Visual Techniques for Exploring Alternatives and Preferences in Group Preferential Choice

by

Emily Hindalong

M.Sc. Bioinformatics, The University of British Columbia, 2015

B.Sc. Cognitive Systems, The University of British Columbia, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Computer Science)

The University of British Columbia

(Vancouver)

January 2018

Abstract

Group Preferential Choice is when two or more individuals must collectively choose among a competing set of alternatives based on their individual preferences. In these situations, it it can be helpful for decision makers to visually model and compare their preferences in order to better understand each others' points of view. Although a number of tools for preference modelling and inspection exist, none are based on a comprehensive understanding of the demands of Group Preferential Choice in particular.

The goal of our work is to understand these demands and explore the space of possible visualizations to support them. We make progress toward this goal in three steps. First, we characterize the scope of Group Preferential Choice by examining a diverse set of real-world scenarios. In particular, we identify sources of variation in preference models, goals, and contexts. Second, we produce a detailed model of abstract tasks to support the goals identified in the first step. Finally, we analytically evaluate various designs with respect to these tasks and conclude with recommendations for different classes of users. We believe that these contributions will help designers produce more effective visual support tools for Group Preferential Choice.

Lay Summary

Sometimes a group of people must make a choice together. For instance, a board of directors may need to agree on a new office location. This can be challenging if there are multiple factors involved, or if the decision makers disagree about what is important. In these situations, effective communication is key.

One way to improve communication is to have each decision maker show his or her preferences graphically. For instance, they might use a bar chart to communicate their ratings of potential office locations. In this work, we try to understand what questions a graphic needs to be able to answer to support effective group decision making. Then, we present a variety of graphical options and discuss how well each one answers these questions.

Preface

Credit for the overarching vision of the project goes to my supervisor, Giuseppe Carenini - it was his idea to develop a design space of visualizations to support Group Preferential Choice. He also came up with the idea of a Preference Model Taxonomy (Section 3.1) and contributed substantially to its development.

Hooman Shariati observed and conducted the interviews for the XpertsCatch case (Section 3.2.6) and the department meeting portion of the Faculty Hiring case (Section 3.2.2).

Soheil Kianzad observed and conducted the interviews for the Gift case (Section 3.2.7).

All other original work is my own. There are no publications based on this work at this time.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisor, Dr. Giuseppe Carenini, for taking me on as a student and helping me find a project that was a good match for my skills and interests. I am grateful for his patient tutelage and enthusiasm for novel research.

Special thanks go to my second reader, Dr. Tamara Munzner, for providing expert guidance at multiple stages of the project. I owe much of my success to her encouragement and advice.

Additional thanks go to Dr. David Poole for advising me during the early stages of the project, Aaron Mishkin for his incisive comments on the first draft of Chapter 3, and Joyce Poon for answering my (many) questions in a timely manner and keeping the Computer Science Graduate Program running smoothly.

Finally, I would like to thank my husband, Michael Gottlieb, for taking care of domestic matters and making sure I took meal breaks - I know it was about as easy as giving a cat a bath.

Chapter 1

Introduction

Group Preferential Choice is when two or more individuals, each with his or her own preferences, must collectively choose among a competing set of alternatives. These situations are common in organizational and public planning. Examples include selecting an office location, hiring a candidate for a position, or choosing a wastewater management system for a city.

Sometimes it is possible for the group to arrive at a satisfactory decision through discussion alone. However, this can be challenging if group members have differences in preferences and opinions. In fact, the group members may not even have a complete understanding of their *own* preferences. To complicate matters further, the group may wish to explicitly model trade-offs among competing criteria. For instance, a company might want to independently assess a job candidate's education, experience, and company fit.

As the number of group members, alternatives, and criteria grows, it becomes increasingly difficult to grapple with the complexities effectively. In fact, organizations often resort to pre-existing solutions because they do not have the resources to tackle this challenge [11] [26]. For instance, a municipality may elect to keep an outdated and costly wastewater management system simply because analyzing the pros and cons of alternatives is too daunting. Clearly, there is an incentive to develop processes and tools to facilitate such analyses.

One viable approach is to have each decision maker explicitly model his or her preferences over the alternatives and criteria. Then, the group members can

compare how alternatives perform under different preference models in order to come to a better understanding of other points of view. There is evidence that explicit preference modelling can encourage reflection, promote transparency and inclusiveness, and ultimately lead to greater satisfaction with the outcome [4].

The benefits of this process depend on how quickly and effectively decision makers can glean insights from their own and others' preferences. It can be difficult to spot interesting differences and trends when the data is represented in text-based formats, such as traditional spreadsheets. Information Visualization solutions are more promising because they leverage the pattern recognition and pre-attentive capabilities of the human visual system [38]. However, not all graphical methods are equally effective, and a poorly-chosen graphic can actually *diminish* the efficacy of the decision making process [2].

Many existing tools for preference modeling and inspection come from the field of Multi-Criteria Decision Making (MCDM), a sub-discipline of Operations Research concerned with practical aspects of multi-criteria decision making [30]. There is considerable evidence that MCDM processes can improve group decision making by enhancing communication among group members [6] and lending transparency and legitimacy to the decision making process [51].

Although MCDA support tools are plentiful, few are able to integrate and display multiple preference models simultaneously [40] [51]. For this reason, most recorded applications of MCDA to group decision making involve joint construction of a single preference model by all members of the group [51]. Tools that *do* allow multiple users to input their preferences, such as M-MACBETH [18], D-Sight [1], and 1000Minds [28], typically show the *aggregate* performance of alternatives over all decision makers using non-interactive charts and tables.

One notable exception is Web-HIPRE, which allows groups to model each decision maker as a separate criterion in the overarching decision problem [39]. Web-HIPRE shows the performance of alternatives using stacked bar charts, where the score for each decision maker is a segment contributing to the total (Figure 1.1). Like other MCDA support tools, the main focus of Web-HIPRE is on the decision analysis process, not the graphical representation.

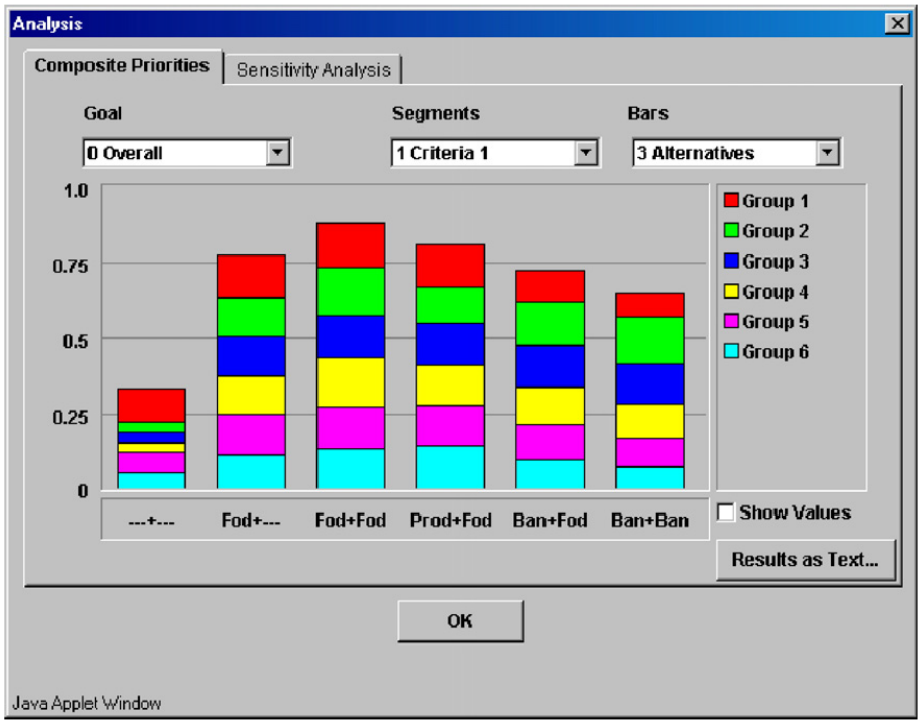


Figure 1.1: Web-HIPRE [40]. The total score for each alternative is represented by bar height, and the contribution of each decision maker to the total is represented by segment height (here, each decision maker is actually a group of people acting as a unit).

A few other tools to support joint preference inspection in Group Preferential Choice put a stronger emphasis on Information Visualization. One is Group ValueCharts (Figure 1.2), which is an interactive visual aid that uses a combination of stacked and multi-bar charts to show how different alternatives perform for different users [4].

Figure 1.2: Group ValueCharts [4]. The top right section shows the score of each alternative for each user. The bars are grouped by alternative and colour-coded by user. The bottom left section shows the criteria hierarchy, and the bottom right section shows the breakdown of scores by criterion. The red outlines show the weights assigned by each user to each criterion.

Group ValueCharts is an extension of ValueCharts, a system that supports elicitation and inspection of linear preference models for individual decision makers [12]. ValueCharts was analytically evaluated based on a task model of *individual* preferential choice [5]. However, this task model may not generalize to Group Preferential Choice.

Another tool, ConsensUs, aims to support the consensus building process by highlighting sources of disagreement [36]. It uses strip plots to encode per-criterion

scores for each alternative and user (Figure 1.3). This tool only allows individual users to compare their preferences against the group average or one other user at a time.

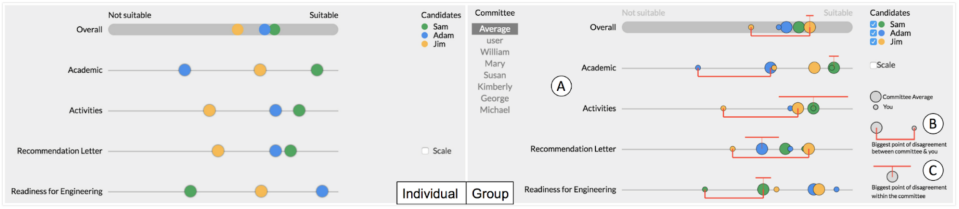


Figure 1.3: ConsensUs [36]. The Individual View (left) allows each user to score alternatives relative to each other on each criterion. The alternatives are colour-coded dots. The Group View (right) shows the individual scores (small dots) in the context of group averages (large dots). Red lines highlight points of disagreement.

A major shortcoming of all available tools and designs is that none (to our knowledge) are grounded in a comprehensive data and task model for Group Preferential Choice. This is not ideal, as the suitability of a design will likely depend on the characteristics of the decision making scenario. However, the diversity of Group Preferential Choice scenarios has not been studied. Future designers of such tools would benefit from a clearer understanding of this diversity and the implications for design. Our work addresses this problem in three steps:

First, we perform an in-depth analysis of seven real-world group preferential choice scenarios in order to characterize the variation in the data, goals, and decision making contexts (Chapter 3). For the analysis of goals, we focus solely on the stage where individual preference models are combined and discussed by the decision makers, a process we call *preference synthesis*. These scenarios were studied using a combination of structured interviews and analysis of secondary sources. The outputs of this analysis are:

1. A precise definition of Group Preferential Choice
2. A taxonomy of commonly-used preference models
3. A summary of the preference synthesis goals across scenarios
4. A summary of the decision making contexts across scenarios

Second, we translate the data and goals identified in Chapter 3 into domain-independent language in order to produce descriptions at various levels of abstraction (Chapter 4). These abstractions are intended to be suitable visualization design and analysis. The data and task abstraction is performed in accordance with the typology of Brehmer and Munzner [7].

Finally, we present a *prescriptive design space* of visualizations to support preference synthesis in the context of Group Preferential Choice (Chapter 5). A prescriptive design space is a set of viable designs for a particular kind of data with recommendations based on goals and contexts. For now, our design space is limited to a subset of all Group Preferential Choice scenarios where there are no explicit criteria and no more than a dozen alternatives and decision makers. The design space is described in terms of the following *design aspects*:

1. Static design aspect - the basic idioms that are available and various options for mapping the dimensions and measures to marks and channels.
2. Dynamic design aspect - the mechanisms for transforming the view.
3. Composite design aspect - the options for arranging and coordinating different views relative to each other.

For inspiration, we look to other prescriptive design space papers, such as Brehmer et al. [9], which presents a design space for timelines in the context of storytelling. This work analyzes over 100 existing timelines in order to identify the major design dimensions, and then offers recommendations based on the narrative points that the storyteller would like to make. Similarly, we provide design recommendations based on the decision making context and the relative importance of various tasks.

We believe this work will provide a sound starting point for designers of Group Preferential Choice support tools. Depending on the situation, designers may wish to create standalone visual aids or integrate visualizations into complete decision support systems. We expect our recommendations to serve in a wide variety of individuals, ranging from project managers who need to produce graphical summaries quickly to designers of MCDA support tools who want to incorporate more effective graphics into their decision analysis software.

Chapter 2

Background and Related Work

2.1 Decision Theoretic Foundations

Decision Theory is a field of study that is concerned with developing abstractions and techniques to support rational decision making. It defines a decision problem as one where a decision maker must select between two or more acts, each of which has an associated outcome [44]. The decision maker is presumed to prefer some outcomes over others. In some cases, the decision maker may not know for certain what the outcome of an act will be. Such scenarios are called decisions under risk.

2.1.1 Utility Theory

In economics, utility is a quantitative measure of satisfaction with a good, service, or situation. In order to apply certain decision theoretic methods to a decision problem, a decision maker must describe her preferences as a utility function [60]. There are two classes of utility functions: ordinal and cardinal.

An ordinal utility function ranks all possible outcomes from most to least preferred without specifying the strength of preference. More formally, it defines a preference relation between each pair of outcomes indicating a preference for one or the other or indifference between the two. The function must satisfy certain conditions, such as completeness, anti-symmetry, and transitivity [44].

A cardinal utility function maps outcomes to values along an interval scale such that preferences are preserved up to positive linear transformations. Cardinal

utility functions are essential for decision making under risk [32]. However, many economists believe they are unnecessary in risk-free decision analysis, since they are more difficult to elicit and do not add much power to the decision analysis [48] [45].

Aside from risk, another major consideration is whether the decision problem has one or multiple attributes [32]. In the single attribute case, the outcome is an atomic value; in the multiple attribute case, it is a composition of values for different attributes. For instance, if a decision maker is looking to buy a house and only cares about cost, the outcome can be expressed in terms of cost only. However, if the decision maker also cares about location, then the outcome needs to be expressed in terms of cost and location.

The optimal decision making strategy for the single attribute case without risk is straightforward - simply choose the option that yields the most preferred outcome on the sole attribute. The multiple attribute case is more complicated, and there is an entire field of study devoted to it.

2.1.2 Multiple Criteria Decision Making

Multi-Criteria Decision Making (MCDM) is a sub-discipline of Operations Research that is concerned with formalizing and developing methods for scenarios where a decision maker must choose among multiple competing alternatives in the presence of multiple competing criteria. Although MCDM draws from Decision Theory, its emphasis is more pragmatic than theoretical. Here, we focus on a subset of MCDM called Multi-Attribute Decision Making (MADM), which is concerned with scenarios where the options are finite and predefined [29].

A number of MADM methods have been developed, but they all require the following key ingredients:

1. A finite set of two or more alternatives
2. A finite set of two or more attributes (also called objectives or criteria)
3. A quantitative model of the individual's preferences over the alternatives and/or attributes

The main way in which these methods differ is in how preferences are elicited, expressed, and combined to produce a final score or ranking over the alternatives.

Multi-Attribute Utility Theory (MAUT)

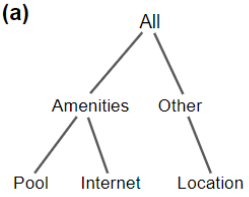
Multi-Attribute Utility Theory is the most popular class of MADM methods and the only one with a solid foundation in Decision Theory [32]. Multi-attribute value theory (MAVT) is a special case of MAUT where there is no risk.

According to MAVT, a decision maker's preferences can be modeled using an additive multi-attribute value function (AMVF) as long as the attributes have *additive independence*, which means that the outcome on one attribute does not affect how the decision maker feels about the possible outcomes on other attributes [32]. An AMVF has three major components (illustrated in Figure 2.1):

- An *attribute tree*, which specifies a decomposition of high-level attributes into lower-level attributes. The attributes at the leaves are called *primitive attributes*, and each primitive attribute has a set of possible outcomes called its *domain*.
- A set of *score functions* for each primitive attribute specifying the value of each possible outcome to the decision maker. The best possible outcome is assigned a score of 1 and the worst a score of 0, and all other outcomes are scored relative to these two.
- An assignment of *weights* to the primitive attributes such that the sum of the weights over all primitive attributes is 1. The weight represents the value of switching from the worst possible outcome to the best possible outcome for that primitive attribute relative to the others.

The final value for each alternative can be computed by taking the weighted sum of the outcome scores on all primitive attributes.

(a)



(b)

	Weight	Score Function
Pool	0.1	Yes: 1 No: 0
Internet	0.7	Highspeed: 1 Lowspeed: 0.5 None: 0
Location	0.2	Downtown: 1 Highway: 0

(c)

	Sheraton	Days Inn	Budget
Pool	yes	**yes**	no
Internet	highspeed	**lowspeed**	none
Location	downtown	**highway**	highway

(d)

Score(Days Inn) =

Weight(pool) * Score(pool, **yes**) +
Weight(internet) * Score(internet, **lowspeed**) +
Weight(location) * Score(location, **highway**) =

0.1 * **1** + 0.7 * **0.5** + 0.2 * **0** = **0.45**

Figure 2.1: A multi-attribute value function for choosing a hotel. A MAVT consists of an attribute tree (a) and a set of weights and score functions for each primitive attribute (b). The score for a each alternative-attribute pair is the value assigned by the score function to the outcome of that alternative on that attribute (c). The score for an alternative (e.g. Days Inn) is the weighted sum of the scores on each attribute (d).

Other Methods

Aside from MAUT, there are two other popular classes of MADM techniques.

Outranking methods such as ELECTRE [49] are among the oldest MADM techniques [59]. They require users to set qualifying and indifference thresholds over the attributes. Then, alternatives are eliminated if they do not meet the qualifying thresholds or if they are *outranked* by at least one alternative - that is, at the same or a lower indifference class on every attribute. Although outranking methods have largely been replaced by more precise methods, they are still sometimes used to winnow the set of alternatives to a reasonable number [59].

The Analytical Hierarchy Process (AHP) is the main contemporary contender to MAUT [50]. In AHP, the decision maker is presented with pairs of alternatives and asked to indicate their degree of preference for one alternative over the other on

each attribute. These comparisons are used to generate a score for each alternative-attribute pair. A similar procedure is used to elicit weights. AHP has been criticized for its susceptibility to rank-reversal, which means that adding a new alternative to the set may cause the relative ranks of two other alternatives to change [59].

2.1.3 Group Multi-Attribute Decision Making

Multi-attribute decision making has been applied in group settings since its initial formulation. Group MADM is similar to individual MADM except that the selection process factors in the preferences of multiple stakeholders. Bose et al. reviewed several early applications of MAUT in group decision making contexts [6]. Based on their findings, they argued that MAU-based models considerably enhance communication and understanding among group members and should be supported in more computer-based group decision support systems.

Salo and Hämäläinen also argued that MADM methods can benefit group decision making by increasing transparency and legitimacy [51]. They analyzed several recent applications of MADM methods to group decision making and identified six basic steps common to all:

1. Clarification of the decision context and identification of group members
2. Explication of decision objectives
3. Generation of decision alternatives
4. Elicitation of preferences
5. Evaluation of decision alternatives
6. Synthesis and communication of decision recommendations

They note that steps 3 and 4 are sometimes reversed but recommend following the suggested order because listing alternatives first makes it easier for people to reason about preferences. They also note that decision makers often revisit earlier steps to refine the decision model.

One of the major theoretical challenges behind Group MADM is how to combine multiple preference models in a way that is both rational and fair. This is the main concern of a philosophical discipline called Social Choice Theory. Arrow's Impossibility Theorem states that it is not possible to aggregate individual

preference rankings into a group preference ranking which is guaranteed to satisfy certain reasonable conditions [3]. This is also true for multi-attribute utility functions unless each decision maker is allowed to define her own objectives [31].

Fortunately, a theoretically-sound aggregate model might not be necessary in most cases. In small group decision settings, it may be more important for decision makers to understand their own and others' preferences on an individual basis so that they can negotiate effectively. To this end, high-quality visualizations have the potential to help decision makers communicate and reason about their preferences.

2.2 Visual and Interactive Techniques For MADM and Related Data

This section summarizes several published techniques for visualizing scores and preferences in the context of MADM, as well as techniques for visualizing similar multi-attribute data, such as rankings, surveys, and product reviews. These techniques are summarized in Table 2.1.

Table 2.1: Eight techniques for visualizing multi-attribute data. Techniques with 'Very High' relevance explicitly support Group MADM. The remaining techniques were assigned 'High' or 'Medium' relevance based on quality and novelty.

	Context	**Main Encodings**	**Relevance**
Group ValueCharts [4]	Group MADM	Stacked bar chart; Tabular bar chart	Very High
ConsensUs [36]	Group MADM	Dot plots in small multiples	Very High
Web-HIPRE [39]	Group MADM	Stacked bar chart	Very High
LineUp [25]	Multi-attribute rankings	Slope graph; Stacked or tabular bar chart	High
SurveyVisualizer [10]	Multi-attribute survey results	Parallel coordinates tree	High
WeightLifter [41]	MADM	Stacked bar chart; Parallel coordinates	Medium
DCPAIRS [20]	MADM	Scatterplot matrix (SPLOM)	Medium
QStack [42]	Multi-attribute rankings	Stacked bar chart	Medium

2.2.1 Group ValueCharts

Group ValueCharts [4] is perhaps the most sophisticated tool designed specifically to support Group MADM. In particular, it is intended for large infrastructure decision problems where trade-offs must be considered between multiple criteria and multiple decision makers' preferences. It aims to make the decision process more participatory, transparent, and comprehensible.

The main encodings are multi-bar charts (also known as grouped bar charts), which show the total score of each alternative for each decision maker, and tabular bar charts (also known as faceted bar charts or small multiples bar charts), which show the breakdown of scores by criteria.

Figure 2.2: Group ValueCharts [4]. A colour-coded multi-bar chart shows the total score of each alternative for each decision maker (top right). A tabular bar chart shows the breakdown of scores for each decision maker by attribute (bottom right). Red outlines show the weights assigned to each attribute by each decision maker. The criteria hierarchy is shown using a rectilinear tree (bottom left).

A key strength of Group ValueCharts is that it is compact and information dense. The multi-bar chart makes it possible to compare the overall performance of each alternative across evaluators, while the tabular bar chart supports comparison on a per-attribute basis. The tabular bar chart also supports direct comparison of weights.

A limitation of Group ValueCharts is that it does not scale beyond a dozen attributes, alternatives, or decision makers. One reason for this is that it does not implement any data reduction strategies to cope with spatial constraints. Another is that colour is used to differentiate decision makers, and people can only differentiate up to around a dozen colour hues [38].

Web ValueCharts

Web ValueCharts is a successor to Group ValueCharts that integrates the capabilities of both the group version and the individual version (ValueCharts [12]) on a web platform. It is intended to bring structured decision support to a broader audience.

Web ValueCharts is a modular system with components to support chart definition, chart management, preference elicitation, and preference inspection. The preference inspection component has two views - the individual view and the group view (Figures 2.3 and 2.4).

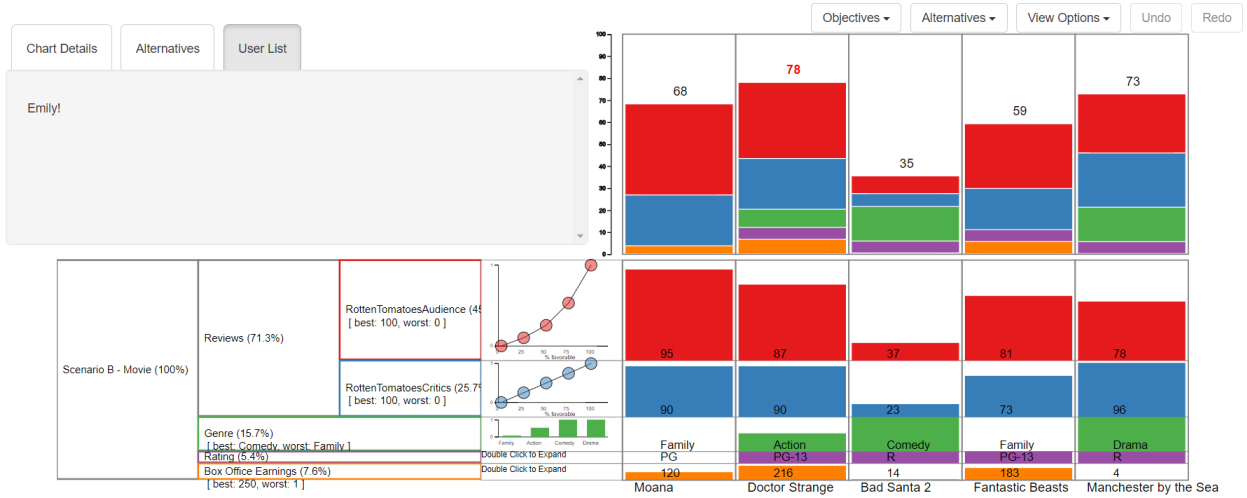


Figure 2.3: Web ValueCharts - Individual View. A colour-coded stacked bar chart shows the total score of each alternative (top right). A tabular bar chart shows the breakdown of scores by attribute (bottom right). A rectilinear tree shows the attribute hierarchy and the score functions for each attribute (bottom left).

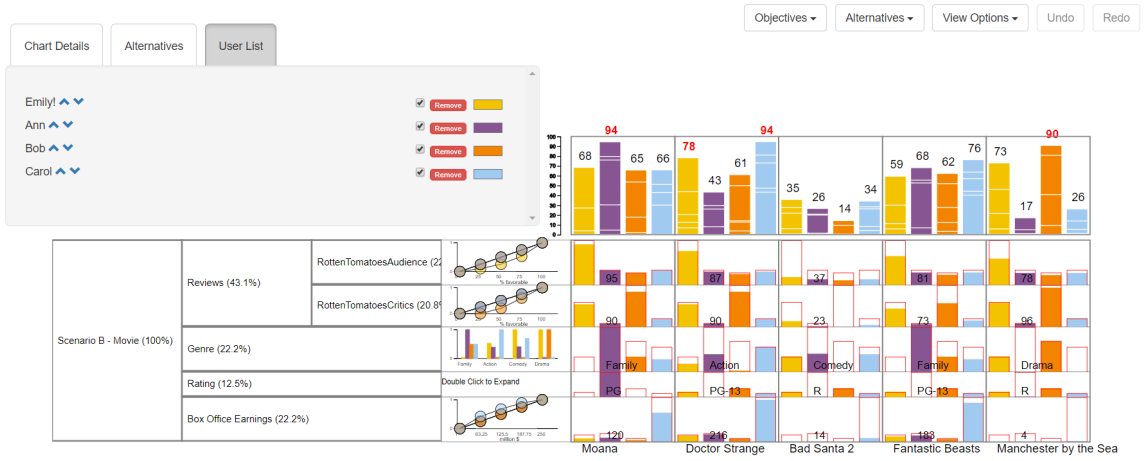


Figure 2.4: Web ValueCharts - Group View. The overall design is the same as for Group ValueCharts.

The individual view (Figure 2.3) allows users to inspect their own scores and preferences in isolation. They can adjust their score functions and weights dynamically by clicking and dragging relevant components. The group view (Figure 2.4) shows the alternative scores and preferences for all users in a single view. The design is identical to that for Group ValueCharts, except that users may also inspect the score functions. Users can select a subset of decision makers to view by toggling the check boxes beside the names.

Both views support manual reordering of alternatives and attributes. There are various view options that can be turned on and off, including the score functions, the outcomes overlay, and the utility scale.

Web ValueCharts supports real-time synchronization of all group members' charts. Users can join a group chart, update their preferences, and even edit the criteria and alternatives in real-time. Although Web ValueCharts improves upon its predecessor in many ways, it has the same limitations when it comes to scalability.

2.2.2 ConsensUs

ConsensUs is another tool that is designed to facilitate multi-criteria group decision making [36]. In particular, it aims to support the consensus-building process by highlighting sources of disagreement. Its primary encoding is the strip plot, which uses point position on an axis to represent values.

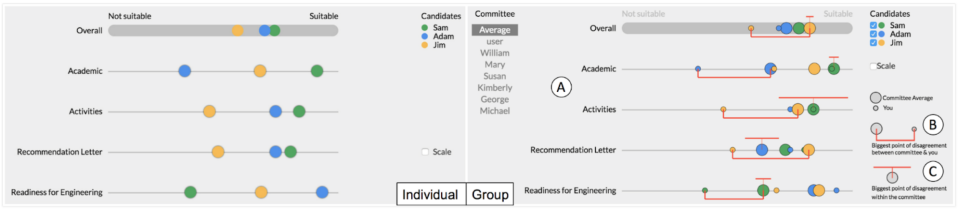


Figure 2.5: ConsensUs - Individual View (left) and Group View (right) [36]. Each view has one strip plot per attribute. The alternatives are colour-coded dots, and their positions represent the scores assigned by the individual (left) or the group average (right).

The solution consists of an individual view and a group view. Individual evaluations are collected via the individual view before being displayed in the group

view. Each evaluator scores each alternative relative to the others using a sliding scale.

The group view has two kinds of colour-coded dots: small dots showing the individual scores and large dots showing the group averages. It also emphasizes two kinds of disagreement for each attribute: the alternative with the largest difference between individual and group score (red line below) and the alternative with the largest variance in score within the group (red line on top). The line length encodes the degree of disagreement.

A few kinds of interaction are available in the group view. Users may click on the large dots to see the scores assigned to that alternative by each evaluator. Users may also filter alternatives (top-right) and change which other user is shown on the large dots (top-left).

Strip plots are notable for their succinctness and compactness relative to bar charts that show the same data [46]. However, there is the risk of occlusion if points have the same or nearly the same value. A weakness of this design in particular is that it only allows users to compare their scores to those of the average user or one other user at a time. Also, the use of colour to differentiate alternatives limits scalability.

2.2.3 Web-HIPRE

Web-HIPRE is one of the oldest interactive support tools for multi-attribute decision analysis [39]. It was originally designed for AHP analysis only but was later extended to support other decision analysis paradigms.

Web-HIPRE's main window (Figure 2.6) shows the attribute hierarchy and alternatives. From there, users can open other windows to inspect their preferences or analyze results.

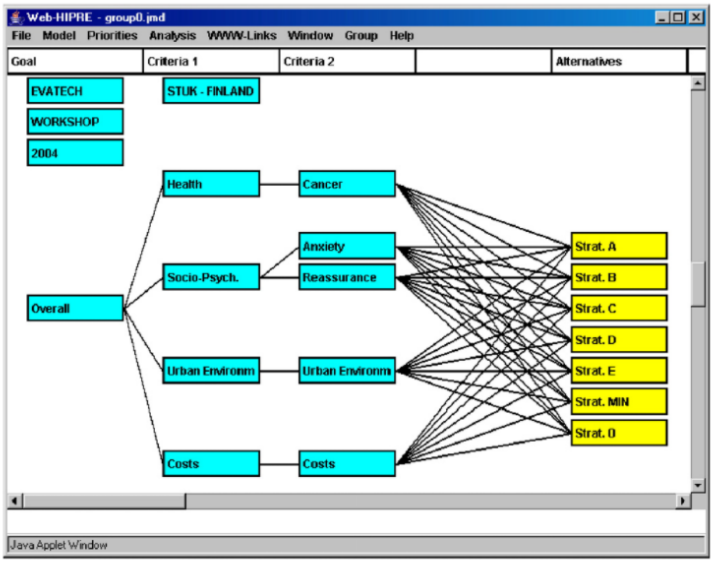


Figure 2.6: Web-HIPRE - Main View [40]. The blue nodes represent the attribute tree. The yellow nodes represent the alternatives.

The Analysis Window (Figure 2.7) uses stacked bar chars to simultaneously show the total score and per-attribute score of each alternative. Users can select which data to map to bars and segments. Effectively, this means that they can reverse the mapping or select a different level of the attribute hierarchy.

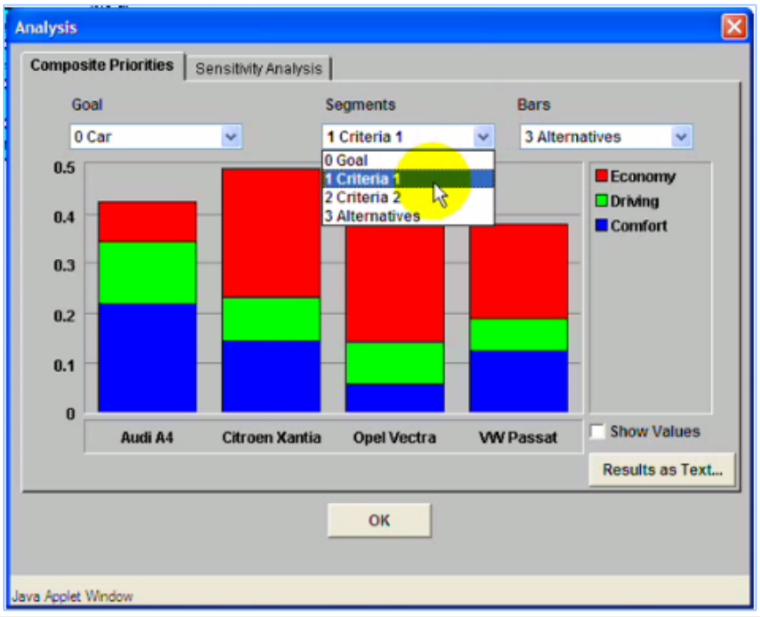


Figure 2.7: Web-HIPRE - Analysis Window [40]. The bars encode alternative scores and the segments encode per-attribute scores.

Web-HIPRE supports multi-attribute group decision making by allowing users to define a new decision problem on top of multiple individual models. The aggregate model treats each user as an attribute in a new decision problem, as shown in Figure 1.1. Web-HIPRE is notable in that it is the only tool that explicitly supports the specification of different weights for different decision makers. However, it is limited in that it has few interactive options and its features are divided over multiple windows.

2.2.4 LineUp

LineUp is an award-winning interactive tool for comparing ranked entities across multiple attributes [25]. It supports a variety of tasks related to rank comparison and sensitivity analysis and is commendable for its power, flexibility, and attention to detail.

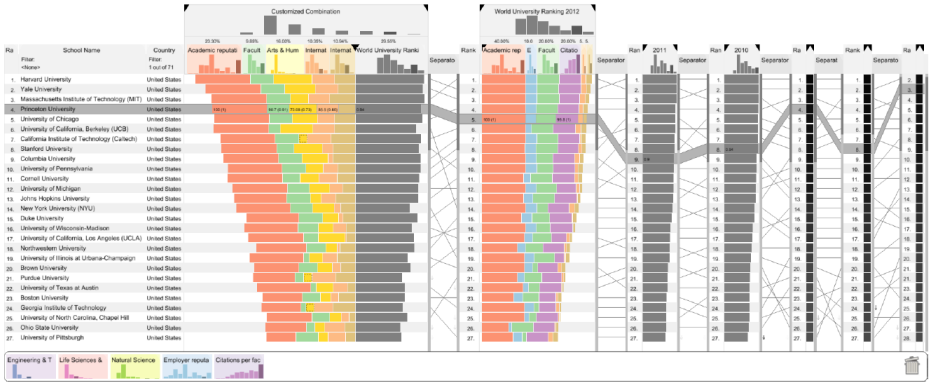


Figure 2.8: LineUp [25]. Each ranked entity is a row and each column is an attribute. Multiple rankings can be compared side-by-side, and same entities are connected with sloped lines. This figure compares seven rankings.

The solution is an elaborate hybrid of bar charts and *slope graphs*, which draw connecting lines between the same entities across different rankings. Each item is a row and each attribute is a column, and a *ranking* is an ordering of items based on the total score over multiple attributes. Categorical attribute columns display text, while numerical attribute columns encode the attribute scores with bars, which are colour-coded by attribute. Histograms above each column show the distribution of scores for that attribute. The slope graph feature can be used to compare two or more rankings side-by-side.

The columns within each ranking can be shown as a stacked bar chart or tabular bar chart based on the user's selection. In this respect, the core idiom is similar to that of ValueCharts. LineUp's extensive list of features allows users to:

- Sort and filter entities by attribute score
- Perform sensitivity analysis on attribute weights and score functions
- Identify missing values
- Scroll through rows or inspect a fish-eye view of the rows (supports scalability on entities)
- Collapse or combine columns (supports scalability on attributes)

- Select one of the following alignment strategies: stacked, diverging stacked, ordered stacked, or tabular

Although LineUp was not designed specifically for Group MADM, it could be adapted to it in a couple ways. First, two or more decision makers' models could be compared in full using the slope-graph component of LineUp. Second, multiple decision makers' models could be condensed into a single model by defining a meta-column over all decision makers.

A possible criticism of LineUp is that it may have *too many* features for typical users. Not all of the features were mandated by the preliminary requirements analysis.

2.2.5 WeightLifter

WeightLifter is a novel visual and interactive technique to help system designers understand the impact of criteria weights on the decision outcome [41].

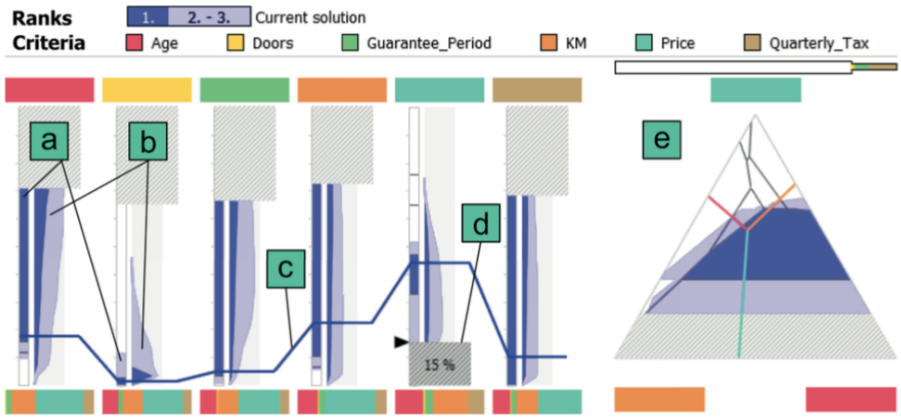


Figure 2.9: WeightLifter [41]. Sliders support exploration of two-way trade-offs between criteria, and a triangle with adjustable line intersections supports exploration of three-way trade-offs.

WeightLifter supports interactive exploration of two-way and three-way trade-offs. Two-ways trade-offs are supported by sliders - users can put any number of criteria on either end and then adjust the slider position. Three-way trade-offs are

supported by a triangle with intersecting lines perpendicular to each edge that the user can adjust. In Figure 2.9, the coloured regions (a) show the points at which the current top solution (c) would change or fall out of the top three, and black lines show the points at which the top solution would change. The sliders also have histograms (b) that show what fraction of the entire weight space given that trade-off has the current solution at the top. Users can constrain the weight space to sub-ranges on the sliders (d).

WeightLifter was integrated with two additional views to support all the tasks identified in the preliminary requirements analysis (Figure 2.10). The Ranked Solution Details view is akin to a simplified version of LineUp - it uses stacked bar charts to show the weighted sum of costs for each alternative over the criteria. Its one unique feature is a strip divided into coloured segments proportional to criteria weights. Each segment also contains a glyph showing the direction of the score function. The Criteria Value View uses parallel coordinates to show criteria outcomes for each alternative. It also allows users to set filters by brushing.

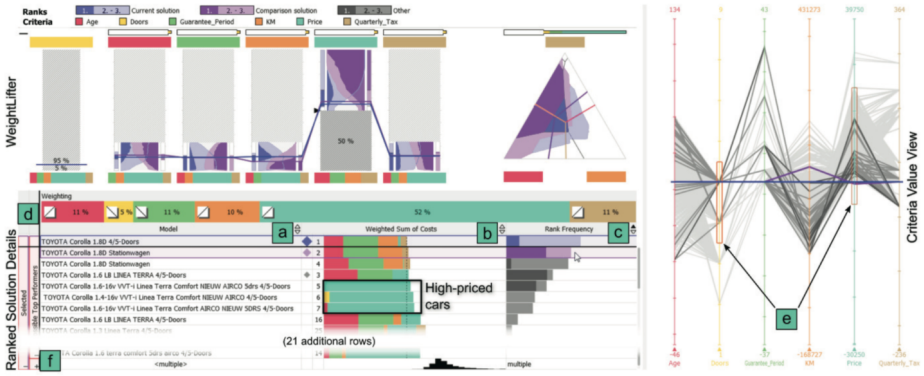


Figure 2.10: WeightLifter plus two additional views [41]. The Ranked Solution Details view allows users to inspect per-criterion scores for the top ranked alternatives. The Criteria Value View shows the criteria outcomes for each alternative.

2.2.6 SurveyVisualizer

SurveyVisualizer is a tool that supports exploration of large, hierarchical satisfaction survey data [10]. It was originally designed to visualize customer satisfaction data for the public transportation system of Zurich. This data consisted of responses to 89 survey questions, which were grouped into 23 quality dimensions and 3 quality indices. The surveys were partitioned into *analysis groups* based on demographic information.

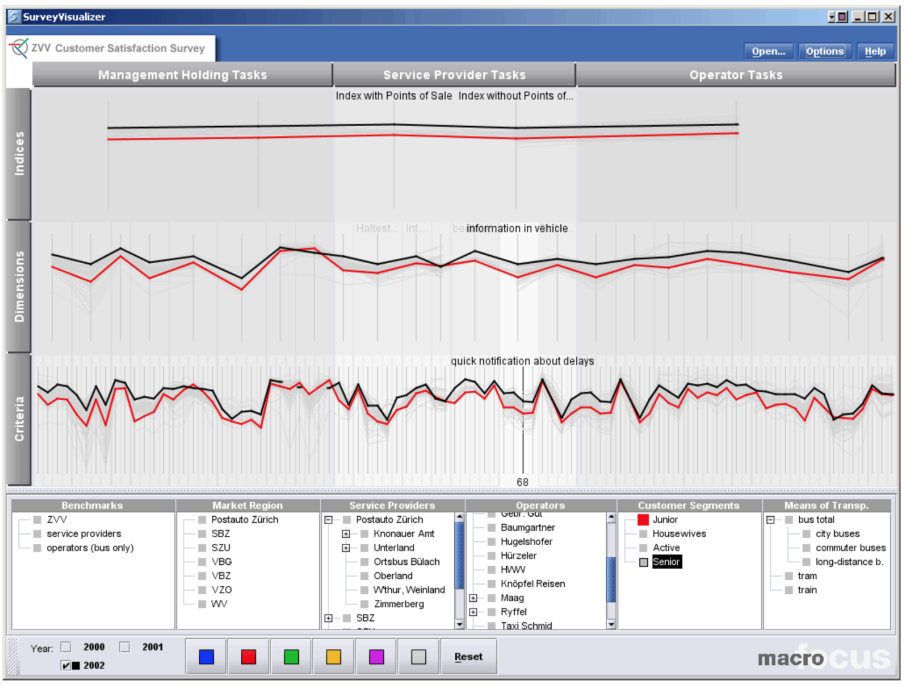


Figure 2.11: SurveyVisualizer [10]. A parallel coordinates tree (top) shows the survey results at three levels of aggregation. Each line corresponds to an analysis group. The analysis group selector (bottom) allows users to control which analysis groups are included.

The basis of SurveyVisualizer is a novel encoding called a Parallel Coordinates Tree, which shows the performance of every analysis group across criteria at three levels of aggregation. The groups are drawn in light grey by default, but individual groups can be emphasized temporarily by: (a) hovering over them, which high-

lights them and brings up details about them, (b) clicking on them, which turns them black temporarily, or (c) assigning them a permanent colour.

The navigation mechanism is the bifocal lens, which allows users to emphasize individual analysis groups. The Parallel Coordinates Tree and the analysis group selectors are coordinated - selecting an analysis group in one causes the same group to be emphasized in the other view.

This work is notable for its novelty and the fact that it has achieved some commercial success [10]. It combines the strengths of two different types of encodings - parallel coordinates and trees. The parallel coordinates component supports inspection of multiple items, while the tree component supports inspection at various levels of aggregation. Parallel coordinates scale well to hundreds of items and are effective for identifying outliers and trends between neighboring attributes, but they are sensitive to the ordering of axes [38]. SurveyVisualizer also makes effective use of linked highlighting, annotation, and the focus plus context design choice.

2.2.7 DCPAIRS

DCPAIRS is a compact tool for individual MCDA that allows users to explore trade-offs between alternatives without using colour to distinguish attributes [20]. This is motivated by the fact that colour is not a scalable identity channel, since people can only distinguish up to around a dozen hues [38]. This work investigates the use of colour for user annotation instead - a novel feature for MCDA tools.

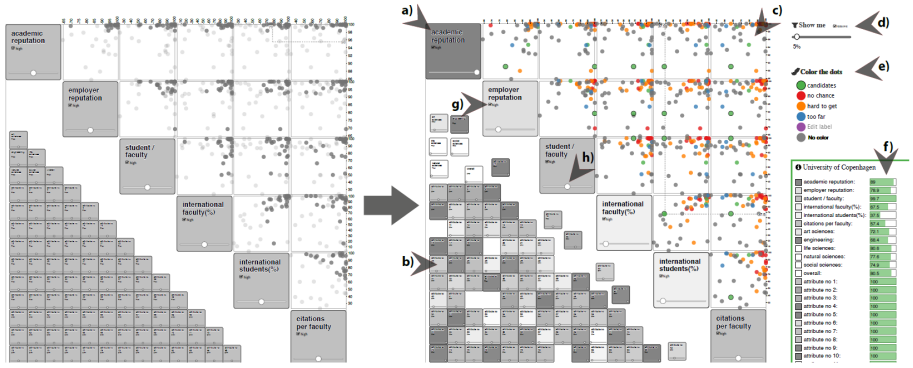


Figure 2.12: DCPAIRS [10]. The six focal attributes are placed on the main diagonal (a), and the remaining attributes are arranged in the lower triangle (b). The points in each scatterplot are the alternatives, and their coordinates encode their weighted scores on each of the two attributes at that intersection (c). Points are coloured according to user-defined annotation groups.

The solution consists of a scatter-plot matrix that shows pairwise trade-offs between six attributes at a time. The six focal attributes are placed on the main diagonal (a), and the remaining attributes are arranged in the lower triangle (b). The points in each scatterplot are the alternatives, and their coordinates encode their weighted scores on each of the two attributes at that intersection (c).

The user can drag-and-drop attribute tiles into one of the six slots and adjust their weights using the sliders on the tiles (h). The current weight of each attribute is redundantly coded in gray-scale. The attribute score functions are positive linear by default, but the user can invert them by toggling ‘high’ and ‘low’ in each tile (g).

When the user clicks on a point, that alternative gets highlighted in all the plots, and the inspector (f) gets populated with the score information for that alternative. The user can interactively assign alternatives to colour-coded groups (e) based on features of interest. Finally, the user can filter alternatives on overall score using the threshold slider (d).

The dominant encoding - the scatter-plot matrix - is limited in that it is only effective at showing pair-wise trade-offs. Furthermore, it does not show the contribution of weighted attribute scores to total score. Nevertheless, the design does have

its strengths, including the annotation feature and the use of details-on-demand and linked highlighting for selected alternatives. It also has relatively high scalability for number of alternatives.

2.2.8 QStack

QStack is a tool for ranking collections in multi-tag datasets based on tag frequency [42]. For instance, a user might want to find photo albums on Flikr with high incidence of the tags 'summer' and 'flowers.'

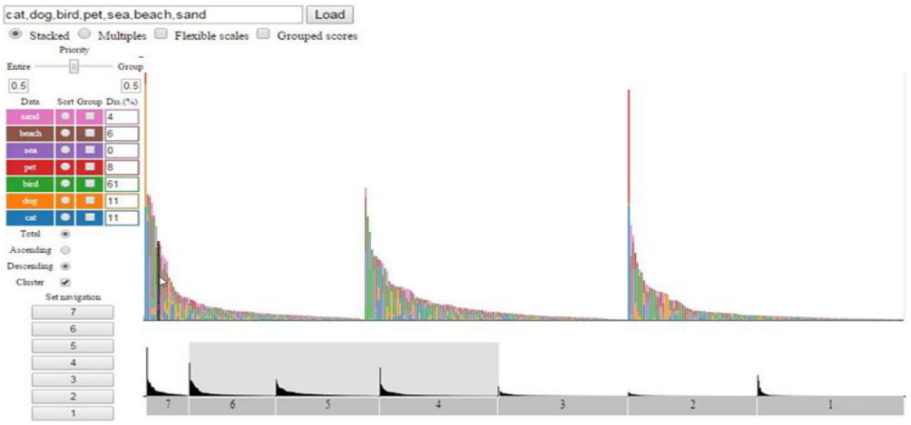


Fig. 1. Basic design of Qstack.

Figure 2.13: QStack [42]. Each bar in the focus view (top) corresponds to a collection, and each segment represents the frequency of a particular tag. The tags are coded by colour (left). The context view (bottom) shows the entire data-set, and the focus view is populated with data from the selected portion.

QStack is similar to ValueCharts and LineUp in that its primary encoding to show score totals is the stacked bar chart. The user enters a set of tags in the search bar, and a set of collections that contain any of these tags is returned. The focus view (Figure 2.13) is then populated with stacked bars, where each bar is a collection and the height of each coloured segment encodes the tag frequency for that collection.

The context view below the focus view shows the total tag frequencies of col-

lections in seven different clusters. Users can brush the context view to select a subset of the data to inspect in the focus view.

Users can sort in ascending or descending order by a particular tag or total tag frequency. When the user hovers over an item, the Distributions column of the tag table (left) is populated with the tag distributions for that item.

For the most part, QStack is simply a weaker version of LineUp, but it is not without its merits. Its primary strength is that it uses the focus plus context design choice to achieve scalability by splitting the view into a focus view and context view.

2.2.9 Lessons from Evaluations

It is important to note that few of these techniques have been thoroughly evaluated, and many have not been evaluated at all.

Group ValueCharts was evaluated in a qualitative study with two groups involved in real-world decision making. The participants expressed a desire to see the average scores and disagreement levels, so these features were added to the tool [4]. The authors of ConsensUs performed a laboratory study and concluded that showing disagreement visually is more effective than showing verbal arguments and just as effective as showing both [36]. Finally, the authors of LineUp also conducted an experiment and discovered that a strong analysis tool enables novices to complete complex tasks faster than experts using Excel or Tableau [25]. The overarching finding in all studies was that people generally react positively to tools of this nature [4] [36] [25] [42].

What has yet to be established is which of the many features implemented by these tools are valuable in various Group Preferential Choice contexts. This is our primary motivation for developing a comprehensive data and task model for preference synthesis in the context of Group Preferential Choice.

2.3 Design Space Analyses

Another major goal of this work is to produce a design space of visual tools for preference synthesis in the context of Group Preferential Choice. For inspiration, we reviewed six papers that can be loosely described as *design space analyses*, but

the papers vary in what that entails. These are summarized in Table 2.2.

Three of these are best described as *design surveys* - they review large bodies of literature on Information Visualization solutions for a particular domain or data class: disease epidemiology [13] traffic data [17], and sentiment in text [33]. They attempt to identify the major dimensions and classify the solutions according to these dimensions. A couple of these works conclude with some broad suggestions for design [13] [33], but none produce a complete set of design recommendations. The result of these works is a *descriptive design space*, which covers what designs currently exist.

Ceneda et al. [14] also has a design survey component, but rather than focusing on a particular domain, it considers a particular *aspect* of InfoVis solutions in general - guidance. Another difference is that it first develops the design space based on previous work and then describes the surveyed works in terms of this space. The result of this work is also a descriptive design space.

Brehmer et al. (2016) [8] is best described as a *design study*, which involves analyzing a specific problem faced by domain experts and developing a visualization solution to address the problem [52]. In this case, the goal was to produce a set of design guidelines for presenting time-oriented data in the energy analysis domain and develop a support tool based on these guidelines. A number of possible designs were proposed and evaluated, and a set of design guidelines was produced. The result of this work is a combination of a *speculative design space*, which describes what designs are possible, and a *prescriptive design space*, which describes what designs are recommended.

Brehmer et al. (2017) combines elements of all of these works [9]. First, it surveys over 100 existing timelines from various sources to produce a descriptive design space of timelines. Then, it considers all combinations of different facets of the design space, resulting in a speculative design space. Finally, the speculative design space is winnowed based on viability, and recommendations are made for different story-telling goals. The final product is a prescriptive design space. Viability was assessed based on existing principles, common sense, and the author's intuition rather than any new empirical data.

Table 2.2: Summary of six design space analyses.

	Type of design space	**# of works surveyed**
Epidemiology Visualization [13]	Descriptive	88
Sentiment Visualization [33]	Descriptive	132
Traffic Data Visualization [17]	Descriptive	Unclear (10s)
Characterizing Guidance [14]	Descriptive	Unclear (10s)
Energy Portfolio Analysis [8]	Speculative, Prescriptive	N/A
Timelines for Storytelling [9]	Descriptive, Speculative, Prescriptive	145

Of the works above, Brehmer et al. (2017) is the closest to our goals, as we intend for our design space to cover all existing viable designs (that we know of), as well as potentially viable new designs.

However, there are some key differences worth noting. First, there are not enough existing tools designed specifically for Group Preferential Choice to support a design survey of the same scope. Hence, our design space may be more speculative. Second, the speculative component of Brehmer et al. (2017) only considers novel *combinations* of dimension values (for example, spiral layout with logarithmic scale), whereas ours may also propose novel dimension *values*. Finally, Group Preferential Choice data is more heterogeneous, which means that our design space will be more complex.

Chapter 3

Characterizing Group Preferential Choice

The goal of this chapter is to characterize sources of variation among real-world Group Preferential Choice scenarios that might have implications for the design of visual support tools for preference synthesis. In particular, we examine the following for each scenario:

1. The nature of the decision problem in terms of alternatives, decision makers, and criteria
2. The nature of the individual preference models
3. The goals of the decision makers during preference synthesis
4. The decision making context

Section 3.1 presents the tentative data model for Group Preferential Choice that is grounded in the existing vocabulary of Multi-Attribute Decision Making (Section 2.1.2). This establishes the scope of our work and constrains which scenarios are suitable for analysis.

Section 3.2 analyzes seven real-world Group Preferential Choice scenarios that roughly conform to this model. These scenarios were selected to cover as much variation as possible.

Section 3.3 proposes revisions to the data model based on this analysis. The revised model is presented in full in Section 3.4.

Section 3.5 collates the scenario-specific goals into scenario-independent goals for preference synthesis in the context of Group Preferential Choice. This list of goals serves as input to the task analysis in Chapter 4.

Finally, Section 3.6 discusses contextual features of the seven scenarios, such as the stakes involved, the expertise of decision makers, and the amount of time invested. It also summarizes the number of alternatives, decision makers, and criteria in each scenario.

3.1 Preliminary Data Model for Group Preferential Choice

We define Group Preferential Choice as a situation where two or more decision makers, each with his or her own explicit preferences,[1] must jointly choose from a set of alternatives.[2] There may or may not be explicit criteria. More formally, the Group Preferential Choice data model has the following elements:

- A set of **Alternatives** $A : \{a_1 ... a_m\}, m \geq 1$
- A set of **Decision Makers** $D : \{d_1 ... d_n\}, n \geq 2$, each of whom has a **Preference Model** (described below)
- A set of **Criteria** $C : \{c_1 ... c_r\}, r \geq 0$
- A set of **Primitive Criteria** $PC \subset C : \{pc_1 ... pc_s\}, s \geq 0$
- A set of **Abstract Criteria** $AC = C \setminus PC$
- A **Criteria Tree** T where the set of nodes in T is equal to C, and the set of leaf nodes in T is equal to PC. This models the criteria hierarchy.

Criteria may be *objective* or *subjective* depending on whether their outcomes are measurable facts or personal judgments. For example, *size of lawn* is an objective criterion, whereas *attractiveness of lawn* is a subjective criterion. Objective

[1] We use *explicit* to differentiate formally-expressed preferences from hidden or informally-expressed preferences (e.g. through conversation).

[2] Here, an *alternative* is an entity that the decision makers evaluate. The number of actual options might be larger, as decision makers might have the option of choosing multiple or no alternatives.

criteria outcomes are the same for all decision makers, while subjective criteria outcomes are individually defined by each decision maker.

For any objective criterion, there are the following additional elements:

- A **Domain** function $dom(pc)$ where $pc \in PC$, which defines the possible outcomes for criterion pc. The domain may be a discrete set (ordered or unordered) or a continuous range.
- An **Outcome** function $out(a, pc) \in dom(pc)$ where $a \in A$ and $pc \in PC$, which defines the outcome of alternative a on criterion pc.

In keeping with the standard definition of Multi-Attribute Decision Making [30] [35], this model excludes scenarios where the number of alternatives is infinite, or where different decision makers have different explicit criteria. We also exclude scenarios where the criteria outcomes are uncertain (that is, decisions under risk).

3.1.1 Preference Model Taxonomy

There are numerous ways that preferences can be modelled in formal decision processes [30] [58]. Here, we describe a few common models that are appropriate for a variety of evaluation contexts. These are organized into a hierarchy of increasing complexity based on *what* is evaluated and *how* the preferences are expressed.

Level P0: The decision makers evaluate the alternatives holistically.

a. Ordinal evaluation. Each decision maker *ranks* the alternatives. Preferences can be modeled as a function $r_d(a) \in [1, |A|]$ where:

1. $a \in A$ and $d \in D$
2. If a_{best} is the most preferred alternative for decision maker d, then $r_d(a_{best}) = 1$
3. $r_d(a_1) < r_d(a_2)$ if and only if d prefers a_1 to a_2

b. Cardinal evaluation. Each decision maker *scores* each alternative along a common linear scale. Preferences can be modeled as a function $s_d(a) \in [min, max]$ where:

1. $a \in A$ and $d \in D$

2. *min* and *max* are the minimum and maximum points on a linear scale common to all decision makers

Level P1: The decision makers evaluate each alternative with respect to each criterion.

a. Ordinal evaluation. Each decision maker *ranks* the alternatives with respect to each criterion. Preferences can be modeled as a function $r_d(a, pc) \in [1, |A|]$ where:

1. $a \in A$, $d \in D$, and $pc \in PC$
2. If a_{best} is the most preferred alternative for decision maker d on criterion pc, then $r_d(a_{best}, pc) = 1$
3. $r_d(a_1, pc) < r_d(a_2, pc)$ if and only if d prefers a_1 to a_2 on criterion pc

b. Cardinal evaluation. Each decision maker *scores* each alternative with respect to each criterion along a common linear scale. Preferences can be modeled as a function $s_d(a, pc) \in [min_{pc}, max_{pc}]$ where:

1. $a \in A$, $d \in D$, and $pc \in PC$
2. min_{pc} and max_{pc} are the minimum and maximum points on a linear scale for pc common to all decision makers

b+w. Same as above, with the addition of *weights* specifying the relative value of switching from the worst to the best outcome on each criterion. This can be modeled as a function $w_d(pc) \in [0, 1]$, where $d \in D$, $pc \in PC$, and

$$\sum_{i=1}^{|PC|} w_d(pc_i) = 1 \tag{3.1}$$

At this level, the raw (unweighted) preferences are specified by the function $uws_d(a, pc)$, while the weighted preferences are specified by the function $s_d(a, pc) = uws_d(a, pc) * w_d(pc)$.

Level P2: The decision makers evaluate each possible outcome of each criterion.

a. Ordinal evaluation. Each decision maker *ranks* the possible outcomes of each criterion. (This is only applicable for criteria with discrete domains.) Preferences can be modeled as a function $r_d(out, pc) \in [1, |dom(pc)|]$ where:

1. $d \in D$, $pc \in PC$, and $out \in dom(pc)$
2. If out_{best} is the most preferred outcome for decision maker d on criterion pc, then $r_d(out_{best}, pc) = 1$
3. $r_d(out_1, pc) < r_d(out_2, pc)$ if and only if d prefers out_1 to out_2 on criterion pc

b. Cardinal evaluation. Each decision maker *scores* each possible outcome of each criterion along a common linear scale.[3] Preferences can be modeled as a function $s_d(out, pc) \in [min_{pc}, max_{pc}]$ where:

1. $d \in D$, $pc \in PC$, and $out \in dom(pc)$
2. min_{pc} and max_{pc} are the minimum and maximum points on a linear scale for pc common to all decision makers

b+w. Same as above, with the addition of *weights* specifying the relative importance of each criterion. More precisely, a weight is the relative value of switching from the worst to the best outcome on each criterion. This can be modeled in the same manner as Level P1b+w.

At this level, the raw (unweighted) preferences are specified by the function $uws_d(out, pc)$, while the weighted preferences are specified by the function $s_d(out, pc) = uws_d(out, pc) * w_d(pc)$.

Recommended Usage

This taxonomy is intended to be *descriptive* in that it captures several models that are used in practice. Here, we briefly discuss a few *prescriptive* considerations pertaining to preference models.

Raw preference data should be collected at the lowest level possible given the criteria. Level P2 is recommended whenever alternative outcomes can be defined globally, which is the case when the criteria are objective. This eliminates the bias associated with direct evaluation of alternatives. Otherwise, preferences must be collected at Level P1 (when the criteria are subjective) or Level P0 (when there are no explicitly-defined criteria). A decision maker's preferences may span multiple levels of the taxonomy if there is a mix of subjective and objective criteria.

[3] This mapping from outcomes to scores is often called a **Score Function**, and the minimum and maximum values are typically 0 and 1.

If different scales are used for different criteria at Levels P1b or P2b, the scores must be normalized. Additionally, in forced-choice scenarios where decision makers must select at least one alternative, it is customary to scale decision makers' scores such that the best and worst alternatives for each criterion receive the minimum and maximum scores on the scale (see Campbell River, Section 3.2.3). This ensures that differences are maximally emphasized in the problem space. However, in scenarios where decision makers may elect to choose none of the alternatives, then absolute performance matters, and the original assessments should be preserved (see Faculty Hiring, Section 3.2.2).

At Level P1b, this can be achieved simply by scaling the scores such that:

1. If a_{best} is the most preferred alternative for decision maker d on criterion pc, then $s_d(a_{best}, pc) = max_{pc}$
2. If a_{worst} is the most preferred alternative for decision maker d on criterion pc, then $s_d(a_{best}, pc) = min_{pc}$
3. $r_d(out_1, pc) < r_d(out_2, pc)$ if and only if d prefers out_1 to out_2 on criterion pc

At Level P2b, this also mandates restricting the domain of each criterion to those represented in the problem space. In other words, there must be a one-to-one relationship between each primitive criterion's domain and the set of outcomes achieved by the alternatives on that criterion. Then, each score function can be scaled such that:

1. If out_{best} is the most preferred outcome for decision maker d on criterion pc, then $s_d(out_{best}, pc) = max_{pc}$
2. If out_{worst} is the most preferred outcome for decision maker d on criterion pc, then $s_d(out_{worst}, pc) = min_{pc}$
3. Corollary: there must be at least two possible outcomes for each criterion, and no decision maker may be completely indifferent to the outcomes of any criterion.

At Level P1a and below, the criteria should be as *independent* as possible. That is, the performance of an alternative on one criterion should not change the way a

decision maker feels about its performance on another criterion. This ensures that the decision maker's total score for each alternative can be attained by summing over the criteria scores. The more independent the criteria, the more accurate the additive model will be.

Conversion Between Taxonomy Levels

Once preferences have been collected at a certain level, it is possible to move *up* the taxonomy by applying simple transformations to the data, as shown in Figure 3.1.

The left-to-right arrows indicate possible conversions between *numeric* levels of the taxonomy, each of which is coded in a different color. Preferences over alternative-criterion pairs (P1) can be derived from preferences over outcomes (P2) simply by looking up the score/rank of the alternative's outcome on that criterion. Preferences over alternatives only (P0) can be derived from preferences over alternative-criterion pairs (P1) by aggregating over criteria. Conversion from P1b to P0b is achieved by mapping the criteria scores to a common scale (if they are not on the same scale already) and summing over the normalized scores. Conversion from P1a to P0a involves combining ranks to form a new ranking. There are a number of established techniques for doing this, but none of them are guaranteed to satisfy all plausible fairness properties [3], so trade-offs must be considered. The dashed arrow is used to convey this ambiguity.

The right-to-left arrows show possible conversions between *alphabetic* levels of the taxonomy (*a*, *b*, and *b+w*). The conversion from *b* levels to *a* a levels is straightforward, since a set of scores implies a ranking. The dotted arrow between Levels P2b to P2a means that this is only applicable for criteria with finite domains. The conversion from *b+w* to *b* involves calculating a weighted criterion score by multiplying the unweighted criterion score by the criterion weight.

Finally, it is possible to move from Level P1a to P0b by converting ranks across criteria into a numeric score for each alternative. One of the simplest and most widely-used methods of doing so is the Borda count, which gives each alternative one point for every alternative it beats on each ballot (in this case, each criterion) [21]. Again, there is no single way to meaningfully convert a set of ranks into a

score, so a dashed arrow is used to capture this ambiguity.

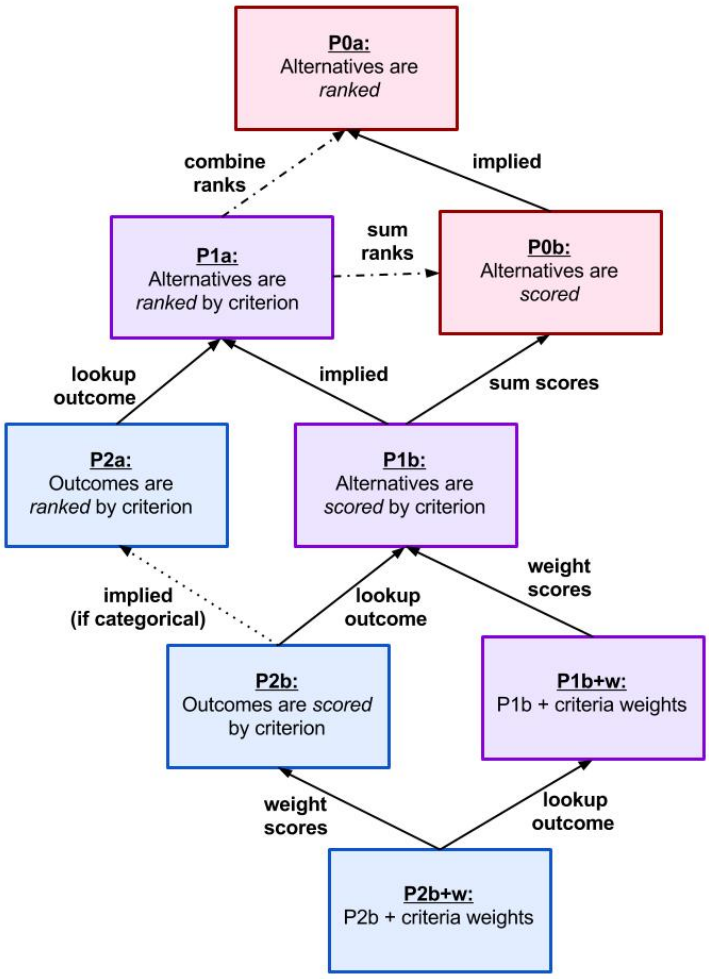


Figure 3.1: Preference Model Taxonomy. The numeric component of each level reflects *what* is evaluated (P0: alternatives, P1: alternatives by criterion, P2: outcomes by criterion), whereas the alphabetic component encodes *how* they are evaluated (a: ordinal, b: cardinal, b + w: cardinal + criteria weights). An upward arrow means that the level below implicitly encodes the level above, as discussed in the text.

Simplifying Assumptions

In order to constrain the scope of the analysis, we make the following tentative assumptions regarding the nature of the preference models:

1. The preferences do not span multiple levels. That is, there in not a mix of objective and subjective criteria or ordinal and cardinal evaluations.
2. All decision makers express their preferences at the same levels of the taxonomy.
3. The preferences are *complete*, that is:
   (a) At level $P0$, every decision maker ranks (or scores) every alternative.[4]
   (b) At level $P1$, every decision maker ranks (or scores) every alternative with respect to every criterion.
   (c) At level $P2$, every decision maker ranks (or scores) every outcome for every criterion.[5]
4. The preferences and weights are treated as *certain*. That is, there is no fuzziness.

Whether or not these assumptions are realistic varies from situation to situation. We revisit this taxonomy and list of assumptions in Section 3.3 after analyzing several real-world scenarios.

3.2 Seven Real-World Scenarios

This section describes seven real-world Group Preferential Choice scenarios. Four were assessed through one-on-one interviews with decision makers and the remainder were drawn from secondary sources. We address the following questions for each scenario:

1. What is the decision problem, and what is the decision-making process?
2. What is the formal description in terms of the data model from Section 3.1?

[4] In the case of P0a, this means that the preference model must specify a *total order* over the alternatives, but not necessarily a *strict total order* (which disallows ties).

[5] In the case of continuous domains, a complete score function may be attained by extrapolating from scores on a few sample points.

3. What are the goals during preference synthesis, and how are they achieved?
4. What are other relevant characteristics of the decision making context? In particular:
   (a) Is this decision made in a professional or casual setting?
   (b) How high are the stakes?[6]
   (c) How often does this decision recur?
   (d) How much time is devoted to preference synthesis?
   (e) Are the decision makers familiar with MCDA?

The first aim of this analysis is to validate and refine the data model. Section 3.3 proposes revisions to the data model to capture all relevant information and sources of variation. The updated model is presented in Section 3.4.

The second aim is to identify key preference synthesis goals and specific tasks that support them. Section 3.5 summarizes these findings.

The final aim is to characterize contextual factors that could inform system design. These findings are summarized in Section 3.6.

3.2.1 Best Paper at a Conference

This scenario was characterized through a one-on-one interview with a faculty member that was involved in selecting the best paper at a conference. Five researchers were tasked with choosing two papers for the best paper award out of four candidates that had been selected by the program chairs.

1. Decision Process

The five researchers met in person to choose the two best papers. They each ranked the papers according to their preferences, with ties permitted. Then, they summarized their rankings and had a discussion.

[6] For this question, the following broad categories suffice for this analysis:
- **Low:** minor impact on a few individuals
- **Medium:** major impact on a small organization or a few individuals
- **High:** major impact on a large organization ($>$ 100 members)
- **Very High:** major impact on multiple large organizations or the general public

2. Formal Data Description

The decision makers were the five researchers, and the alternatives were the four papers. There were no explicit criteria. Each researcher ranked the alternatives, which corresponds to level **P0a** of the Preference Model Taxonomy.

3. Preference Synthesis Goals

The overarching goal was to arrive at a consensus through focused discussion.

To achieve this, the decision makers combined their ranks in a spreadsheet with decision makers on rows, papers on columns, and ranks in cells (Figure 3.2). In the event of ties, the average of the spanned ranks was assigned to each of the tied papers. For instance, if Papers A - D were ranked 1, 2, 2, and 3 respectively, the adjusted ranks would be 1, 2.5, 2.5, and 4. A sum of ranks was computed for each paper to assess overall performance.

	A	B	C	D	E	F	G
1		Person 1	Person 2	Person 3	Person 4	Person 5	**Rank Sum**
2	Paper A	2	1.5	1	3	3	**10.5**
3	Paper B	1	1.5	2	1	1	**6.5**
4	Paper C	3	4	3	3	2	**15**
5	Paper D	4	3	4	3	4	**18**

Figure 3.2: Spreadsheet of ranks for each paper by researcher. (This is not the actual data.)

The researchers used the spreadsheet to identify disagreement among themselves, taking note of papers with high variability in rank. This focused the discussion on contentious points and encouraged the decision makers to reflect on their own assessments.

A pair of papers was chosen that minimized the total rank sum while also adhering to certain constraints (in this case, no two papers from the same author were to be selected). The decision makers agreed that the process was efficient and systematic compared to less formal approaches. The goals and tasks are summarized in Table 3.1.

Table 3.1: Preference Synthesis Goals and Tasks for Best Paper Scenario

	High-Level Goals	**Supporting Activities and Tasks**
G1	Reach consensus	Discussion, focused around **G2 - G5**.
G2	Identify papers with best overall performance	**T1:** Compute rank sum for each paper **T2:** Compare papers with respect to rank sum
G3	Identify disagreement among decision makers	**T3:** Compare paper ranks across decision makers **T4:** Identify rank discrepancies across decision makers and papers
G4	Encourage reflection on individual preferences	**T3:** Compare paper ranks across decision makers **T5:** Identify discrepancies between a particular decision maker's rankings and others' rankings
G5	Understand reasons for disagreement	Discussion

4. Contextual Features

This is a medium-stakes decision made in a professional setting over the course of a one-hour meeting. The decision is made annually, although the exact scenario may vary from year to year. The decision makers do not typically have MCDA knowledge.

3.2.2 Faculty Hiring

This scenario was characterized through one-on-one interviews with four faculty members of a research department at a major university. This department follows a semi-formal process to evaluate candidates for open faculty positions. The process is overseen by a hiring committee consisting of select faculty members and students. Two of the interviewees were members of the hiring committee, and the other two were voting members of the department.

1. Decision Process

The decision process has roughly four stages.

In the first stage, the candidate pool is winnowed via process of elimination. The applications are screened, and select candidates are asked to send letters. A subset of these candidates are contacted for Skype interviews. The candidates that pass the Skype interview are invited to visit the department in person.

In the second stage, the short-listed candidates give a talk at the department,

meet students, and have one-on-one meetings with faculty. The department is invited to evaluate the candidate using a standardized form. Around 50 - 100 opinions are collected this way.

In the third stage, the hiring committee meets to decide who will receive an offer. If more than one candidate is approved for an offer, then the committee decides the order in which the offers will be given.

In the final stage, the committee presents its recommendation at a department meeting. Faculty members may vote to approve, disapprove, or abstain. These votes are consulted by the department head, who makes the final decision.

2. Formal Data Description

The overarching process is composed of many distinct decision problems. However, we focus on Stages 3 and 4, as these make use of formal evaluations collected from the department.

In this context, the alternatives are the short-listed candidates. The explicit criteria are Research, Communication, Compatibility, Maturity, Research Fit, and Teaching Fit. In Stage 3, the decision makers are the members of the hiring committee. In Stage 4, the decision makers are the voting faculty members.[7]

Preferences are collected using a standard form that can be filled out by anyone in the department. It consists of 6-point scales for each of the six criteria. There is also an 'NA' (not applicable) option for each scale. Each scale is accompanied by a text field in which the user can justify their rating. Additionally, each evaluator is asked to rate their confidence in their evaluation as either 'Low', 'Medium', or 'High'. This preference model corresponds to level **P1b** of the taxonomy.

3. Preference Synthesis Goals

Prior to the hiring committee meeting, the quantitative results of the department survey are summarized in the form of histograms, as shown in Figures 3.3 and 3.4.

In Stage 3, committee members consider these results and discuss their own opinions. In addition to the explicit criteria, the committee members consider ad-

[7] Ultimately, the final decision is made by the department head, but this distinction is not critical to this analysis.

ditional factors about the candidates, such as number of papers published at top conferences and how well they complement current faculty members. Faculty members that work in the same area as the candidate (the 'in-area' faculty) are given more weight in the discussion.

Both interviewees on the hiring committee reported that the subjective feedback provided in text fields or expressed in conversation was much more important than the quantitative summaries. Because this is a high-stakes decision with a high level of personal investment, subtleties are taken seriously.

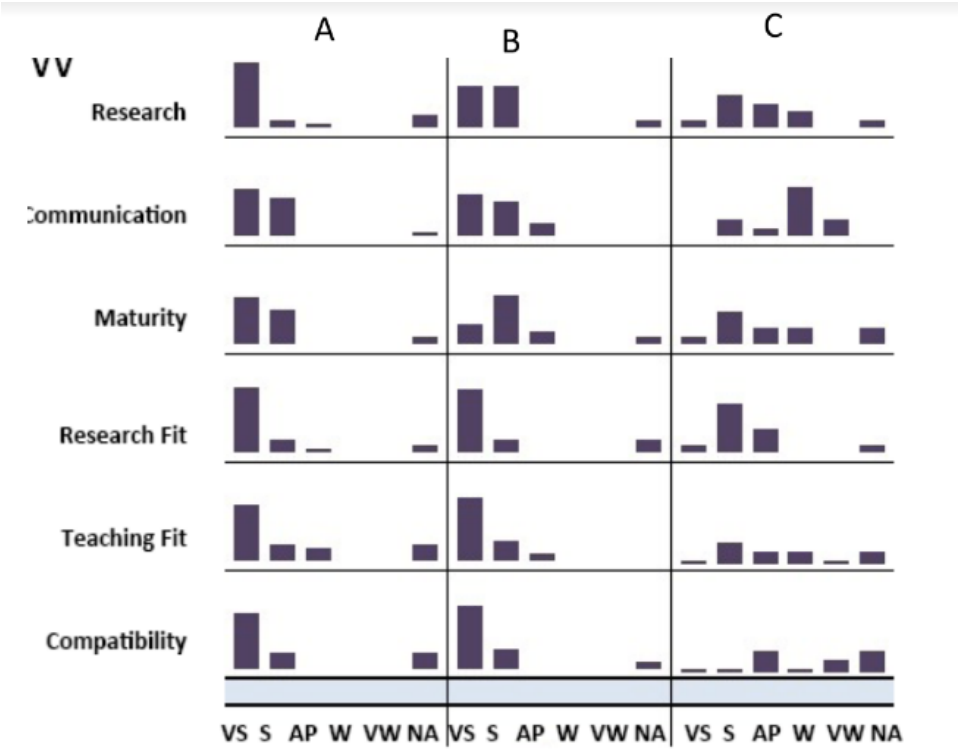


Figure 3.3: Here, the results are summarized as a matrix of histograms with criteria on rows and candidates on columns. Each bar encodes the frequency with which that candidate scored at a particular level for that criterion. The levels are VS: Very Strong, S: Strong, AP: At Par, W: Weak, VW: Very Weak, and NA: Not Applicable.

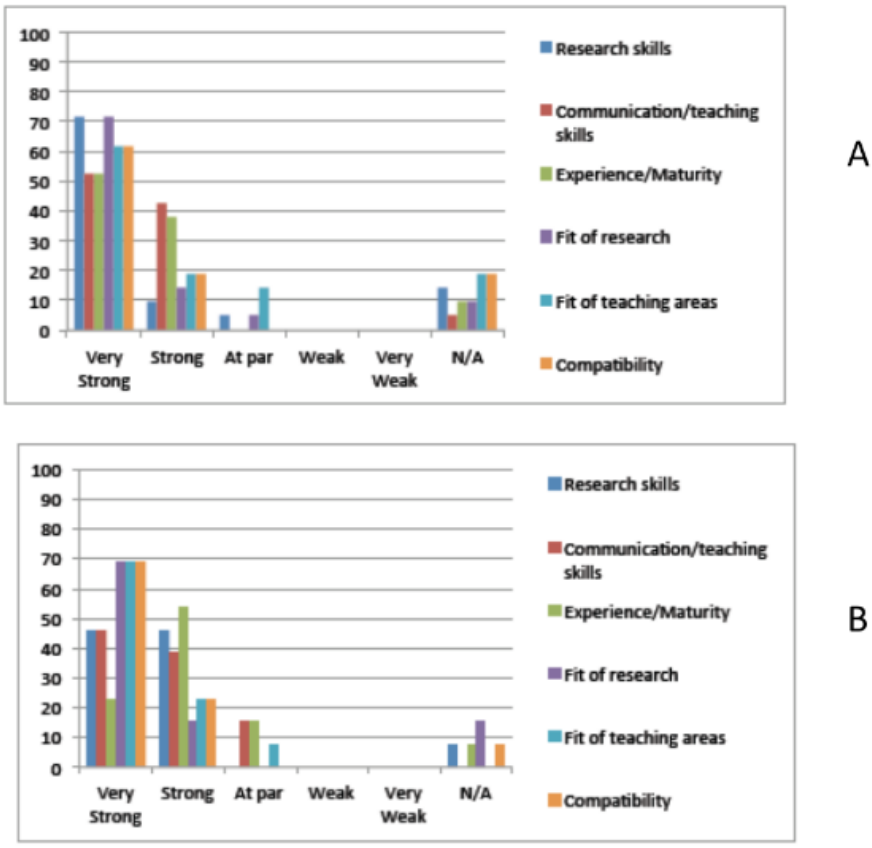


Figure 3.4: Here, the results are summarized separately for candidates A and B. The x-axis groups the results by score level, and the bars are colour-coded by criterion. Each bar encodes the percentage of reviewers that gave that candidate that score for that criterion. The levels are VS: Very Strong, S: Strong, AP: At Par, W: Weak, VW: Very Weak, and NA: Not Applicable.

In Stage 4, Figures 3.3 and 3.4 (or equivalent) are presented to the department along with selected text excerpts in order to justify the hiring committee's recommendation.

Two voting faculty members were interviewed following a department meeting. In order to assess how they used the information presented, they were asked the following questions:

1. Did the visual summary influence your decision? If so, how?
2. Is there any information that would have helped you make your decision?

Both interviewees said that the summary confirmed what they already suspected - that people generally liked the candidate. One interviewee said that the summary influenced him by corroborating his viewpoint. The other said it did not influence her because she was already convinced, but that it might have if it had revealed more controversy.

Both interviewees expressed a desire to see the breakdown of scores by role (student, in-area faculty, and other faculty) and confidence level (low, medium, and high), which was not provided by the visualization. One interviewee would have liked to read specific comments by people that gave negative feedback.

Table 3.2 provides a summary of the preference synthesis goals and tasks. For simplicity, the goals and tasks of Stage 3 and Stage 4 have been combined.

Table 3.2: Preference Synthesis Goals and Tasks for Faculty Hiring Scenario

	High-Level Goals	**Supporting Activities and Tasks**
G1	Reach consensus through approval voting	Facilitated discussion around **G2 - G5**.
G2	Gauge candidate performance across criteria	**T1:** Count frequency of scores for that candidate on each criterion **T2:** Inspect distribution of scores for that candidate on each criterion
G3	Identify discrepancies in performance across candidates	**T3:** Compare distribution of scores across candidates for each criterion
G4	Identify evaluators that might not be satisfied with a particular candidate	**T4:** Count frequency of 'disagree' and 'strongly disagree' outcomes for each criterion and candidate (Identification of individual evaluators not supported)
G5	Identify disagreement across evaluator roles (student, in-area faculty, others)	Not supported
G6	Identify disagreement across evaluator confidence levels (low, medium, high)	Not supported
G7	Understand reasons for evaluator opinions	Consult textual feedback (Stage 3 only)
G8	Understand reasons for voter opinions	Discussion
G9	Give more weight to expert opinions	Implicit (voters do this mentally) Grant experts dedicated time to make a case

4. Contextual Features

This is a high-stakes decision made in a professional setting. The hiring committee meeting lasts about 2 hours, while the department meeting devotes about 1 hour to the faculty hiring segment. This scenario recurs once or twice a month during recruiting season. The decision makers do not typically have MCDA knowledge.

3.2.3 Campbell River Watershed

This scenario was characterized by watching a Webinar prepared by Compass, a Vancouver-based consulting firm that helps organizations tackle high-stakes decision problems using structured decision making techniques. In this scenario, Compass oversaw the selection of a new operation strategy for the Campbell River hydroelectric facilities on Vancouver Island. The process took three years and involved numerous stakeholders, including the Federal and Provincial Government, BC Hydro, local businesses, and First Nations.

1. Decision Process

The Campbell River Watershed is a major hydroelectric facility on Vancouver Island. The region is also one of cultural significance to First Nations peoples, home to multiple salmon species, and a popular recreation destination.

At the time, the Watershed consisted of three reservoirs and three river divisions. The goal was to devise a new operation strategy that would better appeal to a diverse set of interests.

A list of initial issues was collected through a series of public open houses. These issues were pared down and organized by interest group: flooding and erosion, fish and wildlife, recreation, water quality, and financial. Then, special subcommittees were formed for each interest group to identify key objectives and describe them in terms of measurable attributes. This process resulted in twelve objective-attribute pairs. The final set of criteria was produced by listing applicable objectives at each of five watershed locations, yielding a total of fifteen criteria (Figure 3.5). A score function for each attribute was developed by an expert, which would apply to all stakeholders.

Meanwhile, six alternatives were devised by considering feasible strategic ad-

justments at each location in the watershed. The outcomes on each objective were estimated, and these were arranged in a consequence table (Figure 3.5).

Finally, fifteen stakeholders from different interest groups met to evaluate the alternatives with respect to their individual preferences. The process is described below. The best two alternatives were identified in this manner, and these were taken back to the drawing board for refinement. The final choice was made by consensus voting.

		Alternatives					
Objective	Attribute	E	F	G	H	I	J
Upper Campbell / Buttle Lake							
Erosion - Days / Year	weighted days (220 and 221 m)	37	13	4	3	3	3
Recreation - Days / Year	weighted days (217.5, 218.5, 200m by seaso	43	40	106	158	158	158
Effective Littoral Zone	hectares	91	107	93	214	215	220
Lower Campbell / McIvor / Fry							
Erosion - Days / Year	weighted days (177.4 and 178.3 m)	3	27	13	0	0	0
Recreation - Days / Year	weighted days (175.75 - 177.8 by season)	115	43	83	167	170	167
Spawning Habitat - Cutthroat	% Available Habitat	78	18	95	79	79	78
Spawning Habitat - Rainbow	% Available Habitat	26	3	49	49	47	50
Campbell River							
Flooding - Total Days	weighted days (300, 453, 530 cms)	34	48	24	59	59	59
Recreation - Days / Year	weighted days (28 cms - 80 cms)	66	83	51	81	79	81
Total Spill Days - All Species	days (Q>340cms, Sept 22 - April 15)	118	214	102	176	177	176
Spawning Habitat - All Species	% successful redds (Chum as indicator)	55	89	78	59	59	59
Rearing Habitat - All Species	"Average" risk index (scale 0 - 1)	0.53	0.48	0.53	0.50	0.49	0.49
Salmon River							
Canoe Route - Days / Year	days (Q<6cms, April 1 - Oct 22)	162	167	153	204	183	204
All Habitat - All Species	"Average" risk index (scale 0 - 1)	0.54	0.47	0.44	0.48	0.47	0.47
System-Wide							
Power / Financial	Annual Revenue M $ / Year	68.5	64.6	68.6	65.1	65.3	64.1

Figure 3.5: Consequence table for six operation strategies on fifteen criteria (derived by listing applicable objective-attribute pairs for each of five watershed locations).

2. Formal Data Description

The alternatives were the six operation strategies, and the decision makers were the fifteen stakeholders. The criteria were the twelve objectives.

Two types of preferences were collected: holistic and criteria-based. Holistic preferences were obtained by asking users to rank the alternatives in order of preference, with ties allowed. Then, they were asked to assign the highest ranked alternative a score of 100 and score the others relative to that. These preferences correspond to levels **P0a** and **P0b** in the taxonomy.

Criteria-based preferences were obtained by collecting weights using the SMARTER

technique [23]. This corresponds to level **P2b+w** of the taxonomy, where the weights are supplied by individual decision makers and the score functions by an expert. A score for each alternative was calculated using Simple Additive Weighting (SAW) [30] over the weighted criteria scores.

The reason for collecting two types of preferences was to validate the model. Discrepancies between the two outcomes would indicate that one or both models is flawed - either the decision maker did not consider all criteria in her holistic assessment (the holistic model is flawed) or some criteria of interest to the decision maker are missing from the set (the criteria-based model is flawed). The results of this comparison were presented to each user in the form of a line graph (Figure 3.6).

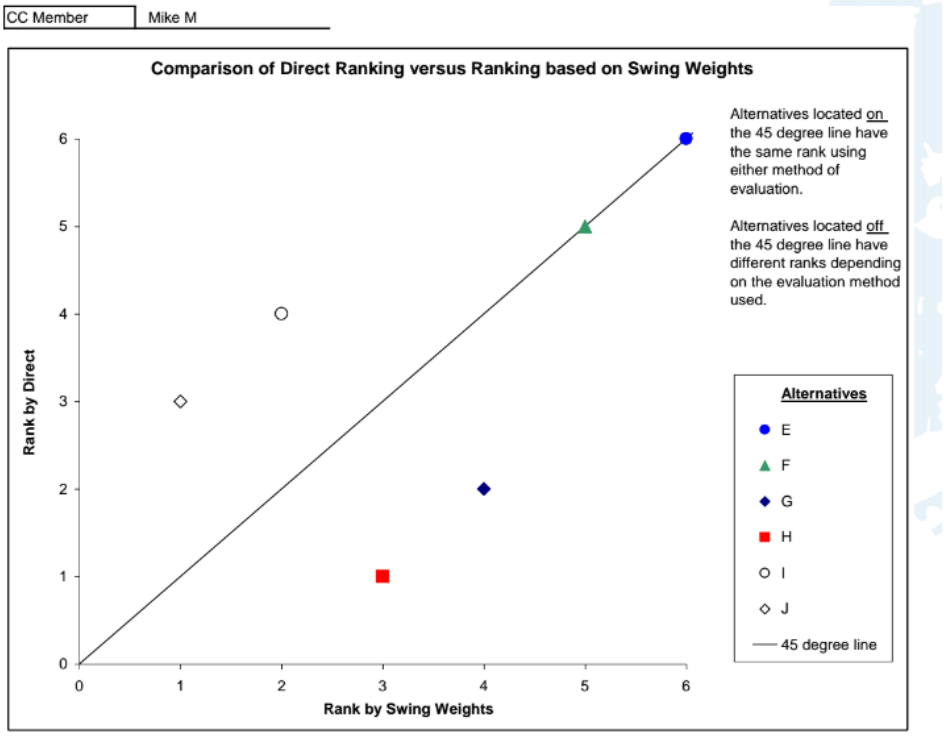


Figure 3.6: Comparing two preference models for a participant.

3. Preference Synthesis Goals

The ultimate goal of preference synthesis was to support negotiation and help the stakeholders reach consensus. To this end, Compass provided each user with two sheets of paper, each featuring a different graphic.

Figure 3.7 shows each person her weights in the context of the range of weights for the whole group. The intent of this was to help the decision makers see how their priorities compare to the rest of the group, and to reveal criteria for which there was a wide range of opinions.

Figure 3.8 shows the ranking of each alternative for each person and scoring method. The purpose of this was to help the decision makers identify high-performing alternatives at a glance, and then to see which decision makers are not content with the top alternatives.

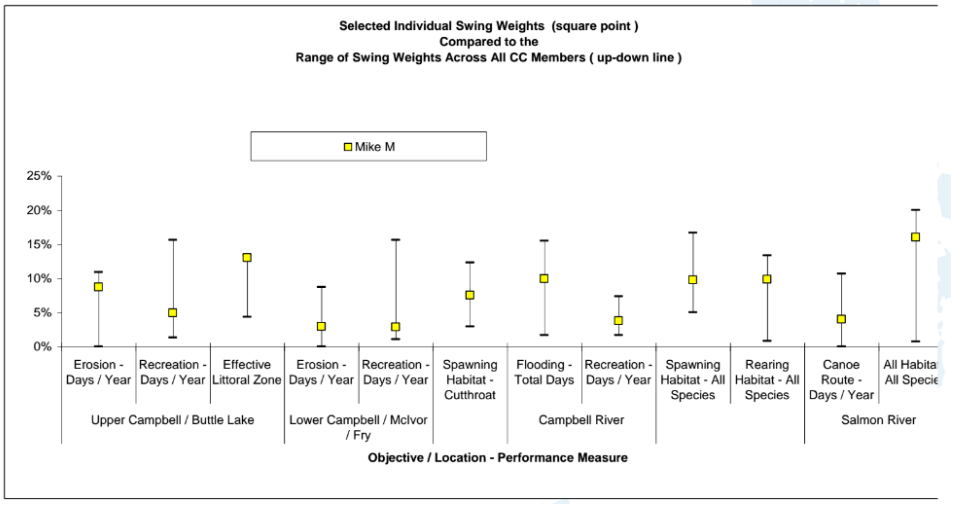


Figure 3.7: The range of weights assigned to each criteria across users. A yellow square denotes the weight assigned by that participant.

Rank of Alternatives by Stakeholder and by Method							
		Alternatives					
Stakeholder	Weighting/ Ranking Method	E	F	G	H	I	J
1	Direct	6	5	2	1	4	3
	Swing	6	5	4	3	2	1
2	Direct	6	5	1	3	4	2
	Swing	6	5	4	3	2	1
3	Direct	6	3	5	1	2	4
	Swing	6	5	2	4	1	3
4	Direct	5	6	4	1	3	2
	Swing	5	6	4	1	3	2
5	Direct	2	3	1	4	4	4
	Swing	5	6	4	2	3	1
6	Direct	3	4	1	2	4	6
	Swing	5	6	1	2	3	4
7	Direct	6	2	1	3	3	3
	Swing	6	5	4	3	2	1
8	Direct	2	3	1	4	4	4
	Swing	6	5	4	3	2	1
9	Direct	2	6	1	5	4	3
	Swing	5	6	1	3	2	4
10	Direct	3	2	1	4	5	6
	Swing	6	5	1	3	2	4
11	Direct	5	6	4	1	2	3
	Swing	5	6	4	1	3	2
12	Direct	6	3	2	4	5	1
	Swing	6	5	4	3	2	1
13	Direct	6	5	4	3	2	1
	Swing	6	5	4	2	3	1
14	Direct	2	5	1	4	3	6
	Swing	2	6	1	4	3	5
15	Direct	2	3	1	4	5	6
	Swing	5	6	4	1	3	2

Figure 3.8: The performance of each alternative for each user's two preference models. The number in each cell represents rank, whereas color encodes score.

After a period of discussion, alternatives G and H were selected for further refinement. The goals and tasks are summarized in Table 3.3.

Table 3.3: Preference Synthesis Goals and Tasks for Campbell River Scenario

	High-Level Goals	**Supporting Activities and Tasks**
G1	Reach consensus	Discussion, focused around **G2 - G6**.
G2	Identify differences in priorities among decision makers	**T1:** Inspect distribution of weights for each criterion
G3	Identify differences in priorities between self and others	**T2:** Compare own weight to distribution of weights for each criterion
G4	Identify strategies with best overall performance	**T3:** Compare strategy scores and ranks across decision makers
G5	Identify decision makers that may not be satisfied with a particular strategy	**T4:** Identify decision makers that assigned a low score to that strategy
G6	Understand reasons for disagreement	Discussion

4. Contextual Features

This was a one-time, very high-stakes decision made in a professional setting. Preference synthesis took an entire day. The decision makers did not have MCDA knowledge themselves, but they were assisted by MCDA experts.

3.2.4 MJS77 Project

Dyer and Miles describe the first recorded application of MCDA methods to a real-world group decision problem [22]. The decision makers were NASA scientists, and the task was to choose a pair of trajectories for the Jupiter/Saturn 1977 (MJS77) Project. The resulting missions were later named Voyager 1 and Voyager 2.

1. Decision Process

The Jet Propulsion Laboratory (JPL) of NASA was tasked with selecting a pair of trajectories (flight paths) for two spacecrafts that would be launched within days of each other. The trajectories had to be chosen jointly, as the merits of one depended on the other.

The choice of trajectory is a major factor in determining the mission's success,

so the JPL recruited a team of eighty scientists for advice. The scientists were divided into ten teams by specialty. Each team was represented by its leader in an inter-team committee called the Science Steering Group (SSG).

An initial set of possible trajectory pairs was developed through back-and-forth consultation between the JPL and science teams. This resulted in a set of 32 candidate pairs.

Then, each science team evaluated each of the candidate pairs by ranking and scoring them holistically, as described below. Each team was permitted to use whatever decision making process it wished. The JPL synthesized the results and presented them to the SSG. The final trajectory was selected by the SSG following a discussion.

2. Formal Data Description

The alternatives were the 32 trajectory pairs, and the decision makers were the ten science teams. Each team ranked the trajectory pairs, with ties permitted. This corresponds to level **P0a** in the taxonomy.

Scores on a cardinal scale were obtained using von Neumann-Morgenstern lotteries [60]. This elicitation method was selected due to its "theoretical consistency, wide acceptance, and ease of implementation" [22]. This yielded an expected utility score of 0 - 1 for each pair. This corresponds to level **P0b** in the taxonomy.

3. Preference Synthesis Goals

The synthesis of preferences was carried out by the JPL. The ranks were aggregated by summing across teams and dividing by the number of trajectory pairs. In the event of ties, the average of the spanned ranks was used. This aggregation method is identical to that used in the Best Paper scenario, with an additional rescaling step at the end.

The JPL tested eight different ways of aggregating the cardinal scores. In particular, they experimented with team weights, normalization procedures, and aggregation techniques. The purpose of this was to perform sensitivity analysis over different collective choice rules. The level of agreement was quantified using Kendall's coefficient of concordance, which came to 0.96. This is very high, and it

suggests that this problem was not especially sensitive to any of these factors.

Finally, the results of this analysis were presented to the SSG in the form of Figures 3.9 and 3.10. Three trajectory pairs (26, 29, and 31) were found to be in the top three for all collective choice rules. The scientists discussed the pros and cons of these three trajectories, and all but one team eventually agreed that trajectory 26 would be acceptable. Trajectory 26 was modified to address the concerns of the disapproving team and then approved by the project manager. The goals and tasks of preference synthesis are summarized in Table 3.4.

Table 3.4: Preference Synthesis Goals and Tasks for MJS77 Scenario

	High-Level Goals	**Supporting Activities and Tasks**
G1	Reach consensus	Facilitated discussion around **G2 - G5**.
G2	Identify high-performing trajectories	**T1:** Identify trajectories ranked in top three across collective choice rules
G3	Identify teams that may not be satisfied with a particular trajectory	**T2:** Identify teams that assigned that trajectory a low ranking (under 10)
G4	Understand sensitivity of outcome to collective choice rule	**T3:** Compare trajectory rankings across nine collective choice rules
G5	Understand reasons for each team's rankings	Discussion

TABLE III

THE COLLECTIVE CHOICE RANKINGS AND VALUES

Collective choice rule	Form	Rank sum		Additive										Nash			
	$u^i(l^i_{32})$			$p_\phi{}^i$		0.0		0.6 or 0.8		$p_\phi{}^i$		0.0		0.6 or 0.8		$p_\phi{}^i$		0.6 or 0.8	
	λ^i weighting			1.0		1.0		1.0		1.0 or 2.0		1.0 or 2.0		1.0 or 2.0					
		Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value	Rank	Value
Trajectory pairs	31	1	0.822	2	0.887	1	0.724	1	0.901	1	0.871	1	0.691	1	0.884	1.5	0.877	1	0.892
	29	2	0.797	3	0.875	3	0.692	3	0.884	3	0.852	3	0.638	3	0.860	3	0.865	3	0.874
	26	3	0.795	1	0.889	2	0.710	2	0.896	2	0.870	2	0.676	2	0.878	1.5	0.877	2	0.886
	27	4	0.719	4	0.856	4	0.641	4	0.871	4	0.833	4	0.596	4	0.848	4	0.839	4	0.856
	5	5	0.683	6	0.791	6	0.555	8	0.841	6.5	0.765	8	0.502	12	0.813	6	0.776	11	0.829
	25	6	0.678	5	0.822	5	0.597	5	0.851	5	0.800	5	0.535	9	0.822	5	0.804	7	0.836
	35	7	0.655	7	0.757	8	0.511	6.5	0.846	6.5	0.765	7	0.517	5.5	0.833	8	0.745	6	0.839
	17	8	0.622	10	0.738	11	0.475	10.5	0.836	10	0.746	10	0.487	8	0.823	10.5	0.725	9.5	0.830
	8	9	0.611	8	0.755	9	0.488	10.5	0.836	8	0.756	11	0.486	10	0.820	7	0.746	9.5	0.830
	10	10	0.605	12	0.728	7	0.514	6.5	0.846	11	0.744	6	0.519	5.5	0.833	14	0.706	5	0.843

Figure 3.9: The scores for the top 10 trajectory pairs on each of nine collective choice rules [22].

TABLE IV

THE SCIENCE TEAM ORDINAL RANKINGS FOR PREFERRED TRAJECTORY PAIRS

		Collective choice ranking		Science team ordinal rankings									
		Rank sum	Additive $u^i(t^i_{22})=p_\phi{}^i$; $\lambda^i=1.0$	RSS	IRIS	ISS	PPS	UVS	CRS	LECP	MAG	PLS	PRA
Trajectory pairs	31	1	2	20.5	3.0	5.0	8.5	6.0	8.0	4.0	3.0	4.0	5.0
	29	2	3	20.5	5.0	19.0	6.5	9.0	3.0	2.0	2.0	4.0	4.0
	26	3	1	20.5	2.0	10.0	11.0	7.0	17.5	3.0	1.0	1.5	2.0
	27	4	4	20.5	1.0	30.0	16.0	3.0	17.5	1.0	4.0	4.0	3.0
	5	5	6	20.5	9.0	28.5	17.0	4.0	13.0	6.0	6.0	1.5	6.0
	25	6	5	20.5	7.0	28.5	25.0	9.0	1.0	5.0	5.0	11.0	1.0
	35	7	7	20.5	21.0	2.0	2.0	13.5	13.0	12.0	9.0	17.5	10.0
	17	8	10	20.5	4.0	13.5	6.5	13.5	24.5	9.0	14.0	17.5	8.0
	8	9	8	20.5	20.0	10.0	5.0	13.5	21.0	11.0	7.0	17.5	9.0
	10	10	12	3.0	11.0	24.0	8.5	19.5	2.0	20.0	24.0	17.5	7.0

Figure 3.10: Ordinal rankings for the top 10 trajectory pairs for each of the 10 science teams (RSS ... PRA) [22]. In the event of ties, the numeric score for each pair is the average of the spanned ranks.

4. Contextual Features

This was a one-time, very high-stakes decision made in a professional setting. Preference synthesis took several days. The decision makers did not have MCDA knowledge themselves, but they were assisted by MCDA experts.

3.2.5 Nuclear Crisis Management

In this study, Mustajoki et al. ran a two day workshop in which a group of participants planned countermeasures for a hypothetical nuclear emergency scenario [40]. It was one of the first attempts to demonstrate the efficacy of MCDA software in group decision making scenarios.

1. Decision Process

The participants in the study were authorities in nuclear emergency planning that would be responsible for devising a plan in the event of a real emergency.

The participants were split into six groups, and each group was assigned to a computer equipped with Web-HIPRE (HIerachical PREference analysis on the Web), a decision support application based on MAVT [32].

The facilitator described the hypothetical scenario: an accident had taken place in a nuclear power plant in Finland. It was now a week later, and the fallout covered a major milk production area. The group was tasked with choosing the best strategy to mitigate damage.

The alternatives had been developed during prior meetings with experts. There were four possible strategies: provision of uncontaminated fodder ('Fod'), processing of milk into other products ('Prod'), banning the milk ('Ban'), and doing nothing ('–'). The alternatives were six realistic pairs of strategies over two time periods: weeks 2 - 5 and 6 - 12 after the accident: ('–+–'), ('Fod+–'), ('Fod+Fod'), ('Prod+Fod'), ('Ban+Fod'), and ('Ban+Ban').

A preliminary set of criteria had also been developed during prior meetings with experts. The conference group deliberated and narrowed these down to seven.

Each group then used the software to supply its preferences, as described below. The results were presented by the facilitator, and then approval voting was carried out for each of the possible alternatives. Two of the alternatives, ('Fod+Fod') and ('Fod+Prod'), were unanimously approved.

2. Formal Data Description

The decision makers were the six teams, and the alternatives were six pairs of strategies.

There were seven criteria arranged hierarchically into three groups: Health (Thyroid Cancer, Other Cancers); Social-Psychological (Reassurance, Anxiety, Industry, Feasibility); and Cost (Cost). These constituted a mix of subjective and objective criteria, although the authors did not specify which were which.

For subjective criteria, the groups directly rated each alternative on a 0 - 1 scale. For objective criteria, a common score function was defined by experts. Weights for criteria were obtained using a SWING weighting technique [61]. Taken together, the preference model is a hybrid of **P1b+w** and **P2b+w**.

3. Preference Synthesis Goals

The overarching goal was to reach consensus through approval voting on the alternatives.

First, the alternative scores for each team were projected onto a screen one by one. The facilitator led a discussion of each, pointing out essential characteristics and explaining how different criteria contribute to the overall score. The facilitator performed sensitivity analysis to demonstrate how changing the criteria weights can affect the outcome.

Individual models were combined by computing a weighted sum of total scores for each alternative. The results of this were projected onto the screen (Figure 3.11). At first, equal weights were assigned to the groups. Then, the facilitator performed sensitivity analysis to demonstrate how changing the weights of the groups can affect the outcome.

Web-HIPRE provides a visual breakdown of the scores of each alternative by group, as seen in Figure 3.11. The teams can also see a breakdown by criteria, or even switch the bars and segments such that the total score for each criterion is broken down by alternative.

The participants discussed the results, paying particular attention to groups whose preferences did not align with the others (Group 2). Eventually, the groups arrived at a consensus, and two of the alternatives, ('Fod+Fod') and ('Fod+Prod'), were unanimously approved.

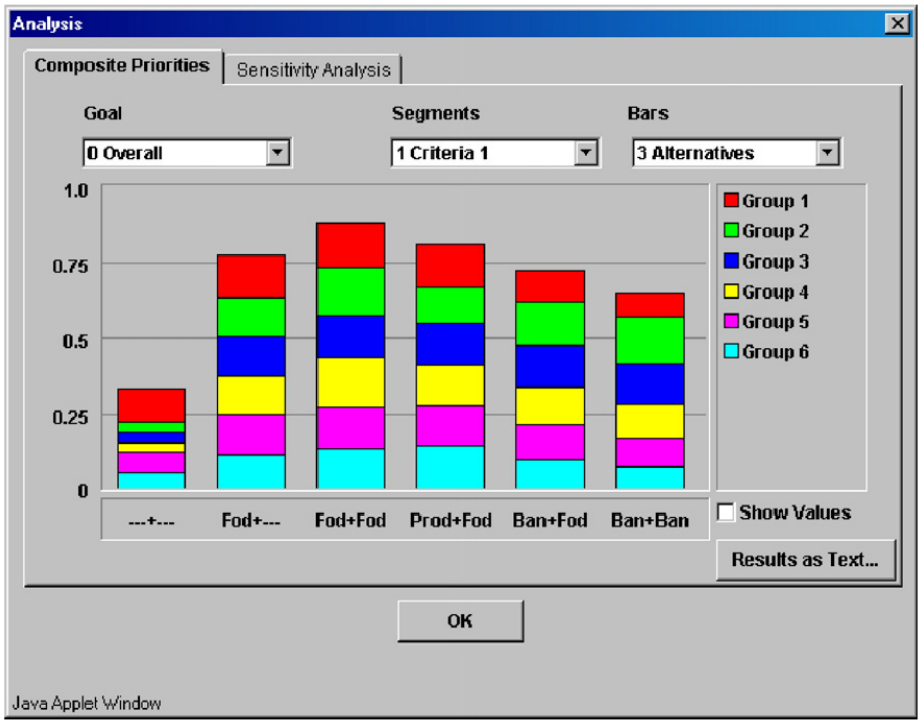


Figure 3.11: Score for each alternative, broken down by group [40].

The process was well-received by the participants, and they were satisfied with their final decision. However, they felt it would be better suited for planning in advance than in the event of a real crisis.

Table 3.5: Preference Synthesis Goals and Tasks for Nuclear Crisis Scenario

	High-Level Goals	**Supporting Activities and Tasks**
G1	Reach consensus through approval voting	Facilitated discussion around **G2 - G5**.
G2	Identify best performing strategy for a particular group	**T1:** Compute total score for each strategy for that group **T2:** Compare strategies with respect to total score for that group **T3:** Compare strategies with respect to criteria scores for that group
G3	Understand effect of criteria weights on outcome for particular group	**T4:** Perform sensitivity analysis on criteria weights for that group
G4	Identify best performing strategies overall	**T5:** Compute total score for each strategy **T6:** Compare total scores of each strategy
G5	Identify disagreements on overall strategy performance	**T7:** Compare strategies with respect to total score for each group
G6	Understand effect of group weights on outcome	**T8:** Perform sensitivity analysis on group weights
G7	Understand contribution of each group to total score for each alternative	**T9:** Inspect breakdown of total scores into scores for each group
G8	Understand reasons for disagreement	Discussion

4. Contextual Features

This was a simulation of a one-time, very high-stakes decision made in a professional setting. Preference synthesis took an entire day. The decision makers did not have MCDA knowledge themselves, but they were assisted by MCDA experts (which may or may not be feasible in the event of a real crisis).

3.2.6 Technology Selection at XpertsCatch

This scenario was characterized by observing a team meeting of the software recruitment start-up, XpertsCatch. During this meeting, the company decided which technology stack to use for their next product. The CTO and two senior employees were interviewed individually after the meeting.

This is not technically Group Preferential Choice since there was no formal preference modelling. However, the interviewees explained that it would have been feasible and useful to express their preferences formally, and they speculated about how they might do so in the future.

1. Decision Process

Prior to the meeting, the senior employees narrowed down their options to two stacks of interoperable technologies along six dimensions: language, database, data format, deploy target, back-end framework, and web server.

During the meeting, the CTO met with the engineering team, which consisted of two senior and two junior developers. The CTO and each engineer cast a vote for one of the two stacks and presented his or her supporting arguments. The CTO made the final decision, putting more weight on the arguments of the senior developers.

2. Formal Data Description

The decision maker was the CTO, and the alternatives were two possible stacks: (Javascript, MongoDB, JSON, Mobile HTML, Express, NodeJS) and (Python, MongoDB, XML, Android, Meteor, Apache).

Criteria and preferences were not explicitly modelled. However, the interviewees said that they implicitly evaluated the stacks based on the six technological dimensions and two whole-system criteria: learning curve and adaptability. Different interviewees expressed different priorities over the criteria. For instance, the CTO cared most about deployment target because it affects the target demographic, whereas the back-end developer cared most about language because it affects his day-to-day productivity.

The interviewees said that if they were to use explicit preference modelling, they would treat each of the six technological dimensions as objective criteria and each of learning curve and adaptability as subjective criteria. Additionally, they would use weights to capture their priorities. They all agreed that explicit preference modelling would have been helpful for their analysis. Such a model would correspond to a hybrid of levels **P2b+w** and **P1b+w**.

3. Preference Synthesis Goals

As there was no formal preference modelling, there was no formal synthesis of preferences. Preferences were shared through conversation, and the expertise of each stakeholder was taken into account. As such, the elicitation, evaluation, and

synthesis phases were intertwined.

Before the final decision was made, the CTO consulted with a developer that had voted for the other stack to confirm that he would accept the decision. He said that he would.

Table 3.6: Preference Synthesis Goals for XpertsCatch Scenario

	High-Level Goals	**Supporting Activities**
G1	Choose best technology stack for the company	Company meeting addressing **G2** - **G6**
G2	Identify most preferred stack for each employee	Elicit votes
G3	Identify most preferred stack overall	Count votes
G4	Understand reasons for each employee's preference	Hear supporting arguments
G5	Identify differences in preferences	Compare supporting arguments
G6	Identify employees that do not prefer a particular stack	Review votes
G7	Differentiate between senior and junior engineers	Implicit (in CTO's head)

4. Contextual Features

This is a medium-stakes decision made in a professional setting over the course of a one-hour meeting. This or similar decisions are made about once a year. The decision makers do not have MCDA knowledge.

3.2.7 Buying a Gift for a Colleague

In this scenario, a research lab chose a gift to buy for a recently-graduated colleague, Oscar.[8] It was characterized by interviewing the person that led the process, as well as two other members of the lab.

Like the XpertsCatch case, this was not technically Group Preferential Choice

[8] All names have been replaced

since there was no formal preference modelling. However, the interviewees were able to speculate about how formal preference models might have been useful.

1. Decision Process

The lab members agreed that each person would contribute $10 - $20 toward the gift. One colleague, Sayid, volunteered to lead the selection process.

First, he made a list of possible gifts within that price range. He did not have time to consult the whole group, so he asked two colleagues that were close friends of Oscar to help him narrow down the list. After brainstorming, they agreed on some criteria: the gift should be around $150, high-quality, long-lasting, useful, and aesthetically appealing. One friend believed that the usefulness of the gift was more important than its aesthetic appeal, whereas the other thought that the colleague might prefer an artistic gift since he did a lot of sketching for his PhD. After conversing for about ten minutes, they narrowed down the list to three options.

Then, Sayid arranged a meeting with the whole lab. He presented the options and explained the criteria that they had considered. The lab then voted on the three options, and the gift with the most votes was purchased.

2. Formal Data Description

The decision makers were the lab members (ten in total), and the alternatives were the three gifts. The criteria were cost, quality, durability, usefulness, and aesthetic appeal. Another criteria, size, was used to screen options: only options that could fit in his backpack were considered. There was no formal preference modelling.

Two interviewees said that formal preference modelling would have been a good way to collect more opinions. There was a mixture of objective and subjective criteria, as well as differences in opinion over which criteria were most important. Therefore, the appropriate preference model would be a combination of levels **P2b+w** and **P1b+w**.

3. Preference Synthesis Goals

As there was no explicit preference modelling, there was no formal synthesis of preferences. Sayid gave a verbal synthesis of his and the two friends' opinions

to the group. He presented the pros and cons of the three options, and the other colleagues used this information to decide how to vote.

Two of these colleagues were interviewed about the process. They were both happy with the outcome, but they wished that more opinions had been collected prior to the meeting. They noticed that not many people actively commented during the meeting, and they suspected this might have been due to the size of the group. One interviewee said that he would not feel comfortable expressing a contrary opinion in front of the others. The other interviewee said that she would put the most weight on the opinions of the organizer (Sayid) and Oscar's friends.

4. Contextual Features

This is a low-stakes decision made in a semi-professional setting over the course of a one-hour meeting. This or similar decisions are made about once a year. The decision makers do not have MCDA knowledge.

3.3 Data Model Revisions

This section proposes adjustments and extensions to the data model from Section 3.1 in order to fully and accurately capture the key aspects of all seven scenarios. These are summarized in Table 3.8. The final, updated data model is presented in Section 3.4.

3.3.1 Participant Roles

Several scenarios indicate that the current definition of *decision maker* is inadequate to capture the complexity of participant roles.

In the Faculty Hiring case, feedback from the department is considered by the hiring committee and voters at the department meeting. In both cases, the preferences of non-voting stakeholders are taken into account. The voters themselves may or may not be evaluators, depending on whether or not they completed the feedback form.

In the XpertsCatch case, the opinions and preferences of four engineers were considered, but ultimately, the CTO was responsible for the final decision.

Finally, in the MJS77 and Nuclear Crisis cases, groups of individuals of operated as single decision-making units.

To address these subtleties, we add the following definitions to our model:

- A **Stakeholder** is an individual or group that is invested in the outcome of the decision.
- A **Decision Maker** is an individual, or a group functioning as an individual, that is responsible for reviewing a collection of preferences and making a decision accordingly, either through voting or acting independently.
- An **Evaluator** is an individual, or a group functioning as an individual, whose preferences are modelled and taken into account by the decision makers.

In light of the definition changes above, all references to **Decision Maker** in the data model definition are replaced with **Evaluator** (Section 3.4).

In the seven scenarios that we analyzed, the decision makers and the evaluators were all stakeholders. However, this might not always be the case, since non-stakeholder preferences may be taken into consideration for additional information.

We limit the definition of Group Preferential Choice to scenarios where at least two of the evaluators are also stakeholders. But other types of problems may call for the synthesis of non-stakeholder preferences exclusively. For example, a shopper might want to inspect a summary of product reviews to inform her selection. This is not Group Preferential Choice, but the data and tasks may be similar, and so many of the same visual techniques may apply.

3.3.2 Criteria

There are situations where a common score function is defined for objective criteria (Campbell River, Nuclear Crisis). In these cases, all evaluators are assigned the same score function for that criterion. This may be appropriate when the relative values of different outcomes are generally agreed upon (e.g. cost) or require expert judgment (e.g. fish population). Our contact at Compass explained that this is normal, and that it is unusual for each evaluator to supply her own score function.

For this reason, we extend the definition of objective criteria to include an optional score function specification. Additionally, we add a new level **P2w** to the Preference Model Taxonomy that only includes weights. It sits below **P2b+w** because the score functions for the evaluators are implicitly encoded as the score functions for the criteria.

3.3.3 Evaluator Groups and Weights

The current data model does not provide a way to partition evaluators into groups. The Faculty Hiring, XpertsCatch, and Gift cases indicate that this would be useful for capturing different classes of evaluators. Furthermore, interviewees in the Faculty Hiring case expressed a desire to see a breakdown of the results by department role *or* confidence level. To support this, the data model would need to permit multiple ways of partitioning the evaluators into groups.

The current data model also does not provide a way to quantify evaluator importance, which may vary for a number of reasons including expertise, authority, or degree of investment in the outcome. For instance, in the Faculty Hiring case, the opinions of experts are valued more than those of non-experts. As such, the data model should support weights for individual evaluators or evaluator groups.

To address these limitations, the following elements have been added to the data model:

- A set of **Group Trees** $GT : gt_1 ... gt_u, |GT| \geq 1$, where:
  - A **Group Tree** gt is a tree where the internal nodes are **Groups** and the leaf nodes are Evaluators from E.
  - If preferences are collected at Level P0b are below:
    - Each Group Tree has a weights function $w(e) \in [0,1]$, where $e \in Evaluators$, and $\sum_{i=1}^{|Evaluators|} w(e_i) = 1$.

Each Group Tree represents a hierarchical partitioning of Evaluators into Groups, analogous to the hierarchical partitioning of Primitive Criteria into Abstract Criteria. We assume that GT always includes a default Group Tree that places every Evaluator under a single Group.

3.3.4 Preference Model Taxonomy

Table 3.7 shows which levels of the taxonomy are covered by which cases, indicated by Xs. In the XpertsCatch and Gift for Colleague columns, the Xs indicate the levels that would have been covered if preferences had been formally modelled (according to the interviewees).

There are no cases for levels P1a and P2a. In fact, a subsequent review of MCDM literature uncovered no recorded cases where this type of preference model was used, even when there is only one decision maker. Nevertheless, these levels will be retained for completeness.

Table 3.7: Coverage of Preference Model Taxonomy. The checkmarks indicate which levels are present in each scenario. Checkmarks with an asterisk are hypothetical.

		Faculty Hiring	Best Paper	Campbell River	Voyager	Nuclear Crisis	XpertsCatch	Gift for Colleague
P0	a		✓	✓	✓			
	b			✓	✓			
P1	a							
	b	✓						
	b+w					✓	✓*	✓*
P2	a							
	b							
	b+w						✓*	✓*
	w			✓		✓		

Finally, we return to the list of potential simplifying assumptions:

1. The preferences do not span multiple levels. That is, there in not a mix of objective and subjective criteria or ordinal and cardinal evaluations.
2. All decision makers express their preferences at the same levels(s) of the taxonomy.
3. The preferences are *complete*, that is:
   (a) At level $P0$, every decision maker ranks (or scores) every alternative.
   (b) At level $P1$, every decision maker ranks (or scores) every alternative with respect to every criterion.
   (c) At level $P2$, every decision maker ranks (or scores) every outcome for every criterion.

4. The preferences and weights are treated as *certain*. That is, there is no fuzziness.

Assumptions 2 and 4 hold in all scenarios.

Assumption 1 is violated by the Nuclear Crisis, XpertsCatch, and Gift scenarios, which each use a mix of objective and subjective criteria. This should not have major implications for visualization design - it simply means that designs may need to handle more heterogeneity.

Assumption 3 is violated by the Faculty Hiring scenario because evaluators can select 'NA' for any of the criteria. Furthermore, in cases where multiple candidates are being considered, evaluators are not required to evaluate all candidates. Missing values could pose a significant challenge for both the mathematical model and the visual design, but the problem does not appear to be ubiquitous in Group Preferential Choice Scenarios. For this reason, we will maintain this assumption going forward and leave the missing values problem to future work.

Table 3.8: Summary of Data Model Issues from Scenarios

Category	Issue	Scenarios	Solution
Participant Roles	Relationship between decision makers and evaluators may not be one-to-one	Faculty Hiring, XpertsCatch	Distinguish between decision maker, stakeholder, and evaluator
	Decision makers and evaluators may be groups functioning as individuals	MJS77, Nuclear Crisis	Revise role definitions accordingly
Criteria	A common score function may be defined for an objective criteria	Campbell River, Nuclear Crisis	Add an optional score function to the objective criterion definition; add level **P2w** to the Preference Model Taxonomy
Evaluator Groups and Weights	Decision makers may want to partition the evaluators into groups, and then assign different weights to different groups	Faculty Hiring, XpertsCatch, Gift	Introduce Group Trees, Groups, and Group Weights
	Decision makers may want to assign weights to individual evaluators	MJS77, Nuclear Crisis, XpertsCatch, Gift	Same as above (assign individual evaluators to their own group)

3.4 Revised Data Model for Group Preferential Choice

We define Group Preferential Choice as a situation where one or more decision makers must jointly choose from a set of alternatives based on two or more stakeholders' preferences over the alternatives.

- A set of **Alternatives** $A : \{a_1 ... a_m\}, m \geq 1$
- A set of **Evaluators** $E : e_1 ... e_n, |E| \geq 2$, each of whom has a **Preference Model** (described below)
- A set of **Criteria** $C : \{c_1 ... c_r\}, r \geq 0$
- A set of **Primitive Criteria** $PC \subset C : \{pc_1 ... pc_s\}, s \geq 0$
- A set of **Abstract Criteria** $AC = C \setminus PC$
- A **Criteria Tree** T where the set of nodes in T is equal to C, and the set of leaf nodes in T is equal to PC. This models the criteria taxonomy.
- A set of **Group Trees** $GT : gt_1 ... gt_u, |GT| \geq 1$, where:
  - A **Group Tree** gt is a tree where the internal nodes are **Groups** and the leaf nodes are Evaluators from E.
  - If preferences are collected at Level P0b are below:
    - Each Group Tree has a weights function $w(e) \in [0,1]$, where $e \in Evaluators$, and $\sum_{i=1}^{|Evaluators|} w(e_i) = 1$.

Criteria may be *objective* or *subjective* depending on whether their outcomes are measurable facts or personal judgments. For any objective criterion, there are the following additional elements:

- A **Domain** function $dom(pc)$ where $pc \in PC$, which defines the possible outcomes for criterion pc. The domain may be a discrete set (ordered or unordered) or a continuous range.
- An **Outcome** function $out(a, pc) \in dom(pc)$ where $a \in A$ and $pc \in PC$, which defines the outcome of alternative a on criterion pc.
- An optional **Score** function $score(out, pc) \in [0,1]$ where $pc \in PC$ and $out \in Domain(pc)$ and $score(out, pc_{worst}) = 0$ and $score(out, pc_{best}) = 1$

3.4.1 Preference Model Taxonomy

Level P0: The evaluators evaluate the alternatives holistically.

a. Ordinal evaluation. Each evaluator *ranks* the alternatives. Preferences can be modeled as a function $r_e(a) \in [1, |A|]$ where:

1. $a \in A$ and $e \in E$
2. If a_{best} is the most preferred alternative for evaluator e, then $r_e(a_{best}) = 1$
3. $r_e(a_1) < r_e(a_2)$ if and only if e prefers a_1 to a_2

b. Cardinal evaluation. Each evaluator *scores* each alternative along a common linear scale. Preferences can be modeled as a function $s_e(a) \in [min, max]$ where:

1. $a \in A$ and $e \in E$
2. *min* and *max* are the minimum and maximum points on a linear scale common to all evaluators

Level P1: The evaluators evaluate each alternative with respect to each criterion.

a. Ordinal evaluation. Each evaluator *ranks* the alternatives with respect to each criterion. Preferences can be modeled as a function $r_e(a, pc) \in [1, |A|]$ where:

1. $a \in A$, $e \in E$, and $pc \in PC$
2. If a_{best} is the most preferred alternative for evaluator e on criterion pc, then $r_e(a_{best}, pc) = 1$
3. $r_e(a_1, pc) < r_e(a_2, pc)$ if and only if e prefers a_1 to a_2 on criterion pc

b. Cardinal evaluation. Each evaluator *scores* each alternative with respect to each criterion along a common linear scale. Preferences can be modeled as a function $s_e(a, pc) \in [min_{pc}, max_{pc}]$ where:

1. $a \in A$, $e \in E$, and $pc \in PC$
2. min_{pc} and max_{pc} are the minimum and maximum points on a linear scale for pc common to all evaluators

b+w. Same as above, with the addition of *weights* specifying the relative value of switching from the worst to the best outcome on each criterion. This can be modeled as a function $w_e(pc) \in [0,1]$, where $e \in E$, $pc \in PC$, and

$$\sum_{i=1}^{|PC|} w_e(pc_i) = 1 \tag{3.2}$$

At this level, the raw (unweighted) preferences are specified by the function $uws_e(a, pc)$, while the weighted preferences are specified by the function $s_e(a, pc) = uws_e(a, pc) * w_e(pc)$.

Level P2: The evaluators evaluate each possible outcome of each criterion.

a. Ordinal evaluation. Each evaluator *ranks* the possible outcomes of each criterion. (This is only applicable for criteria with discrete domains.) Preferences can be modeled as a function $r_e(out, pc) \in [1, |dom(pc)|]$ where:

1. $e \in E$, $pc \in PC$, and $out \in dom(pc)$
2. If out_{best} is the most preferred outcome for evaluator e on criterion pc, then $r_e(out_{best}, pc) = 1$
3. $r_e(out_1, pc) < r_e(out_2, pc)$ if and only if e prefers out_1 to out_2 on criterion pc

b. Cardinal evaluation. Each evaluator *scores* each possible outcome of each criterion along a common linear scale. Preferences can be modeled as a function $s_e(out, pc) \in [min_{pc}, max_{pc}]$ where:

1. $e \in E$, $pc \in PC$, and $out \in dom(pc)$
2. min_{pc} and max_{pc} are the minimum and maximum points on a linear scale for pc common to all evaluators

b+w. Same as above, with the addition of *weights* specifying the relative value of switching from the worst to the best outcome on each criterion. This can be modeled in the same manner as Level P1b+w.

At this level, the raw (unweighted) preferences are specified by the function $uws_e(out, pc)$, while the weighted preferences are specified by the function $s_e(out, pc) = uws_e(out, pc) * w_e(pc)$.

w. Same as above, except with weights *only*. (Assumes that a common score function is defined for each primitive criterion.)

3.5 Summary of Preference Synthesis Goals

This section collates the scenario-specific goals into scenario-independent goals for preference synthesis in the context of Group Preferential Choice.

In all scenarios, the overarching goal is to arrive at consensus or make a well-informed decision that most stakeholders can accept. This is primarily achieved through discussion, with the quantitative summaries serving as a guide. This is a key point - in none of the scenarios did the quantitative summaries completely supplant verbal exchange. Rather, the role of quantitative summaries was to *focus* analysis on points of interest, which can greatly enhance the efficiency of the process. In particular, decision makers used the quantitative summaries to:

1. Discover viable alternatives
2. Discover sources of disagreement
3. Explain individual scores

The first item narrows the scope of analysis to alternatives that show promise. This task is often paired with identifying evaluators that gave these alternatives low scores or ranks (Faculty Hiring, XpertsCatch). Then, these evaluators can explain why they felt this way. If the decision makers have the option of selecting *no* alternatives, this also involves weighing alternatives against the status quo.

The second is concerned with identifying sources of disagreement among evaluators. In order to reach consensus, the decision makers need to understand how their preferences differ so they can negotiate and make compromises. Variations on this goal occur in all seven scenarios.

The third refers to the process of decomposing an individual evaluator's score into its constituents. This is necessary to support the second goal of understanding points of contention, and it also allows evaluators to understand how different aspects of their preferences (e.g. weights, score functions) contribute to their total scores.

These three goals pertain to *understanding* the model - an additional goal is to *validate* the model. In practice, it is not uncommon for evaluators to adjust their preferences after the first round of preference synthesis [51]. The accuracy and robustness of the model can be tested by encouraging reflection (Best Paper, Nuclear Crisis), collecting preferences at multiple levels of the taxonomy (Campbell River), or testing different aggregation techniques (MJS77). The process of observing how changes to inputs influence outputs is called *sensitivity analysis*. If inconsistencies or inadequacies are discovered, evaluators should be given an opportunity to adjust their preferences. In some cases, it may also be necessary for the decision makers to revise the criteria or alternatives.

Finally, quantitative models are seldom sufficient to fully capture individual preferences. So, a final goal is to discover nuances that the explicit preference models do not capture. This is achieved by engaging in discussion (all scenarios) or consulting textual feedback if not all evaluators are present (Faculty Hiring).

Table 3.9 presents these goals in list form and relates them to scenario-specific goals. The scenario goals are indexed by XX.YY, where XX is the scenario ID and YY is the goal ID in that scenario's Goals table. The scenario IDs are:

- BP = Best Paper (Table 3.1)
- FH = Faculty Hiring (Table 3.2)
- CR = Campbell River (Table 3.3)
- MS = MJS77 (Table 3.4)
- NC = Nuclear Crisis (Table 3.5)
- XC = XpertsCatch (Table 3.6)

Table 3.9: Goals for Preference Synthesis in Group Preferential Choice

GENERIC GOAL			SCENARIO GOALS
G1	**Discover Viable Alternatives**		
	a	Discover high-performing alternatives across evaluators/evaluator groups	BP.G2, CR.G4, MS.G2, NC.G4, XC.G3
	b	Discover high-performing alternatives across criteria (aggregated over evaluators)	FH.G2, FH.G3
	c	Discover high-performing alternatives for a single evaluator/evaluator group	XC.G2
G2	**Discover Sources of Disagreement (i.e. find discrepancies across evaluators)**		
	a	Discover and explain disagreement about an alternative (across evaluators/evaluator groups)	BP.G3, FH.G4, FH.G5, FH.G6, CR.G5, MS.G3, NC.G5, XC.G6, XC.G7
	b	Discover differences in preference models (across evaluators/evaluator groups)	CR.G2, CR.G3, XC.G5
G3	**Explain Individual Scores**		
	a	Analyze contribution of different criteria to an alternative's score (for a single evaluator/evaluator group)	NC.G2
	b	Analyze contribution of different parts of the preference model (e.g. weights) to an alternative's score (for a single evaluator/evaluator group)	
	c	Analyze contribution of different evaluators and evaluator weights to an alternative's total score	NC.G7
G4	**Validate Model**		
	a	Understand sensitivity of evaluator scores to evaluator preference models	NC.G3
	b	Understand sensitivity of total scores to group weights	MS.G4, NC.G6
	c	Understand sensitivity of total scores to aggregation method	MS.G4
	d	Discover discrepancies between one's own preferences and others' preferences	BP.G4
G5	**Discover nuances in evaluators' preferences that are not captured by the preference models**		BP.G5, FH.G8, CR.G6, MS.G5, NC.G8, XC.G4

3.6 Summary of Contextual Features and Scale

The scenarios divide roughly into three clusters based on contextual features (Table 3.10). The first cluster consists of very high-stakes, one-time decision problems that the decision makers devote one or more days to analyzing with the help of MCDA experts (MJS77, Nuclear Crisis, and Campbell River). The second consists of high and medium-stakes decision problems that recur annually or monthly that the decision makers devote only a few hours to analyzing (Faculty Hiring, Best Paper, and XpertsCatch). The final cluster is a low-stakes decision made in a more casual setting (Gift).

Table 3.10: Contextual Features of Seven Scenarios

	MJS77 Project	**Nuclear Crisis**	**Campbell River**	**Faculty Hiring**	**Best Paper**	**XpertsCatch**	**Gift for Colleague**
Assessment Method	Journal Article	Journal Article	Webinar	Interview	Interview	Interview + observation	Interview
Work Context	Professional	Professional	Professional	Professional	Professional	Professional	Semi-professional
Frequency	Once	Once	Once	Monthly	Annually	Annually	Once
Stakes	Very High	Very High	Very High	High	Medium	Medium	Low
Preference Model(s)	P0a, P0b	P2w, P1b	P0a, P0b, P2w	P1b	P0a	P2b, P1b	P2b, P1b
Time Allowance	Several days	1 day	1 day	1 - 2 hours	1 hour	1 hour	1 hour
# Evaluators	11	6	15	50 - 100	5 - 10	5	10
# Alternatives	32	4	6	1 - 4	4 - 15	2	3
# Criteria	NA	7	12	6	NA	8	5

Each of these clusters is likely to have somewhat different requirements for its support system. Decision makers in the first cluster may benefit the most from advanced analytic features, since they have the time, incentive, and expertise to take advantage of them. Decision makers in the second cluster are more likely to benefit from systems that are easy to learn and deliver insights quickly. If the system is too complex or cumbersome, decision makers in this cluster may not be willing to put in the effort to learn and use them. The third cluster may have an even greater preference for usability over sophistication.

In most of these scenarios, the decision problem dimensions (number of evaluators, alternatives, and criteria) do not exceed twenty. This is reassuring from a design standpoint, as it suggests that a variety of problems can be addressed without encountering major scalability issues. The two exceptions are the number of alternatives in the MJS77 scenario and the number of evaluators in the Faculty Hiring scenario. In the former case, it is conceivable that the initial list could have been winnowed further prior to complex preference modelling.

The Faculty Hiring scenario, on the other hand, is an outlier in more ways than one. First, the number of evaluators is much higher than in any other scenario. Second, it is the only scenario in which not all evaluators are present during the preference synthesis. This may be why textual data is highly valued in this scenario

- the evaluators are not always around to clarify their preferences in person. It is unclear at this point if the Faculty Hiring scenario is sufficiently different from the others to warrant its own design space.

3.7 Conclusion

The primary goal of this chapter was to characterize sources of variation among Group Preferential Choice scenarios. We conclude with a summary of the similarities and differences that were discovered.

Data

Similarities. By definition, all Group Preferential Choice scenarios have alternatives, evaluators, and a rank or score for each alternative-evaluator combination. In all but two scenarios, the number of evaluators, alternatives, and criteria did not exceed twenty.

Differences. Between the seven scenarios, six different levels of the Preference Model Taxonomy were represented (Table 3.7). Two of the scenarios had non-flat criteria hierarchies (that is, they had abstract criteria other than the implicit root).

Three scenarios had or would have benefited from a non-flat evaluator hierarchy, and four had or would have benefitted from evaluator weights (Table 3.8). In two scenarios, the relationship between decision makers and evaluators was not one-to-one (Table 3.8).

Goals

In Section 3.5, the following goals were identified in three or more scenarios:

- G1a. Discover high-performing alternatives across evaluators/evaluator groups
- G2a. Discover and explain disagreement about an alternative
- G5. Discover nuances in evaluators' preferences that are not capture by the preference models.

The remaining goals were associated with at most two scenarios each.

Context

As described in Section 3.6, the scenarios divide roughly into three clusters with similar features. There is considerable variation between clusters and some variation within.

Chapter 4

Data and Task Abstraction for Preference Synthesis in Group Preferential Choice

The goal of this chapter is to produce an abstract data and task model for preference synthesis in the context of Group Preferential Choice. The resulting model is intended to be broad enough to cover a variety of real-world scenarios but detailed enough to guide requirements analysis for support tools. This model will inform the analysis of potential visual encodings and interactions in Chapter 5.

Section 4.1 describes our existing data model (Section 3.4) in terms of a new abstraction based on tables, which is more suitable for visualization design and analysis.

Section 4.2 develops a task model by relating the goals identified in Section 3.5 to tasks described in terms of a taxonomy by Brehmer and Munzner [7]. Section 4.2.1 describes high-level tasks on Group Preferential Choice data, and Section 4.2.2 decomposes each of these tasks into lower-level tasks on generic data types.

4.1 Data Abstraction

In order to assess the suitability of different visual encodings, it is helpful to describe the data in terms of *multidimensional tables*, which are datasets consisting

of *attributes* and *dimensions*, where an attribute is something that can be measured and a dimension is a set of entities for which an attribute can be defined [38]. The entities of a dimension are called *keys*, and the specific instances of an attribute are called *values*. Types of attributes include categorical, ordinal, and quantitative [38].

Multidimensional tables form the basis of OLAP (Online Analytical Processing), a popular Business Intelligence paradigm that is integral to Analytics tools such as Microsoft Excel and Tableau [16] [53]. In this context, multidimensional tables are called *data cubes* and the term *measure* is used in lieu of attribute. Because attribute is also a synonym for criterion in MCDA, we will also use the term measure instead.

As an example of these concepts, say that the cosmetics department of Macy's Gotham made a $2000 profit on 11-11-2009. In this case, Department, Store Location, and Date are dimensions and Profit is a measure. Cosmetics, Gotham, and 11-11-2009 are keys, and $2000 is the a value for Profit defined by these keys.

Measures can be further divided into *basic measures*, which the user supplies, and *derived measures*, which can be computed from the basic measures. The *dimensionality* of a measure is the set of dimensions whose keys map to a single value for that measure. In the example above, the dimensionality of Profit is {Department, Store Location, Date}.

In Group Preferential Choice, each level of the Preference Model Taxonomy is defined by one or two basic measures, as summarized in Table 4.1. All measures are quantitative except for Outcome, whose type depends on the domain of the criterion. Referring back to the terms and notation introduced in Section 3.8, the dimensions are Evaluators, Criteria, Alternatives, and Outcomes, and their keys are E, PC, A, and $(\bigcup_{i=1}^{|PC|} dom(pc_i))$, respectively.[1]

OLAP also supports the specification of *hierarchies*, which impose hierarchical arrangements on the entities of a dimension [54]. In Group Preferential Choice, the Criteria dimension has a hierarchy that is specified by the Criteria Tree. The Evaluators dimension has one hierarchy for each Group Tree that is defined.

[1] This abstraction makes the simplifying assumption that the *Outcomes* dimensions is independent of the *Criteria* dimension. This is not especially problematic - we can simply treat the nonsensical intersections as undefined.

The derived measures at each level of the taxonomy include all basic measures that are defined by any of its descendants (see Figure 3.1). Additional derived measures can be obtained via *roll-up*, which is the process of aggregating over (that is, factoring out) a dimension or aggregating within a dimension to a higher level of some hierarchy. Applicable to this analysis are the following derived measures:

1. The aggregate of any basic measure except Outcome for an evaluator group (aggregating up the Evaluators hierarchy)

2. The aggregate of AltCritRank, AltCritScore, or CritWeight for an abstract criterion (aggregating up the Criteria hierarchy)

3. The **TotalRank/TotalScore** for an alternative, which is the aggregate of AltRank/AltScore over evaluators (factoring out the Evaluators dimension)

There are numerous ways that values can be aggregated, but we assume that aggregate totals are obtained via summation.

Table 4.1: Basic Measures. The formulae refer to those defined in Section 3.4. The *dimensionality* of a measure is the set of dimensions whose keys map to a single value for that measure. In other words, they are the inputs to the formula for that measure.

Taxonomy Level	**Basic Measure**	**Formula**	**Dimensionality**
P0b and descendants	EvaluatorWeight	$w(g)$	{Evaluators}
P0a	AltRank	$r_e(a)$	{Evaluators, Alternatives}
P0b	AltScore	$s_e(a)$	{Evaluators, Alternatives}
P0b	UnweightedAltScore	$uws_e(a)$	{Evaluators, Alternatives}
P1a	AltCritRank	$r_e(a, pc)$	{Evaluators, Alternatives, Criteria}
P1b	AltCritScore	$s_e(a, pc)$	{Evaluators, Alternatives, Criteria}
P1b+w	UnweightedAltCritScore	$uws_e(a, pc)$	{Evaluators, Alternatives, Criteria}
P2a	OutRank	$r_e(out, pc)$	{Evaluators, Criteria, Outcomes}
P2b	OutScore	$s_e(out, pc)$	{Evaluators, Criteria, Outcomes}
P2b+w	UnweightedOutScore	$uws_e(out, pc)$	{Evaluators, Criteria, Outcomes}
P1b+w, P2b+w	CritWeight	$w_e(pc)$	{Evaluators, Criteria}
P2a/b/b+w	Outcome	$out_a(pc)$	{Alternatives, Criteria}

4.2 Task Abstraction

In the next two sections, we relate each of the goals in Table 3.9 to abstract tasks based on Brehmer and Munzner's typology of visualization tasks [7] (Figure 4.1). This typology is rooted in Munzner's nested model of visualization design, which separates data/task abstraction from consideration of visual encodings/interaction idioms [37]. The idea is that designers should be able to describe *why* a task is performed and *what* data it is performed on independently of *how* it is achieved. At this stage, we are only concerned with the *what* and *why* levels of description, as our aim is to develop a task model that is independent of any particular system.

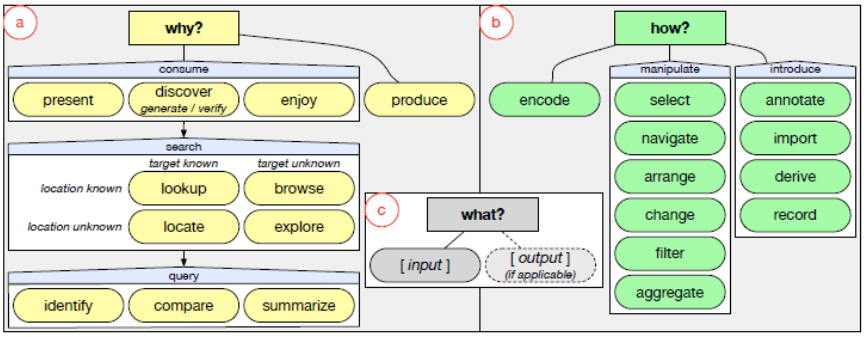


Figure 4.1: Brehmer and Munzner's typology of abstract visualization tasks [7]. The *why* group consists of actions arranged hierarchically from high to low level. The *what* group encapsulates the targets, which are separated into *inputs* and *outputs*.

.

4.2.1 High-Level Task Abstraction

All of the goals in Table 3.9 are instances of the high-level task *Consume: Discover*, which covers many facets of inquiry [7]. The terms *Explain* (G3), *Analyze* (G3a, G3b), *Verify* (G4), and *Understand* (G4a - G4d) are all included in the list of vocabulary related to the Discover task [7].

Tables 4.2 - 4.5 relate each of the sub-goals in Table 3.9 to high-level tasks that support that goal. Although these tasks are lower-level than the goals, they are still high-level from the perspective of the task typology since they, too, fall under

the umbrella of Discover. In these descriptions, *criterion* may refer to a primitive criterion or an abstract criterion, and *evaluator* may refer to a single evaluator or an evaluator group, unless otherwise noted. For abstract criteria and evaluator groups, the respective value will be an aggregate as described in the previous section.

These tasks were identified in two ways: (a) revisiting the scenario-specific goals and coding them as tasks (if they were more specific than the generic goal they were grouped with) and (b) brainstorming tasks that were missing from the scenarios but could clearly support the goal in question.

Table 4.2 shows tasks to support *G1: Discover Viable Alternatives*. Supporting tasks for G1a include finding alternatives with high overall scores (T1) or low variation in scores across evaluators (T2), as these may constitute viable 'compromise' alternatives. Another way to focus the analysis is to identify non-dominated alternatives (T3), which can minimize distraction and interference from others. To narrow the list further, it is essential to be able to consider trade-offs between competitive alternatives (T4). Finally, it may be necessary to looks at absolute pros and cons of one alternative (where the 'pros' are evaluators with high scores and the 'cons' are evaluators with low scores), especially if selecting no alternatives is an option (T6).

G1b is about identifying high-performing alternatives across criteria after evaluators have been factored out. In addition to a high overall scores (T1), consistent performance across criteria may be desirable (T6). As with T3, it may be useful to focus analysis on non-dominated alternatives in criteria-space (T7). Finally, one might want to look at the relative strengths and weaknesses (that is, the trade-offs) between a pair of alternatives (T8), or the absolute strengths and weaknesses of one alternative (T9).

G1c is about identifying high-performing alternatives for a particular evaluator of interest, such as oneself. Tasks T11 - T14 are analogous to Tasks T6 - T9 except that they target a particular evaluator instead of the aggregate over all evaluators.

Table 4.2: Tasks to Support G1: Discover Viable Alternatives. Viable alternatives may include those with high overall scores (T1) or low variation in scores (T2), as these may constitute viable 'compromise' alternatives. Discovering non-dominated alternatives (T3) can focus the analysis on competitive alternatives and minimize distraction and interference from the others.

TASK		**Applicable Levels**
G1a. Discover high-performing alternatives across evaluators		
T1	Discover alternative(s) with best TotalRank/TotalScore	Any
T2	Discover alternatives(s) with low variance in AltRanks/AltScores across evaluators	Any
T3	Discover non-dominated alternatives across evaluators	Any
T4	Discover trade-offs in AltRanks/AltScores between alternatives *a* and *b*	Any
T5	Discover pros and cons in AltRanks/AltScores for alternative *a*	Any
G1b. Discover high-performing alternatives across criteria		
T6	Discover alternatives(s) with low variance in AltCritRanks/AltCritScores across criteria (aggregated over evaluators)	P1a and descendants
T7	Discover non-dominated alternatives across criteria (aggregated over evaluators)	P1a and descendants
T8	Discover trade-offs in AltCritRanks/AltCritScores between alternatives *a* and *b* (aggregated over evaluators)	P1a and descendants
T9	Discover strengths and weaknesses of alternative *a* (aggregated over evaluators)	P1a and descendants
G1c. Discover high-performing alternatives for a single evaluator		
T10	Discover alternative(s) with best AltRank/AltScore for evaluator *e*	Any
T11	Discover alternatives(s) with low variance in AltCritRank/AltCritScore across criteria for evaluator *e*	P1a and descendants
T12	Discover non-dominated alternatives across criteria for evaluator *e*	P1a and descendants
T13	Discover trade-offs in AltCritRanks/AltCritScores between alternatives *a* and *b* for evaluator *e*	P1a and descendants
T14	Discover strengths and weaknesses of alternative *a* for evaluator *e*	P1a and descendants

Table 4.3 shows tasks to support *G2: Discover Sources of Disagreement*. At the highest level, disagreement can be discovered by finding alternatives with high variance in scores, as these are more likely to be controversial (T15). Once an interesting alternative has been identified (either through G1 or G2a), one can zero in on evaluators with dissenting opinions (T16) or criteria that are responsible for the controversy (T17).

Another approach to discovering sources of disagreement is to look directly at the preferences. If weights are included in the model, one can look for criteria with high variance in weights (T18) and then identify dissenters (T19). This can also be done for score functions if they are included (T20 and T21).

Table 4.3: Tasks to Support G2: Discover Sources of Disagreement. These tasks hone in on *where* the disagreement is (T15) and *who* is disagreeing (T16), bringing dissenting viewpoints to light.

TASK		**Applicable Levels**
G2a. Discover and explain disagreement about an alternative		
T15	Discover alternatives(s) with high variance in AltRank/AltScore across evaluators	Any
T16	Discover evaluators that are outliers with respect to AltRank/AltScore for alternative *a*	Any
T17	Discover criteria with high variance in AltCritRank/AltCritScore across evaluators for alternative *a*	P1a and descendants
G2b. Discover differences in preference models		
T18	Discover criteria with high variance in CritWeights across evaluators	P1b+w, P2b+w
T19	Discover evaluators that are outliers with respect to CritWeights for criterion *c*	P1b+w, P2b+w
T20	Discover criteria outcomes with high variance in OutRanks/OutScores across evaluators	P2b and descendants
T21	Discover evaluators that are outliers with respect to OutRanks/OutScores for outcome *o* of primitive criterion *pc*	P2b and descendants

Table 4.4 shows tasks to support *G3: Explain Individual Scores*. Each of these tasks involves breaking down a derived measure into its constituents.

Table 4.4: Tasks to Support G3: Explain Individual Scores. These tasks allow decision makers to analyze constituents of global scores and individual evaluator's scores.

TASK		Applicable Levels
G3a. Analyze contribution of different criteria to an alternative's score		
T22	Analyze breakdown of AltRanks/AltScores into AltCritRanks/AltCritScores for alternative *a* and evaluator *e*	P1a and descendants
G3b. Analyze contribution of different parts of the preference model to an alternative's score		
T23	Analyze breakdown of AltCritScore into UnweightedAltCritScore and CritWeight for alternative *a*, evaluator *e*, and criterion *c*	P1b+w
T24	Analyze breakdown of OutScore into UnweightedOutScore and CritWeight for evaluator *e*, primitive criterion *pc*, and outcome *out*	P2b+w
T25	Understand mapping between AltCritRank/AltCritScore and OutRank/OutScore for a particular evaluator, alternative, and primitive criterion	P2a/b and descendants
T26	Analyze breakdown of AltCritRank/AltCritScore for alternative *a*, evaluator *e*, and abstract criterion *ac*	P1a and descendants
G3c. Analyze contribution of different evaluators and evaluator weights to an alternative's total score		
T27	Analyze breakdown of AltScores into UnweightedAltScore and EvaluatorWeight for alternative *a* and evaluator *e*	P0b and descendants
T28	Analyze breakdown of TotalRanks/TotalScores into AltRanks/AltScores for alternative *a*	Any

Finally, Table 4.5 shows tasks to support *G4: Validate Model*. Tasks T29 - T32 support sensitivity analysis on various aspects of the model. The remaining tasks allow individuals to compare their preference models to those of others, which could inspire them to reevaluate their own preferences.

Table 4.5: Tasks to Support G4: Validate Model. These tasks support sensitivity analysis and encourage comparison of individual preferences with those of others.

TASK		Applicable Levels
G4a. Understand sensitivity of evaluator scores to evaluator preference models		
T29	Discover differences in AltRanks/AltScores for evaluator *e* before and after changing CritWeights	P1b+w, P2b+w
T30	Discover differences in AltRanks/AltScores for evaluator *e* before and after changing non-weight component of preference model	Any
G4b. Understand sensitivity of total scores to evaluator weights		
T31	Discover differences in TotalScores before and after changing Evaluator-Weights	P0b and descendants
G4c. Understand sensitivity of total scores to aggregation method		
T32	Discover differences in TotalRanks/TotalScores from two different aggregation methods	Any
G4d. Discover discrepancies between one's own preferences and others' preferences		
T33	Discover differences in CritWeights for evaluator *e* to CritWeights for other evaluators	Any
T34	Discover differences in non-weight component of preference model (e.g. AltScores at P0b, OutScores at P2b) for evaluator *e* to that of other evaluators	Any

4.2.2 Low-Level Task Abstraction

In this final stage of analysis, we decompose the high-level Discover tasks into low-level Search and Query tasks.

Task Targets

The *what* node in the typology presented in Figure 4.1 represents the targets of the tasks, which include *inputs* and *outputs*.

In Group Preferential Choice, there are two types of targets: values and distributions (which are simply sets of values). Referring back to Table 4.1, the value of a measure is defined by its complete key-set. For instance, an AltScore value is uniquely defined by an alternative and an evaluator. If any of the keys are missing, the result is a distribution.

The codes for various targets are provided in Tables 4.6 and 4.7. The distributions in Table 4.7 are the result of allowing one dimension for the measure in

question to vary and fixing the others. For instance, D2(*john*) is the distribution of AltRanks or AltScores for evaluator *john* over all alternatives.

In each of these tables, a *criterion* may refer to an abstract or primitive criterion. Similarly, an *evaluator* may consist of a single evaluator or multiple evaluators in a group. If it is an abstract criterion or multi-evaluator group, the values in question will be aggregates.

Table 4.6: Target Values for Task Analysis. These include all measures applicable to the given level.

Task Targets - Values		**For**	**Applicable Levels**
V1(a)	TotalRank/TotalScore	$a \in A$	Any
V2(a,e)	AltRank/AltScore	$a \in A, e \in E$	Any
V3(a,e)	UnweightedAltScore	$a \in A, e \in E$	Any
V4(a,e,c)	AltCritRank/AltCritScore	$a \in A, e \in E, c \in C$	Any
V5(a,e,c)	UnweightedAltCritScore	$a \in A, e \in E, c \in C$	P1b+w and descendants
V6(e,pc,o)	OutRank/OutScore	$e \in E, pc \in PC, o \in dom(PC)$	P2a and descendants
V7(e,pc,o)	UnweightedOutScore	$e \in E, pc \in PC, o \in dom(PC)$	P2b+w
V8(c)	CritWeight	$c \in C$	P1b+w, P2b+w
V9(e)	EvaluatorWeight	$e \in E$	P0b and descendants
V10(a,pc)	Outcome	$a \in A$, $pc \in PC$	P2a and descendants

Table 4.7: Target Distributions for Task Analysis. The **Across** column specifies the dimension that varies and the **For** column specifies the dimensions that are fixed.

Task Targets - Distributions		**Across**	**For**	**Applicable Levels**
D1	TotalRanks/TotalScores	Alternatives	All data	Any
D2(e)	AltRanks/AltScores	Alternatives	$e \in E$	P0a/b and descendants
D3(a)	AltRanks/AltScores	Evaluators	$a \in A$	P0a/b and descendants
D4(a,c)	AltCritRanks/AltCritScores	Evaluators	$a \in A, c \in C$	P1a/b and descendants
D5(a,e)	AltCritRanks/AltCritScores	Criteria	$a \in A, e \in E$	P1a/b and descendants
D6(pc,o)	OutRanks/OutScores	Evaluators	$pc \in PC, o \in dom(PC)$	P2a/b and descendants
D7(c)	CritWeights	Evaluators	$c \in C$	P1b+w, P2b+w
D8(e)	CritWeights	Criteria	$e \in E$	P1b+w, P2b+w

Auxiliary Tasks

All of the high-level tasks from the previous section can be accomplished using a combination of just ten auxiliary tasks on these targets. These are defined in terms of an action, an input type, and an output type (Table 4.8).

Brehmer and Munzner define four types of Search tasks based on whether the identify and location of the search target are known [7]. The target of a Locate or Lookup task is an element with a particular *identity*, whereas the target of a Browse or Explore task is an element with particular *features*. The search space for Locate and Explore tasks is the whole data-set, while the search space for Lookup and Browse is restricted.

We use three of these tasks - Locate, Lookup, and Browse. In this context, Locate and Lookup involves finding the value or distribution for a measure given a key-set (e.g. the AltScore for a particular alternative and evaluator), whereas Browse involves looking through distributions or sets of distributions for interesting subsets (e.g. find outliers in a set of AltScores).

Brehmer and Munzner define three types of Query tasks based on the number of items involved: Identify (single item), Compare (two items), and Summarize (3+ items) [7]. Query tasks are often performed on the outputs of a Search tasks. When paired with Locate or Lookup, Query returns *features*; when paired with Browse or Explore, it returns *identities* [7].

Table 4.8: Auxiliary Tasks. In AT3 and AT4, 'matched distributions' means distributions of the same type (i.e. same row in Table 4.7).

AUXILIARY TASKS				
	Action	**Input**	**Output**	**Supported by**
AT1	Query: Identify	A single value or distribution	Its key-set	
AT2	Query: Compare	A pair of values	Difference	
AT3	Query: Compare	A pair of matched distributions	A tuple of differences	AT2
AT4	Query: Compare	A pair of matched distributions	A dominance relation	AT3
AT5	Query: Summarize	A single distribution	A summary of variance	
AT6	Search: Locate	A key-set	A single value or distribution	
AT7	Search: Lookup (in context)	A key-set + a single value or distribution	A single value or distribution	AT6
AT8	Search: Browse	A single distribution	Outliers	AT2
AT9	Search: Browse	A single distribution	Top/bottom values	AT2
AT10	Search: Browse	A set of distributions	Non-dominated distributions	AT4

Equipped with these and the targets from 4.6 and 4.7, we can now describe

how to accomplish each high-level task. Auxiliary tasks and targets are referenced by code and accompanied by a short English description.

- **T1:** Discover alternative(s) with best TotalRank/TotalScore
  1. AT9(D1) $\rightarrow X$ (Get top values in TotalScores distribution)
  2. AT1(x) for $x \in X$ (Identify top values)
- **T2:** Discover alternatives(s) with low variance in AltRanks/AltScores across evaluators
  1. AT5(D3(a)) for $a \in A \rightarrow X$ (Get variance of each AltScores distribution)
  2. AT9(X) $\rightarrow Y$ (Get bottom values)
  3. AT1(y) for $y \in Y$ (Identify bottom values)
- **T3:** Discover non-dominated alternatives across evaluators
  1. AT10($\{$D3(a) for $a \in A\}$) $\rightarrow X$ (Get non-dominated AltScores distributions)
  2. AT1(x) for $x \in X$ (Identify non-dominated AltScores distributions)
- **T4:** Discover trade-offs in AltRanks/AltScores between alternatives a and b
  1. AT6(D3(a)) $\rightarrow X$ (Locate AltScores for a)
  2. AT6(D3(b)) $\rightarrow Y$ (Locate AltScores for b)
  3. AT3(X,Y)) $\rightarrow X$ (Get differences between AltScores distributions)
- **T5:** Discover pros and cons in AltRanks/AltScores for alternative a
  1. AT6(D3(a)) (Locate AltScores for a)
  2. AT6(V2(a,e)) for $e \in E$ (Locate every AltScore for a)
  3. AT2(V2(a,$e1$),V2(a,$e2$)) for $\{a1, a2\} \in E$ (Pairwise compare every AltScore for a)
- **T6:** Discover alternatives(s) with low variance in AltCritRanks/AltCritScores across criteria (aggregated over evaluators)

1. AT5(D5(a,$e = all$)) for $a \in A \rightarrow X$ (Get variance of each AltCritScores distribution)
2. AT9(X) $\rightarrow Y$ (Get bottom values)
3. AT1(y) for $y \in Y$ (Identify bottom values)

- **T7:** Discover non-dominated alternatives across criteria (aggregated over evaluators)
  1. AT10($\{$D5(a,$e = all$) for $a \in A\}$) $\rightarrow X$ (Get non-dominated AltCritScores distributions)
  2. AT1(x) for $x \in X$ (Identify non-dominated AltCritScores distributions)
- **T8:** Discover trade-offs in AltCritRanks/AltCritScores between alternatives a and b (aggregated over evaluators)
  1. AT6(D5(a,$e = all$)) $\rightarrow X$ (Locate AltCritScores for a)
  2. AT6(D5(b,$e = all$)) $\rightarrow Y$ (Locate AltCritScores for b)
  3. AT3(X,Y) (Get differences between AltCritScores distributions)
- **T9:** Discover strengths and weaknesses of alternative a (aggregated over evaluators)
  1. AT6(D5(a,$e = all$)) (Locate AltCritScores for a)
  2. AT6(V4(a,$e = all$,c)) for $c \in PC$ (Locate every AltCritScore for a)
  3. AT2(V4(a,$e = all$,$c1$),V4(a,$e = all$,$c2$)) for $\{c1, c2\} \in PC$ (Pairwise compare every AltCritScore for a)
- **T10:** Discover alternative(s) with best AltRank/AltScore for evaluator e
  1. AT6(D2(e)) $\rightarrow X$ (Locate AltScores for e)
  2. AT9(X) $\rightarrow Y$ (Get top values)
  3. AT1(y) for $y \in Y$ (Identify top values)
- **T11:** Discover alternatives(s) with low variance in AltCritRank/AltCritScore across criteria for evaluator e

1. AT6(D5(x,e)) $\rightarrow X$ (Locate AltCritScores for e)
2. AT5(X) for $a \in A \rightarrow Y$ (Get variance of each AltCritScores distribution)
3. AT9(Y) $\rightarrow Z$ (Get bottom values)
4. AT1(z) for $z \in Z$ (Identify bottom values)

- **T12:** Discover non-dominated alternatives across criteria for evaluator e

1. AT6(D5(x,e)) $\rightarrow X$ (Locate AltCritScores for e)
2. AT10(D5(X)) $\rightarrow Y$ (Get non-dominated AltCritScore distributions)
3. AT1(y) for $y \in Y$ (Identify non-dominated AltCritScore distributions)

- **T13:** Discover trade-offs in AltCritRanks/AltCritScores between alternatives a and b for evaluator e

1. AT6(D5(a,e)) $\rightarrow X$ (Locate AltCritScores for a and e)
2. AT6(D5(b,e)) $\rightarrow Y$ (Locate AltCritScores for b and e)
3. AT3(X,Y) (Get differences between AltCritScores in X and Y)

- **T14:** Discover strengths and weaknesses of alternative a for evaluator e

1. AT6(D5(a,e)) (Locate AltCritScores for a)
2. AT6(V4(a,e,c)) for $c \in PC$ (Locate every AltCritScore for a and e)
3. AT2(V4(a,e,$c1$),V4(a,e,$c2$)) for $\{c1, c2\} \in PC$ (Pairwise compare every AltCritScore for a and e)

- **T15:** Discover alternatives(s) with high variance in AltRanks/AltScores across evaluators

1. AT5(D3(a)) for $a \in A \rightarrow X$ (Get variance of each AltScores distribution)
2. AT9(X) $\rightarrow Y$ (Get top values)
3. AT1(y) for $y \in Y$ (Identify top values)

- **T16:** Discover evaluators that are outliers with respect to AltRank/AltScore for alternative a

1. AT6(D3(a)) $\rightarrow X$ (Locate AltScores for a))
2. AT8(X) $\rightarrow Y$ (Get outliers)
3. AT1(z) for $y \in Y$ (Identify outliers)

- **T17:** Discover criteria with high variance in AltCritRank/AltCritScore across evaluators for alternative a

1. AT6(D4(a,x)) $\rightarrow X$ (Locate AltCritScores for a)
2. AT5(X) for $a \in A \rightarrow Y$ (Get variance of each AltCritScores distribution)
3. AT9(Y) $\rightarrow Z$ (Get top values)
4. AT1(z) for $z \in Z$ (Identify top values)

- **T18:** Discover criteria with high variance in CritWeights across evaluators

1. AT5(D7(pc)) for $pc \in PC \rightarrow X$ (Get variance of CritWeights for each criterion)
2. AT9(X) $\rightarrow Y$ (Get top values)
3. AT1(y) for $y \in Y$ (Identify top values)

- **T19:** Discover evaluators that are outliers with respect to CritWeights for criterion c

1. AT6(D7(c)) $\rightarrow X$ (Locate CritWeights for c))
2. AT8(X) $\rightarrow Y$ (Get outliers)
3. AT1(y) for $y \in Y$ (Identify outliers)

- **T20:** Discover primitive criteria outcomes with high variance in OutRanks/OutScores across evaluators

1. AT5(D6(pc,o)) for $pc \in PC, o \in dom(pc) \rightarrow X$ (Get variance of each OutScores distribution)
2. AT9(X) $\rightarrow Y$ (Get top values)
3. AT1(y) for $y \in Y$ (Identify top values)

- **T21:** Discover evaluators that are outliers with respect to OutRanks/OutScores for outcome o on primitive criterion pc

  1. AT6(D6(pc,o)) $\rightarrow X$ (Locate D6 for pc, o)
  2. AT8(X) $\rightarrow Y$ (Get outliers)
  3. AT1(y) for $y \in Y$ (Identify outliers)

- **T22:** Analyze breakdown of AltRanks/AltScores into AltCritRanks/AltCritScores for alternative a and evaluator e

  1. AT7(D5(a,e),V2(a,e)) (Locate AltCritScores for a, e in context of AltScore for a, e)

- **T23:** Analyze breakdown of AltCritScore into UnweightedAltCritScore and CritWeight for alternative a, evaluator e, and criterion c

  1. AT7(V5(a,e,c),V4(a,e,c)) (Locate UnweightedAltCritScore for a, e, c in context of AltCritScore for a,e,c)
  2. AT7(V8(e,c),V4(a,e,c)) (Locate UnweightedOutScore for e, c in context of AltCritScore for a,e,c)

- **T24:** Analyze breakdown of OutScore into UnweightedOutScore and CritWeight for evaluator e, primitive criterion pc, and outcome o

  1. AT7(V7(e,pc,o),V6(e,pc,o)) (Locate UnweightedOutScore for e, pc, o in context of OutScore for e,pc,o)
  2. AT7(V8(e,pc),V6(e,pc,o)) (Locate UnweightedOutScore for e, pc in context of OutScore for e,pc,o)

- **T25:** Understand mapping between AltCritRank/AltCritScore and OutRank/OutScore for a alternative a, evaluator e, and primitive criterion pc

  1. AT6(V10(a,pc)) $\rightarrow X$ (Locate Outcome for a, pc)
  2. AT6(V7(e,pc,X)) (Locate UnweightedOutScore for e, c, X)

- **T26:** Analyze breakdown of AltCritRank/AltCritScore for alternative a, evaluator e, and abstract criterion ac

1. AT7(V4(a,e,c),V4(a,e,ac)) for $c \in children(ac)$ (Locate AltCritScore for each child of ac in context of AltCritScore for a,e,c)

- **T27:** Analyze breakdown of AltScores into UnweightedAltScore and EvaluatorWeight for alternative a and evaluator e

  1. AT7(V3(a,e),V9(e)) (Lookup UnweightedAltScore for a, e in context of EvaluatorWeight for e)

- **T28:** Analyze breakdown of TotalRanks/TotalScores into AltRanks/AltScores for alternative a

  1. AT7(D2(a),V1(a)) (Lookup AltScores for a in context of TotalScore for a))

- **T29:** Discover differences in AltRanks/AltScores for evaluator e before and after changing CritWeights

  1. AT6(D2(e)_before) $\rightarrow X$ (Locate AltScores for e in the 'before' dataset)
  2. AT6(D2(e)_after) $\rightarrow Y$ (Locate AltScores for e in the 'after' dataset)
  3. AT3(X,Y) (Get differences between AltScores distributions)

- **T30:** Discover differences in AltRanks/AltScores for evaluator e before and after changing non-weight component of preference model

  1. Same as T29

- **T31:** Discover differences in TotalScores before and after changing EvaluatorWeights

  1. AT6(D1)_before) $\rightarrow X$ (Locate D1 in the 'before' dataset)
  2. AT6(D1)_after) $\rightarrow Y$ (Locate D1 in the 'after' dataset)
  3. AT3(X,Y) (Get differences between D1s)

- **T32:** Discover differences in TotalRanks/TotalScores from two different aggregation methods

  1. AT6(D1)_method1) $\rightarrow X$ (Locate TotalScores in the 'method1' dataset)

2. AT6(D1)_method2) $\rightarrow Y$ (Locate TotalScores in the 'method2' dataset)
3. AT3(X,Y) (Get differences between TotalScores distributions)

- **T33:** Discover differences between CritWeights for evaluator e and CritWeights for other evaluators

  1. AT6(D8(e)) $\rightarrow X$ (Locate CrightWeights for e)
  2. AT3(X,D8(e')) for $e' \in E$ (Get differences between CrightWeights for e and every other evaluator)

- **T34:** Discover differences between non-weight component of preference model for evaluator e to that for other evaluators

  1. Analogous to T33 - simply replace CritWeights with the distribution corresponding to the base measure for the taxonomy level

Chapter 5

A Design Space of Visualizations to Support Preference Synthesis in Group Preferential Choice

This chapter presents a design space for visualizations to support inspection and exploration of multiple evaluators' preferences in the context of Group Preferential Choice. This is not intended to cover *all* possible designs, but rather, a viable subset that designers can choose from to suit their needs. We discuss the strengths and weaknesses of the various options, analytically evaluate their efficacy for different tasks, and offer recommendations based on contextual features. Such designs can be used in isolation or integrated into more sophisticated decision support systems.

At this time, we focus solely on Level P0b of the Preference Model Taxonomy (Section 3.8). Furthermore, we limit the design space to Group Preferential Choice scenarios where:

1. There are no more than a dozen alternatives or evaluators.[1]
2. The Evaluator hierarchy is *flat* - that is, there is only one group that contains all evaluators.
3. Preferences are expressed on a scale with *no negative values*. This is impor-

[1] This threshold was selected because colour is effective for encoding up to a dozen distinct identities. Beyond this, other strategies are required.

tant because diverging scales have somewhat different design implications [47].

Chapter 6 will briefly discuss how the design space might be extended to cover other levels of the taxonomy and scenarios that do not meet restrictions above.

The inputs to this analysis are the data and task abstractions developed in Chapter 4, with the exception of the tasks related to sensitivity analysis (T29 - T33), which we leave to future work. At this time, we only consider tasks that do not involve manipulating the underlying data. The design space is described in terms of the following *design aspects*:

1. Static design aspect (Section 5.1) - the basic idioms that are available and various options for mapping the dimensions and measures to marks and channels.
2. Dynamic design aspect (Section 5.2) - the mechanisms for transforming the data and the view, including:
   (a) View transformations, which change *how* the data is shown
   (b) Data transformations, which change *what* data is shown
3. Composite design aspect (Section 5.3) - the options for arranging and coordinating different views relative to each other.

This chapter will use of a running example of seven friends - Beth, Darnell, Janelle, Jessica, Joel, and Taycee - trying to choose a hotel to stay at - Budget, Days Inn, or Fairmont. Each friend scored each hotel on a scale from 0 to 1.

5.1 Static Design Aspect

This section describes the static design aspect for Level P0b of the Preference Model Taxonomy. It introduces the major competitive idioms for presenting small-scale tabular data with categorical keys and numeric values. As such, **it provides the basic building blocks from which the entire design space for all levels of the taxonomy may be built.** We limit our discussion to idioms that encode values using *position on a common scale*, as it is the most effective channel for encoding magnitude [37]. Idioms that use less effective channels in exchange for greater

information density, such as heatmaps, are more appropriate for larger datasets. These are discussed in Chapter 6.

Section 5.1.1 discusses each of these idioms in turn, and Section 5.1.2 performs an analytic evaluation of the idioms based on the tasks identified in Chapter 4.

5.1.1 Major Idioms

We start with the simplest case in which there are no evaluator weights. At this level, evaluators score the alternatives holistically according to their preferences. To recap, the data consists of:

- 2 - 12 Evaluators (E)
- 2 - 12 Alternatives (A)
- $|A|x|E|$ AltScores
- $|A|$ TotalScores

The data abstraction is a two-dimensional table with Evaluators and Alternatives as categorical keys and AltScores as numeric values. The TotalScores are obtained by summing AltScores over Evaluators.

Note that all non-radial designs described in this section can be oriented horizontally or vertically. For succinctness, we show the horizontal orientation only.

Bar-based Idioms

One of the most common ways to represent tabular data is the bar chart [38]. Bar charts redundantly encode values using two perceptual channels: position and length. There are three styles of bar charts that are suitable for presenting two-dimensional tabular data: stacked bar charts, multi-bar charts, and tabular bar charts [25].

Stacked Bar Chart

Stacked bar charts are appropriate when a one-dimensional measure is the sum of a two-dimensional measure, as is the case with TotalScores and AltScores [38] [25].

The stacked bar chart in Figure 5.1 maps alternatives to bars and evaluators to segments. The TotalScore of each alternative is encoded by the length and position

of its bar, while the AltScore for each evaluator is encoded by segment length. To improve discriminability, the segments are typically assigned different colour hues [38].

Because unaligned lengths are more difficult to compare than aligned lengths, the stacked bar chart is not particularly effective for tasks that require comparison of AltScores [38] [55]. However, they *are* effective at supporting TotalScore comparisons while also providing extra information about the relative contribution of each AltScore to the TotalScore.

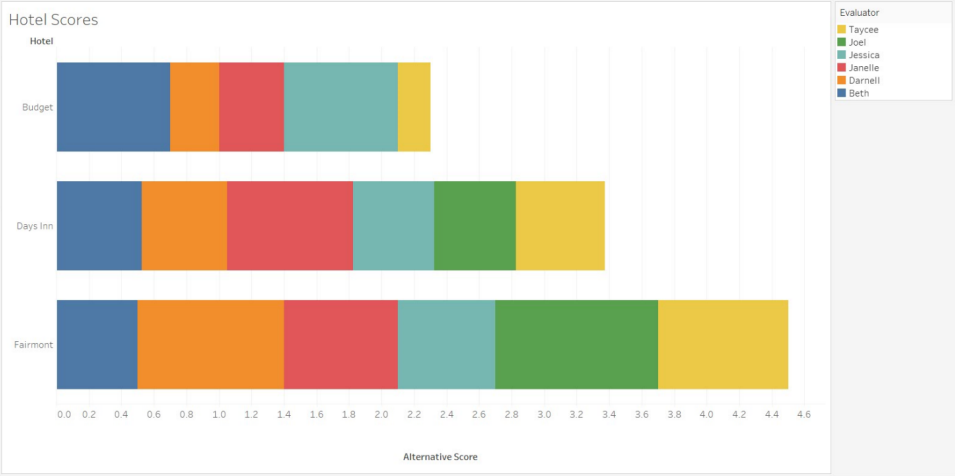


Figure 5.1: Stacked Bar Chart. Each bar encodes the TotalScore for each hotel. The segment lengths correspond the AltScores for each evaluator.

Multi-bar Chart

Multi-bar charts map spatial regions to dimensions in a nested fashion such that all bars are aligned to a common baseline. Additionally, color hue may be mapped to the secondary grouping to facilitate comparison across regions.

Figures 5.2 and 5.3 show the two possible designs given the available mappings from spatial region and color hue to Evaluators and Alternatives.

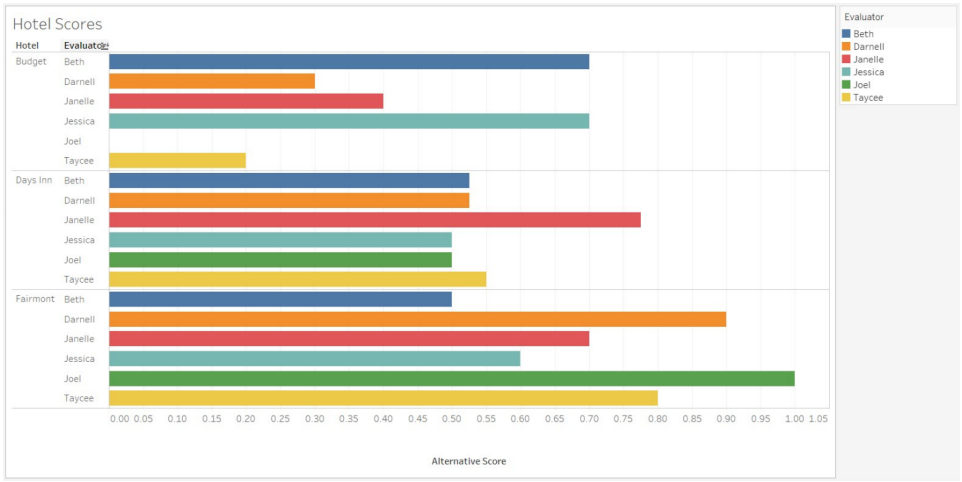


Figure 5.2: Multi-bar Chart Design 1: bars are grouped by alternatives, and colour is mapped to evaluators.

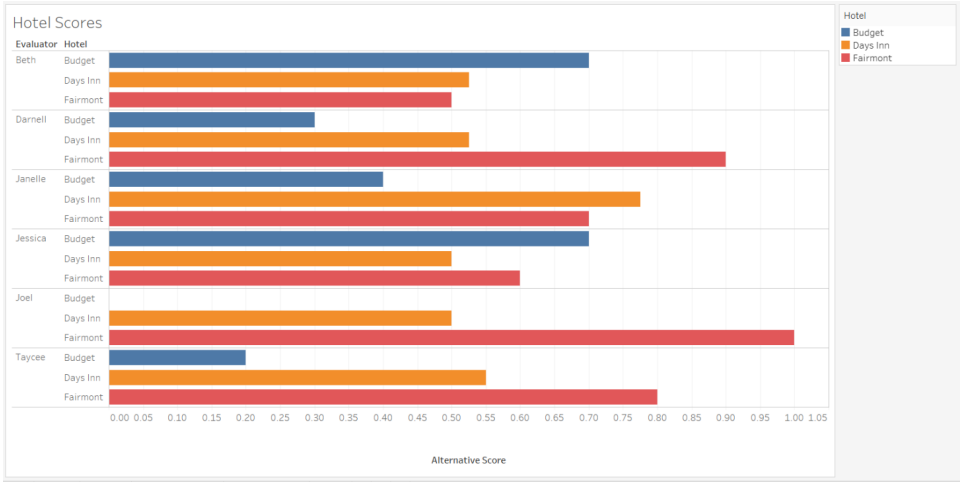


Figure 5.3: Multi-bar Chart Design 2: bars are grouped by evaluators, and colour is mapped to alternatives.

Tabular Bar Chart

Tabular bar charts map dimensions to spatial regions in a grid. There are four possible designs given the available mappings from spatial region and color hue to evaluators and alternatives. Figures 5.4 and 5.5 show the versions that map colour

to the column's dimension (Figures 5.4 and 5.5).

Note that Figure 5.4 pairs nicely with Figure 5.1, since a stacked bar chart can be transformed into a tabular bar chart simply by pulling apart the segments and aligning them to their own baseline. This pairing would allow users to easily transition between the tasks of comparing TotalScores, inspecting the breakdown of TotalScores into AltScores, and comparing AltScores for a particular evaluator.

Figure 5.4: Tabular Bar Chart Design 1: alternatives on rows and evaluators on columns. Colour is mapped to evaluators.

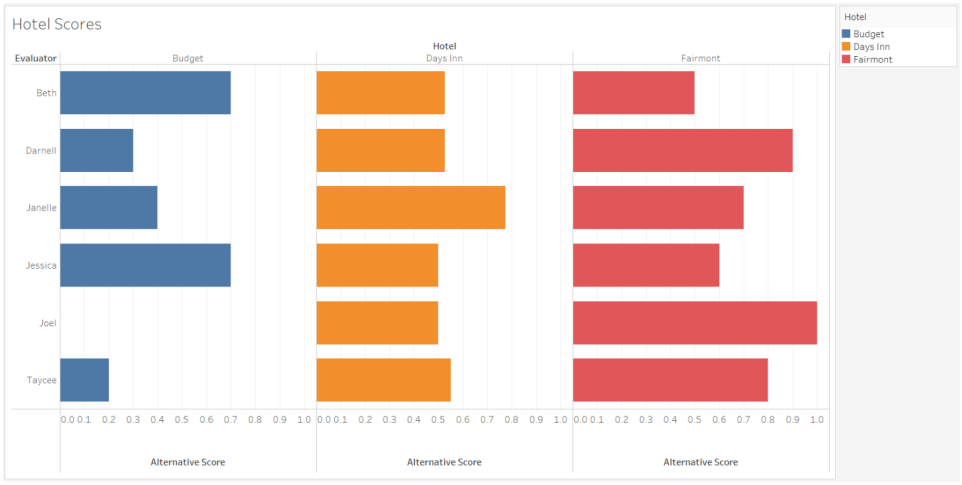


Figure 5.5: Tabular Bar Chart Design 2: evaluators on rows and alternatives on columns. Colour is mapped to alternatives.

Tabular bar charts are more compact than multi-bar charts of the same size, but they are also less precise because the same axis range is compressed and repeated across columns. Another weakness of tabular bar charts is that each column has its own baseline, and so comparisons across columns are less accurate than comparisons across regions in multi-bar charts [55].

Point-based Idioms

Strip Plot

The simplest of the point-based idioms is the *strip plot*, which uses position along a common axis to encode values. Two-dimensional tabular data can be represented as a series of strip plots with one dimension separated by region (each with its own strip plot) and the other distinguished using another channel, typically colour hue.[2] Figures 5.6 and 5.7 show the two possibilities.

[2]Colour hue is the second most effective channel for encoding categorical attributes after spatial region [38]. Another option is mark shape, which is sometimes used redundantly along with colour hue [46].

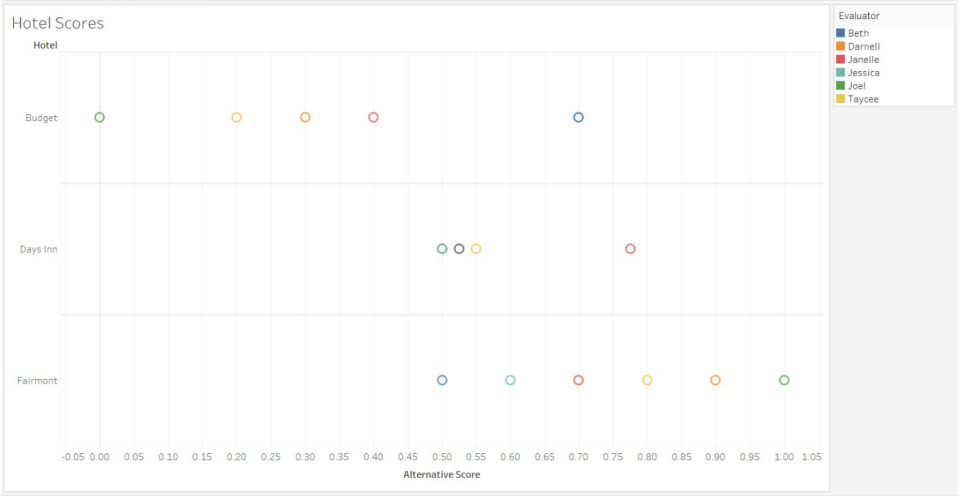


Figure 5.6: Strip Plot Design 1: alternatives on axes and evaluators on points.

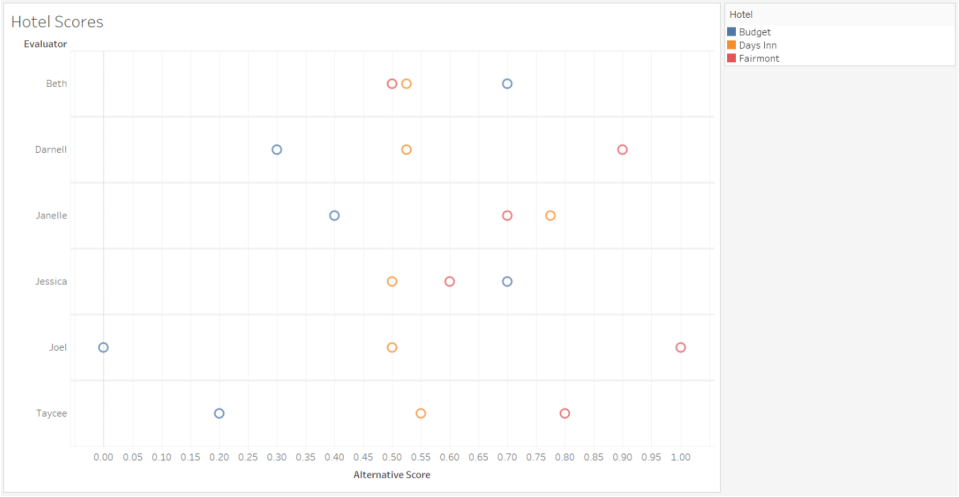


Figure 5.7: Strip Plot Design 2: evaluators on axes and alternatives on points.

The key strength of strip plots relative to bar charts is that they place an entire dimension along a single axis. In doing so, they unite the precision of multi-bar charts with the compactness of tabular bar charts. This property also makes them superior to bar charts for tasks related to spread, such as identifying clusters and outliers, since the user only needs to scan a single spatial dimension to obtain all

relevant information.

However, strip plots are less effective than bar charts at supporting look-up tasks because the secondary dimension is differentiated using colour alone. The necessity of colour also limits their scalability, since people can only differentiate up to around a dozen hues [38]. Their efficacy is contingent on the quality of the colour palette, which should be highly discriminable and accessible to individuals with colour-blindness [38].

Another challenge associated with strip plots is that occlusion may occur if two or more points have the same (or nearly the same) value. This is especially likely to become a problem if a discrete evaluation scale is used. There are several ways to address this challenge, including mark transparency, fill removal, jittering or stacking, or using another channel such as shape to redundantly encode point identity [24] [19]. Perhaps the most scalable option is a combination of stacking and fill removal, which means plotting multiple unfilled points (as in Figure 5.6) in a vertical 'stack' at the same x-coordinate.

Finally, point-based idioms are ill-suited to showing part-whole relationships. An additional plot could be added to Strip Plot Design 2 that shows evaluator averages, but this would not show how the parts contribute to the total.

Strip Plot Enhancements

Strip plots can be augmented in one of two ways to support comparison of distributions across either dimension. First, each axis can be overlaid with distribution information in the form of range plots, box plots, or violin plots. This further increases their efficacy for tasks related to spread along an axis. For succinctness, we will only consider box plots.

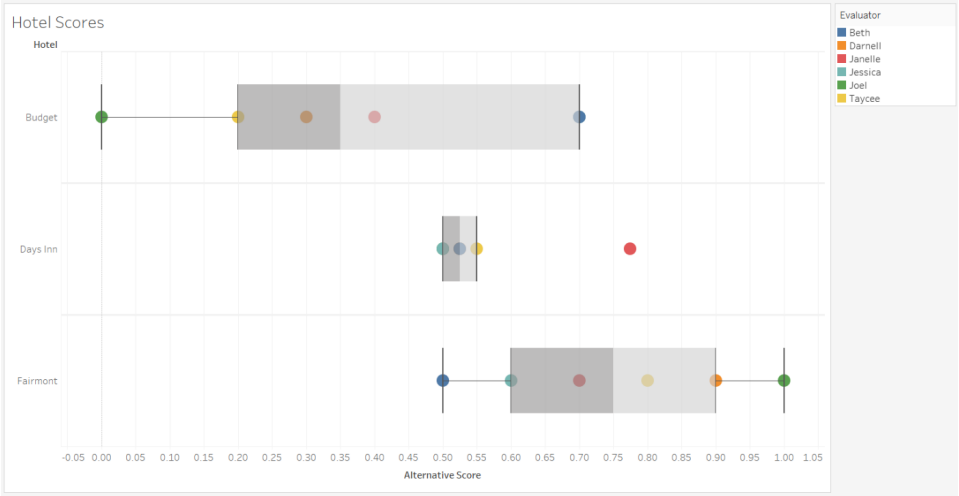


Figure 5.8: Box Plot Design 1 (Note: the gray fill is a feature of Tableau's box plot design. We do not recommend using a fill, as it makes it more difficult to differentiate the colours.)

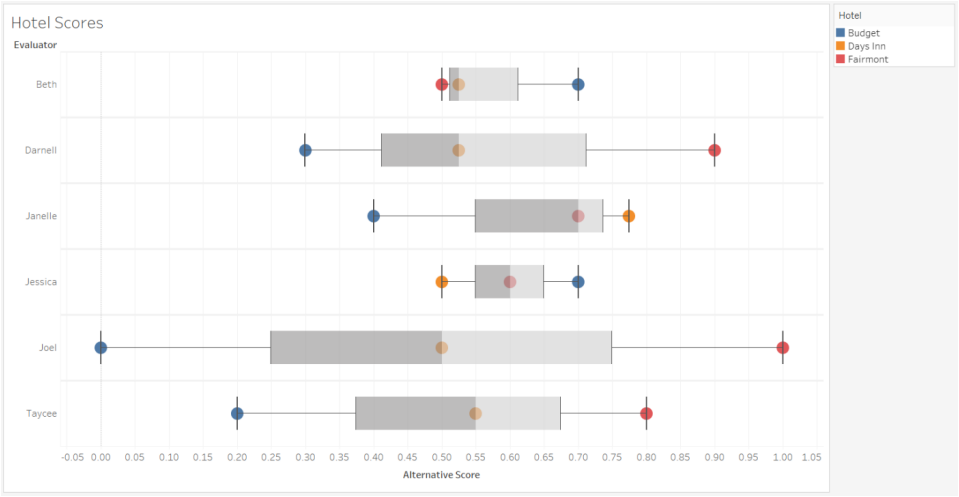


Figure 5.9: Box Plot Design 2

Alternatively, the points corresponding to items in the secondary dimension can be connected with straight lines of the same colour. This enhancement transforms the strip plot into another popular idiom - parallel coordinates. This design supports

tasks related to inspection and comparison *across* axes. However, its effectiveness for these tasks depends on the order of the axes [38].

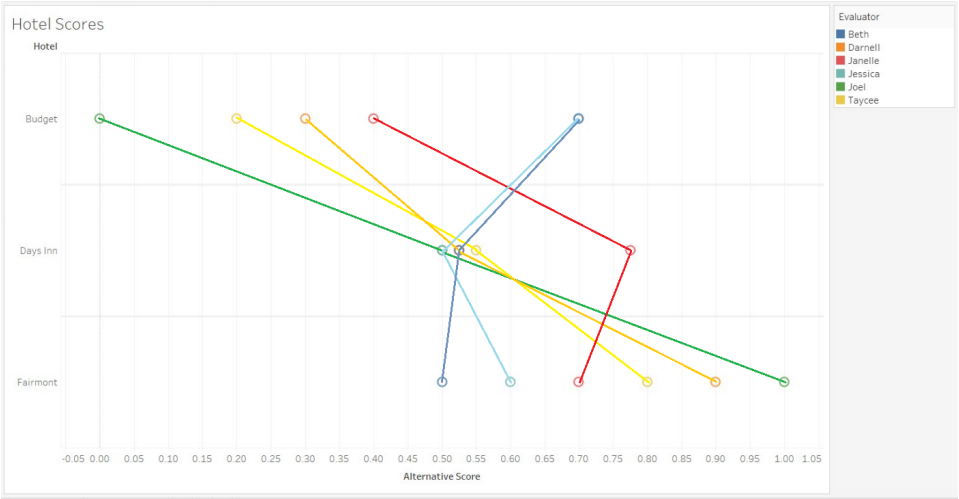


Figure 5.10: Parallel Coordinates Design 1: alternatives on axes and evaluators on lines.

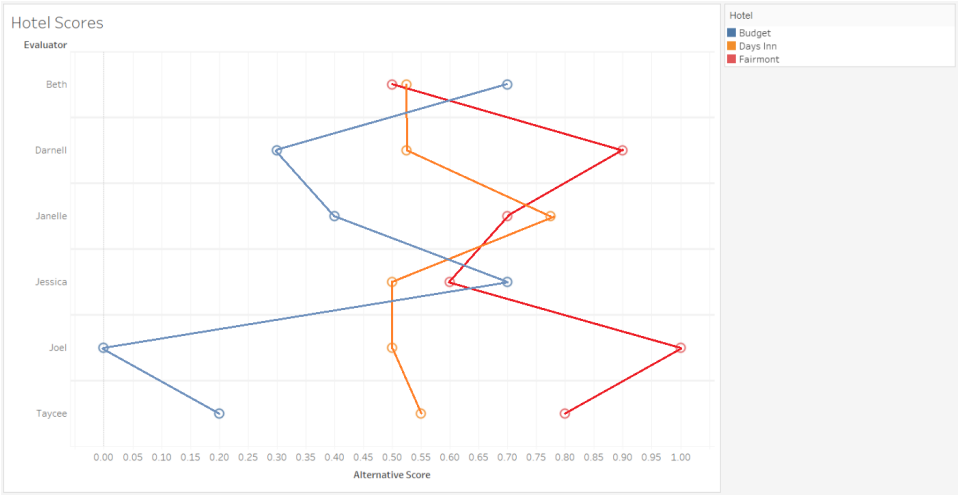


Figure 5.11: Parallel Coordinates Design 2: evaluators on axes and alternatives on lines.

A variation on parallel coordinates is the *radar chart*, which arranges the axes

radially (Figures 5.13 and 5.14). For the most part, radar charts are effective for the same tasks as parallel coordinates. However, they are less effective for comparison of values across axes since the axes are not aligned. Furthermore, their cyclic layout may be misleading if the data itself is not cyclic [38].

Yet another problem with radar charts is that a value of zero on one axis will cause the polygon to collapse on top of the neighboring axes. Figure 5.12 illustrates this problem using a simple example where Ann and Carol have assigned scores of 0 to Days Inn and Budget respectively. Also, crowding gets worse the closer the scores are to 0. For these reasons, the overlap problem for radar charts is much more complicated than it is for strip plots and parallel coordinates.

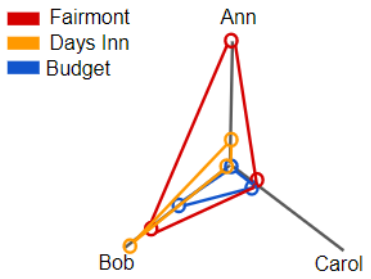


Figure 5.12: Troublesome radar chart.

One benefit of radar charts is that polygon area is roughly proportional to the squared sum of the axis scores. This means that Radar Chart Design 2 (Figure 5.14) roughly encodes TotalScores. Although area is a less effective channel for encoding magnitude than position and length [38], it may be useful to have this information for additional context.

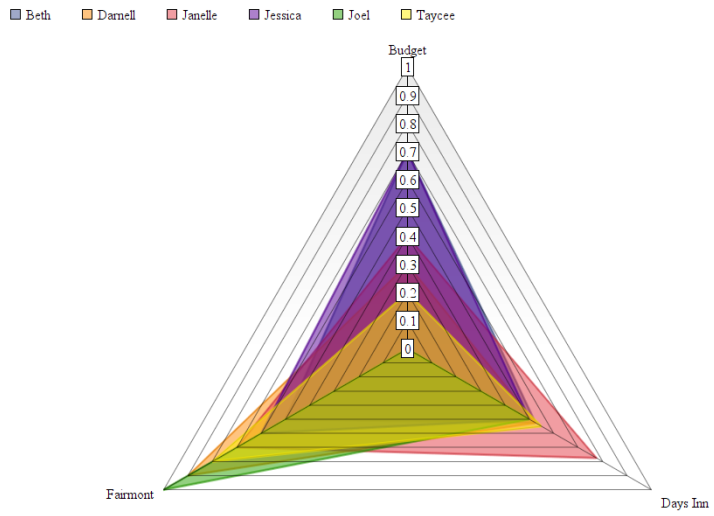


Figure 5.13: Radar Chart Design 1: alternatives on axes and evaluators on polygons. (Note: this figure was generated using onlinecharttool.com, and the polygon fill is a feature of their radar chart design. It is not recommended, as the blending of colours makes it more difficult to identify the boundaries.)

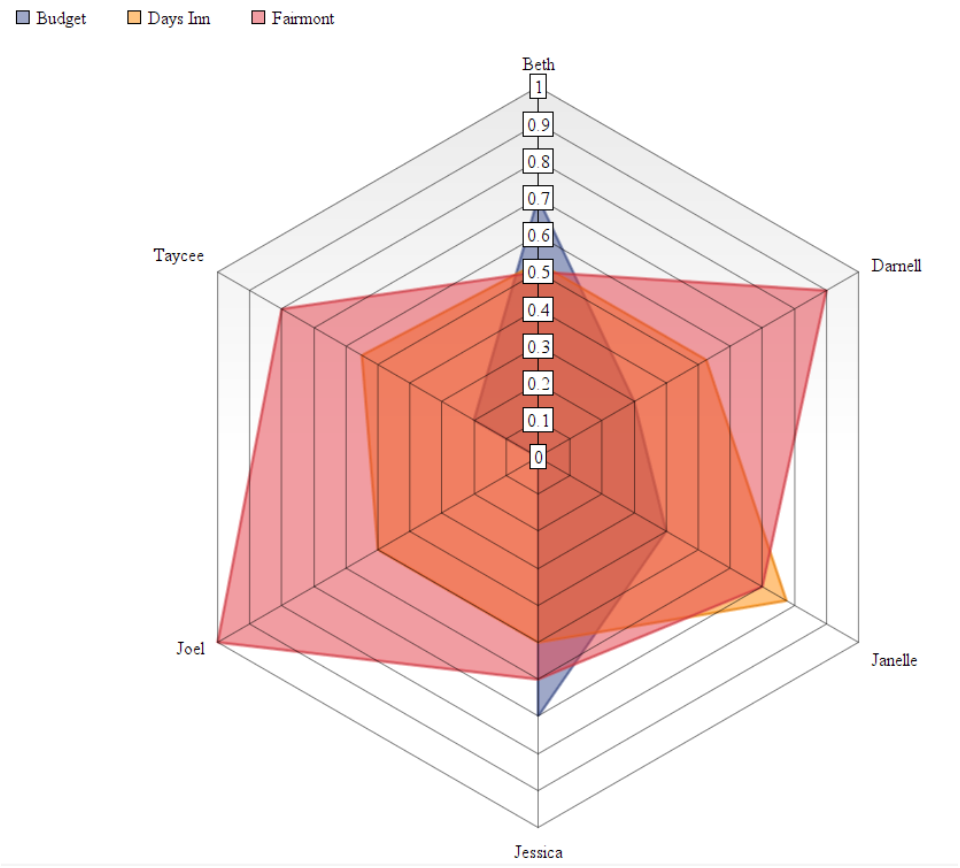


Figure 5.14: Radar Chart Design 2: evaluators on axes and alternatives on polygons.

With Evaluator Weights

Introducing evaluator weights means adding the following measures to the dataset:

- $|A|x|E|$ UnweightedAltScores (scores before applying the weights)
- $|E|$ EvaluatorWeights

Integrated View

The most straightforward way to show the relationship between the original scores (UnweightedAltScores), the weighted scores (AltScores), and the evaluator weights

is with a modified version of Tabular Bar Chart Design 1 (Figure 5.4) where the column widths are proportional to the corresponding EvaluatorWeights. This effectively compresses each axis into an amount of space proportional to the weight of that evaluator. This design is shown in Figure 5.15.

Figure 5.15: Tabular Bar Chart Design 1 with variable column widths. The width of each column encodes the weight of each evaluator. The relative width of each bar within its column encodes the UnweightedAltScore. The absolute width of each bar encodes the AltScore (that is, the product of the UnweightedAltScore and EvaluatorWeight).

This encoding pairs nicely with a stacked bar chart where the segments correspond to the AltScores (Figure 5.16). No other idiom can be as easily adapted to show the part-whole relationship between AltScores and EvaluatorWeights.

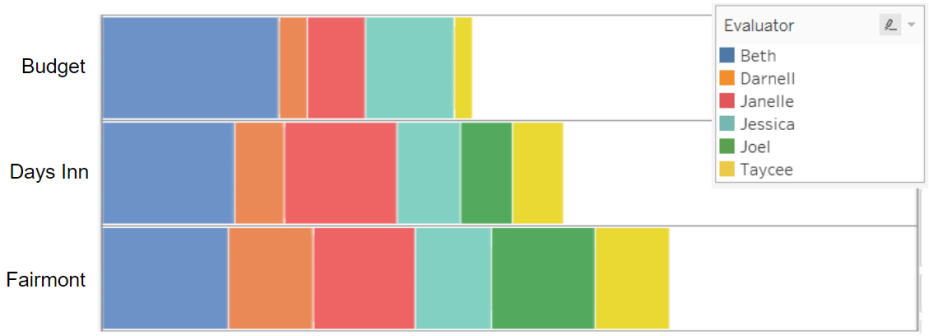


Figure 5.16: Stacked Bar Chart corresponding to Tabular Bar Chart in Figure 5.15. The width of each segment encodes the weighted AltScore, and the length and position of each bar encodes the TotalScore.

Separate Views

An alternative approach is to show the AltScores, UnweightedAltScores, and EvaluatorWeights independently in separate views. This is not recommended, as it obscures the relationship between the measures. We especially advise against showing the AltScores apart from EvaluatorWeights, as this may lead users to erroneously attribute differences in scores to differences in preferences when they are actually due to differences in weights.

However, it may be sensible to *supplement* the integrated view with additional views that better support certain tasks. We will return to this discussion in Section 5.3.

5.1.2 Task-based Evaluation of Encodings

Section 4.2 showed how various tasks identified in our analysis can be decomposed into auxiliary tasks on particular values and distributions. This section performs an in-depth assessment of the suitability of each encoding for each task-input pair that supports some high-level task for Level P0b. Table 5.1 summarizes the possible inputs to each task.

Table 5.1: Possible Inputs for Each Auxiliary Task

	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	AT9	AT10
A single AltScore mark	✓									
A single TotalScore mark							✓			
A single EvaluatorWeight mark							✓			
A pair of AltScore marks for one evaluator		✓								
A pair of AltScore marks for one alternative		✓								
The set of AltScore marks for one evaluator									✓	
The set of AltScore marks for one alternative					✓			✓		
The set of AltScore marks for a pair of alternatives			✓	✓						
The set of AltScore marks for a pair of evaluators			✓							
The set of all AltScore marks										✓
The set of all TotalScore marks									✓	
A single evaluator						✓	✓			
A single alternative						✓	✓			
An evaluator/alternative pair						✓				

Tasks that apply to more than one type of the input (AT2, AT3, AT6, AT7, and AT9) are split into cases in the descriptions below. **Note that much of this evaluation is speculative and will require empirical validation.** The results of this assessment are summarized in Figures 5.20 and 5.21.

AT1: Identify a mark

The input to this task is a single AltScore mark, and the output is the alternative and evaluator it corresponds to.

Bar charts are the most effective for this task because each mark occupies a labeled region, and the user does not need to consult a color key. Furthermore, there is no risk of marks overlapping. Tabular bar charts may be superior to multi-bar charts because they do not nest labels, and this could improve legibility.

Whether or not there are differences among the point-based idioms is less definitive. The connecting lines in parallel coordinates and radar charts may improve identification speed by increasing the salience of the colour. Box plots do not provide anything useful for this task.

AT2: Compare values (Case A: one evaluator, two alternatives)

The input to this task is a pair of AltScore marks for one evaluator, and the output is an approximate difference.

There are numerous factors to consider when ranking the encodings for this

task. Figure 5.17 provides an overview of the key ideas using a simplified version of the hotel problem. The box plots are omitted because the example data set is very small.

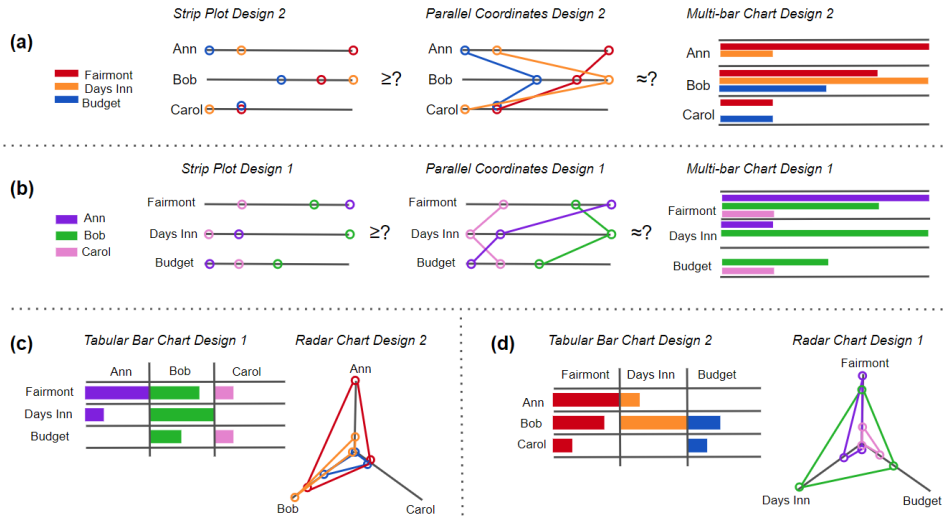


Figure 5.17: What is the best encoding for comparing Budget and Days Inn for Bob? This figure divides the encodings into four efficacy groups according to key principles. (a) Highly effective - comparisons are performed along a single axis or within single region; (b) Less effective - comparisons are made across axes or regions; (c) Less effective - axes are condensed and offer less precision; (d) Least effective - requires comparison of unaligned widths or positions. The rankings within groups (a) and (b) are nuanced, as discussed in the text.

The most critical factor is whether the values to compare are plotted on aligned axes. This is *not* the case for the Stacked Bar Chart, Tabular Bar Chart Design 2, and Radar Chart Design 1, so these are the least effective encodings for this task (Figure 5.17d).

Another important factor is precision, or how much space is allocated to each axis. Tabular Bar Chart Design 1 and Radar Chart Design 2 offer less precision because the axes are shorter relative to the area of the plot (Figure 5.17c). Furthermore, Radar Chart Design 2 may be at a disadvantage because not all axes are perpendicular to the line of site.

Of the remaining point-based idioms, Strip Plot Design 2 is superior to Strip Plot Design 1 (and its derivatives) because the positions to compare are located on the same axis. Similarly, Multi-bar Chart Design 2 is superior to the Multi-bar Chart Design 1 because the comparison is made within rather than across regions [56]. This division is illustrated in Figure 5.17a and 5.17b.

Within these two groups, it is unclear whether the box plot or parallel coordinate overlays for the strip plots would improve or interfere with performance - our intuition is that they might interfere by distorting the perception of distance.

It is also difficult to rank the multi-bar charts relative to the point-based idioms, as there are several factors that may contribute in subtle ways. Bar charts redundantly encode values using both the position and length channels, which may strengthen their efficacy for comparison tasks. However, they are more cluttered than point-based idioms [46], and their efficacy is sensitive to sort order - non-adjacent bars are more difficult to compare than adjacent bars because they are further apart and the viewer must ignore the bars in between [56]. This problem can be mitigated by giving users the ability to filter alternatives.

Point-based idioms are more succinct than multi-bar charts because they do not use length to encode values. Also, there is a more direct relationship between relative positions and relative values - it is simply the distance between the points. Finally, the fact that each plot uses just one spatial dimensions means that ordinal relationships can be identified at a glance simply by checking which point lies to the left or right of the other. Point-based idioms also risk points overlapping, but there are several effective strategies for dealing with this [24] [19].

In light of these factors, we surmise that Strip Plot Design 2 is the best for this task overall.

AT2: Compare values (Case B: one alternative, two evaluators)
The input to this task is a pair of AltScore marks for one alternative, and the output is an approximate difference.

The evaluation of encodings for Case B mirrors that of the Case A with the Design numbers reversed. In other words, the most effective encodings are the Design 1 non-radial point-based idioms and Multi-bar Chart Design 1. We surmise that Strip Plot Design 1 is the best overall.

AT3: Compare distributions (Case A: all evaluators, two alternatives)
The input to this task is the set of AltScore marks for two alternatives, and the output is a rough approximation of the pairwise differences.

This task requires the user to keep two distribution in focus while performing multiple comparisons in sequence. As such, it is a hybrid of AT2 Case A and AT6 Case B, and the ranking of encodings reflects this (Figure 5.18).

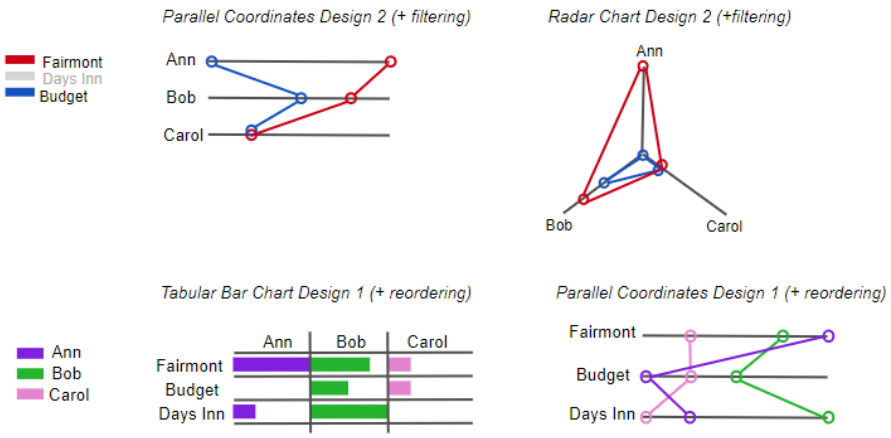


Figure 5.18: What is the best encoding for comparing Fairmont and Budget across all evaluators? Parallel Coordinates Design 2 makes it easy to perform multiple precise comparisons in sequence, especially if filtering is permitted. The other three encodings shown here are also effective, but each has its weaknesses.

Interestingly, the most effective encoding for this task may be Parallel Coordinates Design 2, as the connecting lines make it easy to keep the distributions in focus while the strip plot base makes it easy to perform individual comparisons. Furthermore, trade-offs can be identified at a glance by looking for line intersections. The same is true of Radar Chart Design 2, although the radial layout might make it harder to perform repeat comparisons with accuracy. A drawback of both is visual interference from other lines - this can be mitigated by allowing users to filter alternatives.

Another relatively effective encoding is the Tabular Bar Chart Design 1. The grid structure allows users to compare two rows of bars one column at a time, albeit with less precision than some of the other encodings. This is much easier if the two rows are adjacent. Parallel Coordinates Design 1 is also effective if the plots to be compared are adjacent.

Multi-bar charts are less effective because they require comparisons to be made across regions regardless of how the bars are sorted. Plain strip plots and box plots are also less effective because the absence of a grid or connecting lines makes it difficult to visually isolate each pair for comparison. This is true whether the distributions of interest lie along the plots (Design 1) or across the plots (Design 2). Again, the least effective encodings are those that require comparison of unaligned position and widths - the Stacked Bar Chart, Tabular Bar Chart Design 2, and Radar Chart Design 1.

AT3: Compare distributions (Case B: all alternatives, two evaluators)
The input to this task is the set of AltScore marks for two evaluators, and the output is a rough approximation of the pairwise differences.

The ranking of encodings for Case B mirrors that of the Case A with the Design numbers reversed.

AT4: Identify a dominance relation
The input to this task is a set of AltScore marks for a pair of alternatives, and the output is an assessment of whether one dominates the other.

This task is a special case of AT3 Case A, so the evaluation of encodings is similar. Notice that there is a dominance relationship between Fairmont and Budget in Figure 5.18, so it applies to this task as well.

The best encodings for this task are Parallel Coordinates Design 2 and Radar Chart Design 2, as a dominance relation can be easily identified by checking if the lines intersect. In Radar Charts Design 2, this amounts to checking for *enclosure*. As in AT3 Case A, interference from other lines can be eliminated by filtering alternatives.

The next most effective encoding is Tabular Bar Chart Design 1, as users can identify dominance by comparing two rows, one column at a time. This is easier if

the two rows are adjacent.

Finally, the Design 1 non-radial point-based idioms are also somewhat effective, since a dominance relationship can be identified by checking whether each point lies to the left of the same-coloured point in the other plot. This is easier to do if the plots are adjacent. The connecting lines in Parallel Coordinates Design 1 may help, since all connecting lines will tilt in the same direction or not at all if one alternative dominates the other (see Figure 5.18).

The remaining encodings are not effective for this task for reasons similar to those discussed in AT2 and AT3.

AT5: Summarize variance

The input to this task is a set of AltScores for a single alternative, and the output is a rough approximation of how much variation there is in the set.

Box Plot Design 1 is the best encoding for this task, as it provides direct information about the distribution and range. The next best encoding is Strip Plot Design 1 and its other derivatives, as it enables the user to inspect the range and distribution by scanning a single spatial dimension. Radar Chart Design 1 may be at a slight disadvantage because not all axes are perpendicular to the line of site.

Variance can be roughly assessed using Multi-bar Chart Design 1 by looking at the variation in bar length within a region. This can also be done with Tabular Bar Chart Design 2, albeit with less precision. This is more challenging with Multi-bar Chart Design 2 since the comparisons must be made across regions.

Variance can also be roughly assessed using Parallel Coordinates Design 2 and Radar Chart Design 2 by examining the smoothness of the line or polygon. However, this relationship is sensitive to axis order - the impression of variance may be exaggerated if clusters are split.

The remaining encodings are not effective for this task for reasons similar to those discussed in AT2 and AT3.

AT6: Locate a value for a key-set (Case A: one alternative, one evaluator)

The input to this task is an alternative/evaluator pair and the output is the AltScore value for that pair.

Bar charts are best for look-up tasks for the same reason that they are good for

identification tasks - each mark is assigned to a particular region, so it is possible to look up values without discriminating colour.

Which style of bar chart is best may depend on the size of the dataset and the amount of space allocated to the view. The nesting of labels in multi-bar charts may result in more crowding, but it may also reduce the amount of area the user needs to scan to find the labels of interest.

AT6: Locate a distribution for a key-set (Case B: one alternative)
The input to this task is an alternative and the output is the distribution of AltScores for that alternative.

The best encoding for this task is Tabular Bar Chart 2, since it differentiates alternatives using both contiguous spatial region *and* colour hue. The next best encodings are the Design 1 encodings, as these assign alternatives to contiguous spatial regions.

The Stacked Bar Chart and Multi-bar Chart Design 2 assign alternatives to non-contiguous spatial regions, and users must tune out the bars in between. Parallel Coordinates Design 2 and Radar Chart Design 2 map alternatives to connected lines, but users must tune out the other lines that occupy the same space.

For the remaining encodings, the user must visually group disconnected marks based on colour alone, which is substantially more difficult. Filtering can reduce the amount of interference in all cases.

AT6: Locate a distribution for a key-set (Case C: one evaluator)
The input to this task is an evaluator and the output is the distribution of AltScores for that evaluator.

The evaluations of encodings is the same as in AT6 Case B, except with the Design numbers reversed.

AT7: Look-up value in context (Case A: AltScore in TotalScore)
The input to this task is an evaluator and a TotalScore mark for some alternative, and the output is the AltScore for that evaluator and alternative. In other words, the task is to identify the contribution of some evaluator's AltScore to the TotalScore.

If EvaluatorWeights are defined, then the only applicable encoding for this task

is the Stacked Bar Chart. Otherwise, Radar Chart Design 2 is also weakly effective for this task, since the area of each polygon is roughly proportional to the TotalScore squared.

AT7: Look-up value in context (Case B: UnweightedAltScore in EvaluatorWeight)
The input to this task is an alternative and an EvaluatorWeight mark for some evaluator, and the output is UnweightedAltScore for that alternative. **This task is only applicable when EvaluatorWeights are defined.**

The only applicable encoding for this task is the Tabular Bar Chart with Variable Widths. Using this encoding, the task can be achieved by assessing what fraction of the evaluator's column is filled by the bar.

AT8: Browse for outliers
The input to this task is a set of AltScores for a single alternative, and the output is a set outliers.

Box Plot Design 1 is best for this task, since it encodes outliers explicitly. Strip Plot Design 1 (and its other derivatives) are also effective, since outliers can be identified simply by finding points that are relatively far from the others.

Outliers can be detected in bar charts by identifying bars that are much longer or shorter than others in their region. Large outliers are more perceptually salient than small outliers because they 'stick out' from the others. Multi-bar Chart Design 1 is the most effective of the bar-based idioms due to its precision and the proximity of the bars. Tabular Bar Chart Design 2 is less precise, while Multi-bar Chart Design 2 requires comparison of bars across regions. Sorting based on AltScore could increase the efficacy of bar charts for this task.

Outliers can be detected using Parallel Coordinates Design 2 or Radar Chart Design 2 by looking for non-recurrent spikes. Unlike bar charts, these are not perceptually biased toward large outliers, but they are disadvantaged in that the lines overlap with each other and may interfere perceptually.

The remaining Design 2 point-based idioms are not effective because it is too difficult to visually isolate the distribution of interest (see AT6: Case B). The Stacked Bar Chart and Tabular Bar Chart 1 are the least effective because they require comparison of unaligned widths.

AT9: Browse for top/bottom values (Case A: one evaluator)
The input to this task is a set of AltScores for a single evaluator, and the output is a set of top or bottom values.

The relative strengths of the encodings for identifying top and bottom values are similar to those of task AT8, except this case is concerned with a distribution over alternatives.

The best encoding for this task is Strip Plot Design 2 (and its derivatives), as the top and bottom values are simply the points furthest to the left or right along a single axis. This could be harder with Radar Chart Design 2 because the axis of interest might not be perpendicular to the line of site.

The remainder of the assessment mirrors that of AT8 with the Design numbers reversed. In this case especially, the ability to sort bar charts by AltScore could significantly improve task performance.

AT9: Browse for top/bottom values (Case B: all data)
The input to this task is a set of TotalScores for all alternatives, and the output is a set of top values. If EvaluatorWeights are defined, the only applicable encoding for this task is the Stacked Bar Chart. Otherwise, an 'Average Evaluator' can be added to any of the other plots to show the average scores (which is effectively the same as showing the total scores). In this case, the efficacy of each encoding is the same as for AT9 Case A.

AT10: Browse for non-dominated distributions
The input to this task is all the AltScores, and the output is a set of dominance relationships between the alternatives.

In the worst case, this simply requires performing task AT4 for every pair of alternatives, but this is not necessary most of the time. Dominance relationships can be identified at a glance using Parallel Coordinates Design 2 or Radar Chart Design 2 by looking for lines or sets of lines that do not intersect.

The efficacy of the Design 1 non-radial point-based idioms for this task can be improved by sorting the plots by TotalScore so that fewer comparisons need to be made. The same is true of Tabular Bar Chart Design 1 (Figure 5.19).

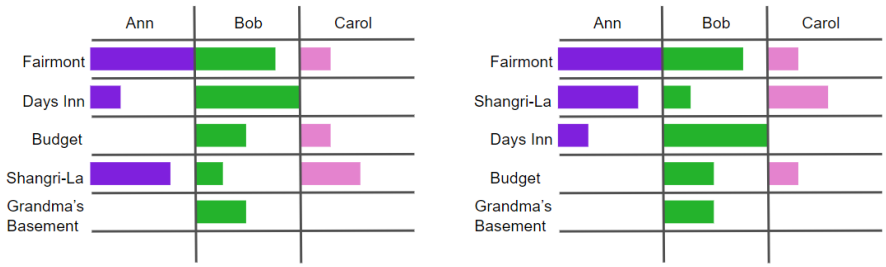


Figure 5.19: It is easier to identify dominated alternatives when the rows are sorted by TotalScore (right) than when they are not (left). When the rows are sorted, each row only needs to be compared to the rows above it. In this example, Grandma's Basement is dominated by Budget, which is dominated by Fairmont.

The remaining encodings are not effective for this task for reasons discussed in AT4.

Summary of Task-based Assessment

Table 5.20 summarizes the results of the task-based assessment when EvaluatorWeights are defined. For each task, the best encodings are assigned a score of 3, strongly effective encodings are assigned a score of 2, weakly effective encodings are assigned a score of 1, and ineffective encodings are assigned a score of 0. Inapplicable encodings are marked with a hyphen.

Table 5.21 summarizes the same information when EvaluatorWeights are *not* defined. Note that the only differences between the two tables are:

1. Table 5.21 does not have a row for Tabular Bart Chart Design 1 with Variable Widths (it is not applicable).
2. Table 5.21 does not have a column for AT7: B (it is not applicable).
3. Some of the scores for AT7: A and AT9: B are different for reasons discussed in the text.

	AT1	AT2: A	AT2: B	AT3: A	AT3: B	AT4	AT5	AT6: A	AT6: B	AT6: C	AT7: A	AT7: B	AT8	AT9: A	AT9: B	AT10	Total Score
Parallel Coordinates 2	1	2	2	3	2	3	1	1	1	2	-	-	1	3	-	3	25
Tabular Bar 1 (w/ weights)	3	1	0	2	0	2	0	3	2	3	-	3	0	1	-	2	22
Parallel Coordinates 1	1	1	2	2	3	2	2	1	2	0	-	-	2	1	-	2	21
Tabular Bar 1	3	1	0	2	0	2	1	3	2	3	-	-	0	1	-	2	20
Radar Chart 2	1	1	1	2	0	3	1	1	1	2	-	-	1	2	-	3	19
Multi-bar 2	2	2	1	1	1	1	1	3	1	2	-	-	1	2	-	1	19
Multi-bar 1	2	1	2	1	1	1	1	3	2	1	-	-	2	1	-	1	19
Box Plot 1	1	1	2	1	1	2	3	1	2	0	-	-	3	0	-	2	19
Strip Plot 1	1	1	3	1	1	2	2	1	2	0	-	-	2	0	-	2	18
Tabular Bar 2	3	0	1	0	2	0	1	3	3	2	-	-	1	0	-	0	16
Strip Plot 2	1	3	1	1	1	1	0	1	0	2	-	-	0	3	-	1	15
Box Plot 2	1	2	1	1	1	1	0	1	0	2	-	-	0	3	-	1	14
Radar Chart 1	1	0	0	0	2	0	2	1	2	2	-	-	2	1	-	0	13
Stacked	1	0	0	0	0	0	0	1	1	2	3	-	0	0	3	0	11

Figure 5.20: Support for each auxiliary task by encoding when EvaluatorWeights are defined. 3 = best, 2 = strongly effective, 1 = weakly effective, 0 = ineffective. Gray cells indicate that the encoding is not applicable to that task. The rows are sorted by the **Total Score** column, which contains the sum of scores for each row.

	AT1	AT2: A	AT2: B	AT3: A	AT3: B	AT4	AT5	AT6: A	AT6: B	AT6: C	AT7: A	AT8	AT9: A	AT9: B	AT10	Total Score
Parallel Coordinates 2	1	2	2	3	2	3	1	1	1	2	-	1	3	3	3	28
Radar Chart 2	1	1	1	2	0	3	1	1	1	2	1	1	2	2	3	22
Parallel Coordinates 1	1	1	2	2	3	2	2	1	2	0	-	2	1	1	2	22
Tabular Bar 1	3	1	0	2	0	2	1	3	2	3	-	0	1	1	2	21
Multi-bar 2	2	2	1	1	1	1	1	3	1	2	-	1	2	2	1	21
Multi-bar 1	2	1	2	1	1	1	1	3	2	1	-	2	1	1	1	20
Box Plot 1	1	1	2	1	1	2	3	1	2	0	-	3	0	0	2	19
Strip Plot 1	1	1	3	1	1	2	2	1	2	0	-	2	0	0	2	18
Strip Plot 2	1	3	1	1	1	1	0	1	0	2	-	0	3	3	1	18
Box Plot 2	1	2	1	1	1	1	0	1	0	2	-	0	3	3	1	17
Tabular Bar 2	3	0	1	0	2	0	1	3	3	2	-	1	0	0	0	16
Radar Chart 1	1	0	0	0	2	0	2	1	2	2	-	2	1	1	0	14
Stacked	1	0	0	0	0	0	0	1	1	2	3	0	0	3	0	11

Figure 5.21: Support for each auxiliary task by encoding when EvaluatorWeights are **not** defined. 3 = best, 2 = strongly effective, 1 = weakly effective, 0 = ineffective. Gray cells indicate that the encoding is not applicable to that task. The rows are sorted by the **Total Score** column, which contains the sum of scores for each row.

What is apparent is that most tasks are strongly supported by at least one of the top two encodings in Table 5.20: Tabular Bar Chart Design 1 (with variable weights) and Parallel Coordinates 2. This suggests that these two encodings can be used in conjunction to support most tasks.

Another observation is that parallel coordinates dominate radar charts except for in AT7 Case A, and then only when there are no EvaluatorWeights. In other words, the only benefit that radar charts confer is that they weakly encode TotalScore via polygon area. In light of this and the numerous problems with radar

charts discussed earlier, we eliminate them from further consideration.

We will return to this discussion in Section 5.3 when we consider how different design choices may be combined to effectively support a variety of tasks.

5.2 Dynamic Design Aspect

This section describes a number of options for transforming the view so that the user can perform multiple analytic tasks in sequence or perform particular tasks more effectively.

5.2.1 View Transformations

Rearrange: Reorder and Sort

Allowing users to manually reorder rows, columns, and plots gives them control over which items are adjacent, and this can improve their performance on comparison tasks (AT2, AT3, and AT4). This is especially true for the bar-based idioms.

Allowing users to sort elements by TotalScore or AltScore for a particular evaluator or alternative can improve their performance on tasks related to identifying top values (AT9) or looking for dominance relationships (AT10). It can also help them perform further analysis on top performing options only. For instance, one evaluator might want to inspect how her top alternatives perform for other evaluators.

Rearrange: Change Mapping

Allowing users to change the mapping from dimensions to regions/marks gives them the flexibility to toggle between Designs 1 and 2 of each idiom. Whether or not this is advised depends on which idioms are already provided and how potential conflicts in the use of colour will be resolved (Section 5.3).

If a multi-bar chart or tabular bar chart is in use, users might also be permitted to select which dimension to map to colour, since it is not strictly dictated by the spatial mapping. This functionality would not greatly add to the users' ability to perform any of the identified tasks. Furthermore, it is not recommended if the tabular bar chart is paired with a stacked bar chart, as this would break the correspondence between the two.

Table 5.2 summarizes which rearrangements are applicable to each encoding. We do not recommend allowing users to manually reorder bars within regions of a multi-bar chart, as this could lead to inconsistency across regions. We also do not recommend allowing users to reorder the segments of a stacked bar chart. However, if the stacked bar chart is paired with a tabular bar chart, then changing the order of columns in the tabular bar chart should change the order of the segments as well.

Table 5.2: Applicable rearrangements for each encoding. Justifiable transformations are shown in green with a checkmark. Applicable but ill-advised transformations are shown in yellow with a question mark. Impossible or nonsensical transformations are shown in gray.

	Manually Reorder Alternatives	Manually Reorder Evaluators	Sort Alternatives by TotalScore	Sort Alternatives by AltScore (for an evaluator)	Sort Evaluators by AltScore (for an alternative)	Swap Region/Mark Mapping	Swap Colour Mapping
Stacked Bar Chart	✓	?	✓	✓	✓		
Multi-bar Chart Design 1	✓	?	✓	✓	✓	✓	✓
Multi-bar Chart Design 2	?	✓	✓	✓	✓	✓	✓
Tabular Bar Chart (Designs 1 and 2)	✓	✓	✓	✓	✓	✓	?
Point-based Design 1	✓		✓	✓		✓	
Point-based Design 2		✓		✓	✓	✓	

Add Emphasis

A final type of view transformation is the ability to emphasize or highlight an entity of interest. This technique alters the appearance of a mark to make it stand out - possible alterations include changing the hue, increasing saturation, or magnifying the mark. *Linked highlighting* adds emphasis to a set of entities that are related to the selected entity. In this case, related entities would be those of the same colour - that is, all other marks for a particular evaluator (in the case of Design 1 encodings) or alternative (in the case of Design 2 encodings). Linked highlighting could improve users' ability to locate distributions (AT6) and compare distributions (AT3), especially in cases where the distributions of interest are spread across regions or axes.

This design choice is coupled with the *select* design choice, which is the mechanism by which users choose items for further action (in this case, highlighting) [38]. One common mechanism that we recommend is *hover*, which selects an item for as long as the mouse hovers over it. It may also be worthwhile to allow users

to select multiple items for highlighting at once - this is typically done via mouse click. This would make it easier for users to keep two or more distributions in focus for comparison tasks.

5.2.2 Data Transformations

Filtering

There are two types of filtering a designer might want to support: filtering on entities and filtering on values.

Filtering on entities is the ability to select a subset of alternatives or evaluators to inspect at any time. This can facilitate any number of tasks by removing distracting elements. Filtering is especially important when working with parallel coordinates or radar charts, since the distributions occupy the same space.

Filtering on values is the ability to exclude alternatives based on TotalScore or AltScore for a particular evaluator. This would allow users to set *satisficing thresholds* that must be met for an alternative to be considered. This feature is not required to support any of the tasks we identified, but it could be useful in scenarios where satisficing thresholds are important.

Both types are filters are applicable to all encodings. There are a number of ways to implement filter controls, such as checklists for categorical entities or range sliders for quantitative entities. Another mechanism for filtering is *brushing*, which allows users to specify a region to filter out or leave in with a drag of the mouse. If this design choice is used, there also needs to be a clear mechanism for reversing the action.

Details-on-demand

Another type of transformation involves augmenting the display with more detailed information. For example, users might want to query the precise value encoded by a bar or mark, as this information may be difficult to glean from the graphical representation alone. Possible implementations of this feature include a label overlay that can be turned on or off or a tool-tip that appears when the user hovers over a mark. The tool-tip could also include the label for the mark in order to expedite identification (AT1).

Other forms of textual information designers might consider making available

on demand include averages, variances, and axis details. If evaluators supplemented their scores with text explanations, this could be displayed whenever a user clicks on the corresponding mark.

5.3 Composite Design Aspect

In this section, we offer recommendations on how different encodings and interactions can be integrated to create a complete interactive tool for preference synthesis in the context of Group Preferential Choice at Level P0b of the taxonomy. **Note that all recommendations are tentative and may be revised as we collect more empirical data.**

We start with some general recommendations that apply to all cases. Then, we present recommendations for each of the three classes of users identified in Section 3.6, starting with the least sophisticated. We recognize that not all cases fall cleanly into one of these three classes, but designers should be able to pick and choose recommendations from each to suit their exact situation.

5.3.1 General Recommendations

Number and Arrangement of Views

It is clear from the task-based assessment that no single encoding is sufficient to support all tasks. For this reason, many of our recommendations employ the *multiform* design choice, in which the same data is faceted into two views that use different encodings [38]. If the intended platform is a desktop of laptop computer, we recommend splitting the window horizontally and populating each half with a single encoding in the horizontal orientation, since this arrangement offers the most precision. It may also be beneficial to allow users to adjust the size of the views in order to devote more screen real-estate to one or the other.

There is a cost associated with multiple views, both in terms of cognitive load and screen real estate [62], so we do not advise supporting more than two views. As the next few sections will demonstrate, it is possible to strongly support every tasks using combinations of just two encodings and a few basic interactions.

Evaluator Weights

For designers of general purpose tools, we recommend including support for evaluator weights, since this feature was desired in the majority of the cases we studied. If EvaluatorWeights are included, then the only viable option is Tabular Bar Chart Design 1 paired with a Stacked Bar Chart, as it is the only combination that supports joint inspection of the three related measures (AT7 Cases A and B) and identifying alternatives with the top weighted scores (AT9 Case B). From this point forward, we will treat these two encodings as a unit due to their complementary nature.

If the designer does not intend to support evaluator weights, then the options are more flexible. In this case, there may be no need to compute total scores in the first place. In fact, it may be more useful to show the *average* scores, since these are on the same scale as individual scores and can be conveyed by adding an 'Average Evaluator' to any plot.

5.3.2 Class C: Casual Users

This class includes users involved in low-stakes decision making in a casual setting. Examples include selecting a gift for a colleague or choosing a hotel to stay at. We now present two viable options that ought to be suitable for this class of users.

Option 1: Tabular Bar Chart Design 1 + Stacked Bar Chart (single view)

This is the simplest option if the designer intends to support evaluator weights, as it only requires one view. It is not effective for tasks that require comparison across evaluators (AT2 Case B, AT5, AT8), and it is only weakly effective for tasks that require comparison across alternatives (AT2 Case A, AT9 Case A). For the latter, the designer might include a text overlay that labels each bar with its value to facilitate more precise comparison. Additionally, users could be given the option to collapse the Stacked Bar Chart to devote more space to the Tabular Bar Chart.

Option 2: Option 1 + Box Plot Design 1 (dual view)

In order to identify potentially strong combinations for a dual-view design, we computed a score for each pair of encodings by taking the sum of the maximum

score on each task. The parallel coordinate designs were excluded from consideration due to the moderate learning curve and the fact that most people are not familiar with them [38] [41]. Including unfamiliar idioms may confuse casual users and make them less likely to stick with the tool.

Of the pairs that were included, the top scoring combinations were:

1. Tabular Bar Chart Design 1 + Box Plot Design 1
2. Tabular Bar Chart Design 1 + Strip Plot Design 2
3. Box Plot Design 1 + Strip Plot Design 2
4. Box Plot Design 1 + Multi-bar Chart Design 2

Of these, only the first uses the same colour mapping in both encodings. This is desirable because it preserves the semantics of colour across views [43]. The remaining pairs would require two distinct, non-overlapping colour pallets. Otherwise, they would risk implying connections between unrelated marks [43]. This limits their scalability to about a dozen entities *in total* (alternatives and evaluators). For this reason, we recommend the first pairing above all others.

When combined with the Stacked Bar Chart (Figure 5.22), this pairing strongly supports all tasks except AT9 Case A, which is weakly supported by the Tabular Bar Chart. This weakness can be mitigated by including sort functionality and text labels for bar values. The one drawback of this design (and multiform designs in general) is that users might get confused shifting attention back and forth between the two views since they use different idioms and the axes do not correspond.

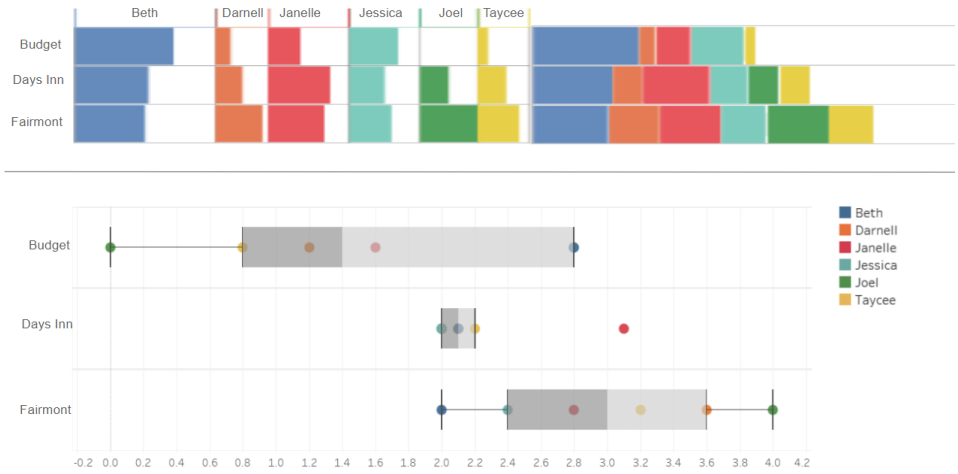


Figure 5.22: Class C Option 2: Dual View with Tabular Bar Chart Design 1 + Stacked Bar Chart (top view) and Box Plot Design 1 (bottom view). **Note that this and other figures in this section are intended for rough illustration only** - we would expect an actual implementation to be more polished and include appropriate interaction controls.

Interactions

At the very least, users should be able to sort rows and plots by TotalScore or AltScore for a particular evaluator, as this is essential for inspecting top values (AT9) and identifying dominance relationships (AT4). Ideally, users should also be able to reorder plots, rows, and columns manually to support particular comparisons of interest. The ability to sort columns by AltScore for a particular alternative is not essential, but would be nice to have. Whenever the columns in the tabular bar chart are reordered, the segments in the corresponding stacked bar chart should be reordered too. We leave it to the designer to choose the mechanism for implementing these features.

Another essential feature is the ability to filter alternatives and evaluators, as this allows users to remove distractions and narrow the scope of analysis. Filtering on values is not essential for small data-sets and may be too advanced for casual users. We recommend a global scope for filter controls in order to preserve consis-

Table 5.3: Recommended Interactive Features for Class C

Essential	Sort alternatives by TotalScore/AltScore for evaluator Filter alternatives and evaluators
Ideal	Manually reorder alternatives and evaluators Tool-tips for dots (AltScore and identify) Label overlay for bars (AltScore)
Nice-to-have	Sort evaluators by AltScore for alternative Linked highlighting (on hover)

tency between views [43].

If linked highlighting is implemented, it should be applied to same-colour marks across *all* views. This will help users stay oriented when shifting attention between views. Highlight-on-hover may be sufficient for casual users.

Finally, we recommend tool-tips for dots that show their identity and value. As previously mentioned, we also recommend text overlays for the bar charts that show the values of the segments and bars. If EvaluatorWeights are defined, the text overlay should specify the AltScore (not the UnweightedAltScore) for consistency between the tabular and stacked bar charts. The text colour should be discernible against the bar colour, and the user should have the ability to turn the overlay on and off.

5.3.3 Class B: Professional Users

This class of users includes professionals involved in medium to high-stakes decision making in a work setting. Examples include faculty hiring and software stack selection. In the cases we studied, this class of decisions was also recurrent, but this may have been an coincidence within our sample.

The space of viable options for this class is somewhat larger than for Class C, since designers may want to provide more or less flexibility depending on the exact work context and expertise of potential users. Here, we describe two possible options that might be worth considering.

Option 1: Dual View with Custom Strip Plot

This design is identical to Option 2 for Class C except that the second view contains a *custom strip plot* that allows users to select:

1. which dimension to map to plots
2. which overlay to apply (box plot, parallel coordinates, or none)

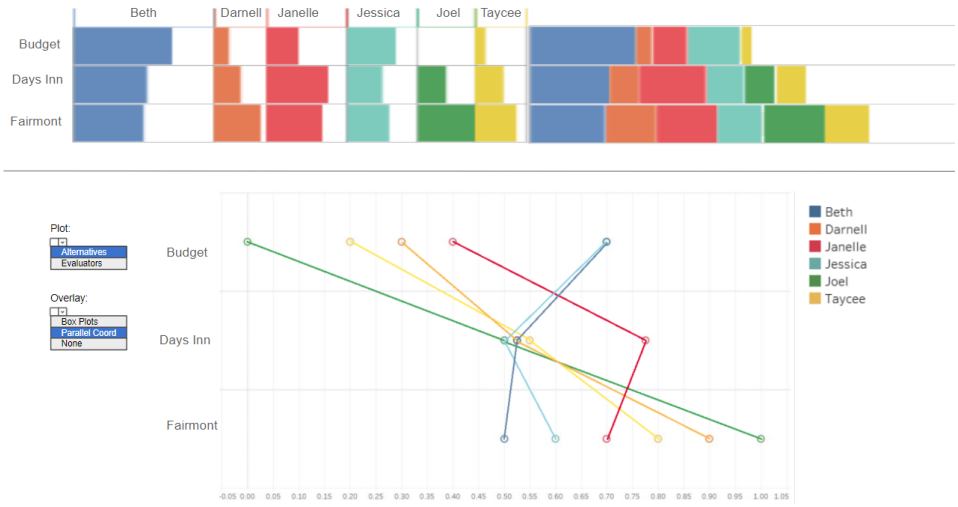


Figure 5.23: Class B Option 1: Dual View with Tabular Bar Chart Design 1 + Stacked Bar Chart (top view) and Custom Strip Plot (bottom view). The user may select a dimension to plot and an overlay. In this example, the user has selected Evaluators with a Parallel Coordinates overlay, producing Parallel Coordinates Design 1.

This design allows users to access the capabilities of all six strip plot-based designs with just a little exploration. All tasks are strongly supported by at least one encoding in this space. The only problem is that it introduces the risk of two different colour mappings in the same window (Figure 5.24). Again, this is not ideal because it reduces scalability by a factor of two, but it might be acceptable if both dimensions are small.

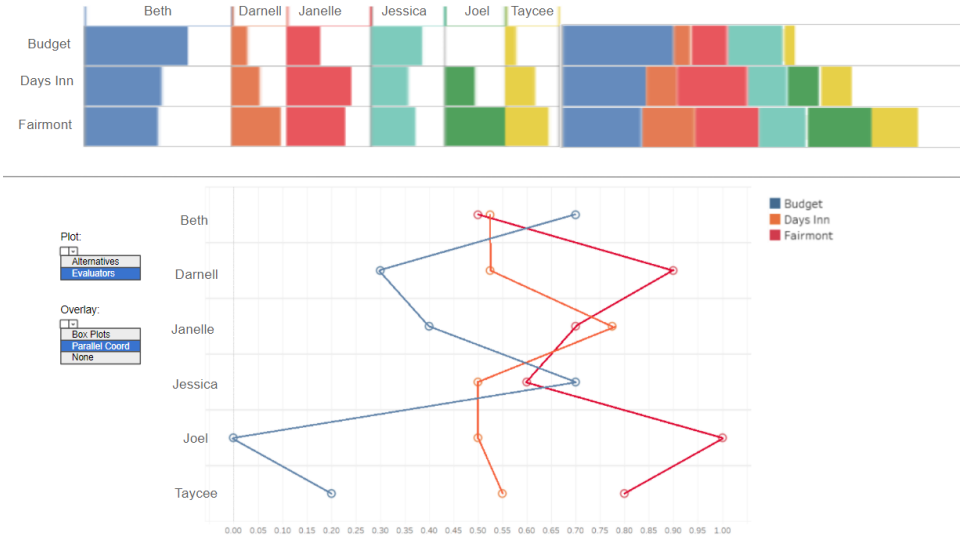


Figure 5.24: The user has constructed Parallel Coordinates Design 2, resulting in two different colour mappings. (In this example, the colour pallets overlap - we recommend using distinct colour pallets.)

A possible solution would be to let users define the colour mapping at a *global* level. That way, they can choose the mapping that is most helpful for their current task while preserving consistency between views. If the Evaluators is selected, then Strip Plot Design 2 dots belonging to the same axis will all have the same colour (and vice versa). To preserve some degree of discriminability in all cases, the designer might also choose to map different *shapes* to the items of the secondary dimension (Figure 5.25). If Alternatives is selected, then the segments of the Stacked Bar Chart will be the same colour. A dividing line can be drawn between them to keep them distinguishable (Figure 5.26).

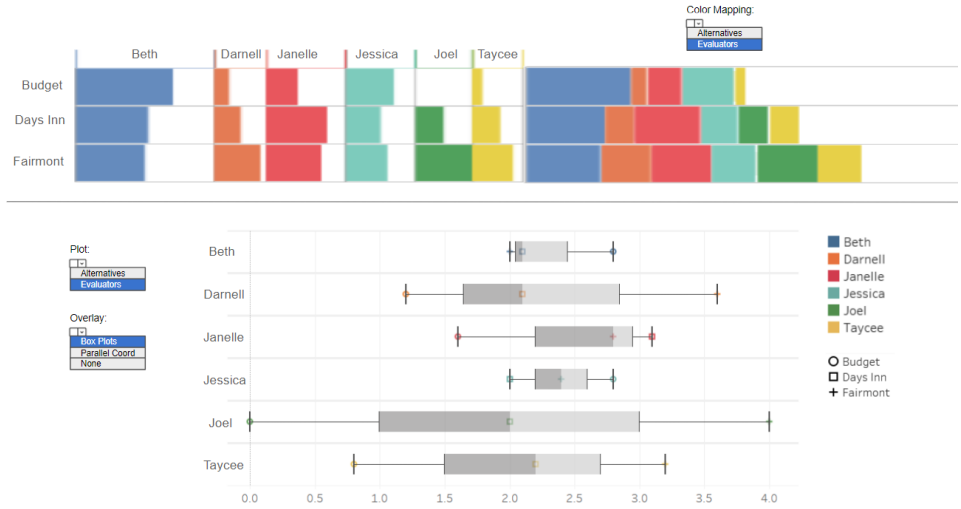


Figure 5.25: The user has selected to map colour to Evaluators. The bottom view contains Box Plot Design 2, where shape is used to preserve some discriminability of hotels along each plot.

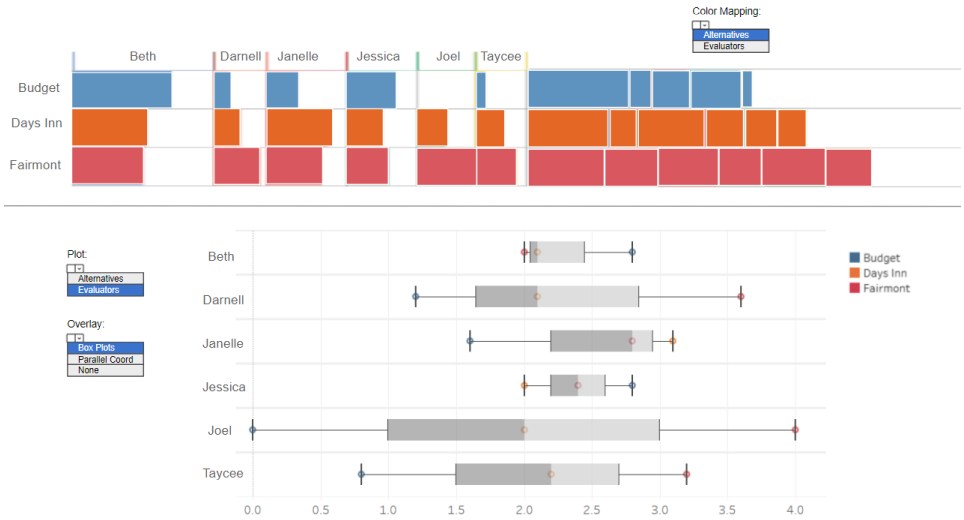


Figure 5.26: The user has selected to map colour to Alternatives, causing the segments of the Stacked Bar Chart to be the same colour. White dividing lines preserve some discriminability.

Option 2: Dual View with Intelligent Plot Selection

Notice that *all* tasks are strongly supported by at least one of the following: Stacked Bar Chart, Tabular Bar Chart Design 1, Box Plot Design 1, and Parallel Coordinates Design 2. In fact, Box Plot Design 1 and Parallel Coordinates Design 2 are the reason that Option 1 achieves complete task coverage.

However, transitioning back and forth between these two encodings in Option 1 requires three toggles - one to change the colour mapping, one to change the dimension mapping in the strip-plot, and one to change the overlay. Furthermore, the user might not realize the complementary power of these two encodings and may end up wasting time with less effective intermediaries.

An alternative approach is to populate the two views with effective, complementary encodings given the selected colour mapping. If colour is mapped to Evaluators, then the secondary view is populated with Box Plot Design 1. Otherwise, it is populated with Parallel Coordinates Design 2. On its own, each pair of designs strongly supports most tasks, but the combination of all four strongly supports for *every* task.

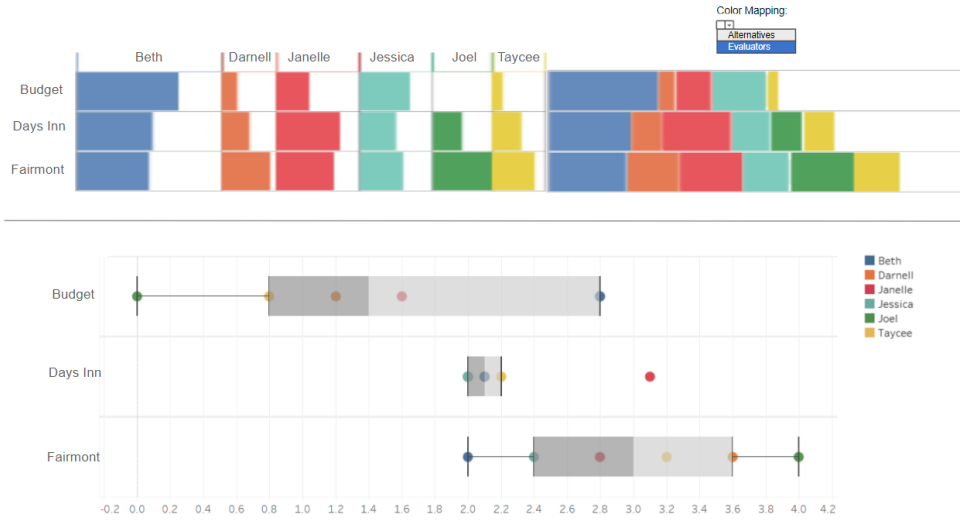


Figure 5.27: Intelligent plot selection when the user has selected to map colour to Evaluators.

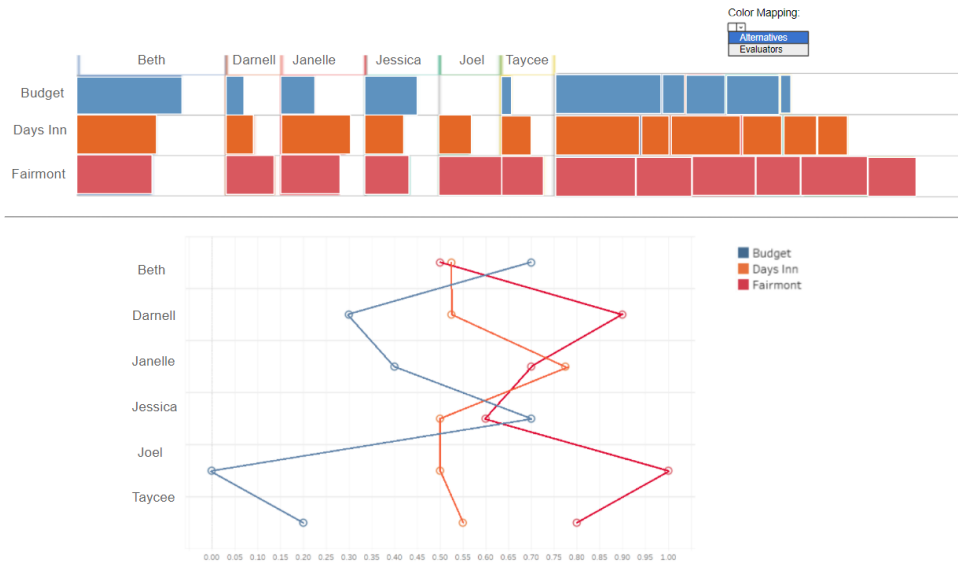


Figure 5.28: Intelligent plot selection when the user has selected to map colour to Alternatives.

One problem with this design is that users might not expect the encoding to change when they toggle the colour mapping. A simple solution would be to change the name of the drop-down or other toggle mechanism to 'Analysis Mode.'

Interactions

The recommended interactions for this group include all of those for Class C with a few additions (Table 5.4). The first addition is the ability to change the colour mapping, which is integral to both suggested designs. The second addition is linked highlighting with multi-select, which would allow users to apply persistent highlighting to items of interest. This could help them keep multiple items in focus while performing complex tasks involving parallel coordinates, such as AT3. The final addition is the ability to filter alternatives based on TotalScore or AltScore, which would enable user to set satisficing thresholds.

Table 5.4: Recommended Interactive Features for Class B. Items in bold are additions to the list for Class C (Table 5.3).

Essential	Sort alternatives by TotalScore/AltScore for evaluator Filter alternatives and evaluators **Swap colour mapping**
Ideal	Manually reorder alternatives and evaluators Tool-tips for dots (AltScore and identify) Label overlay for bars (AltScore) **Linked highlighting + multi-select**
Nice-to-have	Sort evaluators by AltScore for alternative Linked highlighting (on hover) **Filter alternatives on AltScore/TotalScore values**

5.3.4 Class A: Specialized Users

This class of users includes professionals and governing officials involved in very high-stakes decision making that impacts society at large. These users are often aided by consultants with expertise in formal decision processes - these experts are included in this group as well.

This class is the most likely to require sophisticated analysis software. However, this need typically comes hand-in-hand with more sophisticated preference models - that is, expressed at a higher level of the Preference Model Taxonomy. As such, the recommendations for Class A do not differ much from those for Class B at this level. The recommendations will diverge as we extend the design space to higher levels of the taxonomy.

If users in this class *do* express their preferences at Level P0b, then a likely task would be to assess the sensitivity of the final result to aggregation method and evaluator weights, as in the Mariner Jupiter-Saturn project [22]. Tasks related to sensitivity analysis are currently beyond the scope of this analysis - we leave this topic to future work.

Chapter 6

Conclusion

Group Preferential Choice can be challenging due to its multi-variate and interpersonal nature. There is considerable evidence that structured decision processes [6] [51] and individual preference modeling in particular [4] can promote more fruitful analysis and discussion, ultimately leading to greater satisfaction with the outcome.

The potential benefits of individual preference modelling are constrained by how effectively the data is presented to decision makers. Information Visualization solutions have great potential, but only a handful have been attempted [40] [4] [36]. Furthermore, no work thus far has attempted to characterize sources of variation among Group Preferential Choice scenarios.

This work makes progress on these fronts in three major steps, which are summarized in Section 6.1. Section 6.2 critically reflects on the limitations and vision of the work, and Section 6.3 presents possible directions for future work.

6.1 Summary of Contributions

This section summarizes the major contributions of this work and anticipates how they might be used by other academics or designers of Group Preferential Choice support tools. All contributions are works in progress - they may be extended or refined as new information is gathered.

6.1.1 Characterization of Group Preferential Choice

The goal of Chapter 3 was to characterize sources of variation in the data, goals, and decision making contexts of Group Preferential Choice. This was achieved by performing an in-depth analysis of a diverse set of Group Preferential Choice scenarios. The results of this analysis can help designers define the scope of their work by orienting them to the space of possibilities.

Section 3.4 presented a data model for Group Preferential Choice, including a taxonomy of commonly-used preference models. The model was extended to account for new sources of variation that were discovered during the analysis of scenarios. This model is the interface between specific decision problems and the rest of our work - if a decision problem can be described in these terms, then readers can easily identify which tasks and design recommendations are applicable to their situation.

Section 3.5 presented a summary of goals for preference synthesis in the context of Group Preferential Choice. It is worth reiterating that this is not intended to be an exhaustive list. Depending on the exact situation, designers may wish to support additional goals or only a subset of these goals. As noted in Section 3.7, three goals were found in at least three scenarios - these would be good candidates for inclusion in any general-purpose support tool.

Finally, Section 3.6 summarized the variation in contextual features across scenarios. We found that the scenarios form roughly three clusters at different levels of sophistication. This result can help designers define the target audience for their tools by giving them a sense of likely classes of users.

6.1.2 Data and Task Abstraction for Preference Synthesis

The goal of Chapter 4 was to describe the data and goals identified in Chapter 3 in abstract terms that are suitable for visualization design and analysis. This is the bridge between the descriptions of Chapter 3 and the design recommendations in Chapter 5 and beyond.

Section 4.1 described the data in terms of multi-dimensional tables. This abstraction is useful because the pros and cons of different encodings for tabular data are well known [38], and there has been considerable work on representing large

multi-dimensional data sets in particular [16] [54].

Section 4.2.1 presented a list of tasks to support each goal. Some of these tasks were derived from the scenarios we studied, and others were added based on intuition. Again, this list is not intended to be exhaustive and may be iteratively improved as more data is collected. Finally, Section 4.2.2 described each of these tasks in terms of a smaller set of low level tasks from Brehmer and Munzner's task taxonomy [7]. This is useful because it allows potential designs to be evaluated more efficiently.

In addition to providing abstractions for the current set of goals, this analysis also serves as a template for abstracting new goals that are identified in the future.

6.1.3 Design Space for Preference Synthesis

Chapter 5 presented our final contribution, which is a prescriptive design space of visualizations to support preference synthesis in the context of Group Preferential Choice. At this time, the design space is limited to small-scale decision problems where preferences are expressed at Level P0b of the taxonomy - that is, each evaluator simply scores each alternative. Despite the limited scope of the current design space, the analysis underlying its construction lays the foundation upon which a complete design space may be built. As it stands, we believe that designers of Group Preferential Choice support tools will find plenty of useful suggestions regardless of the complexity of their data.

Section 5.1 introduced the major competitive idioms for presenting small-scale tabular data and analytically evaluated their suitability for each auxiliary task. Section 5.2 described how interactivity could be introduced to enhance the efficacy of the static encodings. Finally, Section 5.3 showed how a complete support system could be constructed from the aforementioned elements, with specific recommendations for each of the three contextual classes identified in Chapter 3.

Although the design space is tailored to Group Preferential Choice, many of our recommendations could also be applied to the design of visualizations for other preferential data-sets, including but not limited to rankings, surveys, and evaluations. Furthermore, the task-based assessment of static encodings (Section 5.1.2) applies to tabular data in general.

6.2 Critical Reflections

6.2.1 Goals Elicitation

The procedure we used to elicit scenario goals in Chapter 3 was structured and systematic, but it was not without limitations. According to human-centred design experts, the most effective way to attain a complete and accurate understanding of a situation is using a combination of *in situ* observation and interviews [57]. Due to time constraints, we were only able to do this for two scenarios - Faculty Hiring (department meeting portion) and XpertsCatch.

The Best Paper, Gift, and Faculty Hiring (committee meeting portion) scenarios were assessed through interviews conducted after the fact. This is not as effective, since interviewees may not be able to accurately identify, recall, or communicate key aspects of the situation [27]. The remaining three cases were assessed by reviewing second-hand reports, which is also error-prone due to the degree of separation between the original situation and the analyst. Another potential source of bias is the analyst's interpretation of the data - in our case, this involved compiling a list of scenario goals from the interview notes.

On the one hand, problem characterization is seldom done at all in Information Visualization [37], so any attempt to do so may constitute satisfactory progress. On the other hand, Group Preferential Choice is highly complex and human-centered, and so the risk of some elements getting lost in translation is high. Our hope is that the scenarios we examined are sufficiently rich that they converge upon key points despite the methodological limitations.

There are several immediate actions we could take to validate our model, which are discussed in Section 6.3. However, it is unlikely that we will fully understand the complexity of this problem space until support tools are deployed, which brings us to our next topic.

6.2.2 A More Agile Approach?

Thus far, our approach has been to perform a series of analyses on an entire class of problems in sequence. The strength of this approach is that we now have a solid framework for relating specific scenarios to the overarching problem space. This

is useful because it allows us to iteratively refine our understanding as new data is encountered.

A major challenge with this approach is that the sheer amount of variation within each step makes it easy to lose site of tangible realities. Errors in the early stages of analysis did not always become apparent until later stages, and recovery was sometimes costly due to the layers of complexity and abstraction that needed to be synchronized.

Now that we have a preliminary model in place, it may be worthwhile to switch to an alternative but complementary approach. Specifically, we could embark on a series of design studies following the methodology proposed in Sedlmair et al. [52]. In a design study, the needs of a particular group of users are identified, a visualization solution is implemented and evaluated, and insights are recorded. After several iterations of this process, we could compile our insights and update our data model, task abstractions, and design space recommendations accordingly. This would allow us to achieve breadth while maintaining agility and practical grounding.

6.3 Future Work

6.3.1 Validating the Data and Task Model

There are a number of actions we could take to improve the completeness and accuracy of our data and task models.

The first and easiest would be to have another researcher reproduce the descriptions of each scenario based on our interview notes and second hand sources. Then, the two descriptions could be compared for discrepancies. Another easy option would be to go back and validate the written description of each scenario with interviewees and authors of second hand sources (where possible).

An even better approach would be to collect new data by observing Group Preferential Choice scenarios as they occur. This might be more productive, since we could apply the lessons learned to the new situation. This could be done in the context of complete design studies, as suggested in Section 6.2.

6.3.2 Validating the Task-based Assessment

It is important to emphasize that the task-based assessment in Section 5.1.2 contains a fair amount of speculation. We referred to reliable sources wherever possible, but there were a surprising number of cases where we could not definitively say which encoding was better based on available literature. In particular, there is a scarcity of literature devoted to comparing strip plots and bar charts, and that which does exist invokes general principles such as data-ink maximization rather than empirical data on task efficacy [15] [46].

This could be a rich territory for future research in the field of Vision Science. Questions that one might ask include:

1. Do the connecting lines on parallel coordinates plots affect perception of distance between points along each axis?
2. Under what circumstances do bar charts or strip plots support more accurate comparisons?
3. Do bar charts or parallel coordinates (single line) give a more accurate impression of variance?

It may well be the case that answers to these questions exist but are hard to find due to a scarcity of relevant surveys. In this case, conducting a review of relevant Vision Science literature could be a valuable avenue for future research. Otherwise, we hope that future research in Vision Science will shed light on these questions, as the answers would be valuable to anyone interested in the pros and cons of different ways of presenting tabular data.

6.3.3 Extending the Design Space to Other Levels of the Taxonomy

We are currently working on extending the design space to the remaining levels of the taxonomy while retaining the same constraints, that is:

1. There are no more than a dozen alternatives or evaluators.
2. The Evaluator and Criteria hierarchies are flat.
3. Preferences are expressed on a scale with no negative values.

Recall that the set of *applicable* designs depends on which dimensions and measures are defined (Figure 6.1). With the exception of EvaluatorWeights, this is determined wholly by the level of the Preference Model Taxonomy.

Dimension / Measure	Keys	Datatype	P0b	P1b	P1b+w	P2b	P2b+w
Alternatives (*A*)	---	Categorical	✓	✓	✓	✓	✓
Evaluators (*E*)	---	Categorical	✓	✓	✓	✓	✓
Criteria (*C*)	---	Categorical		✓	✓	✓	✓
Outcomes (*O*)	---	{Categorical, Numeric}				✓	✓
TotalScore	*A*	Cat → Num	✓	✓	✓	✓	✓
EvaluatorWeight	*E*	Cat → Num	?	?	?	?	?
AltScore	*A* x *E*	Cat x Cat → Num	✓	✓	✓	✓	✓
CritWeight	*E* x *C*	Cat x Cat → Num			✓		✓
AltCritScore	*A* x *E* x *C*	Cat x Cat x Cat → Num		✓	✓	✓	✓
UnweightedAltCritScore	*A* x *E* x *C*	Cat x Cat x Cat → Num			✓		✓
Outcome	*A* x *C*	Cat x Cat → {Cat, Num}				✓	✓
OutScore	*E* x *C* x *O*	Cat x Cat x Cat → Num OR Cat x Cat x Num → Num				✓	✓
UnweightedOutScore	*E* x *C* x *O*	Cat x Cat x {Cat, Num} → Num					✓

✓ = present at level
? = optional at level

Figure 6.1: Overview of Dimensions and Measures defined at each level of the Preference Model Taxonomy.

Since each level of the taxonomy implicitly encodes all the levels above it, the design space at each new level is a superset of the design space of the levels above it. Therefore, we will also consider ways to support transitions between different levels of the taxonomy - for instance, factoring out the Criteria dimension to move from P1b to P0b.

6.3.4 Relating Existing Encodings to the Design Space

We have already performed an extensive analysis of each of the tools introduced in Chapter 2. This analysis currently exists as a detailed slide-deck, which is shown in Appendix A. It describes the capabilities of these tools in terms of our data model and identifies the static idioms, mappings, interactive techniques, and other design choices they employ. The next step is to relate this explicitly to the the design space

once it is complete.

6.3.5 Extending the Design Space to Hierarchical and Large Dimensions

In the future, we hope to extend the design space to include hierarchical dimensions and dimensions with more than a dozen items. This promises to be an exciting area of research, as there are a number of interesting possibilities, including:

- More compact encodings for tabular data, such as heatmaps
- Data reduction strategies, such as:
  - Hierarchical aggregation
  - Histograms
  - Focus + context (juxtaposed views or focal lens)
- Hierarchy representation and traversal strategies, such as:
  - Node-link graphs
  - Rectilinear trees
  - Semantic zooming

We have already begun reviewing relevant literature in this area. Liu et al. [34] provides an overview of the pros and cons of different data reduction strategies. The main takeaway is that binned aggregation is the ideal data reduction strategy, since it captures both global trends and outliers. Other data reduction strategies include filtering, sampling, and model-fitting. Filtering and sampling may hide global trends, whereas model-fitting may hide interesting outliers.

Stolte et al. (2002a) [53] presents Polaris, a novel interface for exploring multi-dimensional table, and Stolte et al. (2002b) [54] extends Polaris to support hierarchical dimensions. It proposes basic mechanisms to allow users to drill-down and roll-up hierarchies via drop-down selection. Polaris became the basis for the popular visual analytics tool suite Tableau. When extending the design space to include hierarchical dimensions, we will look to Tableau for guidance due to its long history and widespread use.

Bibliography

[1] *D-Sight*. 4 Rue des Pères Blancs 1040 - Brussels, Belgium, 2015. URL http://www.d-sight.com/. → pages 2

[2] K. Abraham, F. Flager, J. Macedo, D. Gerber, and M. Lepech. Multi-attribute decision-making and data visualization for multi-disciplinary group building project decisions. In *Working Paper Series, Proceedings of the Engineering Project Organization Conference*, Winter Park, CO, 2014. → pages 2

[3] K. J. Arrow. *Social Choice and Individual Values*. Yale University Press, 1963. → pages 12, 37

[4] S. Bajracharya, G. Carenini, B. Chamberlain, K. Chen, D. Klein, D. Poole, H. Taheri, and G. Öberg. Interactive visualization for group decision-analysis. *Submitted to: Decision Support Systems*, 2017. → pages 2, 3, 4, 12, 13, 28, 137

[5] J. Bautista and G. Carenini. An integrated task-based framework for the design and evaluation of visualizations to support preferential choice. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 217–224. ACM, 2006. → pages 4

[6] U. Bose, A. M. Davey, and D. L. Olson. Multi-attribute utility methods in group decision making: Past applications and potential for inclusion in GDSS. *Omega*, 25(6):691–706, 1997. → pages 2, 11, 137

[7] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013. → pages xii, 6, 78, 81, 88, 139

[8] M. Brehmer, J. Ng, K. Tate, and T. Munzner. Matches, mismatches, and methods: Multiple-view workflows for energy portfolio analysis. *IEEE*

Transactions on Visualization and Computer Graphics, 22(1):449–458, 2016. → pages 29, 30

[9] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner. Timelines revisited: A design space and considerations for expressive storytelling. *IEEE Transactions on Visualization and Computer Graphics*, 23(9): 2151–2164, 2017. → pages 6, 29, 30

[10] D. Brodbeck and L. Girardin. Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree. In *IEEE Symposium on Information Visualization*, pages 197–201. IEEE, 2003. → pages 12, 24, 25, 26

[11] V. A. Brown, J. A. Harris, and J. Y. Russell. *Tackling Wicked Problems: Through the Transdisciplinary Imagination*. Earthscan, 2010. → pages 1

[12] G. Carenini and J. Loyd. ValueCharts: Analyzing linear models expressing preferences and evaluations. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 150–157. ACM, 2004. → pages 4, 14

[13] L. N. Carroll, A. P. Au, L. T. Detwiler, T.-c. Fu, I. S. Painter, and N. F. Abernethy. Visualization and analytics tools for infectious disease epidemiology: A systematic review. *Journal of Biomedical Informatics*, 51: 287–298, 2014. → pages 29, 30

[14] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017. → pages 29, 30

[15] J. M. Chambers, W. S. Cleveland, B. Kleiner, P. A. Tukey, et al. *Graphical Methods for Data Analysis*, volume 5. Wadsworth Belmont, CA, 1983. → pages 142

[16] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65–74, 1997. → pages 79, 139

[17] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2970–2984, 2015. → pages 29, 30

[18] C. A. B. E. Costa and J.-C. Vansnick. The MACBETH approach: Basic ideas, software, and an application. In *Advances in Decision Analysis*, pages 131–157. Springer, 1999. → pages 2

[19] T. N. Dang, L. Wilkinson, and A. Anand. Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1044–1052, 2010. → pages 104, 114

[20] E. Dimara, P. Valdivia, and C. Kinkeldey. DCPAIRS: A pairs plot based decision support system. In *EuroVis-19th EG/VGTC Conference on Visualization*, 2017. → pages 12, 25

[21] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited. Technical report, IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, 2001. → pages 37

[22] J. S. Dyer and R. F. Miles Jr. An actual application of collective choice theory to the selection of trajectories for the Mariner Jupiter/Saturn 1977 project. *Operations Research*, 24(2):220–244, 1976. → pages 52, 53, 55, 56, 136

[23] W. Edwards and F. H. Barron. SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement. *Organizational Behavior and Human Decision Processes*, 60(3):306–325, 1994. → pages 49

[24] S. Few and P. Edge. Solutions to the problem of over-plotting in graphs. *Visual Business Intelligence Newsletter*, 2008. → pages 104, 114

[25] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2277–2286, 2013. → pages 12, 20, 21, 28, 98

[26] J. S. Guest, S. J. Skerlos, J. L. Barnard, M. B. Beck, G. T. Daigger, H. Hilger, S. J. Jackson, K. Karvazy, L. Kelly, L. Macpherson, et al. A new planning and design paradigm to achieve sustainable resource recovery from wastewater. *Environmental Science & Technology*, 43(16):6126–6130, 2009. → pages 1

[27] J. T. Hackos and J. Redish. *User and task analysis for interface design*. Wiley, New York, 1998. → pages 140

[28] P. Hansen and F. Ombler. A new method for scoring additive multi-attribute value models using pairwise rankings of alternatives. *Journal of Multi-Criteria Decision Analysis*, 15(3-4):87–107, 2008. → pages 2

[29] I. B. Huang, J. Keisler, and I. Linkov. Multi-criteria decision analysis in environmental sciences: Ten years of applications and trends. *Science of the Total Environment*, 409(19):3578–3594, 2011. → pages 8

[30] C. L. Hwang and K. Yoon. *Multiple Attribute Decision Making: Methods and Applications State-of-the-Art Survey*, volume 186. Springer Science & Business Media, 2012. → pages 2, 33, 49

[31] R. L. Keeney. Foundations for group decision analysis. *Decision Analysis*, 10(2):103–120, 2013. → pages 12

[32] R. L. Keeney and H. Raiffa. *Decision with Multiple Objectives*. Wiley, New York, 1976. → pages 8, 9, 57

[33] K. Kucher, C. Paradis, and A. Kerren. The state of the art in sentiment visualization. In *Computer Graphics Forum*. Wiley Online Library, 2017. → pages 29, 30

[34] Z. Liu, B. Jiang, and J. Heer. imMens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430. Wiley Online Library, 2013. → pages 144

[35] J. Lu and D. Ruan. *Multi-Objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Techniques*, volume 6. Imperial College Press, 2007. → pages 33

[36] N. Mahyar, W. Liu, S. Xiao, J. Browne, M. Yang, and S. P. Dow. ConsesnsUs: Visualizing points of disagreement for multi-criteria collaborative decision making. In *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 17–20. ACM, 2017. → pages 4, 5, 12, 17, 28, 137

[37] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 2009. → pages 81, 97, 140

[38] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014. → pages 2, 14, 25, 79, 98, 99, 102, 104, 106, 107, 124, 126, 128, 138

[39] J. Mustajoki and R. P. Hämäläinen. Web-HIPRE: Global decision support by value tree and AHP analysis. *INFOR: Information Systems and Operational Research*, 38(3):208–220, 2000. → pages 2, 12, 18

[40] J. Mustajoki, R. P. Hämäläinen, and K. Sinkko. Interactive computer support in decision conferencing: Two cases on off-site nuclear emergency management. *Decision Support Systems*, 42(4):2247–2260, 2007. → pages 2, 3, 19, 20, 56, 59, 137

[41] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. WeightLifter: Visual weight space exploration for multi-criteria decision making. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):611–620, 2017. → pages 12, 22, 23, 128

[42] P. G. Pham and M. L. Huang. Qstack: Multi-tag visual rankings. *Journal of Software*, 11(7):695–703, 2016. → pages 12, 27, 28

[43] Z. Qu and J. Hullman. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):468–477, 2018. → pages 128, 130

[44] M. D. Resnik. *Choices: An Introduction to Decision Theory*. University of Minnesota Press, 1987. → pages 7

[45] L. Robbins. *An Essay on the Nature and Significance of Economic Science*. MacMillan, 1932. → pages 8

[46] N. B. Robbins. Dot plots: A useful alternative to bar charts. *Business Intelligence Network Newsletter*, 2006. → pages 18, 102, 114, 142

[47] N. B. Robbins, R. M. Heiberger, et al. Plotting Likert and other rating scales. In *Proceedings of the 2011 Joint Statistical Meeting*, pages 1058–1066, 2011. → pages 97

[48] M. N. Rothbard. *Economic Controversies*. Ludwig von Mises Institute, 2011. → pages 8

[49] B. Roy. Classement et choix en présence de points de vue multiples. *Revue Française D'informatique et de Recherche Opérationnelle*, 2(8):57–75, 1968. → pages 10

[50] T. L. Saaty. What is the analytic hierarchy process? In *Mathematical Models for Decision Support*, pages 109–121. Springer, 1988. → pages 10

[51] A. Salo and R. P. Hämäläinen. Multicriteria decision analysis in group decision processes. In *Handbook of Group Decision and Negotiation*, pages 269–283. Springer, 2010. → pages 2, 11, 73, 137

[52] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012. → pages 29, 141

[53] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002. → pages 79, 144

[54] C. Stolte, D. Tang, and P. Hanrahan. Query, analysis, and visualization of hierarchically structured data using Polaris. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 112–122. ACM, 2002. → pages 79, 139, 144

[55] M. Streit and N. Gehlenborg. Points of view: Bar charts and box plots. *Nature Methods*, 11(2):117–117, 2014. → pages 99, 102

[56] J. Talbot, V. Setlur, and A. Anand. Four experiments on the perception of bar charts. *IEEE Transactions on Visualization and Computer Graphics*, 20 (12):2152–2160, 2014. → pages 114

[57] M. Tory and T. Moller. Human factors in visualization research. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):72–84, 2004. → pages 140

[58] A. Tsoukias, M. Ozturk, and P. Vincke. Preference modelling. *Multiple Criteria Decision Analysis: State of the Art Surveys, International Series in Operations Research and Management Science*, 78:27–71, 2005. → pages 33

[59] M. Velasquez and P. T. Hester. An analysis of multi-criteria decision making methods. *International Journal of Operations Research*, 10(2):56–66, 2013. → pages 10, 11

[60] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior, 2nd rev*. Princeton University Press, 1947. → pages 7, 53

[61] D. Von Winterfeldt and W. Edwards. *Decision Analysis and Behavioral Research*. Cambridge University Press, 1986. → pages 57

[62] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 110–119. ACM, 2000. → pages 126

Appendix A

Analysis of Existing Encodings

Analysis of Existing Encodings

1

Analysis of Existing Encodings

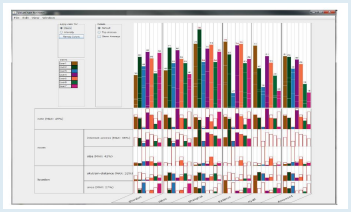
- **Inclusion criteria:** Explicitly visualizes the performance of *alternatives* with respect to multiple criteria and/or multiple people's preferences
- **Excludes:**
  - Visualizations of users (without showing the alternatives)
  - MODM visualization (infinite alternatives, i.e. design space exploration)
- **Class 1:** Interactive Tools (have some interaction)
  - Group and individual MCDA support tools
  - Tools for visualizing related datasets (evaluations, surveys, opinions)
- **Class 2:** Standalone Encodings (no interactions)
  - Encodings used in the scenarios from Ch. 1

2

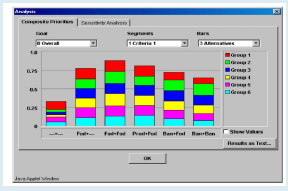
Class 1: Interactive Tools

3

Preview: Group MCDA



Group ValueCharts [1]

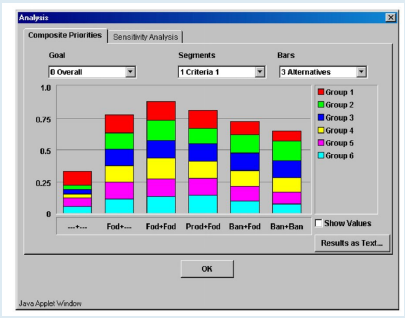


Web-HIPRE (used in Nuclear case) [3,4]

ConsensUs [2]

4

Web-HIPRE [3,4]



5

What data is supported?

Measures:		
	Taxonomy level?	**P2b+w (and above) ***
	Evaluator weights?	✓
Dimensions:		
	Criteria hierarchies?	✓
	Evaluator hierarchies?	✓

6

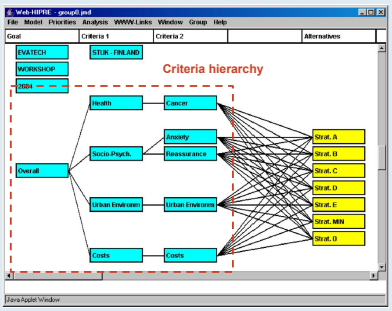
Overall Organization

- Main window shows criteria hierarchy and alternatives
- From there, user can view other windows:
  - Priorities window
  - Analysis window
  - Ratings window
- Group MCDA is is achieved by treating evaluators as criteria in an another decision problem

7

Main Window (Individual)

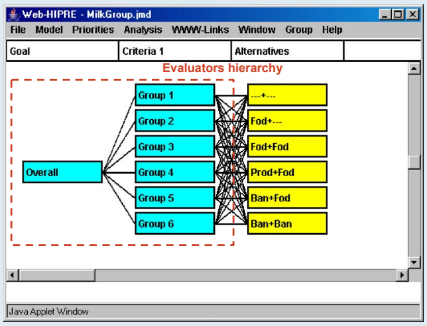
- In individual MCDA context, the main window shows alternatives and criteria hierarchy
- Can open additional windows from here



8

Main Window (Group)

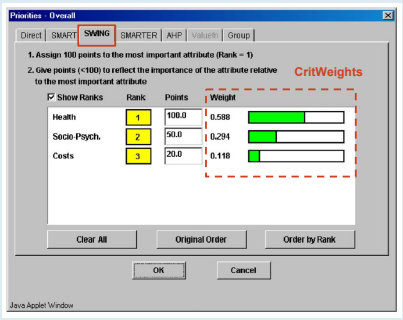
- In group MCDA context, the main window shows alternatives and criteria hierarchy
- Can open additional windows from here



9

Priorities Window

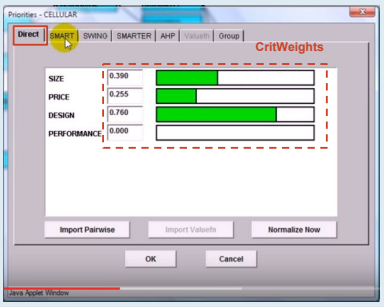
- Priorities window supports preference elicitation
- Can be opened by clicking on a criterion in the main window
- Supports five elicitation methods, each in a different tab



10

Priorities Window

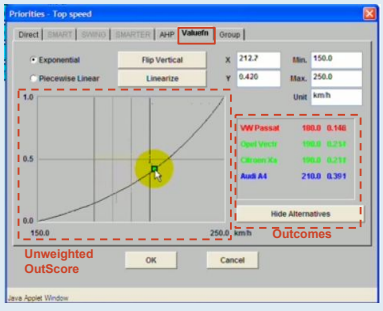
- Priorities window supports preference elicitation
- Can be opened by clicking on a criterion in the main window
- Supports five elicitation methods, each in a different tab



11

Priorities Window

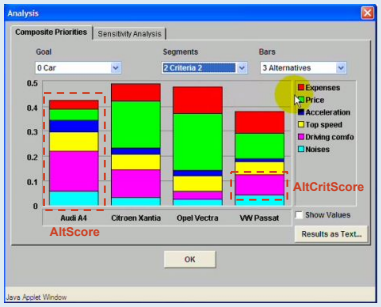
- Priorities window supports preference elicitation
- Can be opened by clicking on a criterion in the main window
- Supports five elicitation methods, each in a different tab
- Can also be used to define or inspect the score function and alternative outcomes



12

Analysis Window (Individual)

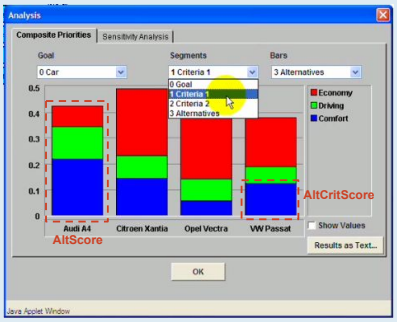
- In individual MCDA context, the analysis window supports *evaluation phase* tasks
- Can map different things to bars and segments:
  - Alternatives
  - Criteria (one level at a time)



13

Analysis Window (Individual)

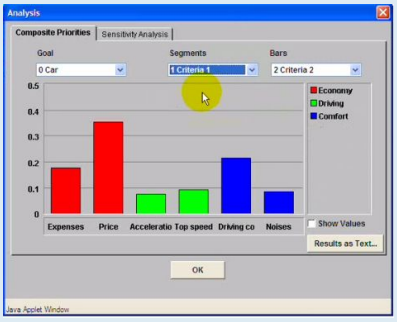
- In individual MCDA context, the analysis window supports *evaluation phase* tasks
- Can map different things to bars and segments:
  - Alternatives
  - Criteria (one level at a time)
- Interaction: roll-up/drill-down criteria hierarchy



14

Analysis Window (Individual)

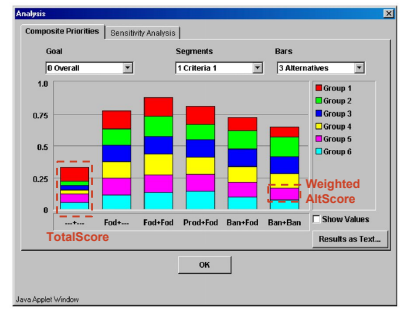
- In individual MCDA context, the analysis window supports *evaluation phase* tasks
- Can map different things to bars and segments:
  - Alternatives
  - Criteria (one level at a time)
- Interaction: roll-up/drill-down criteria hierarchy
- Can also do stuff like...



15

Analysis Window (Group)

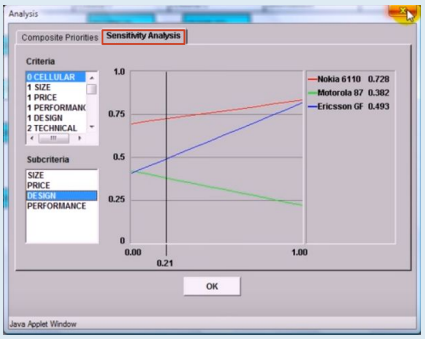
- In group MCDA context, the analysis window supports *synthesis phase* tasks
- Can map different things to bars and segments:
  - Alternatives
  - Evaluators (one level at a time)



16

Analysis Window

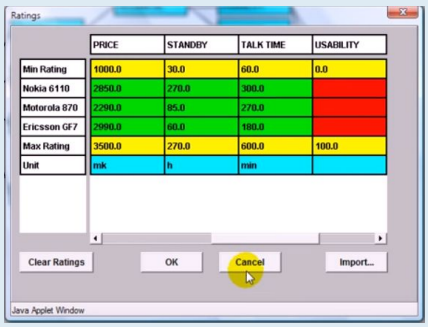
- Another tab allows users to perform sensitivity analysis (i.e. inspect trade-offs):



17

Ratings Window

- Ratings window contains a consequence table
- Colors:
  - Yellow: min/max
  - Blue: unit
  - Green: value present
  - Red: value missing



18

How: Encode (Measures)

Measure Class	Measure	Window	Idiom	Encoding
Scores	TotalScore	Analysis (group)	Aligned stacked bar chart Dimension mappings: customizable	Length of bar
	AltScore			Length of segment
	UnweightedAltScore	Analysis (individual)		Length of bar
	AltCritScore			Length of segment
Weights	CritWeights	Priorities (any weights tab)	Horizontal bar charts + text field	Length of bar, text
	EvaluatorWeights *		Horizontal bar charts + text field	Length of bar, text
Score Functions	UnweightedOutScore	Priorities (ValueFn tab)	Interactive line graph	Point on graph, text coordinates
Outcomes	Outcome		Table (meaning of color unclear)	Color-coded text
		Ratings	Table	Text in color-coded cell

19

How: Encode (Dimensions)

Dimension	Window	Idiom	Encoding
Criteria	Main (individual)	Node-link graph (Nodes color-coded by dimension)	Blue node
Alternatives			Yellow node
Evaluators	Main (group)	Node-link graph (Nodes color-coded by dimension)	Blue node
Alternatives			Yellow node

20

How: Manipulate (Data Changing Interactions)

- **Change weights:**
  - Change values of text fields in Priorities windows
- **Change score function:**
  - Adjust coordinates of a single point on the score function graph in ValueFn tab of the Priorities window (click-and-drag)

21

How: Manipulate (View Changing Interactions)

- **Change mapping:**
  - Swap the selections in the Segments and Bars drop-downs in the Analysis window
- **Change aggregation level:** **(How: Reduce -> Aggregate)**
  - Change selected dimension hierarchy level in one of the three drop-downs in Analysis window
- **Change data shown:** **(How: Reduce -> Filter)**
  - Change selected dimension in one of the three drop-downs in Analysis window

22

Group ValueCharts [1]

23

What data is supported?

Measures:		
	Taxonomy level?	**P2b+w (and above)**
	Evaluator weights?	✗
Dimensions:		
	Criteria hierarchies?	✓
	Evaluator hierarchies?	✗

24

Overall Organization

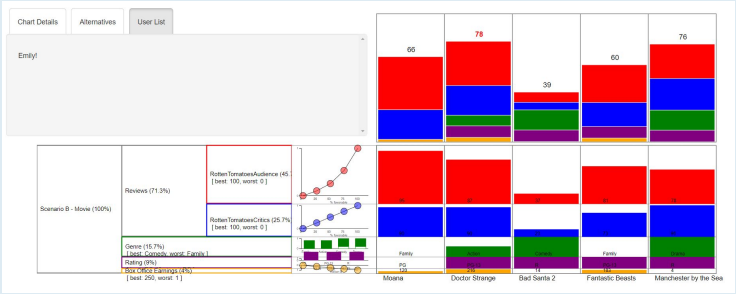
- One main window with two different views:
  - Individual view
  - Group view
- Each view has the following components:
  - Details component (with 3 tabs: *Chart Details*, *Alternatives*, and *User List*)
  - Criteria component
  - Scores component
  - Score functions component

How: Facet -> Partition
How: Facet -> Linked views
How: Facet -> Superimpose

- Another window may be opened to view score functions up close

25

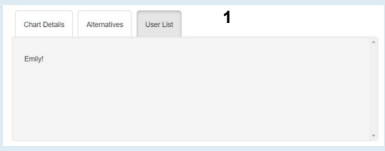
Individual View

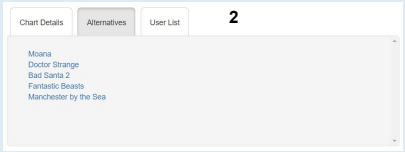


26

Individual View - Details Component

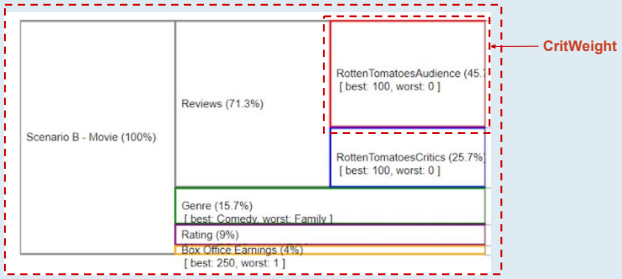
1. ***User List*** **tab**
2. ***Alternatives*** **tab**
3. ***Alternatives*** **tab after clicking an alternative**





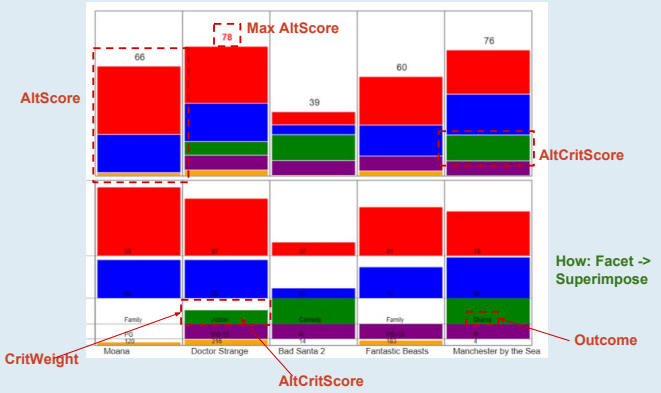
27

Individual View - Criteria Component



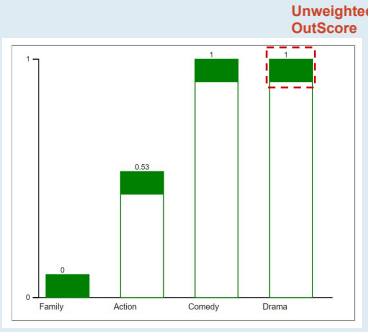
28

Individual View - Scores Component

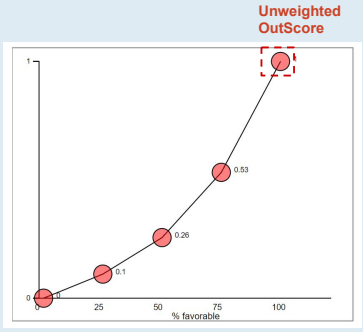


29

Individual View - Score Functions Component



Categorical Score Function



Continuous Score Function

30

How: Encode

Measure Class	Measure	View Component	Idiom	Encoding
Scores	AltScore	Scores	1. Aligned stacked bar chart 2. Tabular bar chart Dimension mappings: Alternatives -> columns Criteria -> rows, colour	Height of bart (1) + text
	AltCritScore			Height of bar (2); height of segment (1)
	Max(AltScore)			Red-coloured text
Weights	CritWeights			Row height
		Criteria	Rectilinear node-link graph	Row height
Outcomes	Outcome	Scores	""	Text label in (2)
		Details	Tabular list	Text
Score Functions	UnweightedOutScore	Score Functions	Interactive bar chart/ line graph	Height of bar/Y-coordinate of dot

31

How: Manipulate (Data Changing Interactions)

- **Change weights:**
  - Adjust height of box in Criteria Component
    - Click-and-drag
    - "Pump" (double-click to inflate/deflate)
- **Change score function:**
  - Adjust y-coordinate of a point/bar in the score functions graph (click-and-drag)

32

How: Manipulate (View Changing Interactions)

- **Change arrangement:**
  - Change orientation (vertical or horizontal)
  - Reorder Objectives (drag-and-drop)
  - Reorder Alternatives (drag-and-drop, alphabetical, or by Objective score)
- **Change data shown:** **(How: Reduce -> Filter)**
  - Choose Alternative to see Outcomes for (click on name in Alternatives tab)
    - This is a special case of filter where exactly one item may be chosen
- **Change elements shown:**
  - Toggle view options (average lines, score functions, outcomes, score labels, utility scale)
- **Change viewpoint:** **(How: Navigate)**
  - Expand score function **(How: Navigate -> Geometric Zoom)**

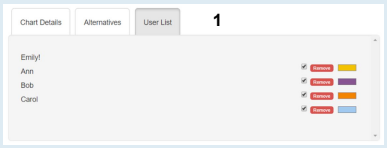
33

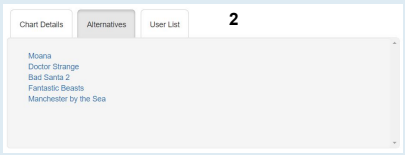
Group View

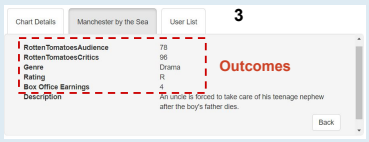
34

Group View - Details Component

1. ***User List*** **tab**
2. ***Alternatives*** **tab**
3. ***Alternatives*** **tab after clicking an alternative**

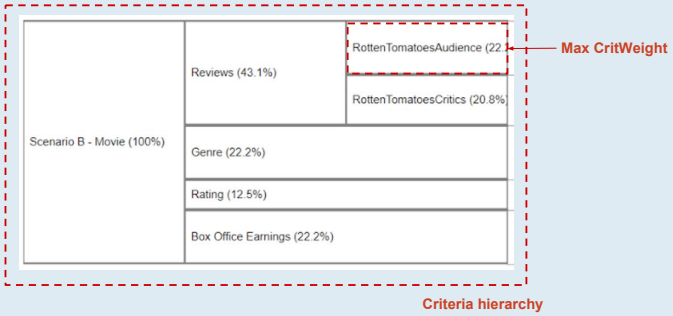






35

Group View - Criteria Component

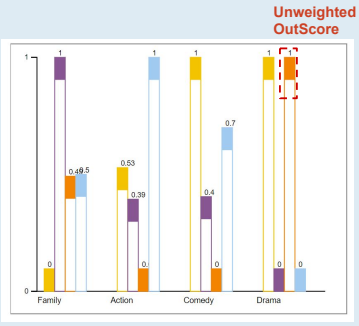


36

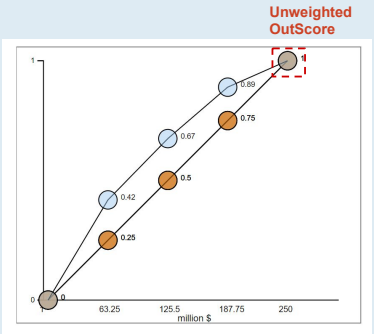
Group View - Scores Component

37

Group View - Score Functions Component



Categorical Score Function



Continuous Score Function

38

How: Encode

Measure Class	Measure	View Component	Idiom	Encoding
Scores	TotalScore (AvgScore)	Scores	1. Aligned multi bar chart 2. Tabular multi-bar chart Dimension mappings: Alternatives -> columns (primary) Criteria -> rows Evaluators -> columns (secondary), colour	Vertical position of horizontal line (1)
	AltScore			Height of bar (1) + text
	AltCritScore			Height of filled bar (2)
	Max(AltScore)			Red-coloured text
Weights	CritWeights			Height of unfilled bar (2)
	Max(CritWeight)			Row height
		Criteria	Rectilinear node-link graph	Row height
Outcomes	Outcome	Scores	""	Text label in tabular bar chart
		Details	Tabular list	Text
Score Functions	UnweightedOutScore	Score Functions	Interactive bar chart/ line graph	Height of bar/Y-coordinate of dot

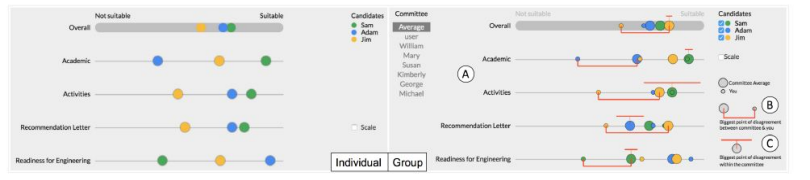
39

How: Manipulate (View Changing Interactions)

- **Change arrangement:**
  - Change orientation (vertical or horizontal)
  - Reorder Objectives (drag-and-drop)
  - Reorder Alternatives (drag-and-drop, alphabetical, or by Objective score)
- **Change mapping:**
  - Change color for user
- **Change data shown:** **(How: Reduce -> Filter)**
  - Filter users (toggle checkboxes)
  - Choose Alternative to see Outcomes for (click on name in Alternatives tab)
    - This is a special case of filter where exactly one item may be chosen
- **Change elements shown:**
  - Toggle view options (average lines, score functions, outcomes, score labels, utility scale)
- **Change viewpoint:** **(How: Navigate)**
  - Expand score function **(How: Navigate -> Geometric Zoom)**

40

ConsensUs [2]



41

What data is supported?

Measures:		
	Taxonomy level?	**P1b**
	Evaluator weights?	✗
Dimensions:		
	Criteria hierarchies?	✗
	Evaluator hierarchies?	✗

42

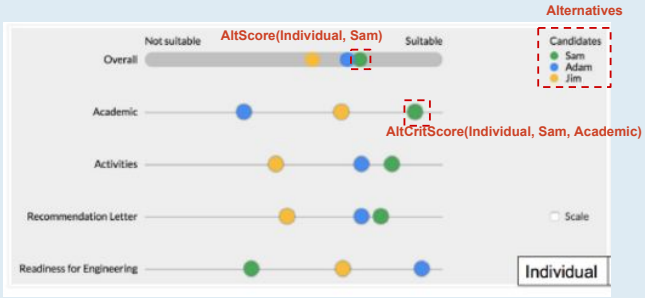
Overall Organization

- One main window with two linked views:
  - Individual view
  - Group view

How: Facet -> Partition
How: Facet -> Linked views

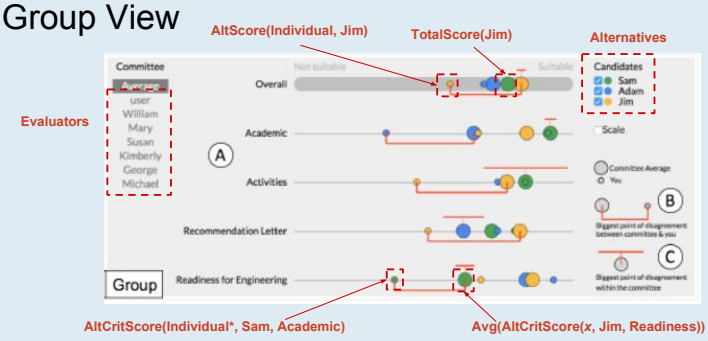
43

Individual View



44

Group View



45

How: Encode

Measure Class	Measure	View	Idiom	Encoding
Scores	AltScore	Group and Individual	Small multiples, dot plot (specifically, Cleveland dot plot) Dimension Mappings: Alternatives -> colour Criteria -> rows Evaluators -> size (two levels)	Position of dot on plot (horizontal axis)
	AltCritScore			
	TotalScore	Group		
	Avg(AltCritScore(*x*, a, c)) *			

46

How: Manipulate (Data Changing Interactions)

- **Change scores:**
  - Adjust position of dot along criteria slider (click-and-drag)

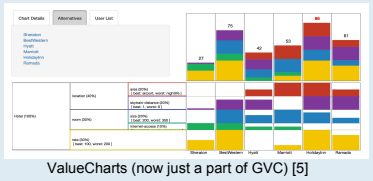
47

How: Manipulate (View Changing Interactions)

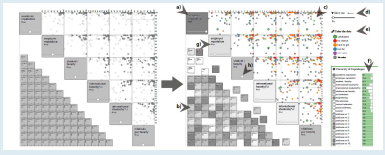
- **Change data shown:** **(How: Reduce -> Filter)**
  - Filter alternatives (toggle checkboxes)
  - Choose an evaluator to map to big dots (click on name in list)
    - This is a special case of filter where exactly one item may be selected
- **Change aggregation level:** **(How: Reduce -> Aggregate)**
  - Drill-down average criterion score to see breakdown by evaluator (click on dot)

48

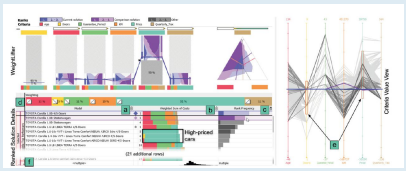
Preview: Individual MCDA



ValueCharts (now just a part of GVC) [5]



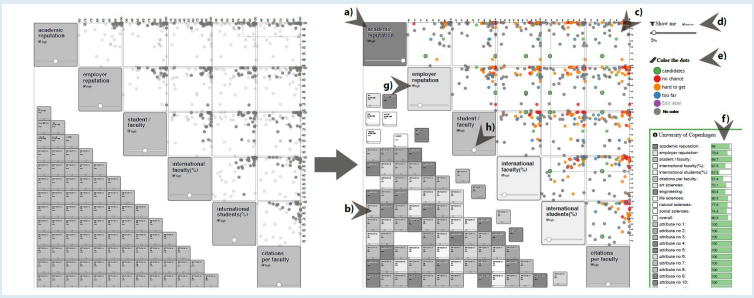
DCPAIRS [6]



WeightLifter [7]

49

DCPAIRS [6]



50

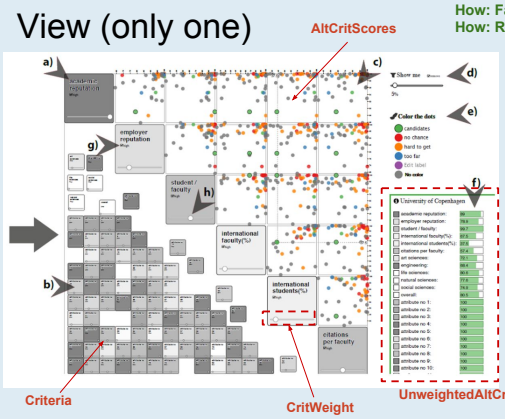
What data is supported?

Measures:		
	Taxonomy level?	**P2+w**
	Evaluator weights?	✗ (Single evaluator)
Dimensions:		
	Criteria hierarchies?	✗
	Evaluator hierarchies?	✗ (Single evaluator)

51

View (only one)

How: Facet -> Partition
How: Facet -> Linked Views
How: Reduce -> Embed -> Focus + Context



- SPLOM (scatter-plot matrix) over six criteria at a time
- Each cell shows trade-off between row/column criteria (each point is an Alternative)
- Other criteria can be swapped in, and are shown as tiles in the bottom corner
- Criteria weight is encoded on slider

52

How: Encode

Measure Class	Measure	Idiom	Encoding
Scores	UnweightedAltCritScore	Scatter-plot matrix Dimension mappings: Alternatives -> spatial coordinates Criteria -> spatial regions	x or y coordinate of point on scatter plot (two criteria per plot) UnweightedAltCritScore(*a*, *c*) is shown on every plot in the row and column for *c*
		Bar chart	Length of bar + text
Weights	CritWeights	---	Position of knob on slider widget; color of tile (grayscale)

53

How: Manipulate (Data Changing Interactions)

- **Change weights:**
  - Adjust position of dot along criteria slider (click-and-drag)
- **Change score function:**
  - Toggle positive linear/negative linear (click text)
- **Define alternative groups:**
  - Assign selected alternatives to a group

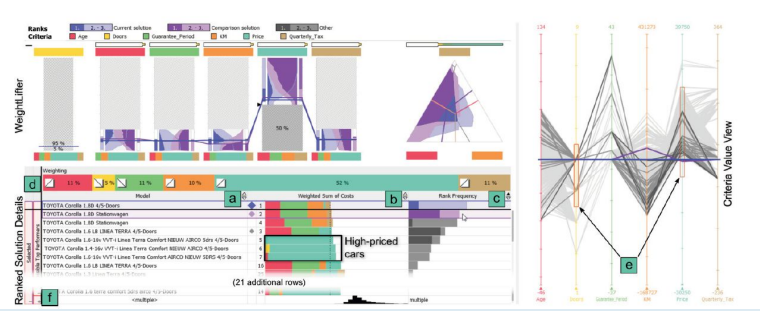
54

How: Manipulate (View Changing Interactions)

- **Change mapping:**
  - Change color for alternative group
- **Change data shown:**
  - Filter alternatives on score (adjust position on range sliders)
  - Choose attributes to put on the main diagonal (drag-and-drop)
    - This is a special case of filter where exactly six items may be chosen
- **Change emphasis:**
  - Highlight selected alternative in all plots (click on point in one plot)

55

WeightLifter [7]



56

What data is supported?

Measures:		
	Taxonomy level?	**P2b+w**
	Evaluator weights?	✗ (Single evaluator)
Dimensions:		
	Criteria hierarchies?	✗
	Evaluator hierarchies?	✗ (Single evaluator)

57

Overall Organization

- One main window with three different views:
  - Ranked Solution Details view
  - Criteria view
  - WeightLifter view

How: Facet -> Linked Views
How: Facet -> Superimpose
How: Reduce -> Embed -> Focus + Context

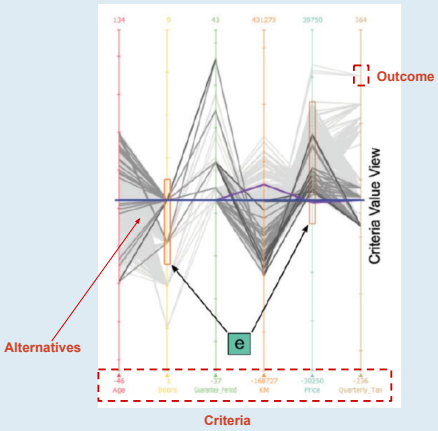
58

Ranked Solution Details View

59

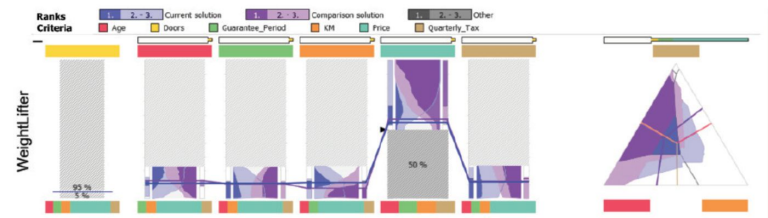
Criteria Value View

- Parallel coordinates plots where:
  - Each line is an Alternative
  - Each axis is a Criterion
  - Each coordinate is an Outcome



60

WeightLifter View



61

How: Encode

Measure Class	Measure	View	Idiom	Encoding
Scores	AltScore	Ranked Solution Details	Table with embedded stacked bars Dimension mappings: Alternatives -> rows Criteria -> colour	Length of bar
	AltCritScore			Length of segment
Weights	CritWeights		Stacked bar (on top of above-mentioned table)	Length of bar, text
Score Functions	UnweightedOutScore			Line graph glyph
Outcomes	Outcome	Criteria Values	Parallel coordinates Dimension mappings: Alternatives -> marks (lines) Criteria -> axes, colour	Coordinate of line for Alternative on axis for Criterion

62

How: Manipulate (Data Changing Interactions)

- **Change weights:**
  - Adjust height of box in Criteria Component
    - Click-and-drag
    - “Pump” (double-click to inflate/deflate)
- **Change score function:**
  - Adjust y-coordinate of a point/bar in the score functions graph (click-and-drag)

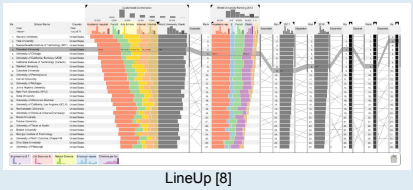
63

How: Manipulate (View Changing Interactions)

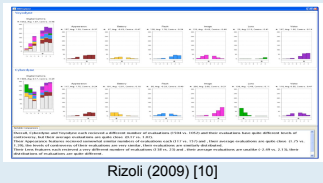
- **Change arrangement:**
  - Reorder Alternatives (by selected criterion score)
- **Change emphasis:**
  - Highlight selected alternative in all views (click on point in one plot)
- **Change data shown:** **(How: Reduce -> Filter)**
  - Filter alternatives (toggle in Ranked Solution Details View)
  - Filter alternatives by criterion value (brush values in Criteria Values View)

64

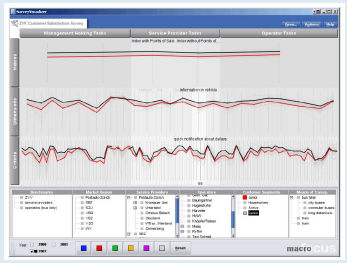
Preview: Related Datasets



LineUp [8]



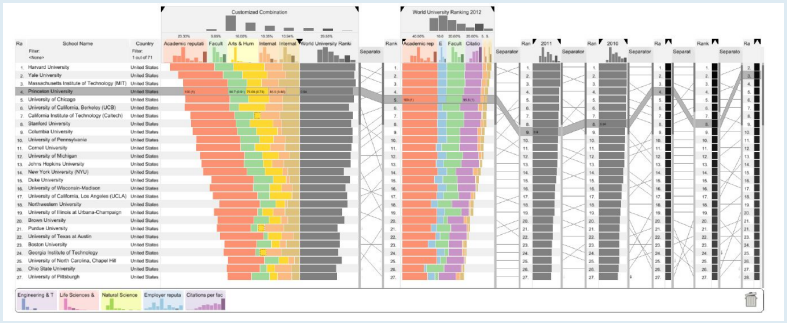
Rizoli (2009) [10]



SurveyVisualizer [9]

65

LineUp [8]



66

What data is supported?

Measures:		
	Taxonomy level?	**P2b+w (analogous)**
	Evaluator weights?	✗ (Single evaluator)
Dimensions:		
	Criteria hierarchies?	✓ (Define on the fly)
	Evaluator hierarchies?	✗ (Single evaluator)

67

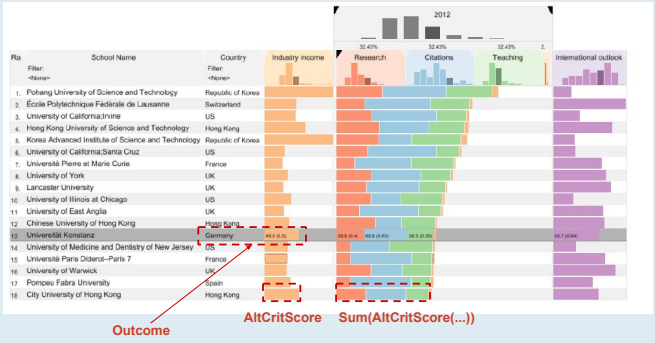
Overall Organization

- One interactive main view that allows users to dynamically define and compare multiple rankings
- Data-mapping (e.g. score function) editor available on demand

How: Facet -> Linked Views
How: Facet -> Superimpose
How: Reduce -> Embed -> Focus + Context -> Distort

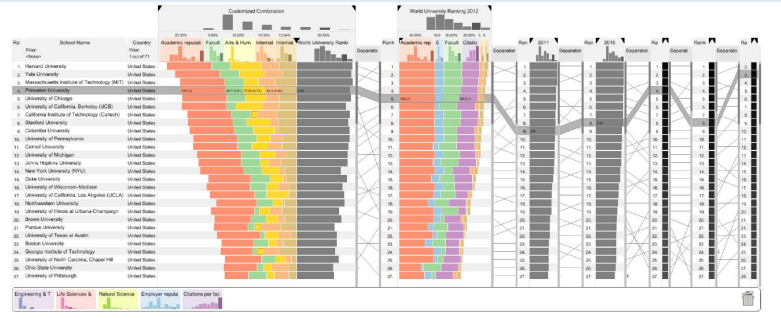
68

Single-Ranking View



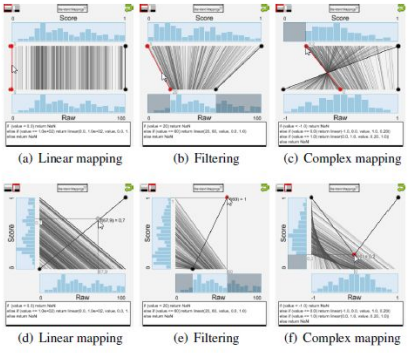
69

Multi-Ranking View



70

Data-Mapping Editors



- Used to define mapping from domain values to scores (i.e. score functions)
- Can also filter values by not mapping them to any score
- (No need to get into the details beyond this)

71

How: Encode

Measure Class	Measure	View	Idiom	Encoding
Ranks (not included in design space analysis)	AltRank	Main	Slope graph/bump chart; Table with embedded bars (with option of aligned, stacked, or stack diverging} Dimension mappings: Alternative -> rows Criteria -> columns, colour	Row order, text label
Scores	AltScore			Length of stacked bar (available on demand)
	AltCritScore			Length of segment
Weights	CritWeights			Column width
Outcomes	Outcome			Text label on segment for AltCritScore (numeric Criteria) Text label in cell for criterion (categorical Criteria)
Score Functions	UnweightedOutScore	Data-Mapping Editor	?	Coordinate on score axis

72

How: Manipulate (Data Changing Interactions)

- **Change weights:**
  - Adjust width of column for criterion (click-and-drag, or manually enter a percentage)
- **Change score function:**
  - Adjust boundaries in data-mapping editor
- **Define meta-criteria:**
  - Assign selected criteria columns to a group

73

How: Manipulate (View Changing Interactions)

- **Change arrangement:**
  - Reorder Alternatives by column or meta-column score (click on header)
  - Change alignment strategy (stacked bars, aligned bars, diverging bars, or sorted bars)
- **Change level of detail:**
  - Expand/collapse criterion column
  - See exact outcomes for an Alternative (hover over row)
- **Change data shown:** **(How: Reduce -> Filter)**
  - Filter alternatives by categorical criterion value (enter text filter in widget in column header)
  - Filter alternatives by numeric criterion value (adjust mappings in Data-Mapping Editor)
  - Filter missing values (checkbox toggle in Data-Mapping Editor)
- **Change emphasis:**
  - Highlight selected alternative in all plots (hover for grey highlighting, click for yellow)

cont...

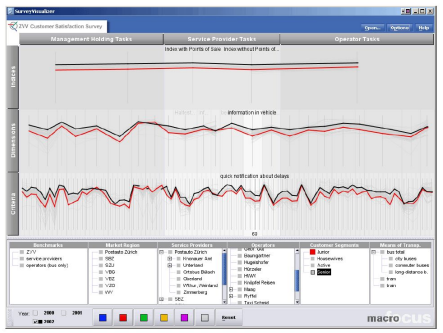
74

How: Manipulate (View Changing Interactions)

- **Change navigation strategy:**
  - Toggle between uniform and fisheye view of rows
- **Change viewpoint:** **(How: Navigate)**
  - Change position of fisheye lens **(How: Navigate -> Pan)**
- **Create new linked viewt:** **(How: Navigate)**
  - Create snapshot of current view (which will appear next to it, connected by a slope graph)

75

SurveyVisualizer [9]



76

What data is supported?

Measures:		
	Taxonomy level?	**P1b (analogous)**
	Evaluator weights?	✗ (Single evaluator)
Dimensions:		
	Criteria hierarchies?	✓
	Evaluator hierarchies?	✗ (Single evaluator)

77

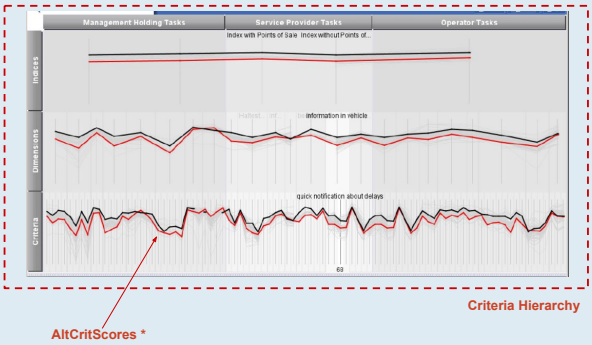
Overall Organization

- One main window with two linked views:
  - Parallel Coordinates Tree View
  - Analysis Group Selector View

How: Facet -> Linked views
How: Reduce -> Embed -> Focus + Context -> Distort

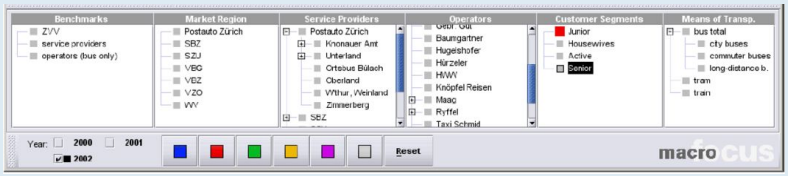
78

Parallel Coordinate Tree



79

Analysis Group Selector



- Narrows down the set of surveys included (here, surveys are analogous to alternatives).
- Nothing relevant to GPC.

80

How: Encode

Measure Class	Measure / Dimension	Idiom	Encoding
Scores	AltCritScore	Parallel coordinate tree (rectilinear tree with embedded parallel coordinates plots) Dimension mappings: Evaluators -> marks (lines) Criteria -> axes	Coordinate of line for Evaluator Group on axis for Criterion
N/A	Criteria		Line; region of rectilinear tree

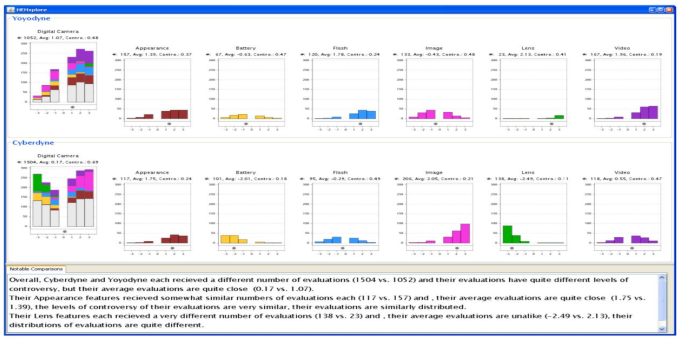
81

How: Manipulate (View Changing Interactions)

- **Change mapping:**
  - Change color for Alternative
- **Change emphasis:**
  - Highlight selected alternative in red (hover over line)
  - Highlight selected alternative in black (click on line)
- **Change data shown:** **(How: Reduce -> Filter)**
  - Filter alternatives by analysis group (expand tree, toggle checkbox)
  - Filter alternatives by criterion value (brush values in Criteria Values View)
- **Change viewpoint:** **(How: Navigate -> Pan)**
  - Change position of bifocal lens

82

Rizoli (2009) [10]



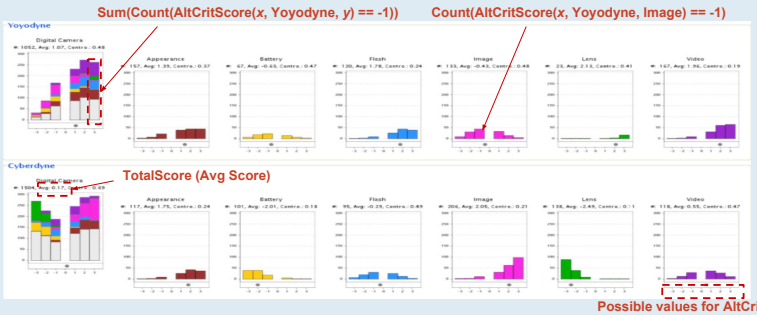
83

What data is supported?

Measures:		
	Taxonomy level?	**P1b (analogous)**
	Evaluator weights?	✗
Dimensions:		
	Criteria hierarchies?	✓
	Evaluator hierarchies?	✗

84

Main View



How: Facet -> Partition

85

How: Encode

Measure Class	Measure	Idiom	Encoding
Scores	Avg(AltCritScore(*x*, a, c))	Small-multiples of histograms Dimension mappings: Alternatives -> rows Criteria -> columns	Text label
	Count(AltCritScore(*x*, a, c) == *value*)		Height of bar
		Stacked bar chart Dimension mappings: Alternatives -> rows Criteria -> colour	Height of segment
	Sum(Count(AltCritScore(*x*, a, *y*) == *value*))		Height of bar
	TotalScore (Avg(AltCritScore(*x*, a, *y*)))		Text label

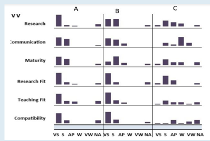
86

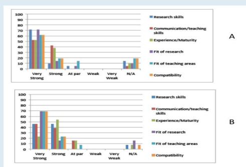
Class 2: Standalone Encodings

87

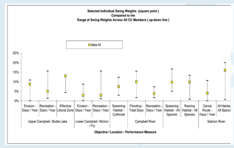
Case Studies

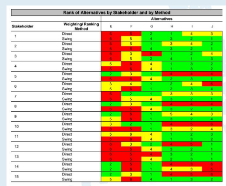
Faculty Hiring





Campbell River





Best Paper

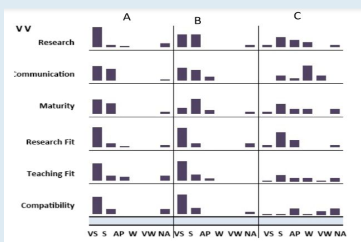
	A	B	C	D	E	F	G
1		Person 1	Person 2	Person 3	Person 4	Person 5	**Rank Sum**
2	Paper A	2	1.5	1	3	3	**10.5**
3	Paper B	1	1.5	2	1	1	**6.5**
4	Paper C	3	4	3	3	2	**15**
5	Paper D	4	3	4	3	4	**18**

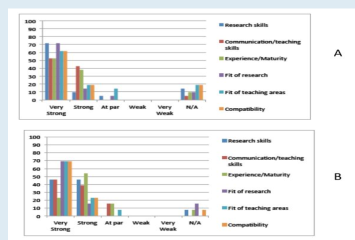
Voyager [11]

TABLE IV
THE SCIENCE TEAM ORDINAL RANKINGS FOR PREFERRED TRAJECTORY PAIRS

	Collective choice ranking		Science team ordinal rankings									
Trajectory pairs	Rank sum	Additive $u^i(t^i_{32}) = p\phi^i$; $\lambda^i = 1.0$	RSS	IRIS	ISS	PPS	UVS	CRS	LECP	MAG	PLS	PRA
31	1	2	20.5	3.0	5.0	8.5	6.0	8.0	4.0	3.0	4.0	5.0
29	2	3	20.5	5.0	19.0	6.5	9.0	3.0	2.0	2.0	4.0	4.0
26	3	1	20.5	2.0	10.0	11.0	7.0	17.5	3.0	1.0	1.5	2.0
27	4	4	20.5	1.0	30.0	16.0	3.0	17.5	1.0	4.0	4.0	3.0
5	5	6	20.5	9.0	28.5	17.0	4.0	13.0	6.0	6.0	1.5	6.0
25	6	5	20.5	7.0	28.5	25.0	9.0	1.0	5.0	5.0	11.0	1.0
35	7	7	20.5	21.0	2.0	2.0	13.5	13.0	12.0	9.0	17.5	10.0
17	8	10	20.5	4.0	13.5	6.5	13.5	24.5	9.0	14.0	17.5	8.0
8	9	8	20.5	20.0	10.0	5.0	13.5	21.0	11.0	7.0	17.5	9.0
10	10	12	3.0	11.0	24.0	8.5	19.5	2.0	20.0	24.0	17.5	7.0

Faculty Hiring





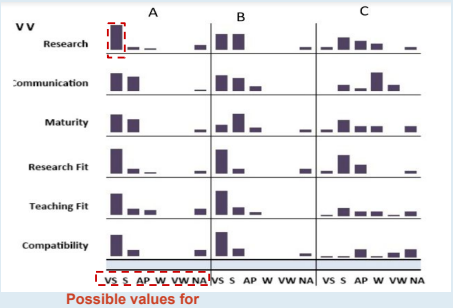
How: Facet -> Partition

What is the data?

Measures:		
	Taxonomy level?	**P1b (with discrete evaluation scale)**
	Evaluator weights?	✗
Dimensions:		
	Criteria hierarchies?	✗
	Evaluator hierarchies?	✗

View 1

Count(AltCritScore(*x*, A, Research) == VS)



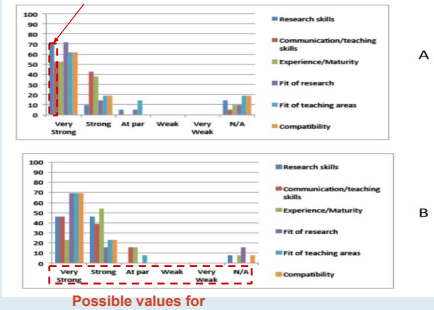
Possible values for AltCritScore (specified by discrete evaluation scale)

- Small-multiples view faceted by criteria on rows, alternatives on columns
- Each cell shows distribution of AltCritScores over possible values of AltCritScore:
  - Bar encodes *count* of AltCritScores with that value
- One bar for each combination of Alternative, Criteria, and possible AltCritScore value (aggregated over Evaluators)

91

View 2

Count(AltCritScore(*x*, A, Research) == VS)



Possible values for AltCritScore (specified by discrete evaluation scale)

- Small-multiples view partitioned by Alternative
- Each view consists of a *multi-bar chart*, partitioned into regions by possible values of AltCritScore
- Each region shows distribution of AltCritScores over criteria
- One bar for each combination of Alternative, Criteria, and possible AltCritScore value (aggregated over Evaluators)

92

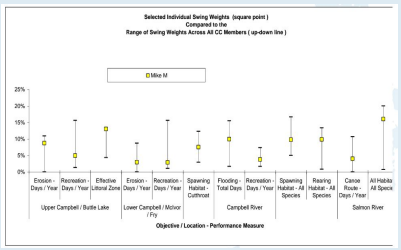
How: Encode

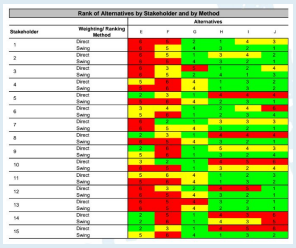
Measure Class	Measure	View	Idiom	Encoding
Scores	Count(AltCritScore(*x*, a, c) == *value*) *	1	Small-multiples Dimension mappings: Alternatives -> columns (primary) Criteria -> rows Outcomes -> columns (secondary)	Height of bar
		2	Small-multiples, multi-bar chart Dimension mappings: Criteria -> columns (secondary), colour Outcomes -> columns (primary)	Height of bar; Text labels on Count axis

View 1 and 2 show the same data in different ways

93

Campbell River





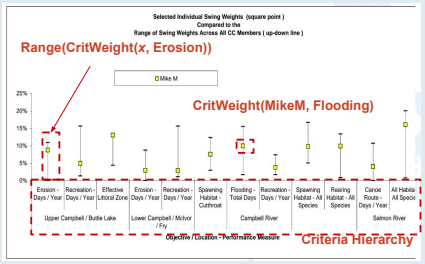
94

What is the data?

Measures:		
	Taxonomy level?	**P0b and P2+w**
	Evaluator weights?	✗
Dimensions:		
	Criteria hierarchies?	✓
	Evaluator hierarchies?	✗

95

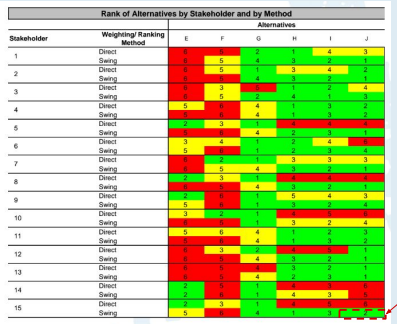
View 1



- Range plot, with one range glyph per Criterion
- Each range glyph shows the range (min and max) of CritWeights over that criterion
- One bar for each combination of Alternative, Criteria, and possible AltCritScore value (aggregated over Evaluators)

96

View 2



- Table with Evaluators/Elicitation method on rows, Alternatives on columns, and AltRanks in cells
- AltScores are sorted into three bins, and a different colour is used for each bin (it is unclear what the scale is)

AltRank(15, J) (number) and AltScore(15, J) (colour)

97

How: Encode

Measure Class	Measure	View	Idiom	Encoding
Ranks (not included in design space analysis)	AltRank	2	Table Dimension mappings: Alternatives -> columns Evaluators -> rows	Text
Scores	AltScore			Colour (three bins)
Weights	Range(CritWeights)	1	Range plot Dimension mappings: Criteria -> columns	Range bar
	CritWeight			Point on range bar
N/A	Criteria Hierarchy			Label groups (a tree, loosely)

Views 1 and 2 show different data

98

Best Paper

Data:

Measures:		
	Taxonomy level?	P0a
	Evaluator weights?	✗
Dimensions:		
	Criteria hierarchies?	✗
	Evaluator hierarchies?	✗

Encoding:

	A	B	C	D	E	F	G
1		Person 1	Person 2	Person 3	Person 4	Person 5	Rank Sum
2	Paper A	2	1.5	1	3	3	10.5
3	Paper B	1	1.5	2	1	1	6.5
4	Paper C	3	4	3	3	2	15
5	Paper D	4	3	4	3	4	18

AltRank(Person 1, Paper D)

Sum(AltRank(x, Paper D)

Analysis:

Measure Class	Measure	Idiom	Encoding
Ranks (not included in design space analysis)	AltRank	Table	Text
	Sum(AltRank)	Dimension mappings: Alternatives -> cols Evaluators -> rows	Text

99

Voyager [11]

Data:

Measures:		
	Taxonomy level?	**P0a, b**
	Evaluator weights?	✓
Dimensions:		
	Criteria hierarchies?	✗
	Evaluator hierarchies?	✗

Encoding:

- **Table II:** Alternatives on rows, Evaluators on columns, AltRank and AltScore in cells
- **Table III:** Alternatives on rows, Collective choice rules (including EvaluatorWeights) on columns, TotalRank and TotalScore in cells

Analysis:

Measure Class	Measure	Idiom	Encoding
Ranks (not included in design space analysis)	TotalRank *	Table (Table III)	Text
	AltRank	Table (Table II)	
Scores	TotalScore *	Table (Table III)	
	AltScore	Table (Table II)	
Weights	EvaluatorWeights	Table (Table III)	

100

References

[1] Bajracharya, S., et al. "Interactive visualization for group decision analysis." *International Journal of Information Technology & Decision Making*. 2016. (in revision)

[2] Mahyar, N., et al. "ConsesnsUs: Visualizing points of disagreement for multi-criteria collaborative decision making." *Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 2017.

[3] Mustajoki, J. and Hamalainen, R.P. "Web-HIPRE: Global decision support by value tree and AHP analysis." *INFOR: Information Systems and Operational Research*, 38.3 (2000): 208–220.

[4] Mustajoki, J., Hamalainen, R.P., and Sinkko, K. "Interactive computer support in decision conferencing: Two cases on off-site nuclear emergency management." *Decision Support Systems*, 42.4 (2007): 2247–2260.

[5] Carenini, G. and Lloyd, J. "ValueCharts: Analyzing linear models expressing preferences and evaluations." *Proceedings of the Working Conference on Advanced Visual Interfaces*, 150–157. ACM, 2004.

[6] Dimara, E., Valdivia, P., and Kinkeldey, C. "DCPAIRS: A pairs plot based decision support system." *EuroVis-19th EG/VGTC Conference on Visualization*. 2017.

[7] Pajer, S., et al. "WeightLifter: Visual weight space exploration for multi-criteria decision making." *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017): 611-620.

[8] Gratzl, S., et al. "Lineup: Visual analysis of multi-attribute rankings." *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013): 2277-2286.

[9] Brodbeck, D. and Girardin, L. "Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree." *Symposium on Information Visualization*. IEEE, 2003.

[10] Carenini, G. and Rizoli, L. "A multimedia interface for facilitating comparisons of opinions." *Proceedings of the 14th International Conference on Intelligent User Interfaces*. ACM, 2009.

[11] Dyer, J.S. and Miles Jr., R.F. "An actual application of collective choice theory to the selection of trajectories for the Mariner Jupiter/Saturn 1977 project." *Operations Research* 60.3 (1976): 220-244.

101