class performance outcome structures (Steinmann, Auping and Kwakkel, 2020; Jafino and Kwakkel, 2021) and nonlinear interactions between policies, SOW, and performance (Trindade, Reed and Characklis, 2019; Quinn *et al.*, 2020). The benefit of this body of literature is exposing DMDU to advanced tools that can address technical challenges. However, a potential limitation is that the resulting scenarios from more complex algorithms may be less interpretable for decision-making (Rudin *et al.*, 2022). In this review, we show how best practices from the field of machine learning can aid the selection of factor mapping algorithms for vulnerability analysis. We discuss interpretability and flexibility and show how evaluating factor mapping methods using *testing* accuracy instead of *training* accuracy can help identify more interpretable scenarios for decision-making.

This review will use a consistent example: reservoir operations in the Colorado River Basin (CRB). The CRB is managed with a system of reservoirs to supply water for 40 million people across seven states, northwest Mexico, and thirty tribal nations (Reclamation, 2012a). Releases from the two largest reservoirs, Lake Mead and Lake Powell, are determined to balance goals for storage (e.g., hydropower, conservation) and delivery (e.g., meeting demands for agriculture and municipalities). Storage and delivery goals are often conflicting, especially during low reservoir conditions. An extended drought since 2000 has threatened the CRB (DOI, 2022). Federal-level policies sought to protect storage but have not prevented historically low storage in 2022. Specifically, policies are delivery reductions for downstream users, as a function of the level of storage in Lake Mead, and how storage in Lakes Powell and Mead are balanced. Current policies expire in 2026, and at time of writing, there is a formal process of negotiating new policies (Reclamation, 2023e). This process must cope with copious policy options, conflicting priorities among stakeholders, and deep uncertainty with respect to future exogenous conditions

(hydrology and demands). The consistent example in this review discusses how vulnerability analysis can survey multiple purposes in the context of CRB reservoir operations.

5.2 A taxonomy of methods

Figure 5-1 provides an overview of the distinct steps of a vulnerability analysis: simulation modelling, defining decision-relevant outcomes, and factor mapping. Each step is discussed in the subsections below.

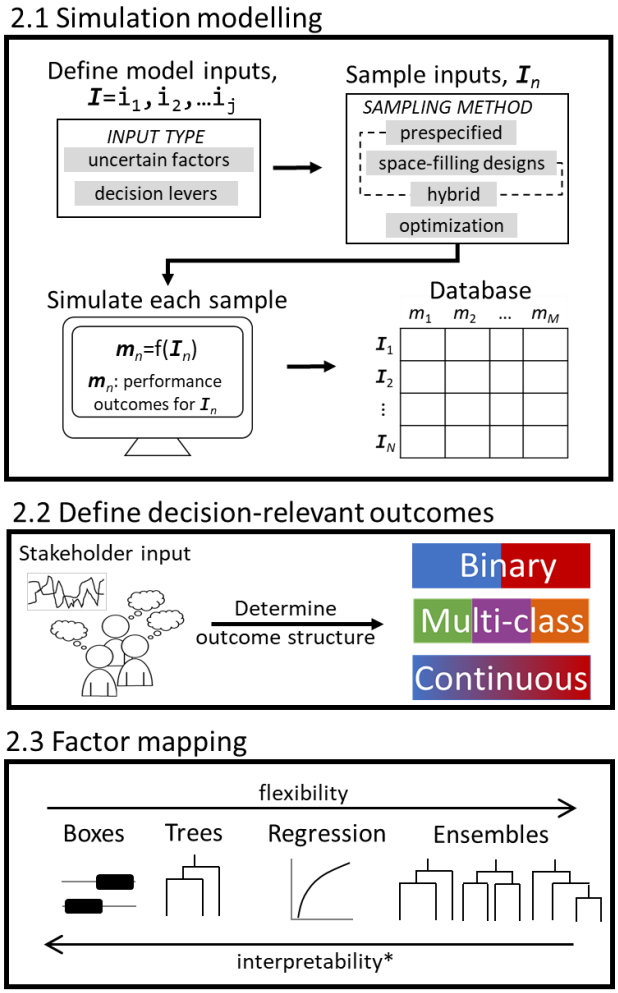


Figure 5-1: The steps of a vulnerability analysis

5.2.1 Simulation modelling

The first step is simulation modelling, which tests the performance outcomes of policies in alternative SOW (Figure 5-1, 2.1). The analyst defines the model inputs (`i` in Figure 5-1). These include

uncertain factors and decision levers, values for which define SOW and policies (Lempert *et al.*, 2006; Kasprzyk *et al.*, 2013). In systems characterized by deep uncertainty, there can be many plausible values for the decision levers and uncertain factors, and also many possible combinations of values. Their possible values are sampled using thousands of input sets (***I*** in Figure 5-1), which are then tested in the model. The model evaluates performance metrics (***m*** in Figure 5-1). The performance metrics often summarize a time-varying state variable, such as annual Lake Powell storage, using a summary statistic, such as the fraction of time the water level is below hydropower intakes (Alexander, 2018; Bonham, J. Kasprzyk and Zagona, 2022a). Multiple metrics can be used to capture different methods for aggregating a time-series (e.g., average of Lake Powell storage) and measure performance for different state variables (i.e., annual deliveries) (Bonham *et al.*, 2023). The outcome of this step is a database of model inputs (SOW and policies) and performance outcomes, which is used in the factor mapping step. The following subsections describe methods for creating policies and SOW.

5.2.1.1 Policies

Policies can be predetermined or generated with optimization, as described by (Herman *et al.*, 2015). Policies may be predetermined due to the existence of only a small number (e.g., 4) of plausible options or because the polices were selected in previous studies. As an example, the current reservoir operation policy in the CRB was chosen from three alternatives analyzed in a 2007 Environmental Impact Assessment (Reclamation, 2007). Under low water supply conditions, one alternative uses large shortages to prioritize reservoir storage, the second used small shortages to prioritize deliveries to the Lower Basin states, and the third (the selected alternative) is a compromise between the former two. Each policy could be modelled by codifying the shortage rules as decision lever values. However, a small number of predetermined policies can leave many plausible values for the decision levers unexplored and fail to identify critical performance tradeoffs (Kasprzyk *et al.*, 2013; Herman *et al.*, 2015).

These challenges can be addressed by creating policies with multi-objective optimization (Hadka and Reed, 2013; Maier *et al.*, 2019). This process couples an optimization algorithm, usually a multi-objective evolutionary algorithm (MOEA), with a simulation model to automatically search for high-performing policy alternatives. For example, Alexander (2018) coupled the Borg MOEA (Hadka and Reed, 2013) with the Colorado River Simulation System to create several hundred shortage policies for Lake Mead. The policies are non-dominated with respect to multiple storage and delivery objectives, meaning no policies were found with equal or better performance in all objectives while being better in at least one. The policies exhibit tradeoffs, meaning improvement in one objective (i.e., storage) comes at the expense of others (i.e., deliveries). All the policies can be tested in a vulnerability analysis, or a small handful can be selected via robustness analysis (Kasprzyk *et al.*, 2013; Bonham, Joseph Kasprzyk, and Edith Zagona, 2023).

5.2.1.2 States of the World

SOW can be predetermined, sampled using a design of experiments, or a hybrid approach. In the predetermined case, planning agencies develop projections of uncertain factors based on climate models (River Management Joint Operating Committee, 2020), paleo-records (Reclamation, 2012a), and population scenarios (Upper Colorado River Commission, 2016). For example, previous studies in the CRB have used demand projections - based on future population scenarios – and streamflow projections – based on historical data and climate change scenarios – in a vulnerability analysis (Reclamation, 2012a; Groves *et al.*, 2013). Because these projections are time-series (e.g., annual streamflow projections for a 40-year planning study), a requirement of this approach is to calculate time-series statistics for each projection, which later become the predictors in the factor mapping step. Said another way, the projections are predetermined, and values for the uncertain factors are calculated using time-series statistics of those projections. For example, Groves et al. (2013) calculated the average annual and driest

eight-year average streamflow for each streamflow projection, and these uncertain factors were later used as inputs to factor mapping.

Uncertain factors can also be sampled using a space-filling design of experiments. Analysts choose the upper and lower bounds for each factor, the number of SOW (i.e., the number of model runs), then an algorithm chooses a set of SOW that maximizes coverage of the uncertainty space (Damblin, Couplet and Iooss, 2013; Joseph, 2016). One common space-filling algorithm is Latin Hypercube Sampling optimized for a space-filling objective (Dupuy, Helbert and Franco, 2015; Carnell, 2022). Space-filling designs are well-suited for vulnerability analysis because they provide a uniform, continuous sampling across each uncertain factor, which enables the factor mapping step to more finely resolve the boundary between decision-relevant outcomes (Section 5.3.2). In our CRB example, this could mean sampling future demand and reservoir inflow continuously such that factor mapping could identify precise values that lead to unacceptable shortages. Since these methods generate new SOW, however, they may not be ideal for agencies looking to use predetermined projections as SOW.

Hybrid sampling methods use predetermined SOW to create space-filling designs. The analyst chooses the number of SOW, then a sampling algorithm selects a subset of SOW from existing data that maximizes coverage of the uncertainty space. Methods include *conditioned* Latin Hypercube Sampling (cLHS) (Minasny and McBratney, 2006), Kennard-Stone sampling (Kennard and Stone, 1969), and Feature Space Coverage Sampling (Wadoux, Brus and Heuvelink, 2019; Wadoux and Brus, 2021). For example, Bonham et al. (2022a) use cLHS to subsample 500 SOW from an existing ensemble of nearly two-million SOW. Subsampling allowed the authors to use existing streamflow projections, continuously sample streamflow and demand values, and do so with a small number of SOW compared to the entire dataset. The hybrid approach also has limitations, however. The subsampled SOW is not exactly the same set of projections the agency is familiar with (it is a subset of those projections), and the range and density of

uncertain factor values sampled by the algorithm is limited to that of the projections being subsampled from.

5.2.2 Define decision-relevant outcomes

After simulation modelling, the analyst works with stakeholders to define decision-relevant outcomes. A common example is when a policy fails to meet a stakeholder-defined performance threshold, such as a reservoir policy failing to store enough water to produce hydropower (Bryant and Lempert, 2010; Alexander, 2018).

Such a split between acceptable and unacceptable performance outcomes is a binary performance outcome *structure*. However, recent studies have demonstrated more flexible performance outcome structures, including multi-class and continuous structures (Quinn *et al.*, 2020; Steinmann, Auping and Kwakkel, 2020). These three performance outcome structures are reviewed below.

5.2.2.1 Binary

When the performance outcome structure is binary, performance is either deemed acceptable (blue) or unacceptable (red, Figure 5-2). Acceptable performance is determined from stakeholder-defined performance thresholds – on a single metric (5-2a) or multiple metrics (5-2b). Although not pictured, other logical conditions can be used (e.g., using OR versus AND).

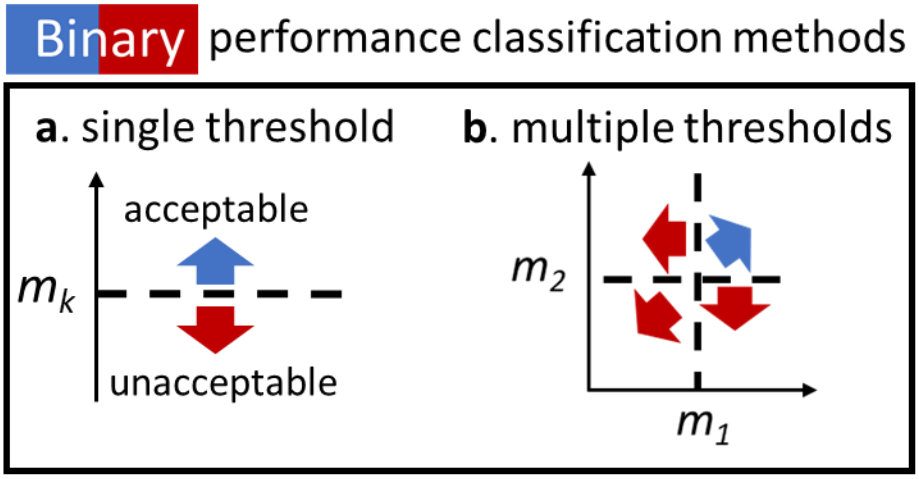


Figure 5-2: Methods for defining binary performance classes

The binary performance outcome structure has the benefit of being easy to understand because it only partitions into two categories. However, the binary transformation loses information about the degree to which the threshold is satisfied or violated, which may be important information for decision-makers. For example, decision-makers in the CRB may want to know not only if hydropower levels are maintained, but the margin of safety a given policy provides. In other cases, there is not a clear threshold between acceptable and unacceptable, but rather multiple 'levels' such as high, moderate, or low performance (Kravits *et al.*, 2021). These cases can be addressed with continuous and multi-class structures.

5.2.2.2 Multi-class

There are three approaches for multi-class performance outcome structures (i.e., more than two classes). The first approach is to define each class manually (Figure 5-3a). As an example, decision-makers in the CRB may be interested high (green), moderate (purple), and low (orange) shortages. Like binary classification, this process can be extended to an arbitrary number of performance metrics and thresholds (Rozenberg *et al.*, 2014; Guivarch, Rozenberg and Schweizer, 2016). However, this process becomes difficult for multiple performance metrics and thresholds.

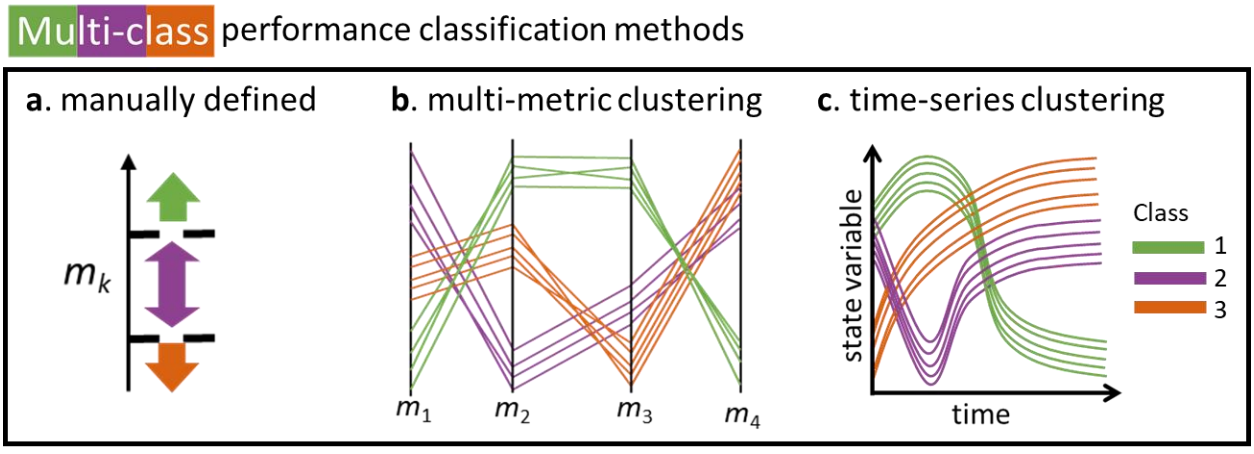


Figure 5-3: Methods for defining multi-class (more than two) performance classes

Multi-metric clustering defines performance classes by finding groups of model outputs with similar performance across multiple metrics**.** The example in Figure 5-3b shows four performance metrics

on a parallel coordinates plot (Inselberg, 2009). Multiple metrics can measure benefits to different stakeholders to parse how policies and uncertain factors can lead to performance inequalities (Jafino and Kwakkel, 2021). Or, the metrics could be different statistics of time-varying state variables (Alexander, 2018). Since performance metrics aggregate a time-series, the use of multiple metrics could mitigate the loss of relevant information. In the CRB, for example, the metrics could measure deliveries to Arizona, California, Nevada, and Mexico. Or, they could describe the maximum, minimum, mean, and frequency of shortages (Alexander, 2018; Bonham *et al.*, 2023). Clustering identifies performance classes within which performance is similar, doing so without prespecified performance thresholds (James *et al.*, 2013, chap. 10.3). For example, class one (green) is characterized by small values for m_1 and m_4 and large values for m_2 and m_3. In contrast, class two (purple) has large values for m_1 and m_4, small values for m_2, and medium values for m_3.

Time-series clustering defines performance classes as similar patterns of performance over time. In Figure 5-3c, each trace corresponds to one model run, showing a state variable changing in time. In the CRB, this could be Lake Mead storage. Decision-makers determine reservoir releases on the basis of storage levels and time of year (e.g., winter vs spring flood space requirements). So, it may be beneficial to define performance classes based on both magnitude and timing, as done with time-series clustering (Steinmann, Auping and Kwakkel, 2020). Class 1 (green) has high values at the beginning of the simulation and low values at the end, while class 3 (orange) has low values at the beginning of the simulation and high values at the end. Previously described performance classification methods use performance metrics, i.e., each trace in Figure 5-3c would be aggregated into a single value using a statistic (e.g., mean). Time-series clustering, however, requires information about state variables at multiple time steps. Time-series clustering can be extended to multiple state variables, such as Lake Mead and Lake Powell storage, using multivariate time-series clustering (Li and Liu, 2021).

5.2.2.3 Continuous

The continuous performance outcome structure uses a continuous function rather than transforming performance into two (binary) or more than two (multi-class) classes. This could mean using the values of the performance metrics directly (Quinn *et al.*, 2020), such as Lake Mead storage, or using a transform on the metrics prior to factor mapping (i.e., some points are deemed acceptable, and unacceptable points have continuous values of performance violation).

This method could be beneficial for cases where decision-makers disagree on performance thresholds or are dissatisfied with variance within binary or multi-class classes. Returning to Figure 5-3b, for example, the range of possible values for m_1 within class one may be too dissimilar for decision-makers, who might instead prefer a prediction of the expected performance value. A potential limitation, however, is that continuous values may be difficult to interpret for decision-making. For example, what storage levels at Lakes Mead and Powell would indicate the need for adaptation versus continuing with the current policy?

5.2.3 Factor mapping

Factor mapping discovers a subset of model inputs, and the values of those inputs, that lead to decision-relevant outcomes – i.e., a mapping between inputs and binary, multi-class, or continuous outputs. This mapping is often called 'scenario discovery', where the inputs and their values define a scenario (Bryant and Lempert, 2010; Steinmann, Auping and Kwakkel, 2020; Jafino and Kwakkel, 2021). A simple example is binary classification with box-shaped factor mapping (Section 5.2.4.2). The factor mapping draws a box around values of the model inputs that result in unacceptable performance, and these conditions are communicated to decision-makers for comparing policies (Groves *et al.*, 2013) or making changes to an existing policy (Dixon, Lempert, LaTourrette and Reville, 2007). Specific use cases of factor mapping are dependent on the purpose, however (Section 5.2.4). Therefore, this section focuses on the mechanics of factor mapping – the "shape" that the factor map uses to best describe the groups of inputs.

Due to the large number of factor mapping methods, concepts of flexibility and interpretability are helpful for understanding how they differ. This differentiation of the methods, and the outputs they create, is important to understand because it influences how the results might be used for policymaking.

5.2.3.1 Flexibility and interpretability

Flexibility describes how a factor mapping method can bend and flex to describe a dataset and make predictions (James *et al.*, 2013; Rudin *et al.*, 2022, chap. 2.1). For vulnerability analysis, flexibility determines the 'scenario shape'. Different factor mapping methods create rectangular, triangular, and arbitrarily complex scenario shapes (Figure 5-1). We elaborate on these differences below. Ideally, more flexible methods are used only when the relationship between model inputs and performance class are so complex that less flexible methods do not accurately describe it (James *et al.*, 2013; Rudin, 2019; Rudin and Radin, 2019; Rudin *et al.*, 2022). We will return to this point on accuracy and method selection in Section 5.4.2.

Interpretability is the extent to which the factor mapping results are easily understood and applied by the intended users (Bryant and Lempert, 2010; Kwakkel, 2019; Rudin *et al.*, 2022). In vulnerability analysis, interpretability is a measure of how well the scenarios are applied by analysts, stakeholders, and decision-makers for the purposes described in Section 5.2.4. Since vulnerability analysis is used to inform policy decisions pertaining to public resources, interpretability is of utmost importance (Rudin *et al.*, 2022). As a rule of thumb, increased flexibility comes at the expense of interpretability (James *et al.*, 2013, chap. 2.1), as illustrated in Figure 5-1 part 2.3. At the same time, interpretability is subjective because it depends on the decision-making context (Rudin *et al.*, 2022). For vulnerability analysis, interpretability has been quantified as the number of model inputs describing the scenario and the number of scenarios presented to decision-makers (Lempert, Bryant and Bankes, 2008; Bryant and Lempert, 2010). It is generally accepted that a scenario defined by one or two model inputs, as opposed

to many, is more interpretable, and so are fewer scenarios compared to more (Kwakkel, 2019; Lee and Shin, 2020). However, less attention has been given to how flexibility impacts interpretability.

The remainder of this section will use flexibility and interpretability to review factor mapping algorithms for binary (Section 5.2.3.2), multi-class (5.2.3.3), and continuous (5.2.3.4) performance outcome structures.

5.2.3.2 Binary

Figure 5-4 arranges binary factor mapping algorithms by flexibility. To the left (less flexible) are PRIM and classification trees, while regression and ensemble methods are located to the right (more flexible). The algorithms are also in approximate order of decreasing interpretability, noting that decision-makers can have different opinions given the subjective nature of interpretability. Two visualizations are shown for each algorithm. The *fitting* visualizations illustrate how the algorithms learn the relationship between inputs and performance classes. In these diagrams, the axes correspond to model inputs ($\mathtt{i}_1$ and $\mathtt{i}_2$), showing only two inputs for simplicity. Depending on the decision-making purpose (Section 5.2.4), the inputs can be uncertain factors or decision levers. Each point corresponds to one model run, colored red (unacceptable) or blue (acceptable) as determined in the performance classification step. The second row of figures, *output*, illustrate common ways of communicating the results, elaborated below.

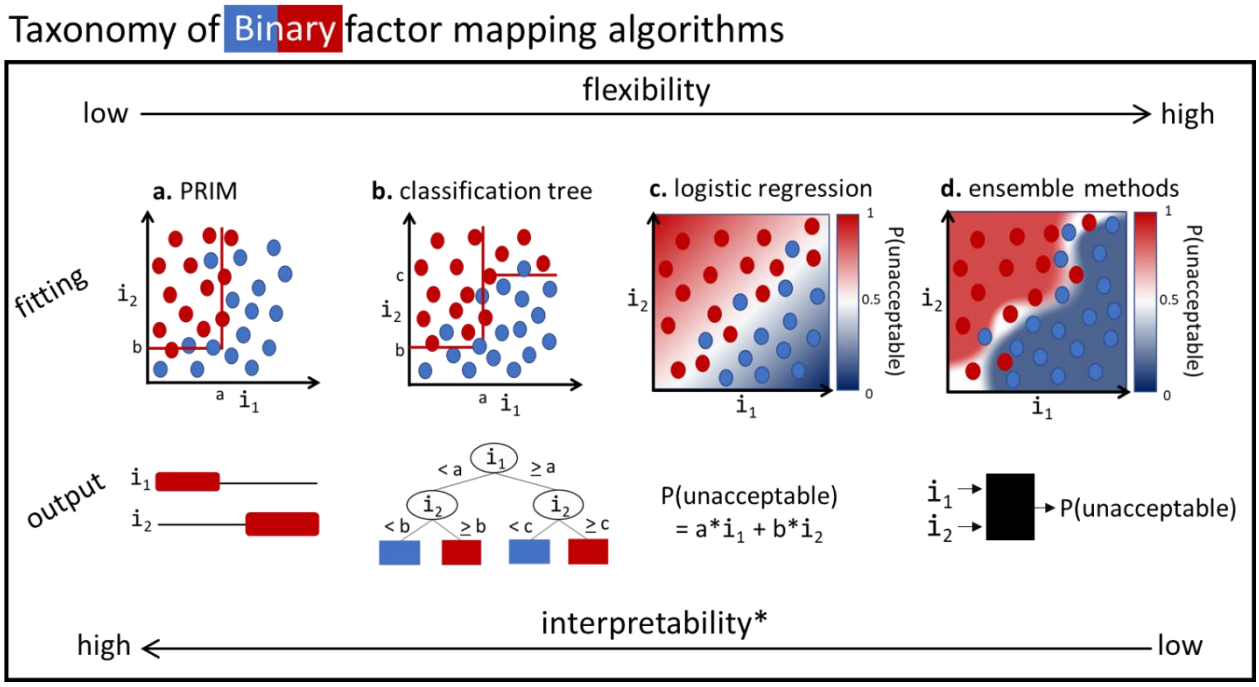


Figure 5-4: Common factor mapping algorithms for binary performance structures, organized by interpretability/flexibility

a. PRIM) The original binary method is the Patient Rule Induction Method (PRIM, Figure 5-4a) (Lempert, Bryant and Bankes, 2008; Bryant and Lempert, 2010). PRIM places constant constraints, i.e. logical rules (Rudin *et al.*, 2022), on the model inputs to find values likely to yield unacceptable performance (Friedman and Fisher, 1999). These logical rules make a box-shaped scenario (Bryant and Lempert, 2010; Kwakkel and Cunningham, 2016). PRIM is often considered interpretable because the logical rules can be communicated to decision-makers for policy creation and comparison (Dixon, Lempert, LaTourrette and Reville, 2007; Dixon, Lempert, LaTourrette, Reville, *et al.*, 2007). In the CRB, for example, a reservoir operation policy could be vulnerable if streamflow is less than 90% of the historical average and demand exceeds 110% average (Figure 5-4a). The results of PRIM are commonly shown like in Figure 5-4a, *output*. Each input has a line, and inputs constrained by PRIM are highlighted (with red, in this example). Not all inputs are necessarily constrained, depending on if that input was statistically significant in separating

acceptable from unacceptable performance outcomes. This visualization is helpful for more than three model inputs, in which case scatter plots like those in the *fitting* row are limited.

Although the high interpretability of PRIM can be useful (Dixon, Lempert, LaTourrette and Reville, 2007; Dixon, Lempert, LaTourrette, Reville, *et al.*, 2007; Kasprzyk *et al.*, 2013; Herman *et al.*, 2014; Reis and Shortridge, 2020), the algorithm has several limitations. The logical rules can be too inflexible when performance class depends on interactions between model inputs (Trindade, Reed and Characklis, 2019; Hadjimichael, Quinn, *et al.*, 2020). Further, PRIM provides a homogenous prediction for all inputs that meet the conditions of the logical rules (unacceptable or acceptable), even though the performance class for inputs located near the boundary are less certain than inputs located well within the boundaries. PRIM can also constrain more inputs than necessary, reducing interpretability (Kwakkel, 2019). Generally, these limitations have been addressed by either modifying PRIM (Table 5-1) or using an alternative factor mapping method. The subsections below describe alternative factor mapping methods while referring to Table 5-1 for related PRIM modifications.

Modification	Limitation of PRIM addressed	Changes to PRIM	Comments	Key References
PCA-PRIM	PRIM can only find rectangular regions of input space leading to uncertainties, which fails to capture how vulnerable regions better described by a triangular shape. This often occurs when there is a linear relationship between input factors.	Preprocess the input data using Principal Component Analysis (PCA). PCA creates new variables from linear combinations of the input factors and then rotates the data so it is orthogonal to these variables. PRIM is then applied, using the new variables as the inputs.	After applying PRIM, the new variables can be transformed back into the original inputs, which enables PRIM to produce triangular scenarios. This process is supported by the EMA Workbench (Kwakkel 2017).	Dalal et al. 2013
Bagging Random Boxes	PRIM is prone to overfitting, meaning scenario boxes may contain insignificant uncertain factors and constraints and may have high misclassification rates when used to classify new data.	PRIM boxes are determined using an ensemble of 'weak' boxes. The ensemble is trained using randomized subsets of the uncertain factors and model simulations. The results are aggregated by ensemble voting or averaging, similar to Random Forests.	Random boxes PRIM outperforms normal PRIM when used to predict class of new data, especially when trained on small data sets.	Kwakkel and Cunningham 2016
Objective functions for categorical inputs	At every peeling iteration, PRIM seeks to maximize density in the resulting box. So, PRIM preferentially places constraints on categorical and discrete data since removing an entire level of data can have greater influence on density than removing a slice of continuous data.	Propose two modifications to the original objective function. 1) divide density by number of data points removed in current iteration, which penalizes PRIM if it data is removed to quickly 2) multiply density by number of data points remaining and divide by number of points being removed, which rewards PRIM for preserving data points and removing data slowly.	Both objective functions tend to result in scenario boxes with greater density and coverage than original PRIM. Further, the peeling tajectory has more boxes to choose from. These objective functions are supported in the EMA Workbench in Python (Kwakkel 2017).	Kwakkel and Jaxa-Rozen 2016
Objective function for multi-class outputs	PRIM designed for binary classification. To work with multiple classes of outcome (e.g., multiple definitions of vulnerability), PRIM must be applied iteratively for each case. The resulting scenario boxes may overlap.	Create a PRIM objective function inspired by Gini Impurity. The objective function is to maximize the reduction in impurity per data point removed during a peeling iteration. Impurity is defined as fraction of data points belonging to different cases.	Kwakkel and Jaxa-Rozen 2016 conclude that PRIM with Gini-Impurity objective function, iterative applications of PRIM, and classification trees all capable of mutliclass vulnerability analysis.	Kwakkel and Jaxa-Rozen 2016
Modified covering process	After selecting a scenario box, PRIM can be used again to discover other vulnerable regions in the input space. This process is called covering. The original covering process would remove data points contained in the selected box. But, PRIM tends to create new boxes that overlap with the original box.	Instead of removing data points in selected scenario boxes, convert them to non-interesting classification. Doing so penalizes PRIM if it makes scenario boxes that overlap the previous box, which encourages PRIM to discover new and diverse scenario boxes.	This process is supported by the EMA Workbench (Kwakkel 2017).	Guivarch et al. 2016
Multi-objective PRIM	PRIM has a single objective function - maximize density. So, the full tradeoff space between density, coverage, and interpretability are not explored. Scenario boxes with superior interpretability may exist with only small tradeoffs to density and coverage.	Run PRIM once with all input factors to determine statistical significance. Remove non-significant factors. Then, perform iterations of PRIM to test all possible number and combinations of uncertain factors. Remove boxes that are dominated with respect to coverage, density, and number of factors.	Kwakkel 2019 Tested Multi-objective PRIM vs. a Multi-Objective Evolutionary Algorithm (MOEA). Found that multi-objective PRIM performs similarly to the MOEA. This modification is supported by the EMA Workbench (Kwakkel 2017).	Kwakkel 2019

Table *5-1:* Summary of modifications to the Patient Rule Induction Method (PRIM)

b. classification trees) Like PRIM, Classification Trees use logical rules to predict performance class, but with some additional flexibility. Classification trees iteratively place constraints on the model inputs, each new constraint building upon previous constraints. Using a CRB example, a reservoir policy could be vulnerable if streamflow is less than 90% and demand exceeds 110%, *or* if demand exceeds 130% (Figure 5-4b). Note that similar results can be obtained via multiple iterations of PRIM ('modified covering process' in Table 5-1, see also (Lempert, Bryant and Bankes, 2008)). The results are often communicated with a decision tree diagram (Almeida *et al.*, 2017; Smith, Kasprzyk and Rajagopalan, 2019; Cohen and Herman, 2021). Like the constraint diagram for PRIM, the tree diagram is easily extended to three or more model inputs.

Although classification trees are more flexible than PRIM, the use of logical rules mean they have similar limitations. Although they can describe and/or type interactions, classification trees can still struggle to accurately capture linear and non-linear interactions (Almeida *et al.*, 2017). Like PRIM, all model runs that fall within a certain set of logical rules are treated as equally likely to have the predicted performance class.

c. logistic regression) Logistic regression captures interactions between model inputs and predicts the probability of unacceptable performance (Figure 5-4c) (Hadjimichael, Quinn, *et al.*, 2020). The probability is a continuous function of both i_1 and i_2. In Figure 5-4c, if both i_1 and i_2 increase at the same rate, the probability of acceptable performance stays at 50%. But, i_2 greater than i_1 increases the probability of unacceptable performance. In the CRB, this could be the relationship for demands that exceed reservoir inflows. The probability information could inform the magnitude of delivery reductions, i.e. - larger delivery reductions may be desired when the probability is 95% compared to 55%. Note that this is the probability of unacceptable performance assuming values for i_1 and i_2 – it is not the probability that the model input values will be observed. Logistic regression can be expanded to non-linear relationships by using higher-order regression equations (e.g., $i_1^2+i_2$) or by adding interaction terms (i.e., $i_1+i_2+i_1*i_2$)

(Hadjimichael, Quinn, *et al.*, 2020). Note that PCA-PRIM (Table 5-1) also describes interactions between model inputs, but does not provide probability information.

The additionally flexibility and probability information of logistic regression can diminish interpretability. The continuous function created by logistic regression may be less interpretable than the logical rules of PRIM or classification trees, i.e., the scenario is not described with if-then statements. This limitation is especially relevant when more than two inputs are included in the regression equation since it is difficult to expand the factor mapping visualizations in Figure 5-4c, *fitting*, to three or more dimensions. Depending on the purpose, e.g., decision-makers comparing policies, the probability information could also be misinterpreted.

d. ensemble methods) Ensemble methods can describe complex interactions between inputs and performance. In Figure 5-4d, the shapes of the unacceptable and acceptable regions are non-linear and non-monotonic, meaning the probability can both increase and decrease as the inputs increase. Ensemble methods use many 'weak' prediction models that work together to predict the performance class. Classification trees are common for this method (Trindade, Reed and Characklis, 2019), but PRIM has also been used ('random boxes' in Table 5-1) (Kwakkel and Cunningham, 2016). Each model in the ensemble is 'weak' because it is uses a random subset of model outputs and input variables (Breiman, 2001; Kwakkel and Cunningham, 2016), or because the complexity of each model is constrained (James *et al.*, 2013, chap. 8.3). The predictions of each weak model are aggregated to provide the ensemble's final prediction. The fraction of weak models that agree on the final prediction is reported as the probability, similar to logistic regression (Trindade, Reed and Characklis, 2019).

Ensemble methods have the same potential limitations as logistic regression with additional interpretability concerns. Because the prediction is determined from a large ensemble of models, the relationship between inputs and performance outcomes cannot be described with a single decision tree, nor is the relationship described with an equation, like in logistic regression. Therefore ensemble methods

are often called ‘black box’ algorithms (Rudin, 2019; Rudin *et al.*, 2022). Like logistic regression, results are communicated using the visualization in Figure 5-4c (Trindade, Reed and Characklis, 2019), but the challenge of visualizing more than two model inputs remains.

5.2.3.3 Multiclass

a. iterative binary algorithm Like binary classes, there are multiple factor mapping methods for multi-class classification. One approach is to iteratively train a binary model for one performance class at a time (Steinmann, Auping and Kwakkel, 2020). This process relabels the performance classes such that there are only two classes in each iteration. As an example, assume classes one through three (green, purple, and orange) as shown in Figure 5-5. To discover conditions that lead to class one, the analyst could relabel all simulations not belonging to class one as *not class one*. This is now a binary problem (*class one* and *not class one*), and binary algorithms such as PRIM, CART, and logistic regression can be applied (Kwakkel and Jaxa-Rozen, 2016; Steinmann, Auping and Kwakkel, 2020).

The iterative binary approach can lead to overlapping scenarios. In other words, the same values for the model inputs can lead to multiple class outcomes, as demonstrated with PRIM boxes in Figure 5-5a. For negotiation and compromise, the regions of overlap could indicate decision lever values that meet the goals of two or more stakeholders (Lempert and Turner, 2020; Bonham, J. Kasprzyk and Zagona, 2022a). However, the overlapping conditions can be less interpretable to decision-makers for the purposes of policy creation and comparison (Section 5.2.4). This is because the same conditions predict more than one performance outcome (Kwakkel and Jaxa-Rozen, 2016; Jafino and Kwakkel, 2021).

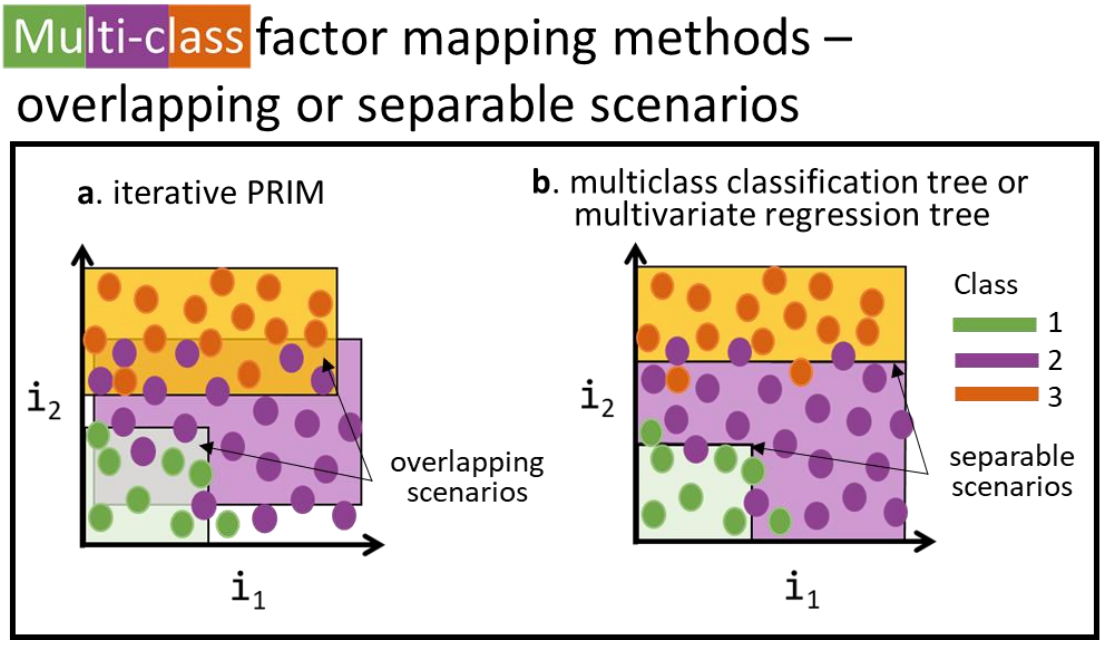


Figure 5-5: Overlapping compared to separable scenarios in multi-class factor mapping

b. multiclass/multivariate algorithms Overlapping scenarios can be avoided with multi-class-specific algorithms, which discover the conditions that lead to each class in a single iteration. The resulting scenarios are non-overlapping, as illustrated with Figure 5-5b (Jafino and Kwakkel, 2021). Many binary methods have multiclass extensions such as multiclass PRIM and classification trees (Kwakkel and Jaxa-Rozen, 2016). A limitation of both the iterative binary and multiclass methods is the requirement for two distinct steps – first defining performance classes via clustering, then factor mapping (Jafino and Kwakkel, 2021).

An alternative approach is to use multivariate *regression* trees (MRT), which discover both the performance classes and the conditions concurrently (Smith, Kasprzyk and Rajagopalan, 2019; Jafino and Kwakkel, 2021). The tree predicts one or more continuous performance metrics (***m***), not prespecified performance classes. The MRT then applies logical rules to the inputs, like a binary classification tree, to find groups of model outputs with similar performance. As explained by Jafino and Kwakkel (2021) this process is similar to the clustering step. i.e., the discovered groups are the performance classes. Effectively, the MRT accomplishes factor mapping and performance clustering concurrently, removing the

need for the pre-clustering step. Like multiclass classification trees, the conditions leading to each class will be mutually exclusive – the same pros and cons described in the previous two paragraphs apply.

5.2.3.4 Continuous

There are numerous methods for continuous factor mapping with analogs to binary and multiclass methods. Figure 5-6 is an example of linear regression (Quinn *et al.*, 2020). The background color shows the predicted performance metric value ($\widehat{m_k}$), and the colored points are performance metric values from the model simulations. More flexible methods, such as non-linear regression and ensemble methods, can also be used. Continuous factor mapping can be extended to multiple performance metrics by training one model per metric (similar to the iterative binary method) or using multivariate methods like multivariate regression trees (Smith, Kasprzyk and Rajagopalan, 2019; Jafino and Kwakkel, 2021). Like binary and multiclass methods, increased flexibility can reduce interpretability.

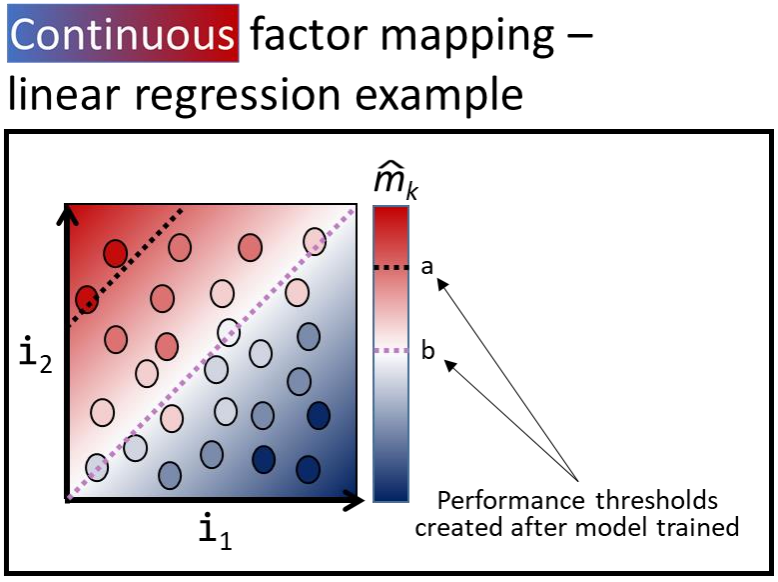


Figure 5-6: Continous factor mapping with linear regression

Continuous factor mapping allows performance thresholds to be defined after the factor mapping is complete. As an example, the performance metric in Figure 5-6 could be average Lake Mead storage. The two performance thresholds, *a* and *b*, could represent the performance goals for irrigators upstream versus downstream of the reservoir (Hadjimichael, Quinn, *et al.*, 2020). Alternatively, the thresholds could

represent how a stakeholder's performance goals could change after seeing what performance outcomes are possible (Kasprzyk *et al.*, 2013; Bonham, Joseph Kasprzyk, and Edith Zagona, 2023). For instance, a stakeholder could update their definition of unacceptable performance from the more lenient threshold of *a* to the more challenging threshold of *b* after seeing that only a few model runs result in performance worse than *a*. In either case, the factor mapping is not affected because it is performed with the continuous values of the performance metrics, independent of performance classes.

5.2.4 Purposes

This review identifies five purposes for vulnerability analysis: scoping, creating policies, comparing policies, negotiation and compromise, and monitoring and adaptation (Figure 5-7). These purposes are synthesized from three bodies of literature – Environmental Impact Assessments, Adaptive Environmental Management, and DMDU. Although specific requirements vary by country, common requirements for Environmental Impact Assessments include scoping and comparing policy alternatives (Yang, 2019, 2023). Monitoring and adaptation is advocated by the field of Adaptive Environmental Management (Holling, 2005), and has been identified by the International Panel on Climate Change as critical for sustainable management in the 21st century (IPCC, 2022). DMDU studies also use vulnerability analysis in the *creation of policies* (Watson and Kasprzyk, 2017) and *negotiation* between decision-makers (Lempert and Turner, 2020; Bonham, J. Kasprzyk and Zagona, 2022a). The following subsections expand on these purposes.

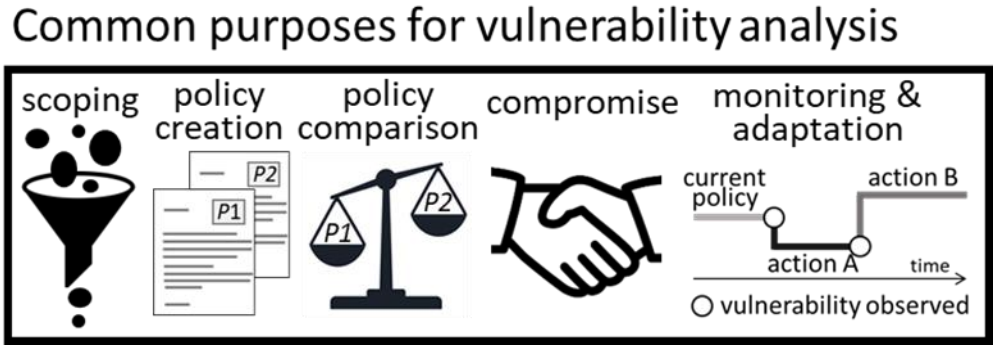


Figure 5-7: common purposes for vulnerability analysis

5.2.4.1 Scoping

Scoping determines the breadth of topics to be considered in an environmental impact assessment (Snell and Cowell, 2006; Reclamation, 2012b). These topics include policy alternatives, uncertain factors, and their impacts on system performance. It is common for the performance of a system to be driven by a small number of decision levers and uncertain factors (Herman *et al.*, 2015; Joseph, 2016), and these impactful factors can be discovered with vulnerability analysis (Almeida *et al.*, 2017). Removing less impactful factors from the scope can reduce the number of model inputs and save computing resources during future simulation modelling. Moreover, narrowing the scope can improve communication between analysts and decision-makers (Reclamation, 2012b, p. 31).

As an example, decision-makers in the CRB must consider the potential impacts of alternative polices and climate change (Smith *et al.*, 2022). Stakeholders have proposed a large list of policy options, including new reservoir operation strategies, modification or removal of dams, market strategies, and interbasin transfers (Rosenberg, 2022; Reclamation, 2023e). These options could be represented as decision levers, then tested in a simulation model. Stakeholders have also expressed concern over the potential impacts of climate change, which could reduce supply via reduced reservoir inflows and increased evaporative losses. Likewise, these uncertain factors can be tested in a model. However, it could be intractable for all of these elements to be covered in an environmental review. Factor mapping could be applied to existing (Reclamation, 2012a; Groves *et al.*, 2013; Bonham, J. Kasprzyk and Zagona, 2022a) or new model runs to provide mathematical justification for what decision levers and uncertainties are within the scope of the review.

5.2.4.2 Creating policies

Vulnerability analysis can be used directly in the process of creating policies. This is accomplished in two ways: refining policies based on identified vulnerable conditions, or including the conditions in a simulation-optimization problem.

For refining policies, the conditions in which a policy is vulnerable are communicated to decision-makers, who then suggest modifications to the policy to improve performance in those conditions (Lempert *et al.*, 2006; Lempert, 2013). In the CRB, a policy could result in unacceptable water shortages if reservoir inflows fall below some percentage of the historical average (Reclamation, 2012a; Groves *et al.*, 2013). That percentage of the historical average could be used to modify the current policy, such as adapting releases based on the historical percentage trigger (Rosenberg, 2022; Reclamation, 2023e). The efficacy of the changes could then be tested with more simulation modelling.

The second approach is to use conditions causing poor performance as model inputs in simulation-optimization that automates the search for new policies (Hadka and Reed, 2013; Watson and Kasprzyk, 2017). In the CRB, this approach would enhance the prior example, where different scenarios of historical streamflow could provide tailored input to a simulation model, where multiple sets of policies can be created via optimization to ameliorate those vulnerable conditions.

5.2.4.3 Probabilistic policy comparison

For this purpose, the resulting vulnerable conditions are compared, where decision-makers can take two sets of vulnerable conditions (i.e., for two different policies) and make a judgement on which set of vulnerable conditions is more likely (Dixon, Lempert, LaTourrette and Reville, 2007; Dixon, Lempert, LaTourrette, Reville, *et al.*, 2007; Groves and Lempert, 2007; Shortridge and Zaitchik, 2018).

Consider the following example with two policies for the CRB, inspired by Groves et al. (2013). Policy one is the ‘status quo’ operating rules and infrastructure, while policy two adds desalination plants. Vulnerability analysis could discover the reservoir inflow values below which each policy will be unacceptable. The results could indicate, for example, that the status quo is vulnerable to reservoir inflow less than 95% the historical average, whereas status quo plus desalination is vulnerable when inflow falls below 90%. If decision-makers believe future inflows may drop below 95%, then they may prefer the

policy with desalination plants. However, the desalination policy could perform poorly with respect to other performance metrics, such as cost or environmental impacts, warranting further analysis.

5.2.4.4 Negotiation and compromise

Negotiation can also be facilitated with vulnerability analysis by discovering policies that meet the goals of multiple stakeholders (Smith, Kasprzyk and Rajagopalan, 2019; Lempert and Turner, 2020; Jafino and Kwakkel, 2021). To do so, decision levers are used as the inputs to the factor mapping algorithm, which identifies values for the decision levers (i.e., policies) that achieve a specified performance outcome. The factor mapping is repeated for each stakeholder's desired performance outcomes to identify their preferred policies. This process could identify policies that simultaneously meet their goals, or it could revela that no such policy exists. In the latter case, the goals could be revised via negotiation (Gold *et al.*, 2019b), and the factor mapping repeated to identify compromise policies.

Consider the following example with two hypothetical stakeholders in the CRB, inspired by Bonham et al. (2022a). Stakeholder one wants to maximize hydropower production while stakeholder two wants to minimize shortages to downstream users. Factor mapping could identify the reservoir release rules (policies) that favor each group. If no policy simultaneously meets both stakeholder's goals, each group would need to compromise on their goal, arriving at a balance between hydropower production and deliveries. Factor mapping could be reapplied to find the reservoir release rules that achieve the compromise.

5.2.4.5 Monitoring and adaptation

Monitoring and adaptation systems track system conditions to indicate when new policy is needed to avoid poor performance outcomes (Groves *et al.*, 2013; Zeff *et al.*, 2016). These systems use signpost variables, triggers, and policy interventions (Haasnoot *et al.*, 2013; Kwakkel and Haasnoot, 2019, chap. 4; Molina-Perez *et al.*, 2019). The signpost variables are monitored in real-time, watching for a

trigger value that mobilizes a policy intervention to avoid unacceptable performance. These policies can include temporary or long-term actions.

Signpost variables and triggers can be informed from vulnerability analysis. Consider the following example, inspired by Groves et al. (2013) and Reclamation (2012a). Vulnerability analysis could identify that unacceptable shortages are likely if average reservoir inflows (the signpost variable) fall below 80% the historical average. A more proactive flow threshold, say 90% the historical average, could be used as the trigger value. If the trigger value is observed, policy interventions such as conservation programs or interbasin transfers could be enacted. A more complex, time-varying analysis of inflows, reservoir levels, and other system conditions could help identify the signpost variables and thresholds that most accurately predict if unacceptable performance is likely, at various lead times (Robinson, Cohen and Herman, 2020).

5.3 Recommendations

5.3.1 Clearly establish the purpose and audience

It is critical for methodological decisions to be informed by the purpose and end-users of the scenarios. For example, consider two purposes: 1) analysts identifying the most impactful model inputs (scoping) and 2) decision-makers comparing policy alternatives. The former application may require less interpretability, so complex performance classifications and flexible factor mapping methods may be appropriate (Trindade, Reed and Characklis, 2019; Steinmann, Auping and Kwakkel, 2020). In the latter case, simple narrative scenarios are used to facilitate policy debate, so simple performance classifications (i.e., binary) and interpretable factor mapping (i.e., PRIM) may be preferred (Dixon, Lempert, LaTourrette and Reville, 2007; Dixon, Lempert, LaTourrette, Reville, *et al.*, 2007).

A defined purpose is often lacking in environmental modelling studies (Sojda *et al.*, 2012). When purpose is defined, it is commonly defined with respect to a methodological innovation, and not a policymaking purpose (Razavi *et al.*, 2021). It is also common for analysts to choose methods they are familiar with, rather than methods best suited for the purpose (Razavi *et al.*, 2021). Defining the purpose

prior to methodological decisions can help avoid these shortfalls (Falconi and Palmer, 2017; Lahtinen, Guillaume and Hämäläinen, 2017).

5.3.2 Consider a space-filling sample of the uncertain factors

There are potential shortfalls of using climate and population projections for use as SOW. First, such projections are often biased toward moderate conditions, which can have unintended consequences during the factor mapping stage because relatively fewer 'challenging' SOW have been tested (Quinn *et al.*, 2020; Reis and Shortridge, 2020). Further, predetermined projections for two or more uncertain factors requires a strategy for combining them into sets of model inputs (i.e., SOW). The most common strategy is to make every possible combination of each uncertain factor, a so-called full-factorial design (Alexander, 2018; Jafino and Kwakkel, 2021). However, this strategy can require a prohibitively large number of simulations, since the number of SOW increases exponentially according to n^u, where n is the number of projections per uncertain factor, and u is the number of uncertain factors (Choi *et al.*, 2021).

With a space-filling design, the analyst chooses how many simulations to do given their computing resources. Further, a full-factorial design can result in large, unsampled regions of the uncertainty space, as demonstrated with the solid line in Figure 5-8. In contrast, space-filling designs minimize such gaps, which can enable the factor mapping to more accurately describe non-linear relationships (Choi *et al.*, 2021). Code for Figure 5-8 is given in Appendix C1. The gaps in the full-factorial can be reduced by increasing the number of values per uncertain factor, but doing so causes an exponential increase in the number of SOW.

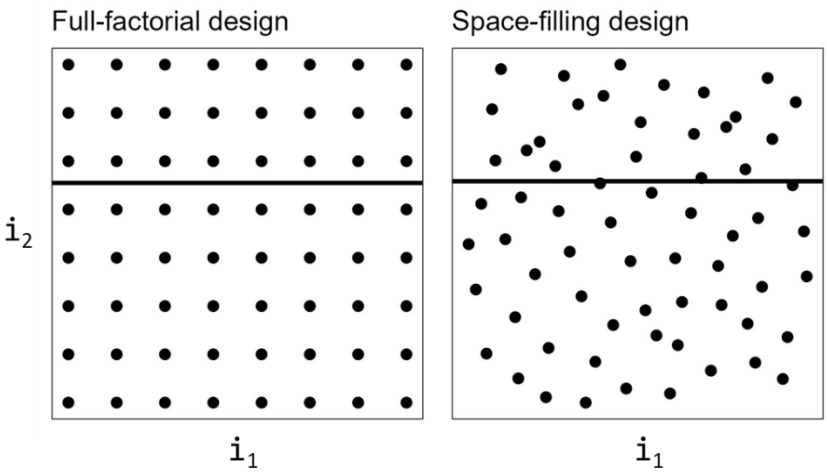


Figure 5-8: Gaps in the sampling space for full-factorial compared to space-filling designs. Both designs include 64 model input sets.

In cases where predetermined projections are to be used, consider using a hybrid method to subsample from a large full-factorial design (Bonham, J. Kasprzyk and Zagona, 2022a; Bonham *et al.*, 2023). Although large SOW ensembles can be simulated with high-performance computing (e.g., Microsoft Azure and Amazon Web Services), doing so may be cost prohibitive and contribute substantially to carbon dioxide emissions (Stevens *et al.*, 2020).

5.3.3 Work with stakeholders to define decision-relevant outcomes

Analysts and stakeholders should collaborate when defining decision-relevant outcomes because it involves numerous preference-informed decisions. These decisions include the performance outcome structure (binary, multi-class, continuous), the performance metrics and thresholds, clustering methods, and number of classes. It is up to the analyst to use cluster validation (Rendón *et al.*, 2011), expert judgment, and feedback from stakeholders in making these decisions. Frequent communication between analysts and stakeholders (Merritt *et al.*, 2017; Stanton and Roelich, 2021) can reveal preferences (Smith, Kasprzyk and Dilling, 2017; Bonham, Joseph Kasprzyk, and Edith Zagona, 2023) and help ensure 'decision-relevant outcomes' are relevant to decision-makers (Falconi and Palmer, 2017).

5.4 Discussion

5.4.1 Model inputs, performance classes, and probability also impact interpretability

Inherent in factor mapping are the model inputs and performance outcomes on which they are trained, so they too impact interpretability. For example, previous CRB studies have used parameters of a synthetic streamflow generator as predictors of water shortages (Hadjimichael, Quinn, *et al.*, 2020; Quinn *et al.*, 2020). Used by analysts, these uncertain factors may be sufficiently interpretable. For other purposes, such as decision-makers comparing policies, they may not be. Interpretability could be improved by transforming them into statistics familiar to decision-makers, such as moving averages of streamflow (Reclamation, 2012a; Groves *et al.*, 2013). Also consider whether the inclusion of more information, such as additional performance classes or probability information, are adding necessary information for decision-makers or if it distracts from the purpose (Miller, 1956; Kasprzyk *et al.*, 2018; Bell *et al.*, 2022).

5.4.2 More flexible factor mapping is not always more accurate

Testing accuracy can help avoid overfitting of scenarios (James *et al.*, 2013, chap. 2; Robinson, Cohen and Herman, 2020; Rudin *et al.*, 2022). Overfitting means the scenarios do poorly at predicting the performance outcomes for new model inputs because the scenario is overly sensitive to noise (Hastie, Tibshirani and Friedman, 2009, chap. 7; James *et al.*, 2013, chap. 2). Testing for overfitting involves splitting model inputs and outputs into training and testing datasets, training the factor mapping algorithms on the former (training accuracy), then evaluating their accuracy on the latter (testing accuracy). This process can be repeated several times to account for randomness in the train-test split, e.g., with k-fold cross-validation (Hastie, Tibshirani and Friedman, 2009, chap. 7).

Following this best practice can identify cases where a more interpretable scenario is as accurate as a more flexible scenario (James *et al.*, 2013, chap. 2; Makridakis, Spiliotis and Assimakopoulos, 2018; Rudin *et al.*, 2022). We provide an example in Figure 5-9, which is based on the numerical simulation

model and performance threshold described in Appendix C2. Part a shows the model inputs, colored by acceptable vs unacceptable performance class. They are split into 80% training (solid) and 20% testing (hollow) sets. In part b, two factor mapping algorithms, logistic regression (a linear model, left) and a random forest (an ensemble method, right), were fit to the training set. Recall that the predicted probability of unacceptable performance is shown as the background color. The training (top) and testing (bottom) accuracies are reported for each. We define accuracy as the fraction of SOW correctly classified by the factor mapping algorithm as acceptable or unacceptable, where the predicted classification is unacceptable if the probability is greater than 0.5 and acceptable, otherwise. More sophisticated accuracy metrics can be used and may yield different results. We use this simple accuracy metric to highlight the benefits of using a train-test split to compare factor mapping algorithms.

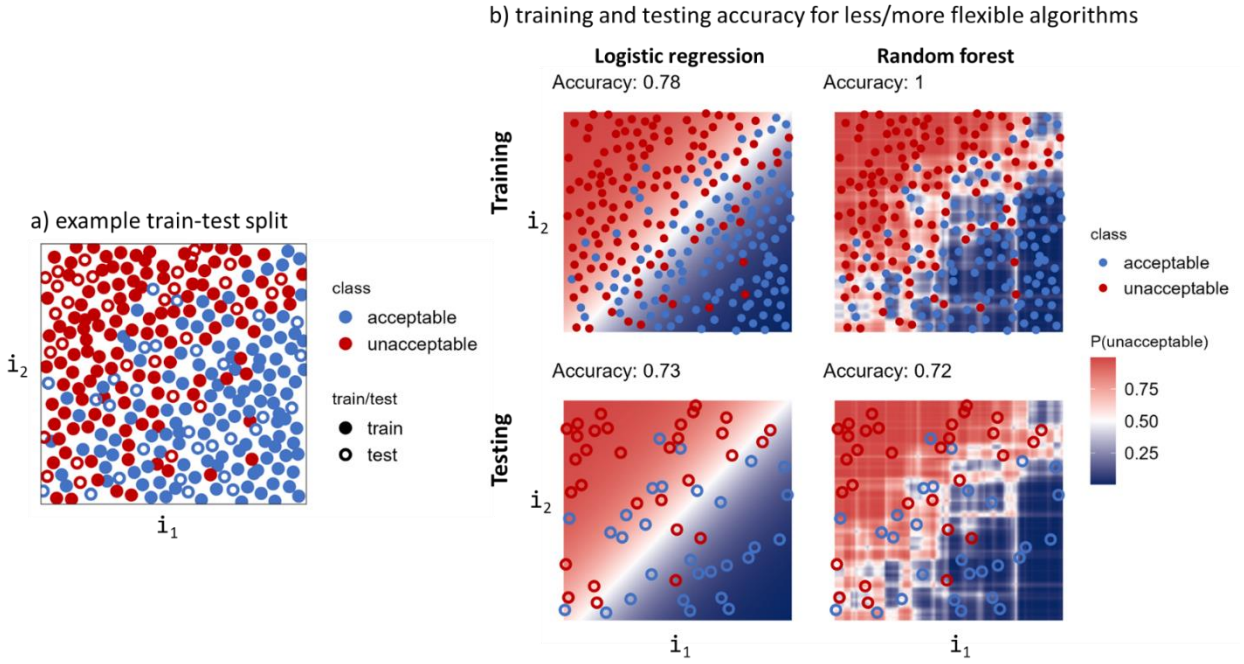


Figure 5-9: Testing accuracy for logistic regression vs random forest for a linear system.a) the dataset is split into 80% training and 20% testing points. b) training (top) and testing (bottom) accuracy are reported for logistic regression (left) and random forest (right)

On training data, both algorithms do well, but random forest outperforms logistic regression (accuracy of 1 versus 0.78). When compared with testing data, however, the methods have nearly identical testing accuracy – 0.73 and 0.72 for logistic regression and the random forest, respectively. The

figure demonstrates, though, that the logistic regression model is more interpretable than the random forest.

This example demonstrates that, for some case studies, more interpretable factor mapping algorithms can be as accurate as more flexible methods. By design, this example used a simulation model with a linear relationship between inputs and performance (Appendix C2). In practice, the relationship may be unknown. The analyst can explore the relationships using data visualizations like scatter matrices (James *et al.*, 2013, chap. 2) and by comparing multiple algorithms as done in this example. Several studies have demonstrated that more flexible algorithms are frequently (but incorrectly) assumed to be more accurate (Makridakis, Spiliotis and Assimakopoulos, 2018; Rudin and Radin, 2019). In vulnerability analysis, this assumption can unnecessarily reduce scenario interpretability.

5.4.3 Vulnerability analyses are often repeated

Repeatable vulnerability analyses can be beneficial for planning agencies. As emphasized in Section 5.3.3, defining decision-relevant outcomes and choosing interpretable methods requires iteration with stakeholders. Further, since a vulnerability analysis is predicated on a simulation model, deeply uncertain factors, and evolving policy alternatives (Lahtinen, Guillaume and Hämäläinen, 2017), it is possible a vulnerability analysis will be repeated. It also important that the analysis can be easily adapted for other systems (McIntosh *et al.*, 2011) since organizations often manage more than one system (e.g., World Bank, Bureau of Reclamation).

There are numerous best practices for creating repeatable analyses. These best practices include: **a)** code modularity – i.e., using separate code chunks to perform each step in the vulnerability analysis, plus prerequisite tasks like data wrangling (Wilson *et al.*, 2014; Pianosi, Sarrazin and Wagener, 2020; Peñuela, Hutton and Pianosi, 2021); **b)** documenting the programming environment (Peñuela, Hutton and Pianosi, 2021); **c)** creating example workflows with a high frequency of in-code comments (Pianosi, Sarrazin and Wagener, 2020; Alsudais, 2021; Hall *et al.*, 2022); **d)** minimizing dependencies on

programming libraries that could become obsolete (Pianosi, Sarrazin and Wagener, 2020). Obsolescence is especially relevant because there exist open source libraries for each step of vulnerability analysis (Bryant and Lempert, 2010; Pedregosa *et al.*, 2011; Hadka, 2015; Kwakkel, 2017; Hadjimichael, Gold, *et al.*, 2020), but these libraries can vary significantly in terms of their maintenance.

5.5 Conclusion

Simulation modelling provides decision support by testing system performance under alternative policies and plausible futures. However, because of the large number of policy options, uncertain factors, and complex system behavior, communicating the relationship between them can be challenging, inhibiting decision-making. Vulnerability analysis uses machine learning techniques to discover concise descriptions of policies and future conditions that cause performance outcomes relevant to decision makers – i.e., scenarios. These scenarios can be used in political debate and motivate the search for policies less vulnerable to challenging futures.

Recently, methods for vulnerability analysis have become increasingly complex to address performance outcomes for multiple interest groups, temporal performance dynamics, and non-linear relationships between policy decisions, uncertain factors, and performance. This means the resulting scenarios may also be less interpretable for decision-making, and that analysts have the difficult task of choosing methods that best address the decision-making purpose.

To provide guidance for analysts, this research establishes a taxonomy of methods, purposes, and recommendations for creating interpretable scenarios. We organize vulnerability analysis methods first by the performance outcome structure (binary, multi-class, and continuous), and we compare the methods by flexibility and interpretability. Vulnerability analysis purposes are identified from the broader environmental management literature, and include scoping, policy creation, policy comparison, negotiation and compromise, and monitoring and adaption. Purpose informs methodological decisions. Finally, we surveyed literature in machine learning, sensitivity analysis, and design of experiments to

identify recommendations. These include space-filling sampling of uncertain factors to improve computationally efficiency and the use of testing accuracy to create more interpretable scenarios. To illustrate purposes, methods, this review used a single environmental system – water supply management in the Colorado River Basin.

This review revealed several opportunities for future research. To reduce the number of computer simulations, future studies could investigate adaptive sampling methods, which use a sequential sampling procedure to discover regions of model input space that result in highly non-linear performance outputs, then strategically samples from those regions to improve factor mapping accuracy (Garud, Karimi and Kraft, 2017). Another challenge is how to account for time-varying model inputs in subsampling and factor mapping algorithms – perhaps the time-series clustering techniques being used for performance classification (Steinmann, Auping and Kwakkel, 2020) could also be applied to model inputs, and the resulting cluster information used as inputs to subsampling and factor mapping. A limitation of non-rule based factor mapping (i.e., logistic regression, ensemble methods) is the challenge of communicating results for more than two model inputs. There is a growing field called 'explainable artificial intelligence' that seeks to improve the interpretability of flexible machine learning methods (Saranya and Subhashini, 2023). Novel methods from this field could also improve the interpretability of flexible methods for vulnerability analysis.

6 Conclusion

6.1 Summary

This thesis contributes novel frameworks and interactive tools to empower participatory DMDU. In Chapter 1, this thesis explained how DMDU uses simulation models to create policy alternatives, explore tradeoffs, evaluate robustness, and discover drivers of vulnerability. The insights provided by DMDU can aid decision-making in the presence of uncertain climate and population changes and conflicting goals held by decision-makers. However, as stressors on environmental systems degrade the benefits received by stakeholders, there is increasing demand for participation in the analysis and choosing of polices that impact them.

Participation in DMDU-based decision support is particularly important because of many (consequential) decisions during the analysis. A DMDU analysis begins with initial input from stakeholders to help define the uncertain factors, decision levers, and performance objectives. These decisions impact the policies and SOW tested in the analysis. Then, input from stakeholders informs how robustness is quantified during robustness analysis. These decisions reflect stakeholders' initial prioritization of performance objectives and tolerance for uncertainty-related risk. The choice of robustness metric impacts which policies are prioritized by decision-makers. Then, during vulnerability analysis, analysts define policy-relevant performance outcomes and choose machine learning methods to identify scenarios. These choices can impact which conditions are discovered to be driving system performance and the interpretability of scenarios for decision-makers.

It is difficult to know, *a priori*, the policy recommendations that result from this sequence of interdependent methodological decisions. Further, it is possible such decisions can lead to undesirable policy recommendations, such as policies that prioritize one stakeholder at the expense of others or scenarios that are uninterpretable for decision-making. The participation of stakeholders and decision-makers throughout the analysis has the potential to identify and correct such problems.

However, there are several barriers to participatory DMDU. First, simulating large SOW ensembles can require significant computing resources, a challenge when faced with computing and time constraints or when feedback from stakeholders warrants additional modelling. Second, choosing robustness metrics is non-trivial because they can exhibit tradeoffs, and current guidance for selecting metrics can overlook such tradeoffs because it depends on the *a priori* preferences of stakeholders. Third, in the presence of tradeoffs, decision-makers may disagree on which policies are most robust, requiring negotiation and compromise to choose policies. However, robustness-informed negotiation requires that decision-makers can interpret complex relationships between multiple policies, many SOW, and robustness tradeoffs. Fourth, there is limited guidance on choosing methods for vulnerability analysis that produce interpretable scenarios for a given decision-making context.

This thesis addressed four barriers to participatory DMDU. Chapter 2 introduced a novel framework for creating SOW ensembles using subsampling and space-filling metrics to improve the computational efficiency of DMDU, especially for practitioners using climate and demand projections for uncertain factors. Chapter 3 introduced an *a posteriori* robustness framework to help stakeholders choose robustness metrics and policies after identifying critical tradeoffs. Chapter 4 used the Self-Organizing Map to synthesize the predominant tradeoffs when choosing between many policies, organize policies according to those tradeoffs, and facilitate negotiation between decision-makers. Chapter 5 reviewed purposes and methods for vulnerability analysis, establishing best practices to help analysts choose methods that are interpretable and relevant to decision-makers.

6.2 Dissemination of work

In addition to journal articles, this research has been disseminated through six oral presentations and one poster at academic conferences. These conferences include AGU Fall Meetings (Bonham, Kasprzyk and E. Zagona, 2020; Bonham, Kasprzyk and Zagona, 2021; Bonham, J. R. Kasprzyk and Zagona, 2022a, 2022b), the DMDU Society Annual Meeting (Bonham, J. Kasprzyk and Zagona, 2022b), the

international Environmental Modelling & Software meeting (Bonham, Kasprzyk and E. A. Zagona, 2020), and the Hydrologic Sciences Symposium at the University of Colorado Boulder (Bonham, Zagona and Kasprzyk, 2021).

Importantly, this research is part of a longer overall project that utilizes DMDU in the Colorado River Basin, documented in a co-authored paper entitled 'Decision Science Can Help Address the Challenges of Long-Term Planning in the Colorado River Basin", published in the *Journal of the American Water Resources Association* (Smith *et al.*, 2022). This paper motivates the need for and describes the evolution of DMDU in the CRB, including research activities prior to and including this thesis.

This research has also been disseminated to practitioners with Reclamation through technical reports and presentations. Preliminary research for Chapters 2, 3, and 5 were disseminated as four technical reports for analysts in the CRB. Multiple presentations on Chapter 5 were given to analysts, decision-makers, and stakeholders in the Columbia River Basin.

In an effort to contribute to DMDU education, the contents of this thesis were used to teach a seven-hour course at the 2023 DMDU Summer School in Mexico City ('DMDU Summer School 2023', 2023). The course was entitled 'Participatory DMDU Methods for Water Policy'. The course was attended by approximately 25 individuals including graduate students, post-doctoral researchers, and practitioners.

6.3 Discussion

6.3.1 Model uncertainty

This thesis discussed how uncertainty with respect to exogenous factors (e.g., streamflow) can be analyzed using a SOW ensemble and simulation model to evaluate policy robustness and vulnerability. However, uncertainty also arises when the outputs of the simulation model (e.g., reservoir storage) differ from observations made in real life, so-called model uncertainty (Kennedy, 2023). Ideally, model uncertainty would be minimized using model calibration techniques prior to a DMDU analysis to isolate the impacts of exogenous uncertainties (Mai, 2023). Because models are representations of complex

systems, however, some degree of model uncertainty is unavoidable. Future research could incorporate model uncertainty in DMDU. For example, robustness analysis could include uncertainty intervals that show by how much the ranking of policies could change as a function of model uncertainty. Likewise, scenarios discovered in vulnerability analysis could use uncertainty intervals to show how model uncertainty could impact precisely what conditions lead to, for example, acceptable versus unacceptable performance.

6.3.2 Policy sets: non-dominated, dominated, and other policies

The case study results in Chapters 2 through 4 use 463 Lake Mead policies generated with multi-objective optimization and that are non-dominated with respect to eight specific performance objectives. Policies in such a tradeoff set are non-dominated because of their performance in a specific set of objectives. However, decision-makers may be interested in policies that are dominated with respect to this specific set of objectives for different reasons, such as additional performance goals. These other goals could include, for example, goals that are difficult to quantify as objectives (e.g., how 'palatable' a policy appears to a stakeholder) or robustness with respect to specific robustness metrics. Moreover, decision-makers may also consider policies not generated from optimization at all.

The methods and tools introduced in Chapters 2-5 can be implemented with policy sets that contain either non-dominated or dominated policies. For example, both dominated and non-dominated policies can be ranked according to robustness to test SOW ensemble size as in Chapter 2, and the web tool in Chapter 3 can show the performance of both non-dominated and dominated policies on parallel coordinate plots. In fact, the parallel coordinate plot in Figure 3-5 shows that many of the 463 policies are dominated with respect to stakeholder-selected robustness metrics (not the eight performance objectives used during optimization), and they are shown on the same parallel coordinate plot as non-dominated policies.

Special considerations may be made within methods that perform mathematical analyses on the policy set. For example, in Chapter 4, the SOM organizes policy clusters according to a two-dimensional coordinate system. In our case study, the coordinate system represented a tradeoff (e.g., improving storage objectives at the expense of delivery objectives). However, if the SOM were applied to a set of policies in which some were dominated, it is possible the coordinate system would represent increasing or decreasing performance without a tradeoff. For example, policies on the left side of the map could have small shortages while policies on the right could have large shortages, but this would not necessarily tradeoff with storage objectives if the low-shortage policies on the left dominated policies on the right (i.e., they performed better with respect to both storage and shortage objectives). In this case, the SOM would still be helpful for organizing policies into similarly performing clusters, describing the major differences in performance between policy clusters, and illuminating policies that strike a compromise between different performance goals.

6.4 Ongoing work

In collaboration with Reclamation, this research is contributing to the analysis of post-2026 policies in the CRB. These efforts include space-filling, subsampling methods for identifying diverse and challenging streamflow projections to use in robustness and vulnerability analyses, utilizing methods from Chapter 2. The Self-Organizing Map framework from Chapter 4 is being applied to new and more complex policy sets that include decision levers for Lakes Mead *and* Powell plus delivery rules that adapt with observed streamflow conditions. There is ongoing development of a web tool that expands on the robustness tradeoffs tool presented in Chapter 3 (Smith *et al.*, 2022; Reclamation, 2023c). Reclamation will use this tool to communicate policy performance, robustness, and vulnerability and solicit preferences from a diverse group of stakeholders including water utilities, state agencies, irrigation districts, environmental agencies, Tribal leadership, etc. In parallel with the app development, Reclamation is holding training sessions to ensure stakeholders can meaningfully engage with the tools (Reclamation,

2023c), and information from this thesis has been used in those sessions. There may also be a desire to integrate other DMDU methods into Reclamation's ongoing processes, including aiding decisions at shorter timescales for which existing probabilistic processes are insufficient to capture the variability in the system. We believe this thesis, combined with ongoing collaboration with Reclamation, mark a significant milestone in the adoption of DMDU for participatory environmental management.

Ongoing work is applying the novel methods and tools in this thesis to other water resources systems. After instructing the course at the 2023 DMDU Summer School, requests have been made for other case studies to insert their data into the robustness tradeoffs tool presented in Chapter 3. Future work will collaborate with interested analysts to adapt the CRB version of the tool as needed. The presentations on methods and purposes for vulnerability analysis given to Reclamation analysts have also contributed to two preliminary vulnerability analyses in the Columbia River Basin.

6.5 Future research opportunities

This thesis has identified several research opportunities to further empower participatory DMDU. To reduce computing requirements, future research could investigate the efficacy of adaptive sampling for creating SOW ensembles. Adaptive sampling begins with a small number of space-filling samples, then receives feedback from the simulation model to strategically sample more from regions of the model input space that lead to highly variable and non-linear model outputs (Garud, Karimi and Kraft, 2017). These methods could reduce the number of SOW required for accurate robustness rankings and vulnerability analysis (i.e., similar results compared to a larger SOW ensemble).

A remaining challenge is how to account for time-varying uncertain factors in the creation of SOW ensembles and vulnerability analysis. Time-varying uncertain factors, e.g., streamflow, have been summarized using scalar values, such as multiplicative factors (Kasprzyk *et al.*, 2013), time-series statistics (Bonham, J. Kasprzyk and Zagona, 2022a), or parameters of statistical models (Quinn *et al.*, 2020). However, these methods lose information about the time-varying nature of uncertain factors, meaning

that sampling methods and vulnerability analyses using these scalar statistics may be missing significant predictors of system performance. As described in Chapter 5, recent research has used time-series clustering to find groups of performance metric values whose temporal dynamics are similar. These methods could be extended to uncertain factors, and this information be used in the creation of SOW and as predictors in vulnerability analysis. The continued development of time-series approaches will also be important within efforts to apply DMDU to different types of problems in the CRB and elsewhere that have different decision horizons.

A potential challenge for *a posteriori* tradeoff analysis is recording and communicating decisions to stakeholders and decision-makers. Because such analyses are exploratory by nature, there is not a prescribed workflow. However, it is important for such analyses to be repeatable and interpretable such that stakeholders can provide feedback. The *a posteriori* robustness framework presented in Chapter 3 tracks the user's decisions during an analysis and provides an activity log to enable repeatability, an example of decision provenance (Chakhchoukh, Boukhelifa and Bezerianos, 2022). More sophisticated methods could further record *why* decisions were made and use this information to recommend policies based on the preferences expressed by stakeholders (Häubl and Trifts, 2000). Further research in these areas could improve the interpretability of DMDU, empowering participation.

Another interpretability challenge is explaining the predictions of flexible machine learning methods in a vulnerability analysis. As described in Chapter 5, flexible machine learning methods like random forests can produce more accurate predictions of performance outcomes, compared to less flexible methods like PRIM, when the relationship between uncertain factors and performance is non-linear. However, it can be unclear why a flexible method makes the predictions that it does, e.g., what are the specific values of streamflow, demand, and other uncertain factors that caused the algorithm to predict 'unacceptable' for one SOW and 'acceptable' for another? If flexible methods are to be used for decision-making, such as determining when adaptation is required to avoid unacceptable performance,

the predictions made by the model must be justifiable (Rudin, 2019). But, justifying the predictions can be challenging due to the model's black-box nature. More interpretable methods, like PRIM, could be used instead, but their accuracy for non-linear systems is limited. Future research could couple flexible vulnerability analysis methods with techniques from *explainable artificial intelligence*, a growing research area in machine learning that uses mathematical and visualization techniques to provide simple explanations of why black-box models make the predictions they do (Saranya and Subhashini, 2023). For cases where interpretable models like PRIM are insufficiently accurate, novel methods from explainable artificial intelligence could improve the interpretability of highly flexible methods for vulnerability analysis.

Congruent with the motivation of this thesis, we encourage future research in these areas to be driven by the decision-making purpose with the goal of empowering participatory DMDU.

7 References

Alexander, E. (2018) *Searching for A Robust Operation of Lake Mead*. M.S. University of Colorado. Available at: https://www.colorado.edu/cadswes/sites/default/files/attached-files/searching_for_a_robust_operation_of_lake_mead_2018.pdf (Accessed: 1 April 2022).

Almeida, S. *et al.* (2017) 'Dealing with deep uncertainties in landslide modelling for disaster risk reduction under climate change', *Natural Hazards and Earth System Sciences*, 17(2), pp. 225–241. Available at: https://doi.org/10.5194/nhess-17-225-2017.

Alsudais, A. (2021) 'In-code citation practices in open research software libraries', *Journal of Informetrics*, 15(2), p. 101139. Available at: https://doi.org/10.1016/j.joi.2021.101139.

Bell, A. *et al.* (2022) 'It's Just Not That Simple: An Empirical Study of the Accuracy-Explainability Trade-off in Machine Learning for Public Policy', in *2022 ACM Conference on Fairness, Accountability, and Transparency*. *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency*, Seoul Republic of Korea: ACM, pp. 248–266. Available at: https://doi.org/10.1145/3531146.3533090.

Ben-Haim, Y. (2004) 'Uncertainty, probability and information-gaps', *Reliability Engineering & System Safety*, 85(1), pp. 249–266. Available at: https://doi.org/10.1016/j.ress.2004.03.015.

Bloom, E. (2014) *Providing Decison Support for Adaptive Strategies using Robust Decison Making: Applications in the Colorado River Basin*. Pardee Rand Graduate School. Available at: https://www.rand.org/pubs/rgs_dissertations/RGSD348.html (Accessed: 1 July 2020).

Blount, S. and Bazerman, M.H. (1996) 'The inconsistent evaluation of absolute versus comparative payoffs in labor supply and bargaining', *Journal of Economic Behavior & Organization*, 30(2), pp. 227–240. Available at: https://doi.org/10.1016/S0167-2681(96)00891-8.

Boelaert, J. *et al.* (2021) 'aweSOM: Interactive Self-Organizing Maps'. Available at: https://CRAN.R-project.org/package=aweSOM (Accessed: 24 November 2021).

Bonham, N. (2023) 'CRB-robustness-app'. Available at: https://github.com/nabocrb/CRB-robustness-app---JWRPM (Accessed: 17 March 2023).

Bonham, N. *et al.* (2023) 'Subsampling and Space-filling Metrics to Test Ensemble Size for Robustness Analysis with a Demonstration in the Colorado River Basin', *Environmental Modelling & Software* [Preprint].

Bonham, N., Joseph Kasprzyk, and Edith Zagona (2023) 'Interactive, Multi-metric Robustness Tradeoffs in the Colorado River Basin', *Journal of Water Resources Planning and Management* [Preprint]. Available at: https://doi.org/10.1061/JWRMD5/WRENG-6199.

Bonham, N., Kasprzyk, J. and Zagona, E. (2020) 'H173-07 - Robust Robustness: A Sensitivity Analysis of MORDM with Competing Assumptions about Future States of the World'. *AGU Fall Meeting*, Online Everywhere, 15 December. Available at: https://agu.confex.com/agu/fm20/meetingapp.cgi/Paper/701174.

Bonham, N., Kasprzyk, J. and Zagona, E. (2021) 'H41H-05 - Retroactive and Future Vulnerability of Lake Mead Operating Policies to Uncertainty in Water Supply, Demand and Storage'. *AGU Fall Meeting*, New

Orleans, LA, 16 December. Available at: https://agu.confex.com/agu/fm21/meetingapp.cgi/Home/0 (Accessed: 29 December 2021).

Bonham, N., Kasprzyk, J. and Zagona, E. (2022a) 'post-MORDM: Mapping policies to synthesize optimization and robustness results for decision-maker compromise', *Environmental Modelling & Software*, 157, p. 105491. Available at: https://doi.org/10.1016/j.envsoft.2022.105491.

Bonham, N., Kasprzyk, J. and Zagona, E. (2022b) 'post-MORDM: mapping policies to synthesize optimization and robustness results for decision-maker compromise'. *DMDU 2022*, Mexico City, Mexico, 9 November.

Bonham, N., Kasprzyk, J. and Zagona, E.A. (2020) 'Evaluation of Scenario Discovery Methods for Multi-Reservoir System Planning'. *international Environmental Modeling and Software*, Brussels, Belgium, 14 September.

Bonham, N., Kasprzyk, J.R. and Zagona, E.A. (2022a) 'post-MORDM: mapping policies to synthesize optimization and robustness results for decision-maker compromise', in. *Fall Meeting 2022*, AGU. Available at: https://agu.confex.com/agu/fm22/meetingapp.cgi/Paper/1112638 (Accessed: 21 October 2023).

Bonham, N., Kasprzyk, J.R. and Zagona, E.A. (2022b) 'Vulnerability in the Colorado River Basin: a critical review of vulnerability analyses to inform the renegotiation of Lake Mead operation policy', in. *Fall Meeting 2022*, AGU. Available at: https://agu.confex.com/agu/fm22/meetingapp.cgi/Paper/1111837 (Accessed: 21 October 2023).

Bonham, N., Zagona, E. and Kasprzyk, J. (2021) 'The Colorado River Basin robustness tradeoffs web application'. *Hydrologic Sciences Symposium*, Boulder, CO, 8 April.

Brams, S.J. and Kilgour, D.M. (2001) 'Fallback Bargaining', *Group Decision and Negotiation*, 10(4), pp. 287–316. Available at: https://doi.org/10.1023/A:1011252808608.

Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32. Available at: https://doi.org/10.1023/A:1010933404324.

Brekke, L. *et al.* (2011) *Addressing Climate Change in Long-Term Water Resources Planning and Management*. U.S. Army Corps of Engineers and Bureau of Reclamation, p. 19. Available at: https://www.usbr.gov/climate/userneeds/docs/Summary-standalone-final.pdf.

Brill, E.D. *et al.* (1990) 'MGA: a decision support system for complex, incompletely defined problems', *IEEE Transactions on Systems, Man, and Cybernetics*, 20(4), pp. 745–757. Available at: https://doi.org/10.1109/21.105076.

Brill, E.D., Chang, S.-Y. and Hopkins, L.D. (1982) 'Modeling to Generate Alternatives: The HSJ Approach and an Illustration Using a Problem in Land Use Planning', *Management Science*, 28(3), pp. 221–235. Available at: http://www.jstor.org/stable/2630877 (Accessed: 24 August 2021).

Brown, C. *et al.* (2012) 'Decision scaling: Linking bottom-up vulnerability analysis with climate projections in the water sector', *Water Resources Research*, 48(9). Available at: https://doi.org/10.1029/2011WR011212.

Brus, D.J. (2019) 'Sampling for digital soil mapping: A tutorial supported by R scripts', *Geoderma*, 338, pp. 464–480. Available at: https://doi.org/10.1016/j.geoderma.2018.07.036.

Bryant, B.P. and Lempert, R.J. (2010) 'Thinking inside the box: A participatory, computer-assisted approach to scenario discovery', *Technological Forecasting and Social Change*, 77(1), pp. 34–49. Available at: https://doi.org/10.1016/j.techfore.2009.08.002.

Buschatzke, T. *et al.* (2019) 'Drought Contingency Plans - Basin States transmittal letter to Congress'. Available at: https://www.usbr.gov/dcp/docs/DroughtContigencyPlansBasinStates-TransmittalLetter-508-DOI.pdf (Accessed: 7 February 2022).

Carnell, R. (2022) 'lhs: Latin Hypercube Samples'. Available at: https://cran.r-project.org/web/packages/lhs/index.html (Accessed: 3 October 2023).

Chakhchoukh, M.R., Boukhelifa, N. and Bezerianos, A. (2022) 'Understanding How In-Visualization Provenance Can Support Trade-off Analysis', *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1. Available at: https://doi.org/10.1109/TVCG.2022.3171074.

Chang, W. *et al.* (2022) 'shiny: Web Application Framework for R'. Available at: https://CRAN.R-project.org/package=shiny (Accessed: 21 February 2023).

Chapman, W.L. *et al.* (1994) 'Arctic sea ice variability: Model sensitivities and a multidecadal simulation', *Journal of Geophysical Research: Oceans*, 99(C1), pp. 919–935. Available at: https://doi.org/10.1029/93JC02564.

Choi, Y. *et al.* (2021) 'Comparison of Factorial and Latin Hypercube Sampling Designs for Meta-Models of Building Heating and Cooling Loads', *Energies*, 14(2), p. 512. Available at: https://doi.org/10.3390/en14020512.

Clark, S., Sisson, Scott.A. and Sharma, A. (2020) 'Tools for enhancing the application of self-organizing maps in water resources research and engineering', *Advances in Water Resources*, 143, p. 103676. Available at: https://doi.org/10.1016/j.advwatres.2020.103676.

Cluster with Self-Organizing Map Neural Network - MATLAB & Simulink (no date) *MathWorks*. Available at: https://www.mathworks.com/help/deeplearning/ug/cluster-with-self-organizing-map-neural-network.html (Accessed: 23 February 2022).

Cohen, J.S. and Herman, J.D. (2021) 'Dynamic Adaptation of Water Resources Systems Under Uncertainty by Learning Policy Structure and Indicators', *Water Resources Research*, 57(11). Available at: https://doi.org/10.1029/2021WR030433.

Colorado River Basin Drought Contingency Plans | Bureau of Reclamation (2019). Available at: https://www.usbr.gov/dcp/ (Accessed: 23 November 2021).

Colorado Springs Utilities (2017) *Integrated water resources plan - final report*. Colorado Springs, CO. Available at: https://www.csu.org/Documents/IWRP.pdf?csf=1&e=0DViR6 (Accessed: 24 November 2021).

Committee to Advise the U.S. Global Change Research Program *et al.* (2021) *Global Change Research Needs and Opportunities for 2022-2031*. Washington, D.C.: National Academies Press, p. 26055. Available at: https://doi.org/10.17226/26055.

Damblin, G., Couplet, M. and Iooss, B. (2013) 'Numerical studies of space-filling designs: optimization of Latin Hypercube Samples and subprojection properties', *Journal of Simulation*, 7(4), pp. 276–289. Available at: https://doi.org/10.1057/jos.2013.16.

Dixon, L., Lempert, R.J., LaTourrette, T. and Reville, R.T. (2007) *The Federal Role in Terrorism Insurance: Evaluating Alternatives in an Uncertain World*. RAND Corporation. Available at: https://www.rand.org/pubs/monographs/MG679.html (Accessed: 20 July 2023).

Dixon, L., Lempert, R.J., LaTourrette, T., Reville, R.T., *et al.* (2007) *Trade-Offs Among Alternative Government Interventions in the Market for Terrorism Insurance: Interim Results*. RAND Corporation. Available at: https://www.rand.org/pubs/documented_briefings/DB525.html (Accessed: 20 July 2023).

'DMDU Summer School 2023' (2023) *DMDU Society*, 24 February. Available at: https://www.deepuncertainty.org/2023/02/24/dmdu-summer-school-2023/ (Accessed: 18 October 2023).

DOI (2022) *Interior Department Initiates Significant Action to Protect Colorado River System*. Available at: https://www.doi.gov/pressreleases/interior-department-initiates-significant-action-protect-colorado-river-system (Accessed: 12 November 2022).

Dupuy, D., Helbert, C. and Franco, J. (2015) 'DiceDesign and DiceEval: Two R Packages for Design and Analysis of Computer Experiments', *Journal of Statistical Software*, 65, pp. 1–38. Available at: https://doi.org/10.18637/jss.v065.i11.

Falconi, S.M. and Palmer, R.N. (2017) 'An interdisciplinary framework for participatory modeling design and evaluation—What makes models effective participatory decision tools?', *Water Resources Research*, 53(2), pp. 1625–1645. Available at: https://doi.org/10.1002/2016WR019373.

Farinosi, F. *et al.* (2018) 'An innovative approach to the assessment of hydro-political risk: A spatially explicit, data driven indicator of hydro-political issues', *Global Environmental Change*, 52, pp. 286–313. Available at: https://doi.org/10.1016/j.gloenvcha.2018.07.001.

Frequently Asked Questions (FAQs) - U.S. Energy Information Administration (EIA) (2020). Available at: https://www.eia.gov/tools/faqs/faq.php (Accessed: 7 February 2022).

Friedman, J. and Fisher, N. (1999) 'Bump hunting in high-dimensional data-Discussion', *Statistics and Computing*, 9, pp. 156–162.

Gangopadhyay, S. *et al.* (2022) 'Tree Rings Reveal Unmatched 2nd Century Drought in the Colorado River Basin', *Geophysical Research Letters*, 49(11), p. e2022GL098781. Available at: https://doi.org/10.1029/2022GL098781.

Garud, S.S., Karimi, I.A. and Kraft, M. (2017) 'Design of computer experiments: A review', *Computers & Chemical Engineering*, 106, pp. 71–95. Available at: https://doi.org/10.1016/j.compchemeng.2017.05.010.

Ghojogh, B. *et al.* (2019) 'Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review', *arXiv:1905.02845 [cs, stat]* [Preprint]. Available at: http://arxiv.org/abs/1905.02845 (Accessed: 15 February 2022).

Giuliani, M. *et al.* (2014) 'Many-objective reservoir policy identification and refinement to reduce policy inertia and myopia in water management', *Water Resources Research*, 50(4), pp. 3355–3377. Available at: https://doi.org/10.1002/2013WR014700.

Gold, D.F. *et al.* (2019a) 'Identifying Actionable Compromises: Navigating Multi-City Robustness Conflicts to Discover Cooperative Safe Operating Spaces for Regional Water Supply Portfolios', *Water Resources Research*, 55(11), pp. 9024–9050. Available at: https://doi.org/10.1029/2019WR025462.

Gold, D.F. *et al.* (2019b) 'Identifying Actionable Compromises: Navigating Multi-City Robustness Conflicts to Discover Cooperative Safe Operating Spaces for Regional Water Supply Portfolios', *Water Resources Research*, 55(11), pp. 9024–9050. Available at: https://doi.org/10.1029/2019WR025462.

Groves, D.G. *et al.* (2013) *Adapting to a changing Colorado River: making future water deliveries more reliable through robust management strategies*. Santa Monica, CA: RAND. Available at: https://www.rand.org/pubs/research_reports/RR242.html.

Groves, D.G. and Bloom, E. (2013) *Robust Water-Management Strategies for the California Water Plan Update 2013: Proof-of-Concept Analysis.* Santa Monica, CA: RAND Corporation. Available at: 2013. https://www.rand.org/pubs/research_reports/RR182.html (Accessed: 1 July 2020).

Groves, D.G. and Lempert, R.J. (2007) 'A new analytic method for finding policy-relevant scenarios', *Global Environmental Change*, 17(1), pp. 73–85. Available at: https://doi.org/10.1016/j.gloenvcha.2006.11.006.

Guivarch, C., Rozenberg, J. and Schweizer, V. (2016) 'The diversity of socio-economic pathways and CO2 emissions scenarios: Insights from the investigation of a scenarios database', *Environmental Modelling & Software*, 80, pp. 336–353. Available at: https://doi.org/10.1016/j.envsoft.2016.03.006.

Haasnoot, M. *et al.* (2013) 'Dynamic adaptive policy pathways: A method for crafting robust decisions for a deeply uncertain world', *Global Environmental Change*, 23(2), pp. 485–498. Available at: https://doi.org/10.1016/j.gloenvcha.2012.12.006.

Hadjimichael, A., Quinn, J., *et al.* (2020) 'Defining robustness, vulnerabilities, and consequential scenarios for diverse stakeholder interests in institutionally complex river basins', *Earth's Future* [Preprint]. Available at: https://doi.org/10.1029/2020EF001503.

Hadjimichael, A., Gold, D., *et al.* (2020) 'Rhodium: Python Library for Many-Objective Robust Decision Making and Exploratory Modeling', *Journal of Open Research Software*, 8(1), p. 12. Available at: https://doi.org/10.5334/jors.293.

Hadka, D. (2015) 'Introducing OpenMORDM', *Water Programming: A Collaborative Research Blog*, 1 October. Available at: https://waterprogramming.wordpress.com/2015/10/01/introducing-openmordm/ (Accessed: 28 October 2019).

Hadka, D. and Reed, P. (2013) 'Borg: An Auto-Adaptive Many-Objective Evolutionary Computing Framework', *Evolutionary Computation*, 21(2), pp. 231–259. Available at: https://doi.org/10.1162/EVCO_a_00075.

Hall, C.A. *et al.* (2022) 'A hydrologist's guide to open science', *Hydrology and Earth System Sciences*, 26(3), pp. 647–664. Available at: https://doi.org/10.5194/hess-26-647-2022.

Hashimoto, T., Stedinger, J.R. and Loucks, D.P. (1982) 'Reliability, resiliency, and vulnerability criteria for water resource system performance evaluation', *Water Resources Research*, 18(1), pp. 14–20. Available at: https://doi.org/10.1029/WR018i001p00014.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd edn. New York: Springer-Verlag (Springer Series in Statistics). Available at: https://doi.org/10.1007/978-0-387-84858-7.

Häubl, G. and Trifts, V. (2000) 'Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids', *Marketing Science*, 19(1), pp. 4–21. Available at: https://doi.org/10.1287/mksc.19.1.4.15178.

Herman, J.D. *et al.* (2014) 'Beyond optimality: Multistakeholder robustness tradeoffs for regional water portfolio planning under deep uncertainty', *Water Resources Research*, 50(10), pp. 7692–7713. Available at: https://doi.org/10.1002/2014WR015338.

Herman, J.D. *et al.* (2015) 'How Should Robustness Be Defined for Water Systems Planning under Change?', *Journal of Water Resources Planning and Management*, 141(10), p. 04015012. Available at: https://doi.org/10.1061/(ASCE)WR.1943-5452.0000509.

High Performance Computing – HPC | Microsoft Azure (no date). Available at: https://azure.microsoft.com/en-us/solutions/high-performance-computing (Accessed: 5 October 2023).

High Performance Computing (HPC) | AWS (no date) *Amazon Web Services, Inc.* Available at: https://aws.amazon.com/hpc/ (Accessed: 5 October 2023).

Hill, R.R. and Miller, J.O. (2017) 'A history of United States military simulation', in *2017 Winter Simulation Conference (WSC)*. *2017 Winter Simulation Conference (WSC)*, Las Vegas, NV: IEEE, pp. 346–364. Available at: https://doi.org/10.1109/WSC.2017.8247799.

Hira, Z.M. and Gillies, D.F. (2015) 'A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data', *Advances in Bioinformatics*, 2015, pp. 1–13. Available at: https://doi.org/10.1155/2015/198363.

Holling, C.S. (ed.) (2005) *Adaptive environmental assessment and management*. Reprint of the 1978 ed. Caldwell, NJ: Blackburn Press. Available at: https://pure.iiasa.ac.at/id/eprint/823/1/XB-78-103.pdf.

Inselberg, A. (2009) *Parallel Coordinates*. New York, NY: Springer New York. Available at: https://doi.org/10.1007/978-0-387-68628-8.

International Boundary and Water Commission (2012) 'Minute 319: Interim international cooperative measures in the Colorado River Basin through 2017 and extension of minute 318 cooperative measures

to address the continued effects of the April 2010 earthquake in the Mexicali Valley, Baja California'. Available at: https://www.ibwc.gov/Files/Minutes/Minute_319.pdf.

International Boundary and Water Commission (2017) 'Minute 323: Extension of cooperative measures and adoption of a binational water scarcity contingency plan in the Colorado River Basin'. Available at: https://www.usbr.gov/lc/region/g4000/4200Rpts/DecreeRpt/2018/43.pdf.

IPCC (2021) *Summary for Policymakers. In: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press. Available at: https://www.ipcc.ch/report/ar6/wg1/downloads/report/IPCC_AR6_WGI_SPM_final.pdf (Accessed: 11 January 2022).

IPCC (2022) *Climate Change 2022: Impacts, Adaptation and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Available at: https://report.ipcc.ch/ar6/wg2/IPCC_AR6_WGII_FullReport.pdf.

Jafino, B.A. and Kwakkel, J.H. (2021) 'A novel concurrent approach for multiclass scenario discovery using Multivariate Regression Trees: Exploring spatial inequality patterns in the Vietnam Mekong Delta under uncertainty', *Environmental Modelling & Software*, 145, p. 105177. Available at: https://doi.org/10.1016/j.envsoft.2021.105177.

James, G. *et al.* (2013) *An Introduction to Statistical Learning*. New York, NY: Springer New York (Springer Texts in Statistics). Available at: https://doi.org/10.1007/978-1-4614-7138-7.

James, T. *et al.* (2014) 'The economic importance of the Colorado River to the basin region'. Available at: https://businessforwater.org/wp-content/uploads/2016/12/PTF-Final-121814.pdf.

Jones, D.R., Schonlau, M. and Welch, W.J. (1998) 'Efficient Global Optimization of Expensive Black-Box Functions', *Journal of Global Optimization*, 13(4), pp. 455–492. Available at: https://doi.org/10.1023/A:1008306431147.

Joseph, V.R. (2016) 'Space-filling designs for computer experiments: A review', *Quality Engineering*, 28(1), pp. 28–35. Available at: https://doi.org/10.1080/08982112.2015.1100447.

Kansara, S., Parashar, S. and Xue, Z. (2015) 'Effective Decision Making and Data Visualization Using Partitive Clustering and Principal Component Analysis (PCA) for High Dimensional Pareto Frontier Data', *SAE International Journal of Materials and Manufacturing*, 8(2), pp. 336–343. Available at: https://doi.org/10.4271/2015-01-0460.

Kasprzyk, J. and Garcia, M. (2023) 'Guiding Questions for Water Resources Systems Analysis Research', *Journal of Water Resources Planning and Management*, 149(8), p. 01823001. Available at: https://doi.org/10.1061/JWRMD5.WRENG-6198.

Kasprzyk, J.R. *et al.* (2012) 'Many-objective de Novo water supply portfolio planning under deep uncertainty', *Environmental Modelling & Software*, 34, pp. 87–104. Available at: https://doi.org/10.1016/j.envsoft.2011.04.003.

Kasprzyk, J.R. *et al.* (2013) 'Many objective robust decision making for complex environmental systems undergoing change', *Environmental Modelling & Software*, 42, pp. 55–71. Available at: https://doi.org/10.1016/j.envsoft.2012.12.007.

Kasprzyk, J.R. *et al.* (2018) 'Defining the Role of Water Resources Systems Analysis in a Changing Future', *Journal of Water Resources Planning and Management*, 144(12), p. 01818003. Available at: https://doi.org/10.1061/(ASCE)WR.1943-5452.0001010.

Kendall, M.G. (1938) 'A New Measure of Rank Correlation', *Biometrika*, 30(1/2), p. 81. Available at: https://doi.org/10.2307/2332226.

Kendall, M.G. (1945) 'THE TREATMENT OF TIES IN RANKING PROBLEMS', *Biometrika*, 33(3), pp. 239–251. Available at: https://doi.org/10.1093/biomet/33.3.239.

Kennard, R.W. and Stone, L.A. (1969) 'Computer Aided Design of Experiments', *Technometrics*, 11(1), pp. 137–148. Available at: https://doi.org/10.1080/00401706.1969.10490666.

Kennedy, M.C. (2023) 'Chapter 42 - Exposure assessment: modeling approaches including probabilistic methods, uncertainty analysis, and aggregate exposure from multiple sources', in M.E. Knowles et al. (eds) *Present Knowledge in Food Safety*. Academic Press, pp. 614–632. Available at: https://doi.org/10.1016/B978-0-12-819470-6.00032-9.

Khalid, S., Khalil, T. and Nasreen, S. (2014) 'A survey of feature selection and feature extraction techniques in machine learning', in *2014 Science and Information Conference*. *2014 Science and Information Conference*, pp. 372–378. Available at: https://doi.org/10.1109/SAI.2014.6918213.

Knight, F. (1921) *Risk, uncertainty and profit*. Boston, MA: Houghton Mifflin. Available at: https://www.worldcat.org/title/risk-uncertainty-and-profit/oclc/1410657.

Kohonen, T. (1982) 'Self-organized formation of topologically correct feature maps', *Biological Cybernetics*, 43(1), pp. 59–69. Available at: https://doi.org/10.1007/BF00337288.

Kohonen, T. (1990) 'The self-organizing map', *Proceedings of the IEEE*, 78(9), pp. 1464–1480. Available at: https://doi.org/10.1109/5.58325.

Kohonen, T. (2001) 'The Basic SOM', in T. Kohonen (ed.) *Self-Organizing Maps*. Berlin, Heidelberg: Springer (Springer Series in Information Sciences), pp. 105–176. Available at: https://doi.org/10.1007/978-3-642-56927-2_3.

Kohonen, T. (2013) 'Essentials of the self-organizing map', *Neural Networks*, 37, pp. 52–65. Available at: https://doi.org/10.1016/j.neunet.2012.09.018.

Koishi, M. and Shida, Z. (2006) 'Multi-Objective Design Problem of Tire Wear and Visualization of Its Pareto Solutions2', *Tire Science and Technology*, 34(3), pp. 170–194. Available at: https://doi.org/10.2346/1.2345640.

Kollat, J.B. and Reed, P. (2007) 'A framework for Visually Interactive Decision-making and Design using Evolutionary Multi-objective Optimization (VIDEO)', *Environmental Modelling & Software*, 22(12), pp. 1691–1704. Available at: https://doi.org/10.1016/j.envsoft.2007.02.001.

Kravits, J. *et al.* (2021) 'Screening Tool for Dam Hazard Potential Classification Using Machine Learning and Multiobjective Parameter Tuning', *Journal of Water Resources Planning and Management*, 147(10), p. 04021064. Available at: https://doi.org/10.1061/(ASCE)WR.1943-5452.0001414.

Kwakkel, J.H. (2017) 'The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making', *Environmental Modelling & Software*, 96, pp. 239–250. Available at: https://doi.org/10.1016/j.envsoft.2017.06.054.

Kwakkel, J.H. (2019) 'A generalized many-objective optimization approach for scenario discovery', *FUTURES & FORESIGHT SCIENCE*, 1(2), p. e8. Available at: https://doi.org/10.1002/ffo2.8.

Kwakkel, J.H. and Cunningham, S.C. (2016) 'Improving scenario discovery by bagging random boxes', *Technological Forecasting and Social Change*, 111, pp. 124–134. Available at: https://doi.org/10.1016/j.techfore.2016.06.014.

Kwakkel, J.H. and Haasnoot, M. (2019) 'Supporting DMDU: A Taxonomy of Approaches and Tools', in V.A.W.J. Marchau et al. (eds) *Decision Making under Deep Uncertainty*. Cham: Springer International Publishing, pp. 355–374. Available at: https://doi.org/10.1007/978-3-030-05252-2_15.

Kwakkel, J.H. and Jaxa-Rozen, M. (2016) 'Improving scenario discovery for handling heterogeneous uncertainties and multinomial classified outcomes', *Environmental Modelling & Software*, 79, pp. 311–321. Available at: https://doi.org/10.1016/j.envsoft.2015.11.020.

Lahtinen, T.J., Guillaume, J.H.A. and Hämäläinen, R.P. (2017) 'Why pay attention to paths in the practice of environmental modelling?', *Environmental Modelling & Software*, 92, pp. 74–81. Available at: https://doi.org/10.1016/j.envsoft.2017.02.019.

LeCompte, D.C. (1999) 'Seven, Plus or Minus Two, is too much to Bear: Three (or Fewer) is the Real Magic Number', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 43(3), pp. 289–292. Available at: https://doi.org/10.1177/154193129904300334.

Lee, I. and Shin, Y.J. (2020) 'Machine learning for enterprises: Applications, algorithm selection, and challenges', *Business Horizons*, 63(2), pp. 157–170. Available at: https://doi.org/10.1016/j.bushor.2019.10.005.

Lempert, R. (2013) 'Scenarios that illuminate vulnerabilities and robust responses', *Climatic Change*, 117(4), pp. 627–646. Available at: https://doi.org/10.1007/s10584-012-0574-6.

Lempert, R.J. *et al.* (2006) 'A General, Analytic Method for Generating Robust Strategies and Narrative Scenarios', *Management Science*, 52(4), pp. 514–528. Available at: https://doi.org/10.1287/mnsc.1050.0472.

Lempert, R.J., Bryant, B.P. and Bankes, S.C. (2008) 'Comparing Algorithms for Scenario Discovery'. Available at: https://www.rand.org/pubs/working_papers/WR557.html.

Lempert, R.J. and Collins, M.T. (2007) 'Managing the Risk of Uncertain Threshold Responses: Comparison of Robust, Optimum, and Precautionary Approaches', *Risk Analysis*, 27(4), pp. 1009–1026. Available at: https://doi.org/10.1111/j.1539-6924.2007.00940.x.

Lempert, R.J., Popper, S.W. and Bankes, S.C. (2003) *Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis*. RAND Corporation. Available at: https://www.rand.org/pubs/monograph_reports/MR1626.html (Accessed: 10 January 2022).

Lempert, R.J. and Turner, S. (2020) 'Engaging Multiple Worldviews With Quantitative Decision Support: A Robust Decision-Making Demonstration Using the Lake Model', *Risk Analysis*, p. risa.13579. Available at: https://doi.org/10.1111/risa.13579.

Levy, S. and Steinberg, D.M. (2010) 'Computer experiments: a review', *AStA Advances in Statistical Analysis*, 94(4), pp. 311–324. Available at: https://doi.org/10.1007/s10182-010-0147-9.

Li, H. and Liu, Z. (2021) 'Multivariate time series clustering based on complex network', *Pattern Recognition*, 115, p. 107919. Available at: https://doi.org/10.1016/j.patcog.2021.107919.

Li, Y. and Kinzelbach, W. (2020) 'Resolving Conflicts between Irrigation Agriculture and Ecohydrology Using Many-Objective Robust Decision Making', *Journal of Water Resources Planning and Management*, 146(9), p. 05020014. Available at: https://doi.org/10.1061/(ASCE)WR.1943-5452.0001261.

Li, Z., Liao, H. and Coit, D.W. (2009) 'A two-stage approach for multi-objective decision making with applications to system reliability optimization', *Reliability Engineering & System Safety*, 94(10), pp. 1585–1592. Available at: https://doi.org/10.1016/j.ress.2009.02.022.

Loeppky, J.L., Sacks, J. and Welch, W.J. (2009) 'Choosing the Sample Size of a Computer Experiment: A Practical Guide', *Technometrics*, 51(4), pp. 366–376. Available at: https://doi.org/10.1198/TECH.2009.08040.

Lukas, J. and Payton, E. (2020) *Colorado River Basin Climate and Hydrology: State of the Science*. Boulder, CO: Western Water Assessment. Available at: https://scholar.colorado.edu/concern/reports/8w32r663z (Accessed: 23 November 2021).

Ma, T. *et al.* (2020) 'Comparison of conditioned Latin hypercube and feature space coverage sampling for predicting soil classes using simulation from soil maps', *Geoderma*, 370, p. 114366. Available at: https://doi.org/10.1016/j.geoderma.2020.114366.

Madani, K., Shalikarian, L. and Naeeni, S.T.O. (2011) 'Resolving hydro-environmental conflicts under uncertainty using Fallback Bargaining procedures'. Available at: http://www.ipcbee.com/vol8/43-S10035.pdf (Accessed: 1 April 2022).

Mai, J. (2023) 'Ten strategies towards successful calibration of environmental models', *Journal of Hydrology*, 620, p. 129414. Available at: https://doi.org/10.1016/j.jhydrol.2023.129414.

Maier, H.R. *et al.* (2019) 'Introductory overview: Optimization using evolutionary algorithms and other metaheuristics', *Environmental Modelling & Software*, 114, pp. 195–213. Available at: https://doi.org/10.1016/j.envsoft.2018.11.018.

Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2018) 'Statistical and Machine Learning forecasting methods: Concerns and ways forward', *PLOS ONE*, 13(3), p. e0194889. Available at: https://doi.org/10.1371/journal.pone.0194889.

Marchau, V.A.W.J. *et al.* (eds) (2019) *Decision Making under Deep Uncertainty: From Theory to Practice*. Cham: Springer International Publishing. Available at: https://doi.org/10.1007/978-3-030-05252-2.

McIntosh, B.S. *et al.* (2011) 'Environmental decision support systems (EDSS) development – Challenges and best practices', *Environmental Modelling & Software*, 26(12), pp. 1389–1402. Available at: https://doi.org/10.1016/j.envsoft.2011.09.009.

McLeod, A.I. (2022) 'Kendall: Kendall Rank Correlation and Mann-Kendall Trend Test'. Available at: https://cran.r-project.org/web/packages/Kendall/index.html (Accessed: 9 June 2023).

McPhail, C. *et al.* (2018) 'Robustness Metrics: How Are They Calculated, When Should They Be Used and Why Do They Give Different Results?', *Earth's Future*, 6(2), pp. 169–191. Available at: https://doi.org/10.1002/2017EF000649.

McPhail, C. *et al.* (2020) 'Impact of Scenario Selection on Robustness', *Water Resources Research*, 56(9). Available at: https://doi.org/10.1029/2019WR026515.

McPhail, C. *et al.* (2021) 'Guidance framework and software for understanding and achieving system robustness', *Environmental Modelling & Software*, 142, p. 105059. Available at: https://doi.org/10.1016/j.envsoft.2021.105059.

Means, E. *et al.* (2010) *Decision Support Planning Methods: Incorporating Climate Change Uncertainties into Water Planning*. Water Utility Climate Alliance, p. 113. Available at: https://www.wucaonline.org/assets/pdf/pubs-whitepaper-012110.pdf (Accessed: 9 June 2023).

Merritt, W.S. *et al.* (2017) 'Realizing modelling outcomes: A synthesis of success factors and their use in a retrospective analysis of 15 Australian water resource projects', *Environmental Modelling & Software*, 94, pp. 63–72. Available at: https://doi.org/10.1016/j.envsoft.2017.03.021.

Miller, G.A. (1956) 'The magical number seven, plus or minus two: Some limits on our capacity for processing information', *Psychological Review*, 63(2), pp. 81–97. Available at: https://doi.org/10.1037/h0043158.

Minasny, B. and McBratney, A.B. (2006) 'A conditioned Latin hypercube method for sampling in the presence of ancillary information', *Computers & Geosciences*, 32(9), pp. 1378–1388. Available at: https://doi.org/10.1016/j.cageo.2005.12.009.

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

Moallemi, E.A. *et al.* (2021) 'Evaluating Participatory Modeling Methods for Co-creating Pathways to Sustainability', *Earth's Future*, 9(3), p. e2020EF001843. Available at: https://doi.org/10.1029/2020EF001843.

Molina-Perez, E. *et al.* (2019) *Developing a robust water strategy for Monterrey, Mexico: diversification and adaptation for coping with climate, economic, and technological uncertainties*. Available at: https://www.rand.org/pubs/research_reports/RR3017.html (Accessed: 1 April 2022).

Mosnier, D., Gillot, F. and Ichchou, M. (2013) 'Integrated workflow for multi-objective evolutionary optimization of the vehicle tyre parameters', *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 227(2), pp. 222–233. Available at: https://doi.org/10.1177/0954407012450821.

Nikas, A., Doukas, H. and Papandreou, A. (2019) 'A Detailed Overview and Consistent Classification of Climate-Economy Models', in H. Doukas, A. Flamos, and J. Lieu (eds) *Understanding Risks and Uncertainties in Energy and Climate Policy*. Cham: Springer International Publishing, pp. 1–54. Available at: https://doi.org/10.1007/978-3-030-03152-7_1.

Obayashi, S. and Sasaki, D. (2003) 'Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map', in C.M. Fonseca et al. (eds) *Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 796–809. Available at: https://doi.org/10.1007/3-540-36970-8_56.

Pedregosa, F. *et al.* (2011) 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, 12(85), pp. 2825–2830. Available at: http://jmlr.org/papers/v12/pedregosa11a.html (Accessed: 23 October 2023).

Peñuela, A., Hutton, C. and Pianosi, F. (2021) 'An open-source package with interactive Jupyter Notebooks to enhance the accessibility of reservoir operations simulation and optimisation', *Environmental Modelling & Software*, 145, p. 105188. Available at: https://doi.org/10.1016/j.envsoft.2021.105188.

Pianosi, F., Sarrazin, F. and Wagener, T. (2020) 'How successfully is open-source research software adopted? Results and implications of surveying the users of a sensitivity analysis toolbox', *Environmental Modelling and Software*, 124(104579). Available at: https://doi.org/10.1016/j.envsoft.2019.104579.

Quinn, J.D. *et al.* (2017) 'Rival framings: A framework for discovering how problem formulation uncertainties shape risk management trade-offs in water resources systems', *Water Resources Research*, 53(8), pp. 7208–7233. Available at: https://doi.org/10.1002/2017WR020524.

Quinn, J.D. *et al.* (2018) 'Exploring How Changing Monsoonal Dynamics and Human Pressures Challenge Multireservoir Management for Flood Protection, Hydropower Production, and Agricultural Water Supply', *Water Resources Research*, 54(7), pp. 4638–4662. Available at: https://doi.org/10.1029/2018WR022743.

Quinn, J.D. *et al.* (2020) 'Can Exploratory Modeling of Water Scarcity Vulnerabilities and Robustness Be Scenario Neutral?', *Earth's Future*, 8(11). Available at: https://doi.org/10.1029/2020EF001650.

R Core Team (2022) 'R: A language and environment for statistical computing.' Vienna, Australia: R Foundation for Statistical Computing. Available at: https://www.r-project.org/ (Accessed: 19 March 2021).

R Core Team (2023) 'R: A language and environment for statistical computing.' Vienna, Australia: R Foundation for Statistical Computing. Available at: https://www.r-project.org/ (Accessed: 19 March 2021).

Ragan, E. *et al.* (2016) 'Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes', *IEEE Transactions on Visualization and Computer Graphics*, 22(1). Available at: https://doi.org/10.1109/TVCG.2015.2467551.

Raseman, W.J. *et al.* (2020) 'Multi-objective optimization of water treatment operations for disinfection byproduct control', *Environmental Science: Water Research & Technology*, 6(3), pp. 702–714. Available at: https://doi.org/10.1039/C9EW00850K.

Raseman, W.J., Jacobson, J. and Kasprzyk, J.R. (2019) 'Parasol: an open source, interactive parallel coordinates library for multi-objective decision making', *Environmental Modelling & Software*, 116, pp. 153–163. Available at: https://doi.org/10.1016/j.envsoft.2019.03.005.

Razavi, S. *et al.* (2021) 'The Future of Sensitivity Analysis: An essential discipline for systems modeling and policy support', *Environmental Modelling & Software*, 137, p. 104954. Available at: https://doi.org/10.1016/j.envsoft.2020.104954.

Reclamation (2007) *Colorado River Interim Guidelines for Lower Basin Shortage and Coordinated Operations for Lake Powell and Mead - Final Environmental Impact Statement*. Available at: https://www.usbr.gov/lc/region/programs/strategies/FEIS/ExecSumm.pdf (Accessed: 22 February 2020).

Reclamation (2011) *Lake Mead Area and Capacity Tables*. Available at: https://www.usbr.gov/lc/region/g4000/LM_AreaCapacityTables2009.pdf (Accessed: 18 October 2022).

Reclamation (2012a) *Colorado River Basin Water Supply and Demand Study*. Available at: https://www.usbr.gov/lc/region/programs/crbstudy/finalreport/Study%20Report/CRBS_Study_Report_FINAL.pdf (Accessed: 30 August 2019).

Reclamation (2012b) *Reclamation's NEPA Handbook*. U.S. Department of the Interior Bureau of Reclamation, p. 292. Available at: https://www.usbr.gov/nepa/docs/NEPA_Handbook2012.pdf (Accessed: 1 August 2023).

Reclamation (2015) *Law of the River| Lower Colorado Region | Bureau of Reclamation*. Available at: https://www.usbr.gov/lc/region/pao/lawofrvr.html (Accessed: 7 February 2022).

Reclamation (2018a) *Colorado River Basin Ten Tribes Partnership Tribal Water Study Report*. USBR. Available at: https://www.usbr.gov/lc/region/programs/crbstudy/tws/docs/Ch.%205.1%20Ute%20Tribe%20Current-Future%20Water%20Use%2012-13-2018.pdf (Accessed: 16 March 2020).

Reclamation (2018b) *Hoover Dam | Bureau of Reclamation*. Available at: https://www.usbr.gov/lc/hooverdam/faqs/powerfaq.html (Accessed: 7 February 2022).

Reclamation (2020) *5-Year Probabilistic Projections*. Available at: https://www.usbr.gov/lc/region/g4000/riverops/crss-5year-projections.html (Accessed: 11 January 2022).

Reclamation (2021a) *Lower Colorado Water Supply Report*. Available at: https://www.usbr.gov/lc/region/g4000/weekly.pdf (Accessed: 23 November 2021).

Reclamation (2021b) *Power Office | Upper Colorado Basin*. Available at: https://usbr.gov/uc/power/ (Accessed: 7 February 2022).

Reclamation (2022) *General Modeling Information*. Available at: https://www.usbr.gov/lc/region/g4000/riverops/model-info.html.

Reclamation (2023a) *Lower Colorado water supply report*. Available at: https://www.usbr.gov/lc/region/g4000/weekly.pdf.

Reclamation (2023b) *Notice of Intent To Prepare an Environmental Impact Statement and Notice To Solicit Comments and Hold Public Scoping Meetings on the Development of Post-2026 Operational Guidelines and Strategies for Lake Powell and Lake Mead*, *Federal Register*. Available at: https://www.federalregister.gov/documents/2023/06/16/2023-12923/notice-of-intent-to-prepare-an-environmental-impact-statement-and-notice-to-solicit-comments-and (Accessed: 20 June 2023).

Reclamation (2023c) *Post-2026 Colorado River Reservoir Operational Strategies for Lake Powell and Lake Mead*. Available at: https://www.usbr.gov/ColoradoRiverBasin/Post2026Ops.html (Accessed: 6 March 2023).

Reclamation (2023d) *Reclamation announces 2024 operating conditions for Lake Powell and Lake Mead*, *Newsroom*. Available at: https://www.usbr.gov/newsroom/ (Accessed: 19 October 2023).

Reclamation (2023e) *Summary of pre-scoping comments for development of post-2026 Colorado River reservoir operations*. Available at: https://www.usbr.gov/ColoradoRiverBasin/documents/Post-2026_Pre-Scoping%20Comment%20Summary%20Final_Updated1.30.2023_508.pdf.

Reis, J. and Shortridge, J. (2020) 'Impact of Uncertainty Parameter Distribution on Robust Decision Making Outcomes for Climate Change Adaptation under Deep Uncertainty', *Risk Analysis*, 40(3), pp. 494–511. Available at: https://doi.org/10.1111/risa.13405.

Reis, J. and Shortridge, J. (2021) 'Robust decision outcomes with induced correlations in climatic and economic parameters', *Mitigation and Adaptation Strategies for Global Change*, 27(1), p. 7. Available at: https://doi.org/10.1007/s11027-021-09970-5.

Rendón, E. *et al.* (2011) 'Internal versus External cluster validation indexes', *International Journal of Computers and Communications*, 5(1), p. 8. Available at: http://universitypress.org.uk/journals/cc/20-463.pdf (Accessed: 1 April 2022).

River Management Joint Operating Committee (2020) *Climate and Hydrology Datasets for RMJOC Long-Term Planning Studies: Second Edition (RMJOC-II)*. Available at: https://www.bpa.gov/-/media/Aep/power/hydropower-data-studies/rmjoc-ll-report-part-l.pdf (Accessed: 10 October 2022).

Robinson, B., Cohen, J.S. and Herman, J.D. (2020) 'Detecting early warning signals of long-term water supply vulnerability using machine learning', *Environmental Modelling & Software*, 131, p. 104781. Available at: https://doi.org/10.1016/j.envsoft.2020.104781.

Root, J.C. and Jones, D. (2022) *Elevation-area-capacity relationships of Lake Powell in 2018 and estimated loss of storage capacity since 1963*, *Elevation-area-capacity relationships of Lake Powell in*

2018 and estimated loss of storage capacity since 1963. USGS Numbered Series 2022–5017. Reston, VA: U.S. Geological Survey, p. 21. Available at: https://doi.org/10.3133/sir20225017.

Rosenberg, D.E. (2022) 'Adapt Lake Mead Releases to Inflow to Give Managers More Flexibility to Slow Reservoir Drawdown', *Journal of Water Resources Planning and Management*, 148(10), p. 02522006. Available at: https://doi.org/10.1061/(ASCE)WR.1943-5452.0001592.

Roudier, P. (2020) 'R package: Conditioned Latin Hypercube Sampling'. Available at: https://cran.r-project.org/web/packages/clhs/clhs.pdf (Accessed: 17 July 2020).

Roudier, P. *et al.* (2021) 'clhs: Conditioned Latin Hypercube Sampling'. Available at: https://cran.r-project.org/web/packages/clhs/index.html (Accessed: 13 June 2023).

Roy, B. (1990) 'Decision-aid and decision-making', *European Journal of Operational Research*, 45(2–3), pp. 324–331. Available at: https://doi.org/10.1016/0377-2217(90)90196-I.

Rozenberg, J. *et al.* (2014) 'Building SSPs for climate policy analysis: a scenario elicitation methodology to map the space of possible future challenges to mitigation and adaptation', *Climatic Change*, 122(3), pp. 509–522. Available at: https://doi.org/10.1007/s10584-013-0904-3.

Rudin, C. (2019) 'Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead', *Nature Machine Intelligence*, 1(5), pp. 206–215. Available at: https://doi.org/10.1038/s42256-019-0048-x.

Rudin, C. *et al.* (2022) 'Interpretable machine learning: Fundamental principles and 10 grand challenges', *Statistics Surveys*, 16(none), pp. 1–85. Available at: https://doi.org/10.1214/21-SS133.

Rudin, C. and Radin, J. (2019) 'Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition', *Harvard Data Science Review*, 1(2). Available at: https://doi.org/10.1162/99608f92.5a8a3a3d.

Saaty, T.L. and Ozdemir, M.S. (2003) 'Why the magic number seven plus or minus two', *Mathematical and Computer Modelling*, 38(3), pp. 233–244. Available at: https://doi.org/10.1016/S0895-7177(03)90083-5.

Salehabadi, H. *et al.* (2022) 'An Assessment of Potential Severe Droughts in the Colorado River Basin', *JAWRA Journal of the American Water Resources Association*, n/a(n/a). Available at: https://doi.org/10.1111/1752-1688.13061.

Saranya, A. and Subhashini, R. (2023) 'A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends', *Decision Analytics Journal*, 7, p. 100230. Available at: https://doi.org/10.1016/j.dajour.2023.100230.

Schmidt, K. *et al.* (2014) 'A comparison of calibration sampling schemes at the field scale', *Geoderma*, 232–234, pp. 243–256. Available at: https://doi.org/10.1016/j.geoderma.2014.05.013.

Shortridge, J.E. and Zaitchik, B.F. (2018) 'Characterizing climate change risks by linking robust decision frameworks and uncertain probabilistic projections', *Climatic Change*, 151(3–4), pp. 525–539. Available at: https://doi.org/10.1007/s10584-018-2324-x.

Sievert, C. *et al.* (2022) 'plotly: Create Interactive Web Graphics via "plotly.js"'. Available at: https://CRAN.R-project.org/package=plotly (Accessed: 21 February 2023).

Sievert, C. *et al.* (2023) 'flexdashboard: R Markdown Format for Flexible Dashboards'. Available at: https://CRAN.R-project.org/package=flexdashboard (Accessed: 21 February 2023).

Smith, R. (2021) 'sklearn-som: A simple planar self organizing map'. Available at: https://github.com/rileypsmith/sklearn-som (Accessed: 23 February 2022).

Smith, R. *et al.* (2022) 'Decision Science Can Help Address the Challenges of Long-Term Planning in the Colorado River Basin', *JAWRA Journal of the American Water Resources Association*, 58(5), pp. 735–745. Available at: https://doi.org/10.1111/1752-1688.12985.

Smith, R., Kasprzyk, J. and Basdekas, L. (2018) 'Experimenting with Water Supply Planning Objectives Using the Eldorado Utility Planning Model Multireservoir Testbed', *Journal of Water Resources Planning and Management*, 144(8). Available at: https://doi.org/10.1061/(ASCE)WR.1943-5452.0000962.

Smith, R., Kasprzyk, J. and Dilling, L. (2017) 'Participatory Framework for Assessment and Improvement of Tools (ParFAIT): Increasing the impact and relevance of water management decision support research', *Environmental Modelling & Software*, 95, pp. 432–446. Available at: https://doi.org/10.1016/j.envsoft.2017.05.004.

Smith, R., Kasprzyk, J. and Dilling, L. (2019) 'Testing the potential of Multiobjective Evolutionary Algorithms (MOEAs) with Colorado water managers', *Environmental Modelling & Software*, 117, pp. 149–163. Available at: https://doi.org/10.1016/j.envsoft.2019.03.011.

Smith, R., Kasprzyk, J. and Rajagopalan, B. (2019) 'Using multivariate regression trees and multiobjective tradeoff sets to reveal fundamental insights about water resources systems', *Environmental Modelling & Software*, 120, p. 104498. Available at: https://doi.org/10.1016/j.envsoft.2019.104498.

Snell, T. and Cowell, R. (2006) 'Scoping in environmental impact assessment: Balancing precaution and efficiency?', *Environmental Impact Assessment Review*, 26(4), pp. 359–376. Available at: https://doi.org/10.1016/j.eiar.2005.06.003.

Sojda, R. *et al.* (2012) 'Identifying the decision to be supported : a review of papers from Environmental Modelling and Software', in. *International Environmetnal Modelling and Software 2012*, Leipzig, Germany, pp. 73–80. Available at: https://www.researchgate.net/publication/235352901_Identifying_the_decision_to_be_supported_a_review_of_papers_from_Environmental_Modelling_and_Software.

Stanton, M.C.B. and Roelich, K. (2021) 'Decision making under deep uncertainties: A review of the applicability of methods in practice', *Technological Forecasting and Social Change*, 171, p. 120939. Available at: https://doi.org/10.1016/j.techfore.2021.120939.

Steinmann, P., Auping, W.L. and Kwakkel, J.H. (2020) 'Behavior-based scenario discovery using time series clustering', *Technological Forecasting and Social Change*, 156, p. 120052. Available at: https://doi.org/10.1016/j.techfore.2020.120052.

Stevens, A.R.H. *et al.* (2020) 'The imperative to reduce carbon emissions in astronomy', *Nature Astronomy*, 4(9), pp. 843–851. Available at: https://doi.org/10.1038/s41550-020-1169-1.

Thompson, L. and Loewenstein, G. (1992) 'Egocentric interpretations of fairness and interpersonal conflict', *Organizational Behavior and Human Decision Processes*, 51(2), pp. 176–197. Available at: https://doi.org/10.1016/0749-5978(92)90010-5.

Trindade, B.C., Reed, P.M. and Characklis, G.W. (2019) 'Deeply uncertain pathways: Integrated multi-city regional water supply infrastructure investment and portfolio management', *Advances in Water Resources*, 134, p. 103442. Available at: https://doi.org/10.1016/j.advwatres.2019.103442.

Tsay, C.-J. and Bazerman, M.H. (2009) 'A Decision-Making Perspective to Negotiation: A Review of the Past and a Look to the Future', *Negotiation Journal*, 25(4), pp. 467–480. Available at: https://doi.org/10.1111/j.1571-9979.2009.00239.x.

UN General Assembly (2015) *Transforming our world: the 2030 Agenda for Sustainable Development | Department of Economic and Social Affairs*. New York. Available at: https://sdgs.un.org/2030agenda (Accessed: 11 January 2022).

Upper Colorado River Commission (2016) '2016 Upper Colorado River Basin depletion demand schedules'. Available at: http://www.ucrcommission.com/RepDoc/DepSchedules/CurFutDemandSchedule.pdf.

Vettigli, G. (2021) 'MiniSom: Minimalistic implementation of the Self Organizing Maps (SOM)'. Available at: https://github.com/JustGlowing/minisom (Accessed: 23 February 2022).

Wadoux, A.M.J. -C. and Brus, D.J. (2021) 'How to compare sampling designs for mapping?', *European Journal of Soil Science*, 72(1), pp. 35–46. Available at: https://doi.org/10.1111/ejss.12962.

Wadoux, A.M.J.-C., Brus, D.J. and Heuvelink, G.B.M. (2019) 'Sampling design optimization for soil mapping with random forest', *Geoderma*, 355, p. 113913. Available at: https://doi.org/10.1016/j.geoderma.2019.113913.

Watson, A.A. and Kasprzyk, J.R. (2017) 'Incorporating deeply uncertain factors into the many objective search process', *Environmental Modelling & Software*, 89, pp. 159–171. Available at: https://doi.org/10.1016/j.envsoft.2016.12.001.

Wehrens, R. and Buydens, L.M.C. (2007) 'Self- and Super-organizing Maps in R: The kohonen Package', *Journal of Statistical Software*, 21(1), pp. 1–19. Available at: https://doi.org/10.18637/jss.v021.i05.

Wehrens, R. and Kruisselbrink, J. (2019) 'kohonen: Supervised and Unsupervised Self-Organising Maps'. Available at: https://CRAN.R-project.org/package=kohonen (Accessed: 1 April 2022).

Wheeler, K.G. *et al.* (2018) 'Exploring Cooperative Transboundary River Management Strategies for the Eastern Nile Basin', *Water Resources Research*, 54(11), pp. 9224–9254. Available at: https://doi.org/10.1029/2017WR022149.

Wheeler, K.G. *et al.* (2022) 'What will it take to stabilize the Colorado River?', *Science*, 377(6604), pp. 373–375. Available at: https://doi.org/10.1126/science.abo4452.

Wickham, H., François, R., *et al.* (2023) 'dplyr: A Grammar of Data Manipulation'. Available at: https://CRAN.R-project.org/package=dplyr (Accessed: 21 February 2023).

Wickham, H., Chang, W., *et al.* (2023) 'ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics'. Available at: https://CRAN.R-project.org/package=ggplot2 (Accessed: 21 February 2023).

Wilson, G. *et al.* (2014) 'Best Practices for Scientific Computing', *PLOS Biology*, 12(1), p. e1001745. Available at: https://doi.org/10.1371/journal.pbio.1001745.

Woodruff, M.J., Reed, P.M. and Simpson, T.W. (2013) 'Many objective visual analytics: rethinking the design of complex engineered systems', *Structural and Multidisciplinary Optimization*, 48(1), pp. 201–219. Available at: https://doi.org/10.1007/s00158-013-0891-z.

Worsham, L. *et al.* (2012) 'A Comparison of Three Field Sampling Methods to Estimate Soil Carbon Content', *Forest Science*, 58(5), pp. 513–522. Available at: https://doi.org/10.5849/forsci.11-084.

Yang, T. (2019) 'The Emergence of the Environmental Impact Assessment Duty as a Global Legal Norm and General Principle of Law', *HASTINGS LAW JOURNAL*, 70(2). Available at: https://www.hastingslawjournal.org/the-emergence-of-the-environmental-impact-assessment-duty-as-a-global-legal-norm-and-general-principle-of-law/ (Accessed: 3 October 2023).

Yang, T. (2023) 'NEPA's Conquest of the World', *Natural Resources & Environment*, 37(4), pp. 29–35. Available at: https://research.ebsco.com/c/3czfwv/viewer/html/vxj7kineu5 (Accessed: 3 October 2023).

Yarlagadda, B. *et al.* (2023) 'Trade and Climate Mitigation Interactions Create Agro-Economic Opportunities With Social and Environmental Trade-Offs in Latin America and the Caribbean', *Earth's Future*, 11(4), p. e2022EF003063. Available at: https://doi.org/10.1029/2022EF003063.

Zagona, E.A. *et al.* (2001) 'Riverware: A Generalized Tool for Complex Reservoir System Modeling1', *JAWRA Journal of the American Water Resources Association*, 37(4), pp. 913–929. Available at: https://doi.org/10.1111/j.1752-1688.2001.tb05522.x.

Zatarain Salazar, J., Castelletti, A. and Giuliani, M. (2022) 'Multi-Objective Robust Planning Tools', in *Oxford Research Encyclopedia of Environmental Science*. Available at: https://doi.org/10.1093/acrefore/9780199389414.013.626.

Zeff, H.B. *et al.* (2014) 'Navigating financial and supply reliability tradeoffs in regional drought management portfolios', *Water Resources Research*, 50(6), pp. 4906–4923. Available at: https://doi.org/10.1002/2013WR015126.

Zeff, H.B. *et al.* (2016) 'Cooperative drought adaptation: Integrating infrastructure development, conservation, and water transfers into adaptive policy pathways: COOPERATION THROUGH INTEGRATED ADAPTIVE PATHWAYS', *Water Resources Research*, 52(9), pp. 7327–7346. Available at: https://doi.org/10.1002/2016WR018771.

Zeleny, M. (1989) 'Cognitive Equilibirum: A New Paradigm of Decision Making?', *Human Systems Management*, 8(3), pp. 185–188. Available at: https://doi.org/10.3233/HSM-1989-8301.

Zhang, S. *et al.* (2018) 'Visualization and Data Mining of Multi-Objective Electric Machine Optimizations with Self-Organizing Maps: A Case Study on Switched Reluctance Machines', in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*. *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*, Portland, OR: IEEE, pp. 4296–4302. Available at: https://doi.org/10.1109/ECCE.2018.8558399.

A. Supplementary Material for Subsampling and Space-filling Metrics to Test Ensemble Size for Robustness Analysis with a Demonstration in the Colorado River Basin

A.1. Streamflow features used as inputs to cLHS

To subsample from the existing 500 scenarios, we calculated 4 streamflow features for every streamflow time series. These 4 features were then used as inputs to cLHS (in addition to Lake Mead pool elevation, Lake Powell pool elevation, and demand). The 4 features are the average of the driest 20-year period, wettest 20-year period, driest 2-year period, and wettest 2-year period. The two features that capture the driest periods are calculated as $argmin(AVG_W)$, where AVG_W is the moving window average of the previous W years (either 2 or 20 years). The features that capture the wettest periods are calculated as $argmax(AVG_W)$. The figure below shows an example calculation of the driest and wettest 2-year averages. The annual flow, in Million-Acre Feet (MAF), is shown in the 'Q (MAF)' column, and the trailing 2-year average is in the '2yr AVG' column. The wettest and driest 2-year periods are highlighted in blue and red, respectively.

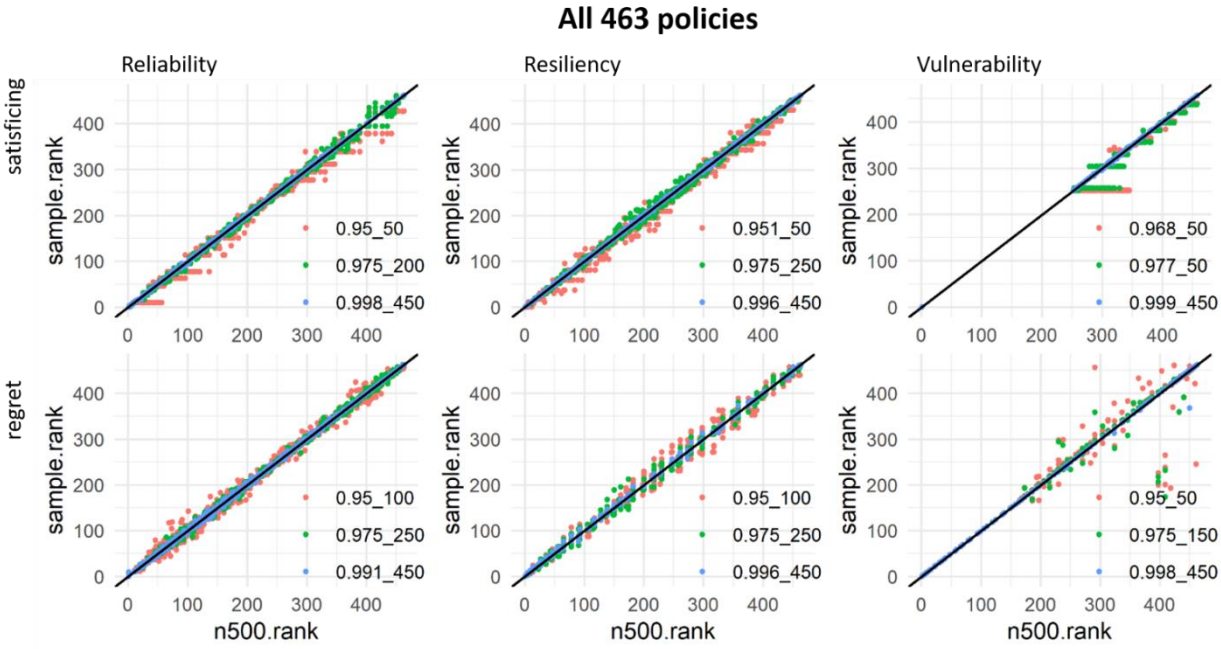
2-yr moving window

Year	Q (MAF)	2yr AVG	
1	18.7	NA	
2	20.9	19.8	wettest 2 year period
3	11.7	16.3	
4	22.2	17.0	
5	14.6	18.4	
6	15.7	15.1	driest 2 year period
7	18.6	17.1	
8	14.5	16.6	
9	21.4	17.9	
10	13.6	17.5	

A.2. Rank diagrams illustrating why we chose 0.975 correlation as 'accurate' threshold

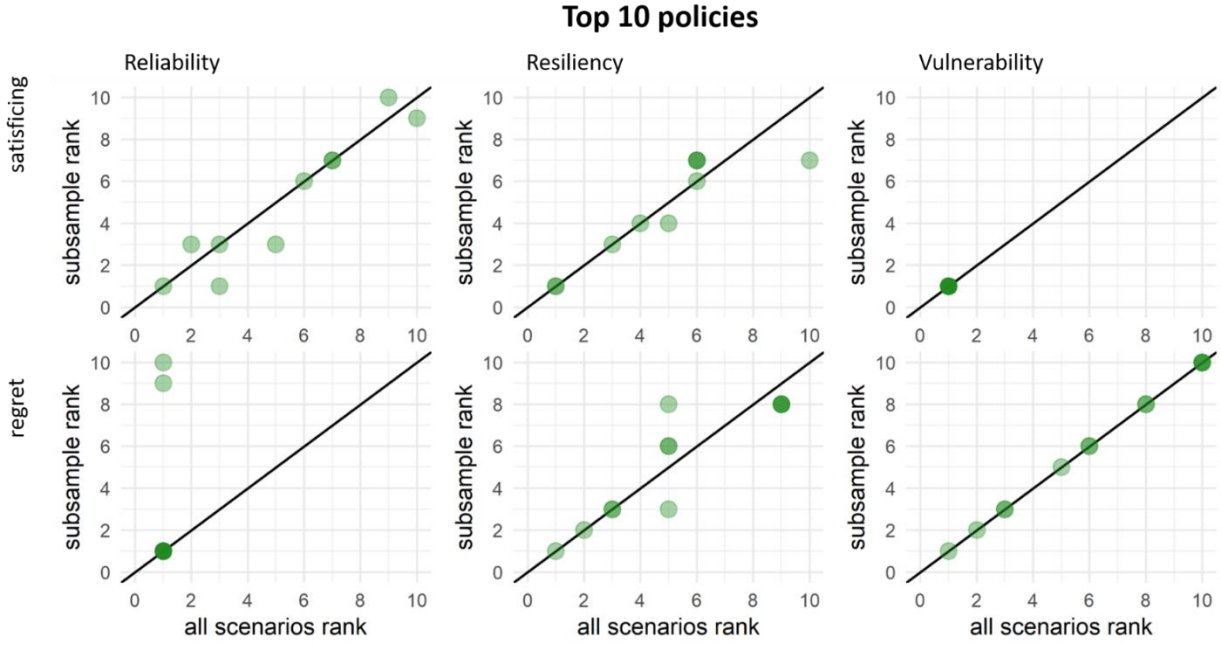
We defined 'accurate ranking' as scenario sets that achieved a rank correlation of 0.975 or greater. This decision is based upon visual inspection of scatter plots of subsample rank (y axis) vs all scenarios rank (x axis, labelled n500.rank), as shown in the figure below. There are six plots, one per robustness

metric, where the top row shows satisficing- type metrics, the bottom row is regret from best, and the columns are the three performance objectives. For each robustness metric, we selected three representative scenario sets with correlations closest to 0.95, 0.975, and 1.00 (maximum rank correlation). These are shown in red, green, and blue, respectively. The legend of each plot uses the format correlation_number-scenarios, e.g., 0.95_50 in the *reliability.satisficing* legend (top left plot) means the scenario set achieved a rank correlation of 0.95, and the scenario set has 50 scenarios in it. Policies that are ranked identically between the subsample and all scenarios are located on the black, 1-to-1 line, and the orthogonal distance between a point and the line is the number of positions by which the policy is misranked.



From this plot, we observed that scenario sets with correlation of 0.95 can have many policies that are incorrectly tied, especially for the satisficing metrics. This is shown by horizontal runs of red points. For example, we observe several such ties in the *reliability.satisficing*, *resiliency.satisficing*, and *vulnerability.satisficing* plots.

In contrast, these erroneous ties are resolved for the scenario sets with rank correlations of 0.975 and above (i.e., we do not observe horizontal runs of green or blue points). Some exceptions are seen in the *vulnerability.satisficing* plot, which shows horizontal runs at roughly rank 250 and 300. However, these ties are not very interesting, since they occur for policies that are not ranked near the top.



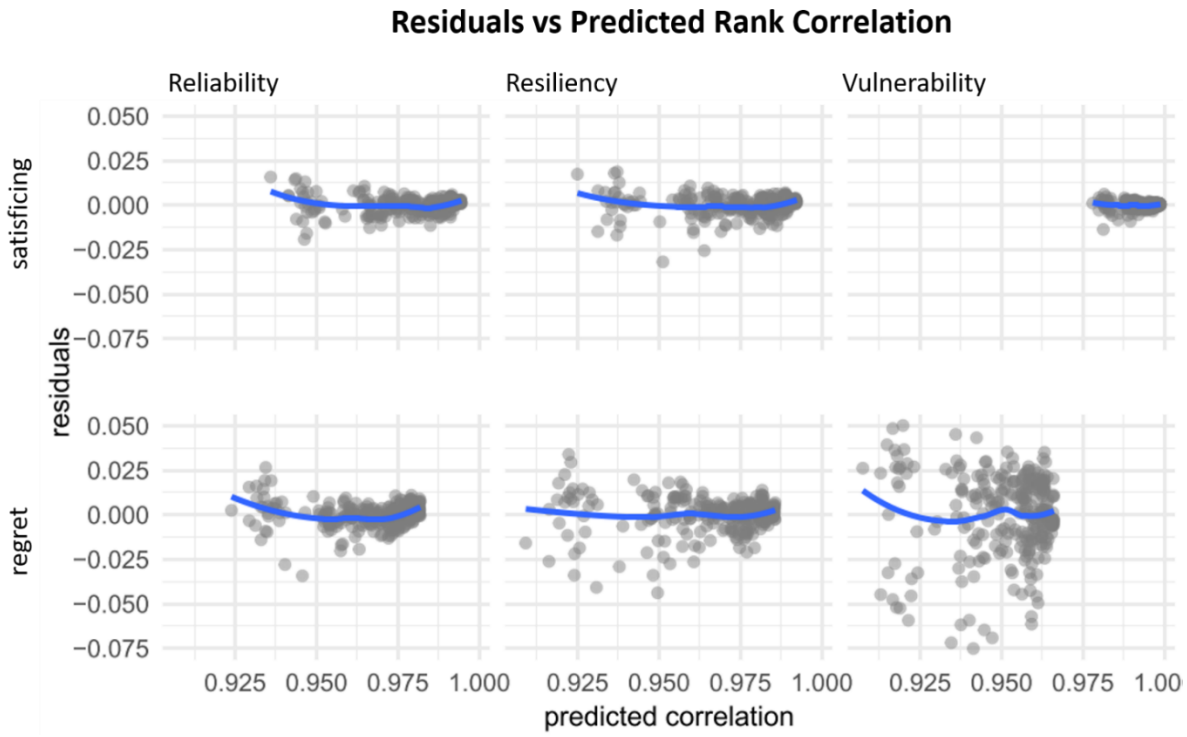
We are most concerned with how accurately a scenario set ranks the most robust policies, as these most robust policies would ideally be prioritized by decision-makers. The plot below zooms in on the plot above to the top 10 policies, showing only the scenario set with rank correlation of 0.975. The dots are plotted with a transparency such that policies with the same x-y coordinates (e.g., they are correctly tied) are darker green. These plots reveal that our selected scenario sets with correlation of 0.975 accurately identify the most robust policy (rank 1 is located on the diagonal line). Moreover, the subsampled scenario sets agree with all scenarios on which policies belong in the top 10. Lastly, policies in the top 10 are correctly ranked and/or misranked by only 1-3 positions (the orthogonal distance from a point to the line is 1-3 positions). There are few exceptions, which are caused by ties. Consider *reliability.regret* (bottom-left), for example. The dark green point at position 1-1 indicates multiple policies

are correctly tied for most robust. Policies ranked 9 and 10 using the subsample are ranked 1 using all scenarios. Although the vertical distance suggests the ranking is wrong by 8 plus positions, this is simply because 8 policies are given rank 1, then the next most robust policy is given rank 9. The selected scenario sets provide a useful and accurate description of which policies are most robust; therefore, we define accurate ranking as any scenario set that meets or exceeds this 0.975 correlation threshold.

A.3. Residual plots of rank correlation vs MSTmean linear models

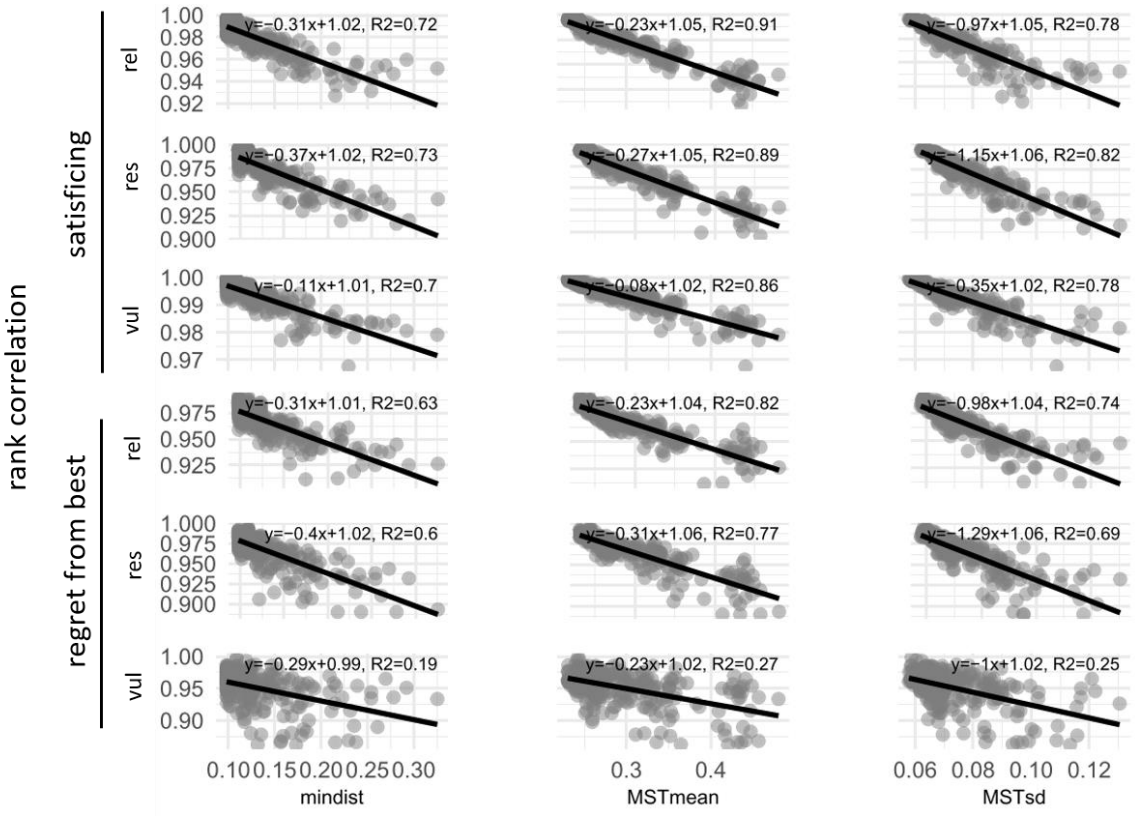
For the space-filling metric with highest R^2 (MSTmean), we checked that the residuals had approximately a mean of 0 and that the errors were homoscedastic. These are assumptions of linear models upon which prediction intervals are calculated. Residuals vs. predicted correlation are shown in the figure below, each subplot being for one robustness metric. As before, the rows indicate the robustness type, and columns indicate objective type. Each point corresponds to one subsampled scenario set, the y axis is the residual between the scenario set's actual rank correlation and the linear model's predicted correlation, and the x axis is the predicted correlation. The blue line is a loess fit to highlight any trends in the residuals. The blue line shows that the average residual is approximately zero for each model, and there is no strong trend as a function of the predicted correlation. We note that the residuals can be

slightly larger for smaller predicted correlations; however, most residuals are of similar magnitude regardless of the predicted correlation (i.e., they are homoscedastic).



A.4. Linear models using mindist and MSTsd as predictors

In the main text, we presented the linear models that use MSTmean because it achieved higher R^2 than mindist and MSTsd, as shown in the figure below. The columns are for mindist, MSTmean, and MSTsd, respectively, and each row is a robustness metric. In the subplots, each point is a scenario set, the y axis is rank correlation, and the x-axis is the value of the space-filling metric. The line shows the fitted linear model, and each model's R^2 is reported. For every robustness metric, the highest R^2 is obtained by the MSTmean model, followed by MSTsd and mindist, respectively.


rank correlation
satisficing
regret from best
rel
res
vul
y=−0.31x+1.02, R2=0.72
y=−0.23x+1.05, R2=0.91
y=−0.97x+1.05, R2=0.78
y=−0.37x+1.02, R2=0.73
y=−0.27x+1.05, R2=0.89
y=−1.15x+1.06, R2=0.82
y=−0.11x+1.01, R2=0.7
y=−0.08x+1.02, R2=0.86
y=−0.35x+1.02, R2=0.78
y=−0.31x+1.01, R2=0.63
y=−0.23x+1.04, R2=0.82
y=−0.98x+1.04, R2=0.74
y=−0.4x+1.02, R2=0.6
y=−0.31x+1.06, R2=0.77
y=−1.29x+1.06, R2=0.69
y=−0.29x+0.99, R2=0.19
y=−0.23x+1.02, R2=0.27
y=−1x+1.02, R2=0.25
mindist
MSTmean
MSTsd

B. Supplementary Material for post-MORDM: mapping policies to synthesize optimization and robustness results for decision-maker compromise

B.1. The SOM batch update function

The SOM update function is iterated over every neuron until the neurons stabilize. The governing equation for the batch version of the update function is

Eq. (A.1)
$$m_j = \frac{\sum w_k v_k}{\sum w_k}$$

where $\boldsymbol{m}_j$ is the (updated) prototype vector of neuron j, k is an index for each data point whose BMU is within the neighborhood of neuron j, $\boldsymbol{v}_k$ is data point k as defined by its vector of feature values, and w is a weight (Hastie, Tibshirani and Friedman, 2009, chap. 14.4; Kohonen, 2013). The weight applied to every data point can be 1, or it can decrease with increasing map distance between the data point's BMU and $\boldsymbol{m}_j$ according to a user-defined neighborhood function shape. Effectively, all data points within a neuron's neighborhood contribute to the updated neuron's prototype vector, and closer data points contribute more (in the case where the weight is not constant).

We tersely describe several hyperparameters that the user must define when using the batch update function.

Neighborhood radius: the neighborhood radius defines how many neurons are considered to be in the neighborhood of a neuron and thus affect the neuron's updated prototype vector. The radius is measured in two-dimensional map space. A smaller radius tends to result in better QE, whereas a larger radius results in better TE.

Neighborhood function (also called neighborhood shape)**:** Neighborhood function determines the value of the weight, w_k, as a function of the map distance between the neuron being updated and the BMU of v_k. Common neighborhood functions include bubble (also called uniform), Gaussian, and parabolic.

Distance function: The distance function determines how distance is measured in data space, and thus controls the assignment of data points to BMUs. Common distance functions include Euclidean, Manhattan, and sum of squared distance.

Edge neuron behavior: Neurons on the outer edges of the SOM have less neighbors than other neurons, thus they can be unequally 'pulled' by their neighbors into the middle of the SOM. Thus, edge behavior can be defined as toroidal (as opposed to planar), connecting the neurons on the left and right (and bottom/top) in a torus shape to avoid this inward-pull effect.

For further details, we refer the reader to (Clark, Sisson and Sharma, 2020).

B.2. SOM quality metrics: percent of variance explained and topographic error

The number of neurons and hyperparameter set are evaluated by percent of variance explained (PVE) and topographic error (TE). PVE quantifies how well the SOM neurons represent the input data. PVE is calculated from quantization error, scaled 0 to 100% using the total variance of the data set, according to the equation

Eq. (A.2.1) $$percent\ variance\ explained = 100 - \frac{100 * q_e}{\sigma^2}$$

where σ^2 is the total variance of the data and q_e is the quantization error calculated as

Eq. (A.2.1) $$q_e = \frac{1}{P}\sum_{p=1}^{P} \quad dist(m_c, v_p)$$

where *P* is the total number of data points, *p* is the data point index, $\boldsymbol{v}_p$ is the feature vector of data point *p*, and $\boldsymbol{m}_c$ is the vector of the closest neuron to $\boldsymbol{v}_p$ (Clark, Sisson and Sharma, 2020; Boelaert *et al.*, 2021). dist() is the distance function, which can be Euclidean, Manhattan, or sum of squares. The range of percent variance explained is 0 to 100%, with 100% percent being ideal.

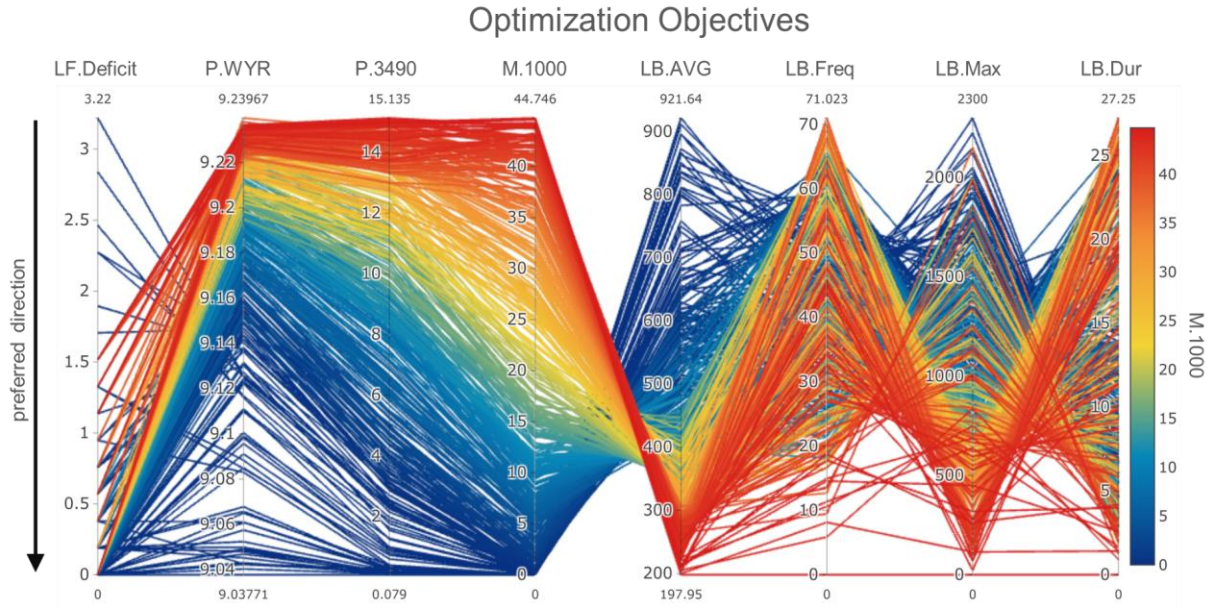
TE measures how well the mapping of data onto the SOM preserves the data's topologic patterns. TE is calculated according to

Eq. (A.2.2) $$topographic\ error = \frac{1}{P}\sum_{p=1}^{P} \quad u_{v_p}$$

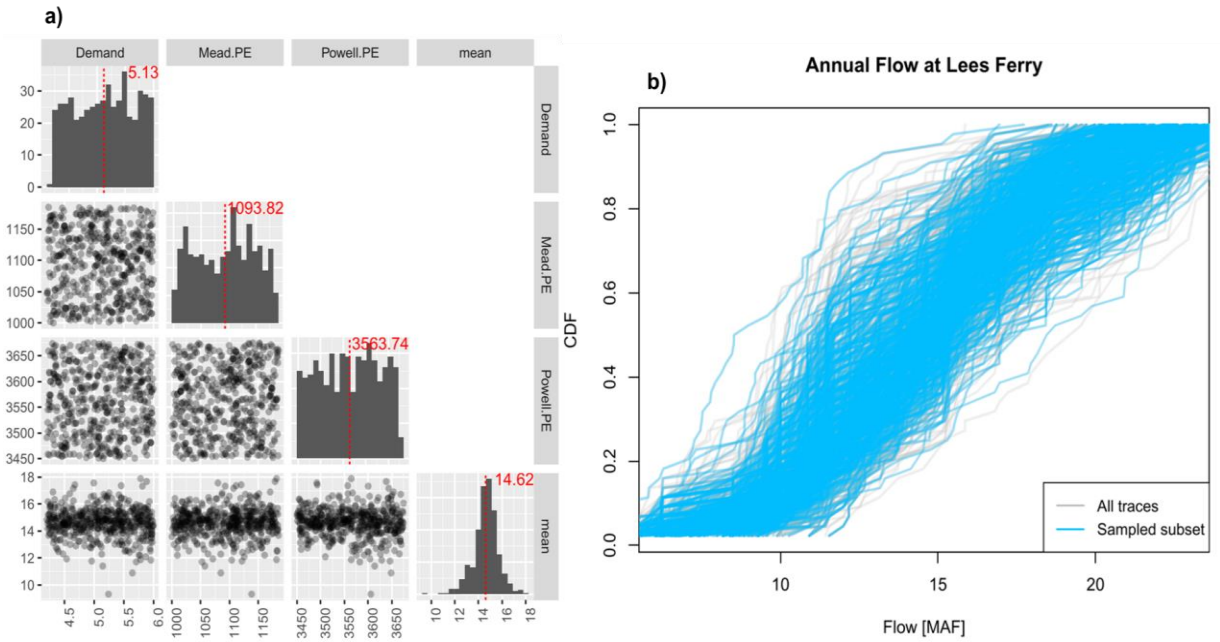
where u_{v_p} = 0 if the first and second closest neuron to data point $\boldsymbol{v}_p$ according to the distance function are not adjacent neighbors in the SOM (i.e, an instance of topographic error), and 1 otherwise. Topographic error ranges from 0 to 1, where 0 means no instances of topographic error occur, and 1 means every data point is characterized by a topographic error. TE and PVE typically conflict, where a map with more neurons results in better representation of the data points at the expense of topologic preservation (Clark, Sisson and Sharma, 2020). Therefore, both metrics are considered when selecting the number of neurons and hyperparameter set.

B.3. Parallel coordinates plot of Lake Mead polices from MOEA-optimization

Section 4.4.3.1 uses radar plots and a SOM topology map to visualize the tradeoffs that policies exhibit with respect to optimization objectives. Because parallel axis plots are commonly used for exploring the tradeoffs of a non-dominated policy set, we have provided this common visualization here. This plot shows the optimization objectives of 463 Lake Mead operation policies created with the Borg MOEA. Each parallel axis is an objective, and each colored trace is a policy. Policies are colored by M.1000. The preferred direction is downward for all axes. For a description of the objectives and units, see Table 4-2.



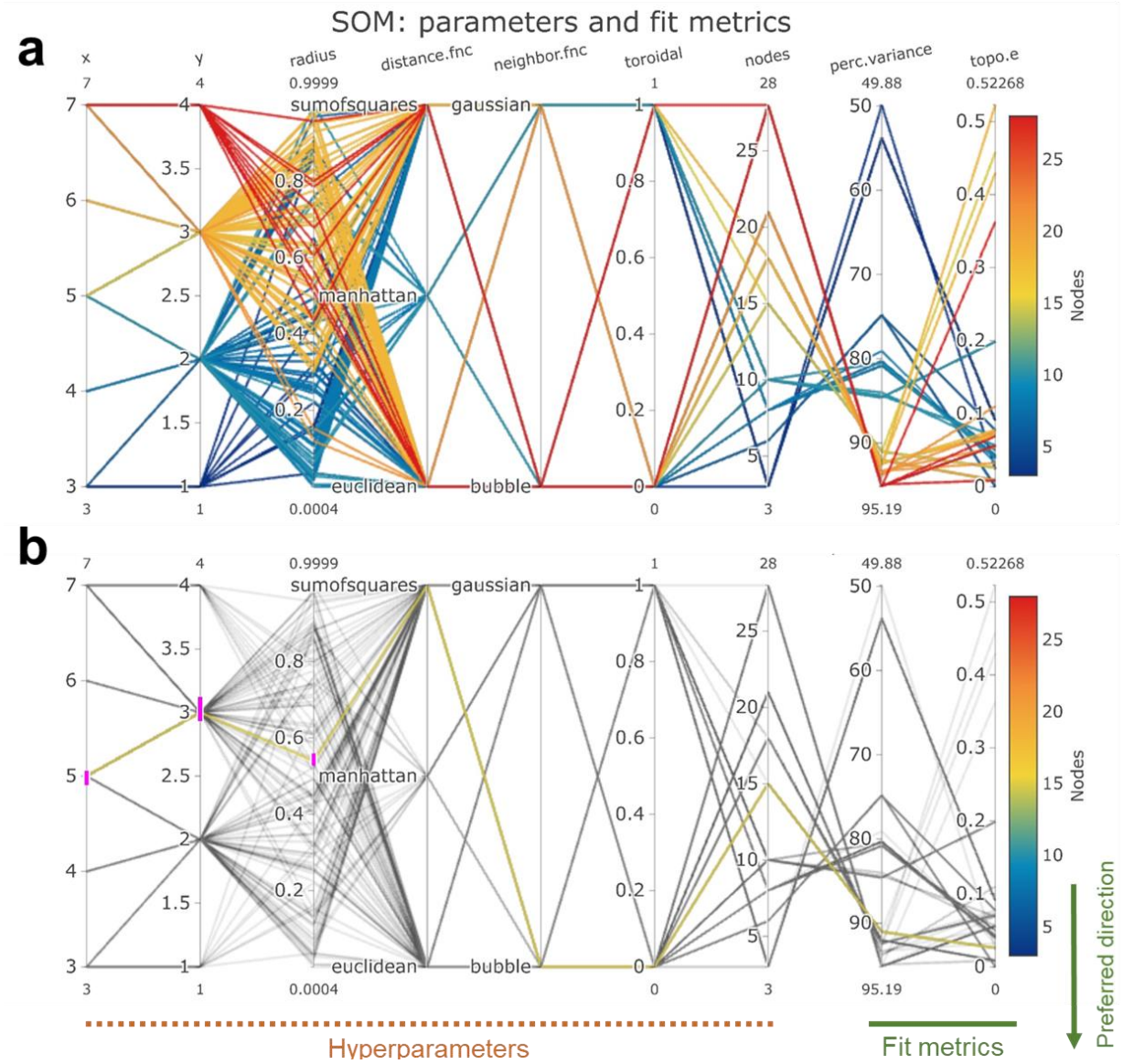
B.4. 500-member State of the World (SOW) ensemble sampled with conditioned Latin Hypercube Sampling (cLHS)



a) Scatter-matrix of uncertainty metrics sampled in the SOW ensemble. Demand is Upper Basin annual depletion in million-acre feet (MAF) held constant over the simulation. Mead.PE and Powell.PE are the pool elevations at Lakes Mead, respectively, at the beginning of the simulation in feet above mean sea level. Mean is the annual cumulative natural flow averaged over the 44-year simulation at Lees Ferry, Arizona, in MAF. The lower triangle shows values for each SOW, and the diagonal shows a histogram with the average in red.

b) The Cumulative Distribution Function (CDF) of hydrology traces used in the SOW ensemble. The plot shows the CDF of the annual cumulative natural flow measured at Lees Ferry, Arizona. Blue traces are those sampled via cLHS for use in the SOW ensemble. Light gray traces are traces in the cumulative 1963 traces contained in the Observed Resampled, GCM, Paleo Resampled, and Paleo Conditioned ensembles but not included in the SOW ensemble.

B.5. Grid search of SOM size and hyperparameters



a) To select the SOM size and hyperparameters in section 4.4, we trained SOM on the optimization layer using 1000 parameter sets derived from Latin Hypercube Sampling. The hyperparameters include: 1) neighborhood radius from 0 – 1, measured as the fraction of total two-dimensional map distance, 2) distance function, including Euclidean, Manhattan, and sum of squares, 3) neighborhood function, either Gaussian or bubble, 4) edge behavior, either toroidal (1) or planar (0), 5) total number of neurons, from 3-28 (neuron axis is labeled as 'nodes').

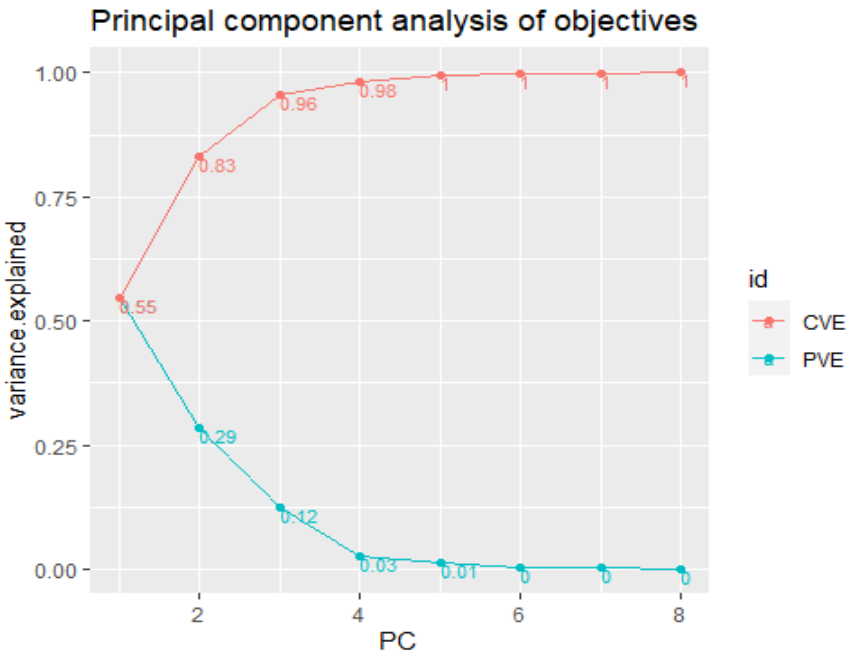
The range of neurons was informed by calculating the Davies-Bouldin index (DB) for k-means clusters of k = 2 to 100 clusters. DB measures cluster performance, where large separation between different clusters and small separation within a cluster is preferred (Xiao, Lu and Li, 2017). Because SOM is a constrained form of k-means clustering that collapses to k-means clustering at the end of the training procedure, we used DB to identify a reasonable range of neurons to test (Clark, Sisson and Sharma, 2020). We found DB to be smallest at k=13 except where k exceeded 80, but only offering a small improvement. Note that the ratio of neurons in the x to y dimensions are calculated from the total number of neurons by setting the ratio equal to the ratio of the first and second eigenvalues of the objective layer according to the recommendations of (Kohonen, 2001; Clark, Sisson and Sharma, 2020). For more information on the eigenvalues calculated in this case study, see part c below.

Davies-Bouldin index was calculated in R using the clusterSIM package (R Core Team, 2021; Walesiak and Dudek, 2021). SOM training was performed with the kohonen package, and fit metrics were calculated using the aweSOM package (Kruisselbrink, 2019; Boelaert *et al.*, 2021; R Core Team, 2021). To reduce the number of parameter sets under consideration, we applied a non-domination filter considering minimization of the number of neurons, maximization of percent variance explained, and minimization of topographic error. We performed this task with the ecr package (Bossek *et al.*, 2017).

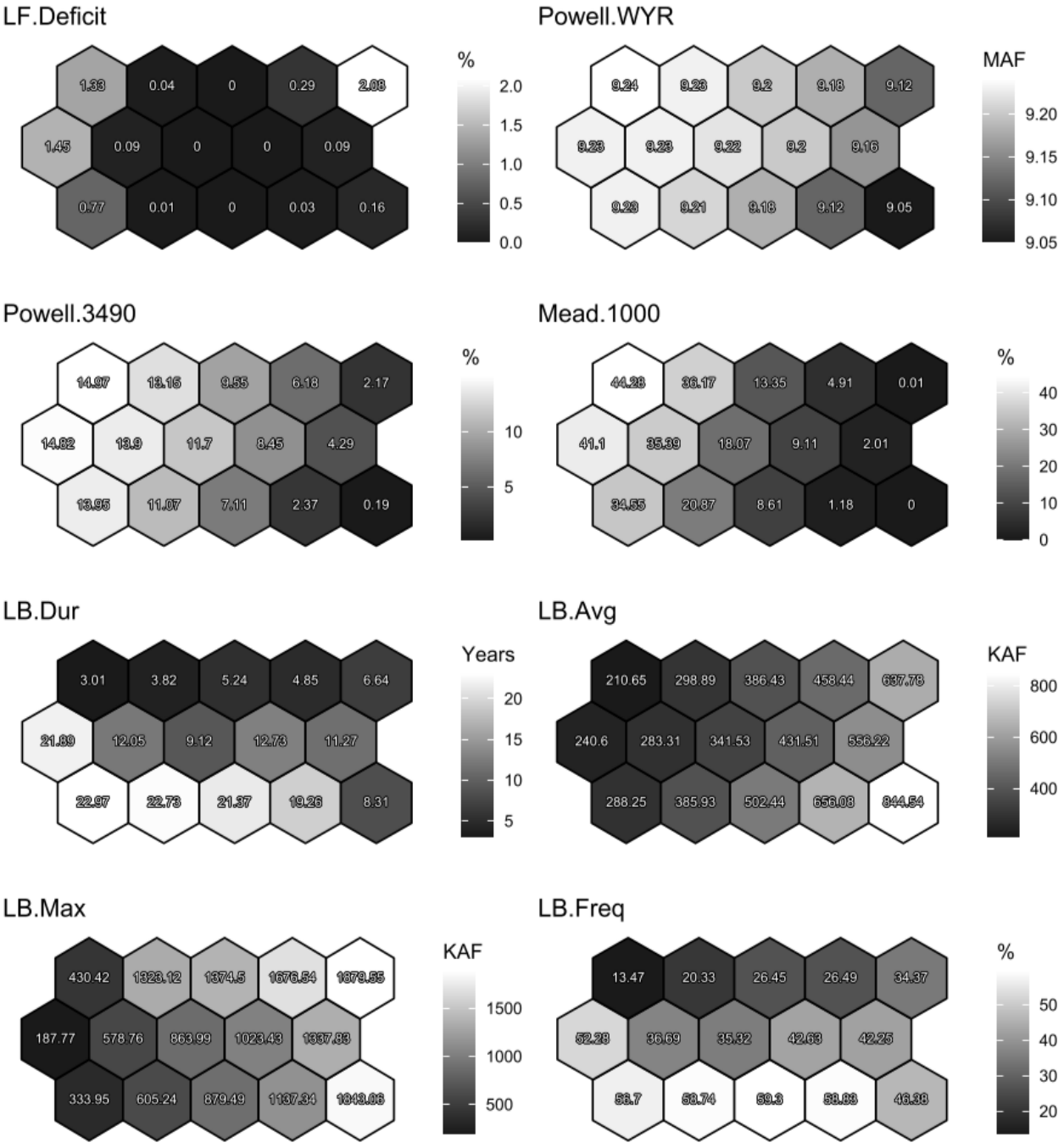
b) We used the interactive brushing features of plotly parallel coordinate plots to select the final parameter set, highlighted in yellow (Sievert *et al.*, 2021). The SOM consists of 15 neurons, 5 in the x direction and 3 in the y direction. Larger maps had relatively small improvements in TE or PVE, whereas smaller maps had relatively large decrease in performance. Several parameter sets with identical x and y dimensions, distance function (sum of squares), neighborhood function (Gaussian), and edge behavior (planar, 0) but slightly different neighborhood sizes resulted in the same SOM and thus the same PVE and TE. We chose the parameter set with the smallest neighborhood distance.

All code required to create a SOM, including the steps discussed in Section 4.3.1.1 and a grid search of SOM size and hyperparameters, are available on GitHub: https://github.com/nabocrb/post-MORDM

c) We have included the variance explained by each principal component (PC) in the figure below. The proportion of variance explained (PVE) by each PC is calculated as the PC's eigenvalue divided by the sum of all eigenvalues, since an eigenvalue is the variance of a principal component (James *et al.*, 2013, chap. 10.2). As indicated in the figure, PC1 and PC2 explain 83% of the cumulative variance (cumulative variance explained, or CVE). Thus, in our case study, arranging the SOM along PC1 and PC2 is especially helpful for visualizing tradeoffs and supporting negotiation. However, it is possible that PC1 and PC2 will explain less CVE in other applications. In this case, aligning the SOM along PC1 and PC2 is still valid, but the user may find that individual neurons, or neighborhoods of neurons, capture other patterns of the feature space (such as non-linear patterns or patterns described by PC3, for example). We could imagine a case where a DM may be more interested in the tradeoffs represented by, say, PC1 and PC3. Then, the SOM could be initialized along PC1 and PC3, instead.

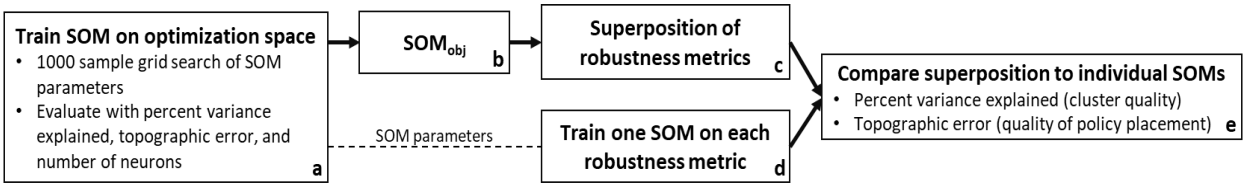

Principal component analysis of objectives
variance.explained
1.00
0.75
0.50
0.25
0.00
0.55
0.83
0.96
0.98
1
1
1
1
0.29
0.12
0.03
0.01
0
0
0
2
4
6
8
PC
id
CVE
PVE

B.6. Component planes of performance objectives



Component planes are topology maps that show one dimension per map, plotting the average value per neuron. In this figure, each component plane shows the objective value averaged over the policies per neuron. Component planes are helpful for visualizing the patterns of one objective in the topology map.

B.7. Method for testing quality of robustness superposition method



Because this research presents the first superposition of robustness metrics onto a SOM trained to to the optimization layer, we performed an experiment to affirm the quality of the resulting robustness topology maps. First, we establish our SOM using the parameter set used in Section 4.4, described in Appendix B.5. This establishes the assignment of each Lake Mead policy to a neuron. We then superposition three robustness metrics, mean, 90% maximin, and 90% regret from best, creating three robustness topology maps (Appendix B.11 – B.13, below). For descriptions and example calculations, see B.8-B.9. For a baseline to compare the superposition method to, we also train a new SOM for each robustness metric, using the same SOM parameters (d). Finally, we compare the percent variance explained and topographic error. The results are summarized in Figure B.10.

B.8. Description of robustness metrics tested

Robustness metric descriptions		
Name	Definition	Interpretation
Mean (Laplace's Principle of Insufficient Reason)	The performance averaged over the SOW ensemble.	The units and interpretation of each performance objective are maintained (e.g. smaller values are desired for minimization objectives)
90^{th} percentile maximin	The 90^{th} percentile worst performance obtained by a solution in the SOW ensemble.	The units and interpretation of each performance objective are maintained (e.g. smaller values are desired for minimization objectives)
90^{th} percentile regret from best	The 90^{th} percentile deviation of a policy from the best performing policy in each SOW.	0 is ideal, meaning the policy was the best performing in every SOW. Larger values indicate greater regret. Units are maintained.

B.9. Example calculations for the tested robustness metrics

Mean

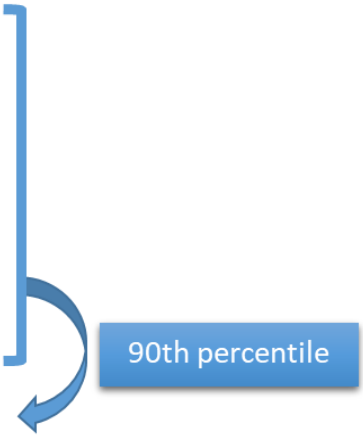
Data			Calculations
Policy ID	SOW	LB.AVG(KAF)	mean
1	1	16	244
	2	155	
	3	0	
	4	839	
	5	64	
	6	300	
	7	107	
	8	644	
	9	154	
	10	164	
		mean	244

90th percentile maximin

Data			Calculations
Policy ID	SOW	LB.AVG (KAF)	90th percentile
1	1	16	664
	2	155	
	3	0	
	4	839	
	5	64	
	6	300	
	7	107	
	8	644	
	9	154	
	10	164	
		90th % maximin	664

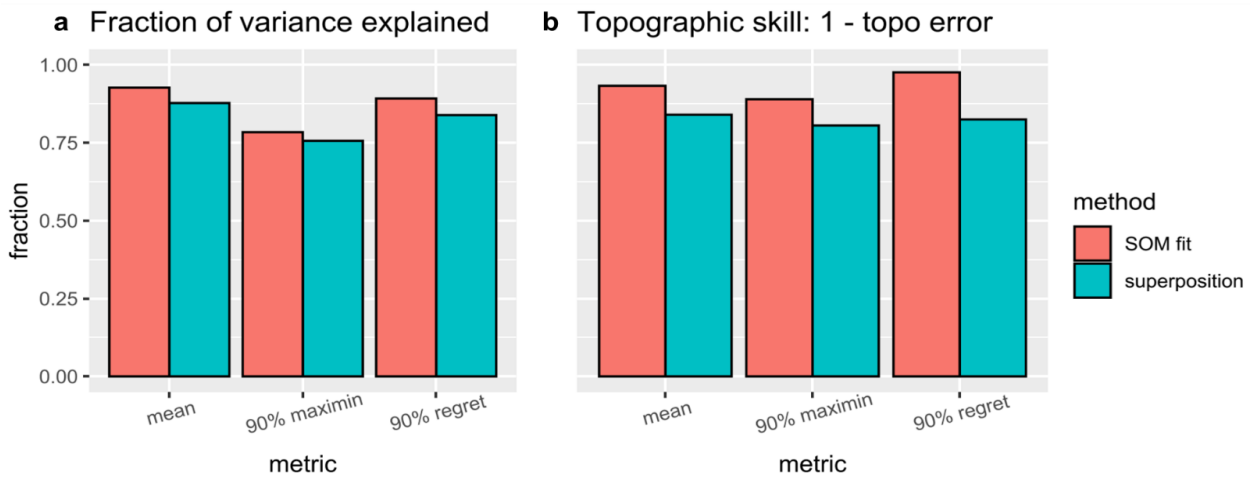
90th percentile regret from best

Data				Calculations (ID: 1)	
SOW	LB.AVG (KAF)			Best	sol'n - best
	Policy ID: 1	Policy ID: 2	Policy ID: 3		
1	16	0	0	0	16
2	155	291	127	127	28
3	0	0	0	0	0
4	839	835	812	812	27
5	64	162	32	32	32
6	300	551	302	300	0
7	107	227	95	95	11
8	644	1036	670	644	0
9	154	291	143	143	11
10	164	324	127	127	37
			90th% regret from best		32.3



Each example calculation is for a hypothetical policy 1 that was simulated in 10 SOW. We perform the calculation using the LB.AVG objective. 90th percentile regret from best requires data for other policies, so we include data for hypothetical policies 2 and 3. The calculation first involves finding the best performance in each SOW, then subtracting this value from the performance of policy 1. The final value is obtained by taking the 90th percentile.

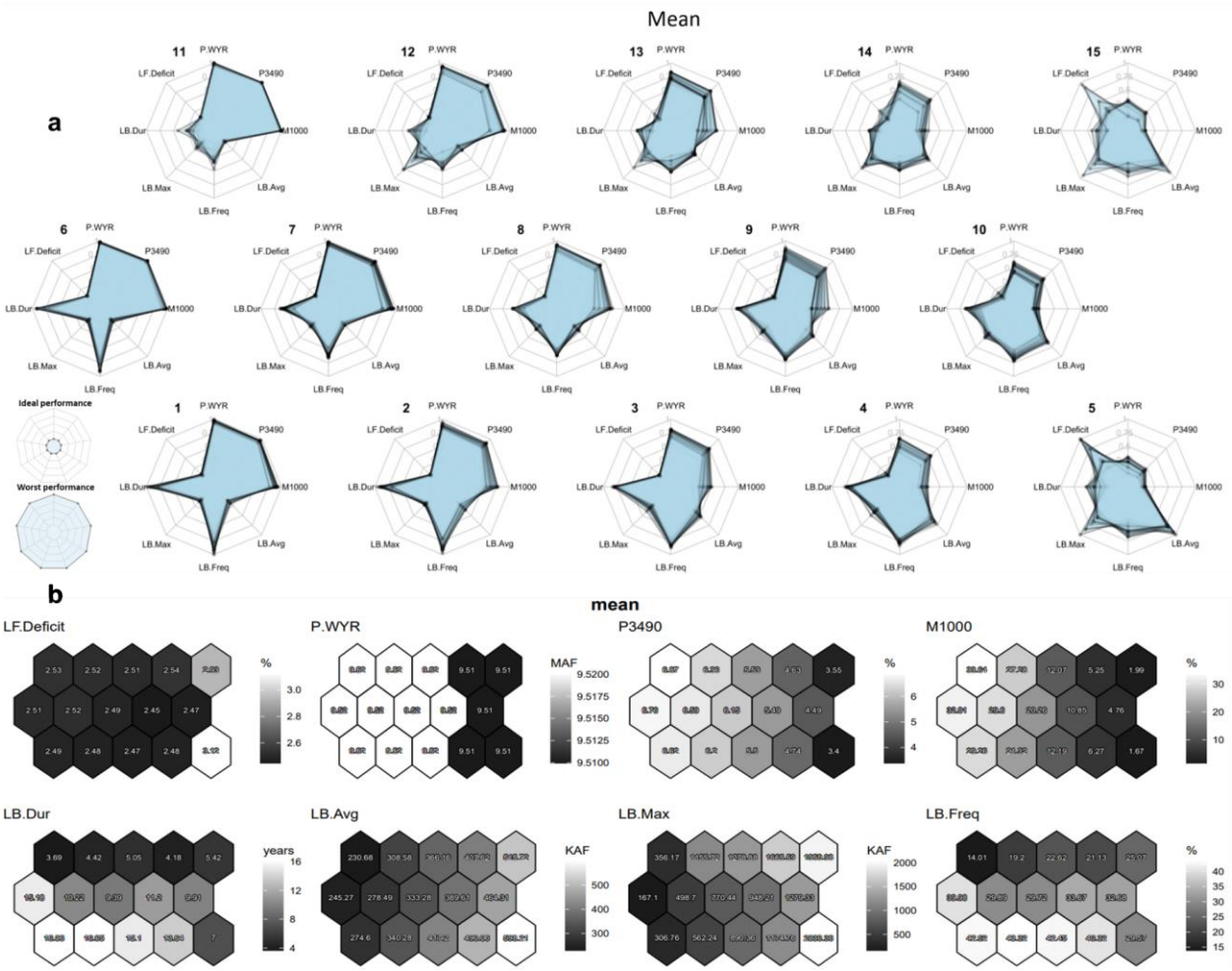
B.10. Skill of robustness metric superposition method



a) Fraction of variance explained of training a new a new SOM to each robustness metric (salmon color) compared to the robustness superposition method (blue). Fraction of variance explained for both methods exceeds 0.75. Note that fraction of variance explained is the same as percent of variance explained in A2 divided by 100.

b) Topographic skill of training a new SOM to each metric compared to the superposition method. We plot topographic skill, 1 – topographic error, such that up is the desired direction for both plots a and b. Topographic skill exceeds 0.75 for both methods. We conclude robustness superposition attains satisfactory skill for implementation in the post-MORDM framework.

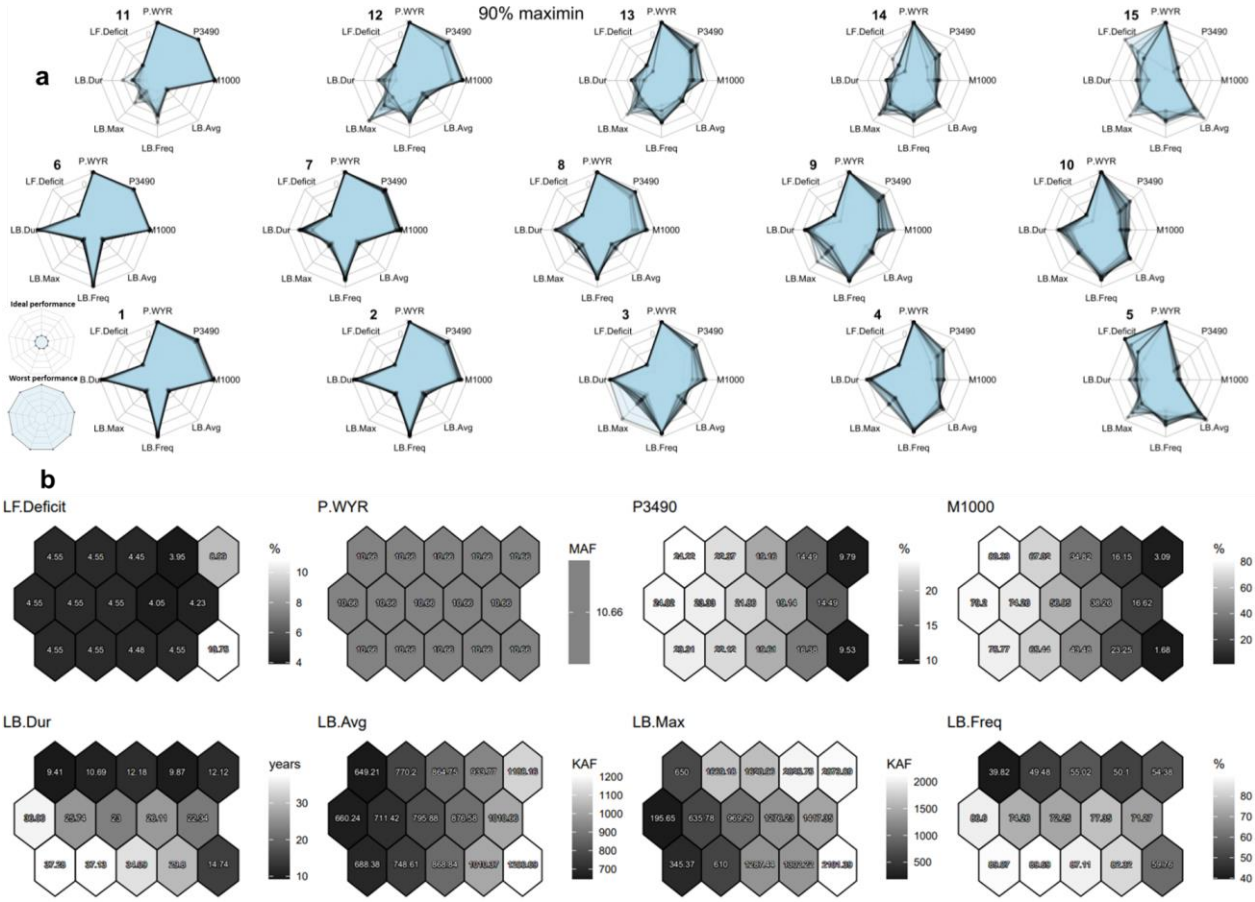
B.11. Superposition results for mean (Laplace's Principle of Insufficient Reason)



a) The resulting radar plot topology map of the superposition method applied to the mean robustness metric.

b) The robustness component planes of each objective, showing the value averaged over policies in each neuron.

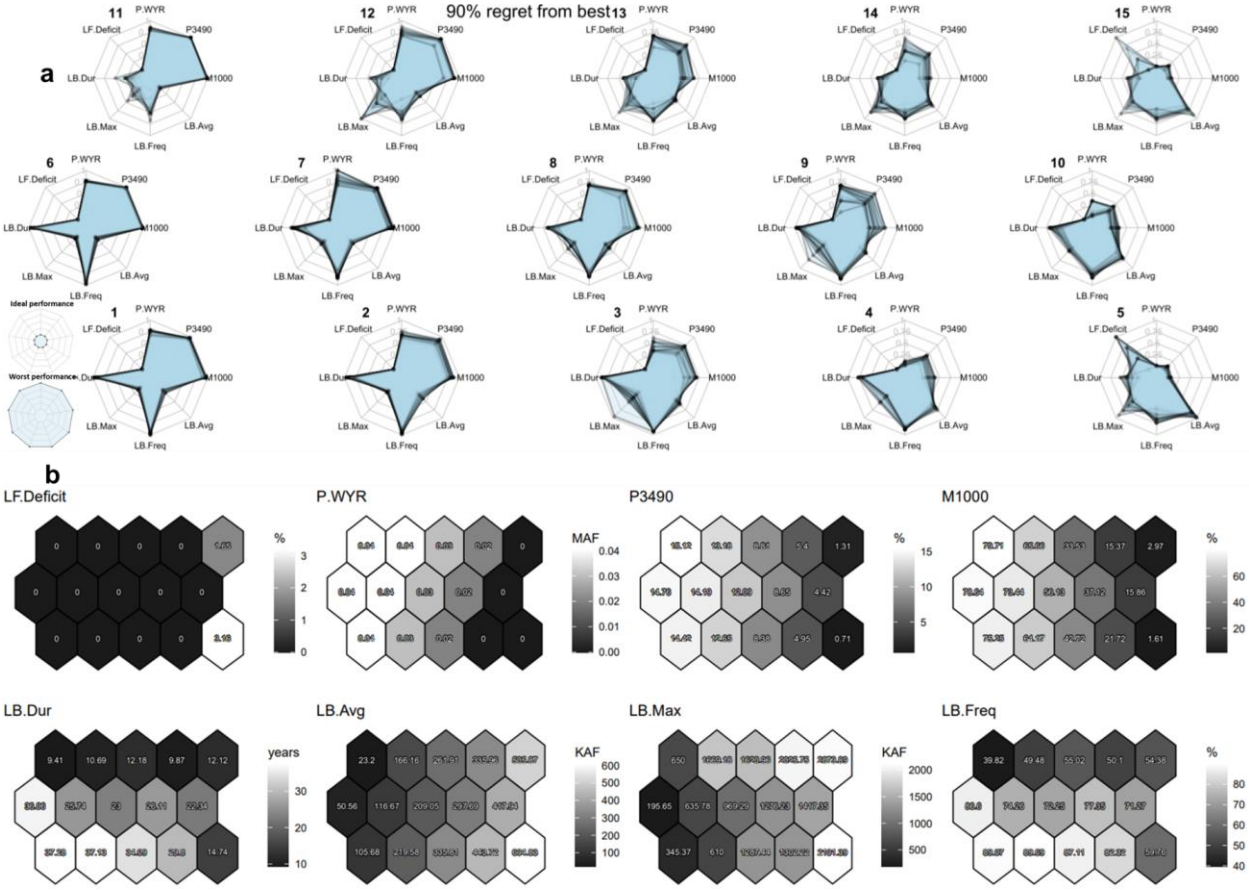
B.12. Superposition results for 90th percentile maximin



a) The resulting radar plot topology map of the superposition method applied to the 90th percentile maximum robustness metric.

b) 90th percentile maximin component planes.

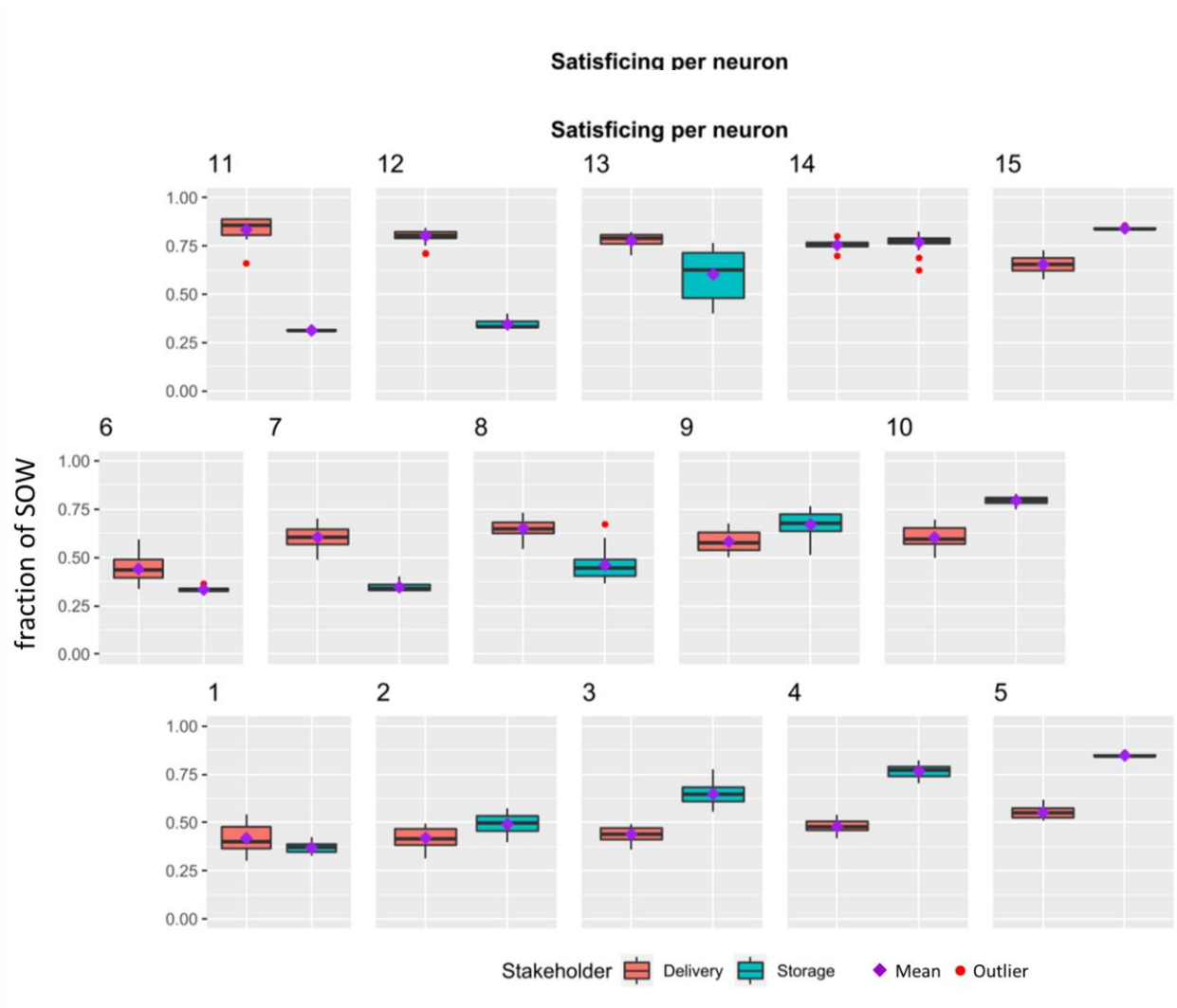
B.13. Superposition results for 90th percentile regret from best



a) The resulting radar plot topology map of the superposition method applied to the 90th percentile regret from best.

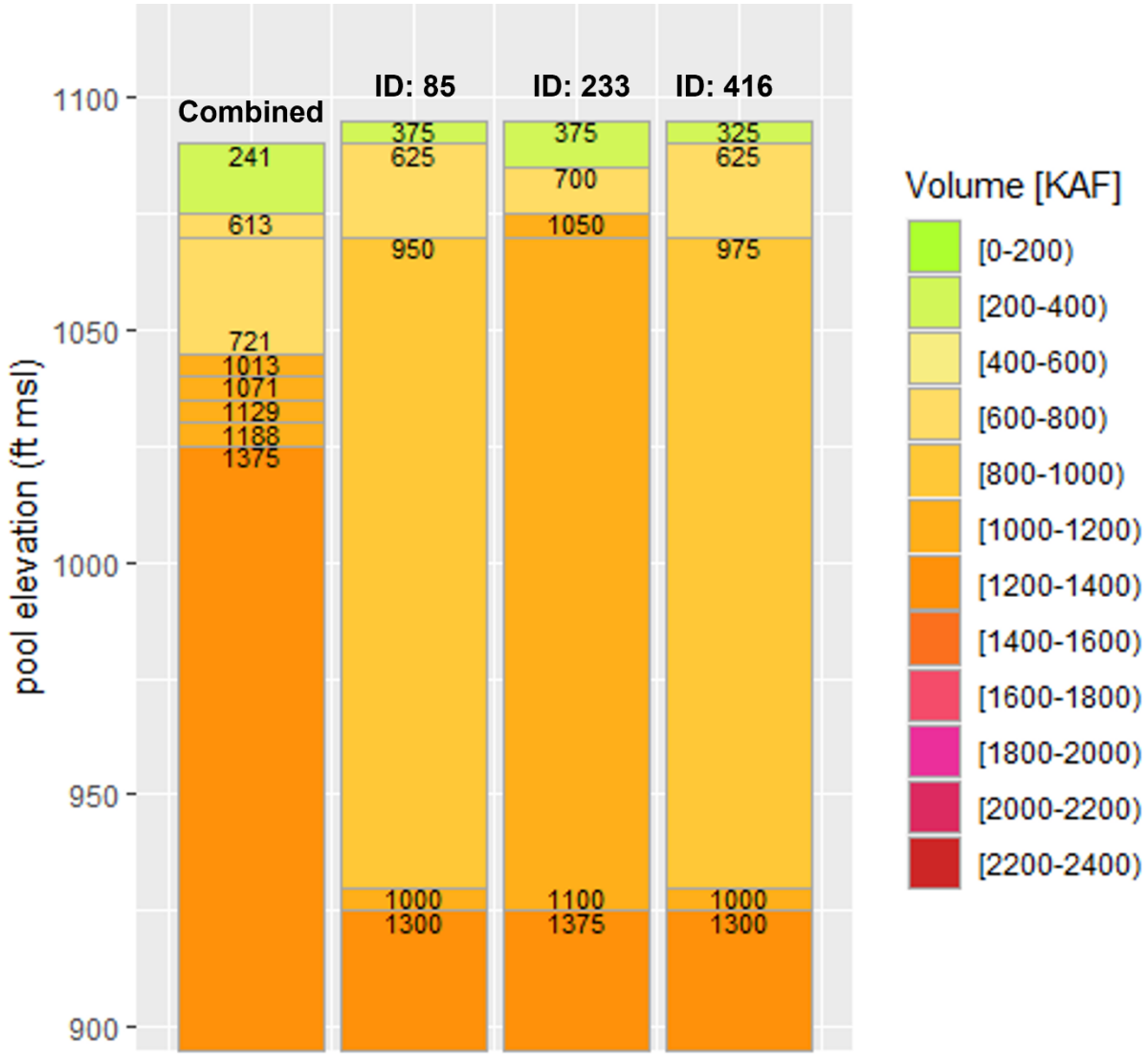
b) 90th percentile regret from best component planes.

B.14. Stakeholder robustness satisficing map using boxplots



The satisficing values of the Delivery (orange) and Storage (blue) decision makers, plotted with boxplots on the SOM. Figure 8 in the text shows neurons 9-10 and 13-15, which were included in the decision makers' negotiation area. This figure shows all neurons.

B.15. Combined shortage operation vs similar policies in neuron 3



Section 4.4.4 uses neurons 9 and 3 to compare the neurons in the feasible negotiation space to the current, combined Lake Mead policy. We use neuron 3 to represent the combined Lake Mead policy because of similar T1e, T1V, and maxV. The average T1e, T1V, and maxV of neuron 3 are 1082 feet msl, 764 KAF, and 1302 KAF, respectively, compared to 1090 feet msl, 241 KAF, and 1375 KAF for the combined policy. Although the average T1V of neuron 3 is 523 KAF greater than the combined policy, there exists three policies within neuron 3 whose T1V values range from 325 to 375 KAF, which is more comparable

to the combined operation (IDs 85, 233, 416, see above figure). These MOEA-derived policies implement T1e only five feet higher than the combined operation (1090 feet msl vs 1095 feet msl). T1V is larger for the MOEA-derived policies, but their T1V values are more similar than the average in neuron 3. The maxV of these three policies and the overall average of neuron 3 is also similar to the combined operation. Interestingly, no neuron (or individual policy) closely mimics all shortage tiers of the combined operation. Neuron 3 is the most similar when comparing T1e, T1V, and maxV, but, from visual inspection of Figure 6, it appears the combined operation enacts maxV at a higher elevation than do policies in neuron 3.

References

Boelaert, J. *et al.* (2021) 'aweSOM: Interactive Self-Organizing Maps'. Available at: https://CRAN.R-project.org/package=aweSOM (Accessed: 24 November 2021).

Bossek, J. *et al.* (2017) 'ecr: Evolutionary Computation in R'. Available at: https://CRAN.R-project.org/package=ecr (Accessed: 13 January 2022).

Clark, S., Sisson, Scott.A. and Sharma, A. (2020) 'Tools for enhancing the application of self-organizing maps in water resources research and engineering', *Advances in Water Resources*, 143, p. 103676. Available at: https://doi.org/10.1016/j.advwatres.2020.103676.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. 2nd edn. New York: Springer-Verlag (Springer Series in Statistics). Available at: https://doi.org/10.1007/978-0-387-84858-7.

James, G. *et al.* (2013) *An Introduction to Statistical Learning*. New York, NY: Springer New York (Springer Texts in Statistics). Available at: https://doi.org/10.1007/978-1-4614-7138-7.

Kohonen, T. (2001) 'The Basic SOM', in T. Kohonen (ed.) *Self-Organizing Maps*. Berlin, Heidelberg: Springer (Springer Series in Information Sciences), pp. 105–176. Available at: https://doi.org/10.1007/978-3-642-56927-2_3.

Kohonen, T. (2013) 'Essentials of the self-organizing map', *Neural Networks*, 37, pp. 52–65. Available at: https://doi.org/10.1016/j.neunet.2012.09.018.

Kruisselbrink, R.W. and J. (2019) *kohonen: Supervised and Unsupervised Self-Organising Maps*. Available at: https://CRAN.R-project.org/package=kohonen (Accessed: 8 December 2020).

R Core Team (2021) *R: A language and environment for statistical computing.* Available at: https://www.r-project.org/ (Accessed: 19 March 2021).

Sievert, C. *et al.* (2021) 'plotly: Create Interactive Web Graphics via "plotly.js"'. Available at: https://CRAN.R-project.org/package=plotly (Accessed: 19 March 2021).

Walesiak, M. and Dudek, A. (2021) ‘clusterSim: Searching for Optimal Clustering Procedure for a Data Set’. Available at: https://CRAN.R-project.org/package=clusterSim (Accessed: 25 January 2022).

Xiao, J., Lu, J. and Li, X. (2017) ‘Davies Bouldin Index based hierarchical initialization K-means’, *Intelligent Data Analysis*, 21(6), pp. 1327–1338. Available at: https://doi.org/10.3233/IDA-163129.

C. Supplementary material for Taxonomy of purposes, methods, and recommendations for vulnerability analysis

C.1. Details of the full-factorial vs space-filling design in Fig. 9.

Both the full-factorial and space-filling design use 64 samples. Values for i1 and i2 range from 0 to 1. The full-factorial is created by making a uniformly-spaced sequence of 8 points for both i1 and i2, then making all combinations of those points, resulting in 64 samples. The space-filling design was created with the improvedLHS function of the lhs package in the R programming language. The code is available on the corresponding author's GitHub: https://github.com/nabocrb/Vulnerability-analysis-examples-GitHub

C.2. Details of the training versus testing accuracy example in Fig. 10

The example uses a linear simulation model: m1 = i1-i2+noise. We simulated 300 Latin Hypercube samples of i1 and i2, where the values range from 0 to 1. The noise represents smaller impacts on performance from other model inputs and any randomness inherit in the system being modelled. The noise is randomly sampled from a normal distribution with a mean of 0 and standard deviation of 0.3. We defined unacceptable performance as being less than the average value of m1. The dataset was randomly split into 80% training and 20% testing (Fig. 10a). A logistic regression model (a linear method) and random forest (a flexible ensemble method) were then trained and evaluated for testing accuracy (Fig. 10b).

The code is available on the corresponding author's GitHub: https://github.com/nabocrb/Vulnerability-analysis-examples-GitHub