

WeightLifter: Visual Weight Space Exploration for Multi-Criteria Decision Making

Stephan Pajer Marc Streit Thomas Torsney-Weir Florian Spechtenhauser Torsten Möller Harald Piringer

Abstract— A common strategy in Multi-Criteria Decision Making (MCDM) is to rank alternative solutions by weighted summary scores. Weights, however, are often abstract to the decision maker and can only be set by vague intuition. While previous work supports a point-wise exploration of weight spaces, we argue that MCDM can benefit from a regional and global visual analysis of weight spaces. Our main contribution is *WeightLifter*, a novel interactive visualization technique for weight-based MCDM that facilitates the exploration of weight spaces with up to ten criteria. Our technique enables users to better understand the sensitivity of a decision to changes of weights, to efficiently localize weight regions where a given solution ranks high, and to filter out solutions which do not rank high enough for any plausible combination of weights. We provide a comprehensive requirement analysis for weight-based MCDM and describe an interactive workflow that meets these requirements. For evaluation, we describe a usage scenario of *WeightLifter* in automotive engineering and report qualitative feedback from users of a deployed version as well as preliminary feedback from decision makers in multiple domains. This feedback confirms that *WeightLifter* increases both the efficiency of weight-based MCDM and the awareness of uncertainty in the ultimate decisions.

Index Terms—Visual analysis, decision making, multi-objective optimization, interactive ranking, rank sensitivity

1 INTRODUCTION

In our daily lives as well as in professional settings, decisions typically require the consideration of multiple – often conflicting – criteria. Buying a car, for example, involves personally assessing multiple car models regarding criteria such as cost, comfort, safety, and fuel economy. More comfortable and safer cars usually come at a higher price. In portfolio management, stocks enabling high returns typically involve high risks. In product design, competing criteria include performance, reliability, production costs, and form factors.

Multi-Criteria Decision Making (MCDM, also known as Multi-Criteria Decision Analysis) is thus a challenge of ubiquitous importance and has long been a dedicated field of research [40]. As a sub-field of Operations Research, MCDM focuses on structuring and solving decision and planning problems for which unique optimal solutions do not exist but where the superiority of solutions depends on the decision maker's preferences. In this paper, we focus on discrete MCDM problems comprising a finite set of alternative solutions.

In order to support decision makers facing such problems, the MCDM literature discriminates between various decision strategies [27]. As one of the most frequently applied strategies, *Additive Weighting* assigns weights to the criteria in order to express the preference for each solution by a weighted summary score for subsequent ranking (see Sec. 4.1 for a formal description). This strategy makes it straightforward to obtain a unique decision—typically the solution ranking first. Moreover, the explicit and quantitative representation of preference as weights supports reproducibility.

In practice, however, this seemingly precise and transparent strategy conceals significant sources of uncertainty. In many cases, the weights are not known precisely, but set by vague intuition. For example, a car buyer may prefer safety over fuel economy and costs, but might be unsure about precise weights for these criteria. Weights are often rather abstract to the decision maker. Moreover, Additive Weighting typically requires to normalize the criteria before summa-

tion, which represents an additional source of ambiguity. In general, reducing decisions to a single score may hide interesting options and the ranks of solutions may be highly sensitive to changes in weights. These uncertainties may significantly reduce the confidence in the ultimate decision.

Decision makers may thus benefit from suitable tools that support them in exploring the relationship between weights and the corresponding (top) ranked solutions. While this seems an opportunity for interactive visualization, few tools explicitly address general MCDM problems and even fewer tools allow the user to see the effect of manipulating weights [38]. A notable exception, *LineUp* [16], enables the user to compare entire rankings corresponding to user-defined weightings and served as inspiration and starting point of our work. *LineUp* employs a point-wise approach to the exploration of the space of possible combinations of weights (referred to as *weight space*). In the context of MCDM, this approach supports only a cumbersome and indirect assessment of the sensitivity of a top-ranked solution to variations in weights and does not support expressing a vague intuition about weightings. We argue that MCDM can benefit from a more comprehensive exploration of the weight space.

Our main contribution is *WeightLifter*, a new interactive visualization technique that increases both the efficiency of MCDM using Additive Weighting and the confidence in the ultimate decisions. Its main novel aspect is a global visual representation and exploration of weight spaces with up to ten criteria. This enables, for example, visualizing the sensitivity of a decision to weight changes, an efficient localization of weight regions where a given solution ranks high, and the filtering of solutions which do not rank high enough for any plausible combination of weights. Additional contributions of the paper include:

- A comprehensive requirement analysis of weight-based MCDM.
- A concept for integrating *WeightLifter* with other views to support an interactive workflow for MCDM.
- A description of a usage scenario of *WeightLifter* in automotive engineering.
- A report of qualitative feedback from engineers using a deployed version of *WeightLifter*.

2 REQUIREMENT ANALYSIS

In order to ensure that *WeightLifter* is applicable to a broad set of MCDM problems, we identified the requirements from multiple sources: We had several discussions with decision makers from different fields (automotive engineering, health policy making, energy transmission operation, and aviation infrastructure planning), reflected on experiences with our previous work on *LineUp* [16], and conducted

- *Stephan Pajer, Florian Spechtenhauser, and Harald Piringer are with the VRVis Research Center. E-Mail: {pajer, fspechten, hp}@vrvis.at.*
- *Thomas Torsney-Weir and Torsten Möller are with the University of Vienna. E-Mail: {thomas.torsney-weir, torsten.moeller}@univie.ac.at.*
- *Marc Streit is with Johannes Kepler University Linz. E-Mail: marc.streit@jku.at.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

thorough research studies on limitations of existing approaches from MCDM [35, 40] (Sec. 3). We refined these requirements in multiple iterations, following the nested model for visualization design and validation [26]. We grouped the resulting requirements into three high-level goals: (1) high decision confidence, (2) efficient decision making, and (3) scalability to problems involving many criteria and a large number of solutions.

As a guiding example throughout this paper, consider a fictional purchase situation: Assume that Jane wants to buy a used car from a list of 1,442 available models, which thus correspond to the possible ‘solutions’ in this example. She bases her decision on six criteria: *Price* (in EUR), *Age* (in months), *Mileage* (in kilometers), number of *Doors*, *Guarantee Period* (in months), and quarterly *Tax* (in EUR). Jane wants an affordable, reliable family car, but could not quantify her preferences as precise criteria weights. She thus wants to reduce the 1,442 models to a short list which still covers a range of good choices for different realistic preferences.

2.1 High Decision Confidence

R1: Analyzing trade-offs. It should be intuitive to express preferences (i.e., weights) and to explore the effect of variations in preferences. For this task, we found that users often have particular trade-offs in mind, including trade-offs between two criteria, between one criterion and all others, and between groups of criteria. *Jane may want to explore the trade-off between Price and all other criteria.*

R2: Visualizing weight-related sensitivities of decisions. The sensitivity of a decision along interesting trade-offs (R1) should be directly perceptible. *To increase her decision confidence, Jane wants to know if certain variations of weights would have a significant impact on the best choice of car model.*

R3: Localizing favorable weight space regions. It should be efficient to localize and characterize the region in the weight space where a particular solution is considered a good or even the best choice. *Aware of her vague intuition of weights, Jane might prefer localizing cars that are robustly considered good choices for many different weightings and to match this short list of candidates against her intuitive preferences.*

2.2 Efficient Decision Making

R4: Filtering decision-irrelevant weight space regions. In most cases, large parts of the weight space represent preferences which are irrelevant for the decision maker. *As Price is an important criterion to Jane, she is not interested in expensive cars that only become good choices when the weight on Price is low.* To focus the analysis, users should thus be able to specify which criteria weightings are meaningful for them.

R5: Focus on decision-relevant solutions. The set of solutions which is presented to the user in detail should only be as large as necessary. Specifically, it should not contain solutions which fail to be good choices for any plausible weighting (R4) or violate other filter criteria (R9). *In the guiding example, only 24 cars are considered an optimal choice for any possible combination of weights. This enables Jane to focus her attention on a small set early in the decision making process without requiring her to specify criteria preferences.*

R6: Dynamic transparent ranking of solutions. In addition to an instant identification of the best choice for a given weighting of criteria, the user should be presented a ranking of all other decision-relevant solutions as context information. The visual encoding should explain the cause of the ranking and should convey for which criteria a solution has sub-optimal values. *This enables Jane to assess, for example, if the best-ranking model is in fact much better than other candidates for a given weighting, and how multiple roughly equivalent alternatives differ regarding their per-criterion strengths.*

R7: Comparison of solutions regarding favorable weight regions. The technique should support a selection of solutions for a detailed comparison. In particular, it should provide an effective visual encoding when comparing the favorable regions in weight space for multiple solutions. *After initial steps for reducing the number of possible car candidates (R5 and R4), Jane wants to further shorten*

the list by excluding models based on direct comparisons between the remaining cars. In addition to the actual data values of the criteria, Jane wants to compare different car models by their robustness in being considered a good choice (R3).

2.3 Problem Scalability

R8: Scalability and flexibility regarding criteria. The technique should avoid any upper limit on the number of the criteria while being intuitive also for problems with only two or three criteria. It should be applicable to criteria of different types and scaling. *Jane’s criteria include ones to be minimized (e.g., Price) as well as maximized (e.g., Guarantee Period) and are partly discrete (e.g., number of Doors).*

R9: Incorporating additional decision making strategies. The technique should support mixing the Additive Weighting strategy with other strategies, e.g., Elimination by Aspects [38] for pre-filtering solutions based on actual criteria values rather than expressing filters solely in terms of weights (as in R4). *Jane has a hard upper limit on the Price of her new car regardless of all other criteria.*

3 RELATED WORK

Weight space-aware MCDM is related to parameter space exploration in general (Sec. 3.1) and other work on MCDM (Sec. 3.2).

3.1 Parameter Space Exploration

For many types of complex computations, analyzing the correspondence between inputs and outputs is crucial. An increasing number of visualization systems aims to replace the traditional exploration based on trial-and-error by more global strategies [33]. Examples include meteorological simulations [29], disease simulations [1], image analysis [30], and simulations in engineering and design [8, 13, 23]. The analysis of parameter spaces comprises multiple tasks, which are also relevant in the context of this paper.

Sensitivity analysis refers to understanding what variations of outputs can be expected with changes of inputs [33]. While sensitivities can be expressed on a global scale [32], most visualization systems focus on local sensitivities, e.g., in scatter plots [11]. In several cases, local sensitivities are defined around a focal point, which can be used for slicing higher-dimensional parameter spaces [4]. HyperSlice [39], for example, shows all 2D orthogonal slices of a scalar function around a focal point. This work inspired our concept of navigating weight spaces (R1, Sec. 4). While most previous work addresses Cartesian parameter spaces, sensitivity analysis in our context refers to the stability of top-ranks (R2) within barycentric coordinate spaces.

As another task, *partitioning* the output space reveals what different types of outputs can be achieved [33]. Bergner et al. [5] cluster fuel cell performance graphs as the outputs of a model. Mapping the clusters back into the input space reveals parameter regions that lead to similar results. In our work, the partitioning of weight spaces reveals segments where the top-ranked solution does not change (R3).

Another motivation for a systematic parameter space analysis concerns the identification of input parameter values which *optimize* the outputs in some sense. While the optimization regarding multiple quantitative objective functions is a classical problem of Operations Research (Sec. 3.2), an assessment of optimality often also involves qualitative judgments of complex data like time series [1, 18] segmented image data [37], animations [8], and 3D geometry [13]. This explains a need for systems which combine human-oriented assessments of outputs with mathematically defined decision making strategies such as Additive Weighting (R9). An example addressing both aspects in the context of light design is LiteVis [34].

3.2 Multi-Criteria Decision Making

In many decades of research, the field of Multi-Criteria Decision Making (MCDM) has investigated approaches for solving decision problems that involve multiple criteria [20] (remark: we use the term *criterion* in this paper, while the term *objective* is also common in the literature). A single best solution does generally not exist for MCDM problems. Instead, a solution is considered Pareto-optimal if no other solution exists that is better for some criteria without being worse for

others. Much work focuses on problems where the criteria are given as functions on a continuous decision domain. In this case, many approaches address the (semi-)automated creation of Pareto-optimal solutions, e.g., the NIMBUS method [25]. As a common classification of methods, the decision makers preferences can be given a priori, progressively, or a posteriori to creating the solutions [17]. In this paper, we focus on MCDM for a finite number of discrete solutions, which corresponds to the a posteriori case.

Velasquez and Hester survey the MCDM literature along eleven methods [40]. A main distinction concerns the strategy and required precision for specifying preference information by the decision maker. ELECTRE [31] and PROMETHEE [7], for example, ask for ordering the criteria by their preference, but do not support a direct identification of strengths and weaknesses of the alternative solutions. Additive Weighting allows a compensation between criteria but requires precise weights for each criterion. As argued in Sec. , the need to define precise weights may mismatch an often vague intuition about the relative importance of the criteria. Therefore, some methods extend Additive Weighting by including uncertainties regarding weights, e.g., based on multidimensional integrals [35] and random changes of weights [9]. While these methods are related to our goals, they are mathematical approaches to MCDM and do not address human-factors regarding transparent presentation (R6) and effective solution comparison (R7).

In visualization, several approaches address the representation of a discrete set of Pareto-optimal solutions, known as the Pareto Frontier. Korhonen and Wallenius [19] note that visualizing the Pareto Frontier for more than three criteria is difficult. Common approaches include scatter plot matrices of bi-objective slices [21] and extensions to parallel coordinates [2], which are the prevalent technique for visualizing Pareto Frontiers [3]. More recently, Chen et al. [12] propose a modification of self-organizing maps where anchor points correspond to criteria. While this creates a comprehensible overview of the Pareto Frontier, it introduces projection errors when more than three criteria are involved. In general, visualizing the Pareto Frontier implicitly facilitates MCDM by representing trade-offs, but does not provide explicit suggestions, e.g., by ranking the solutions (R6). Therefore, reaching a decision may still be hard for a large number of Pareto-optimal solutions [24] and requires additional strategies.

A recent study of 21 visualization systems [38] found that Lexicographic and Elimination by Aspects were by far the most commonly supported decision making strategies [27]. Vismon [6], for example, supports decision making in fisheries. The user can eliminate regions of the decision space by constraining on indicators which represent aggregations of complex simulation outputs. Only a few visualization tools support Additive Weighting as decision making strategy. ParaGlide [5] employs weighting by investigating a distance metric rather than by adjusting the weights directly. More related to our goals, ValueCharts [10] shows a ranking of solutions by weighting multiple criteria. Users may perform a sensitivity analysis of the ranks by changing the weights interactively. Conceptually similar, LineUp [16] employs Additive Weighting to let the user control the creation and comparison of rankings. The approach has quickly gained popularity, e.g., for ranking documents [14] and light designs [34]. However, the focus of LineUp is on the inspection of rankings in their entirety, not on the efficient identification of a small set of decision-relevant solutions (R5). Moreover, all described visualization techniques are limited to a point-wise exploration of weight spaces. They neither provide a visualization of local weight-related sensitivities (R2, with the exception of ValueCharts [10]), nor a global analysis of the weight space. The goal of our work is thus to bridge the gap between global exploration strategies for parameter spaces in other contexts [33] and the task of weight-based MCDM.

4 SENSITIVITY-AWARE WEIGHT SPACE EXPLORATION

This section presents WeightLifter, a novel technique to weight space analysis in the context of MCDM (see Fig. 1). WeightLifter addresses the interactive visualization of the weight space itself for navigation (R1, Sec. 4.2 and 4.3), weight-related sensitivity analysis (R2, Sec. 4.4), global solution characterization (R3, Sec. 4.5), and weight-

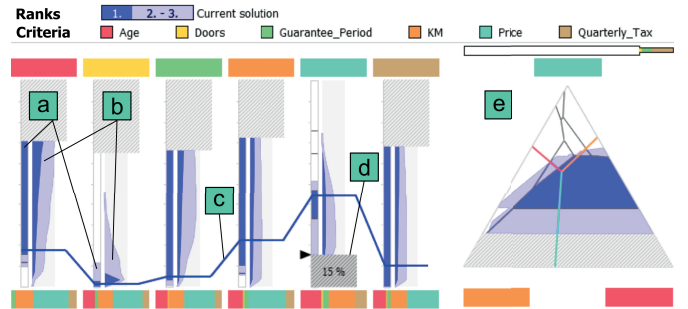


Fig. 1: WeightLifter for an MCDM problem of six criteria. Two-way trade-offs show the rank sensitivities (a) and the global rank distributions (b) of the top-ranking solution for the current weighting (c) and support defining constraints on the weights (d). A barycentric triangle (e) displays three-way trade-offs.

related filtering (R4, Sec. 4.6). WeightLifter is conceptually independent of any surrounding system or other linked views. For addressing the other requirements of MCDM problems, however, WeightLifter is suitable and intended to be used in conjunction with other views (Sec. 5) in order to enable a holistic MCDM workflow (Sec. 6).

4.1 Problem Formalization

Discrete MCDM can be formalized as choosing from a set of m possible solutions $\{S_1, \dots, S_m\}$ on the basis of n different criteria $\{C_1, \dots, C_n\}$. To support criteria of heterogeneous types and scales (R8), each criterion C_j has an associated *cost function* f_j that maps the criterion values of the solutions to the interval $[0, \infty)$. The cost functions thus express the cost in terms of dissatisfaction incurred by values of the criteria. A cost of zero indicates perfect satisfaction. Cost functions f_j with an increasing gradient can be used for numerical criteria that should be minimized (i.e., larger values cause larger dissatisfaction), while decreasing f_j express a desired maximization of C_j . In many cases, cost functions have a simple (e.g., linear) structure. A simple cost function for, e.g., *Mileage* assigns the value of the car with minimal *Mileage* to a cost of zero and linearly interpolates to assign the maximal *Mileage* to a cost of one. All examples in this paper apply corresponding linear cost functions for all criteria.

Consistent with *Additive Weighting* [40], the cost values of the criteria are weighted by a vector \mathbf{w} , referred to as *current weighting*, and summed up to attain an *overall cost* $v(S_i, \mathbf{w})$ per solution S_i .

$$v(S_i, \mathbf{w}) = \sum_{j=1}^n f_j(S_i)w_j$$

Weights are non-negative and normalized. The set of all possible combinations of weights is called *weight space* W and corresponds to an $(n-1)$ -simplex in geometry, i.e., a line for two criteria, a triangle for three (Fig. 2a, c), and a tetrahedron for four criteria (Fig. 2b).

$$W = \{\mathbf{w} \in \mathbb{R}^n : w_j \geq 0, \sum_{j=1}^n w_j = 1\}$$

We call the solution with the minimal overall cost for the current weighting \mathbf{w} *current solution*. The current solution may change with respect to changes of \mathbf{w} . WeightLifter analyzes the effect of changes to \mathbf{w} on changes of the current solution.

Some MCDM problems benefit from inspecting more than the top-ranking solution, especially when multiple solutions have similar overall costs. E.g., cars ranking high for large regions of W may be interesting even if they never rank first. For considering these potentially interesting alternatives, we denote a user-defined number of additional ranks as *rank context*, e.g. the second and third-ranking car. As for the current solution, the rank context may change on variations of \mathbf{w} .

4.2 Navigation of Two-way Trade-offs

As a starting point, consider a decision involving only two criteria, e.g., *Price* and *Mileage* of a car. In this case, a simple slider intuitively represents all possible trade-offs between these two criteria.

WeightLifter uses this familiar slider metaphor as a basis for visualizing trade-offs also for MCDM problems involving more than

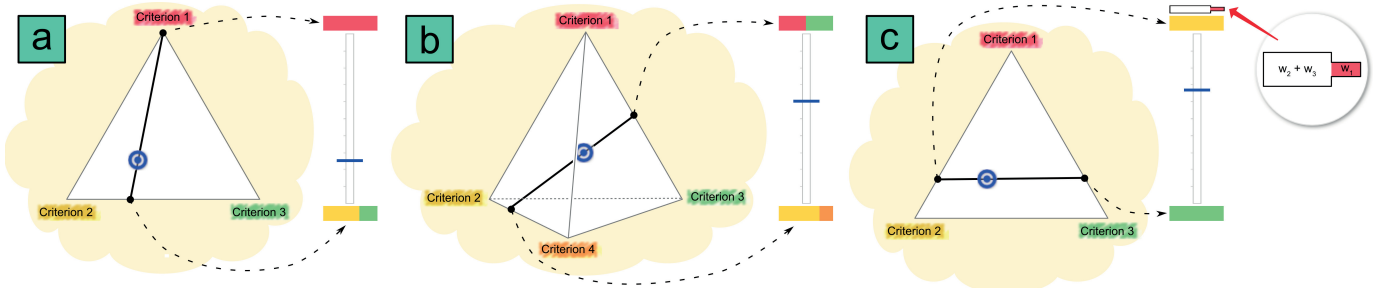


Fig. 2: Two-way trade-offs within the sketched weight space and the corresponding sliders of WeightLifter. (a) one criterion and the remaining two criteria, (b) two groupings of criteria, and (c) two criteria (with w_1 held constant at 25%). The blue dot indicates the current weighting.

two criteria. As inspired by HyperSlice [39], the general idea is to break down the multidimensional MCDM problem to a set of coordinated visual representations of low-dimensional trade-offs (Fig. 1). In particular, we use sliders to represent two-way trade-offs, i.e., trade-offs between two disjoint sets of criteria. Common cases of two-way trade-offs include trade-offs between two criteria (e.g., *Price* versus *Mileage*), between one criterion and all others (e.g., *Price* versus the combination of all other criteria), and between groups of related criteria (e.g., the cost-related criteria *Price* and *Tax* versus the wear-related criteria *Age* and *Mileage*).

Each position within a slider corresponds to a relative weighting of the criteria at its ends. The position of the current weighting is shown by a blue line crossing the sliders (see Fig. 1c). For example, a blue line crossing the center of the slider indicates that the sum of the criteria weights is equal for both ends. Dragging this line towards one end increases the current weights for the criteria at this end by the same amount by which the weights at the opposite end are reduced. The proportion of weights between the criteria at the same end is preserved. Furthermore, the weights of non-involved criteria are not affected. Each slider could be understood as representing a straight line through the weight space passing through the current weighting.

Fig. 2 illustrates particular sliders for three exemplary MCDM problems. Fig. 2a and Fig. 2c show cases with three criteria while Fig. 2b shows a case with four criteria. Each sub-figure sketches the entire weight space of the corresponding MCDM problem, and indicates one possible two-way trade-off in this space as well as its visual slider representation in WeightLifter. The slider in Fig. 2a has criterion 1 (red) at the top end, and the combination of criterion 2 (yellow) and criterion 3 (green) at the bottom. Dragging the blue line towards criterion 1 increases its current weight by reducing the weights of criteria 2 and 3, which preserve their proportion to each other. The slider in Fig. 2b has two criteria at either end. In both Fig. 2a and b, the sliders involve all criteria of the respective MCDM problems and thus represent 100% weight. In Fig. 2c, the slider represents the trade-off between criteria 2 and 3, but not criterion 1. In this case, dragging the blue line does not affect the weight of criterion 1. We call the sum of all weights represented by a slider its *Overall Trade-off Weight*.

Regarding the visual encoding, each slider shows the proportion of weights between the assigned criteria at either end as horizontal stacked bars. We employ hue to discriminate the criteria, which has proven a suitable visual variable for this purpose in prior work [16]. As illustrated in Fig. 2c, we also indicate sliders where the Overall Trade-off Weight is less than 100%. In this case, the length of a thick white bar above the slider indicates the Overall Trade-off Weight while thinner colored bars show weights that are not affected by that slider. This representation is important as it conveys the overall impact of sliders on the current weighting. We also experimented with encoding the Overall Trade-off Weight as the height or width of sliders, but found that these alternatives severely reduced the interpretability of sliders.

As the default configuration, WeightLifter starts with one vertical slider per criterion to represent the trade-off between that criterion and the combination of all other criteria. In Fig. 1, for example, the left-most slider represents the trade-off of *Age* vs. the remaining five criteria *Doors*, *Guarantee Period*, *Mileage*, *Price*, and *Tax*. The second slider represents *Doors* vs. all other five criteria, etc. As an advantage

of such one-vs.-all-other sliders, the position of the current weighting directly represents the weight of the single criterion. For example, *Price* has the highest weight of all criteria in Fig. 1. This configuration of all sliders could be understood as a mixing desk metaphor.

The number of potentially interesting trade-offs increases rapidly with the dimensionality of the MCDM problem. The user can add sliders on demand which are appended to the right (before the triangle, discussed in Sec. 4.3). For example, the user could be interested in trade-offs between particular pairs of criteria, as shown in Fig. 2c. New sliders can be configured to represent any two-way trade-off by dragging criteria from other sliders or the color legend to areas at the top or bottom of the slider. This way, the user may define combinations of criteria at the end points of sliders in an ad-hoc fashion. In general, the user can add criteria to or remove criteria from any slider any time, and may also remove sliders entirely.

Sliders showing the same set of criteria are considered to semantically belong together. In particular, this applies to all sliders of the default configuration, as each of them comprises all criteria. Related sliders that are placed adjacent to each other are drawn closer together and use a continuous blue line to display the current weighting. Fig. 6 shows an example with five connected and one disconnected slider.

4.3 Navigation of Three-way Trade-offs

It is often helpful to explicitly represent all weight combinations between three sets of criteria. Even for MCDM problems involving more than three criteria, three-way trade-offs may already cover the most important degrees of freedom, e.g., if some criteria are more important than others or if criteria can be grouped into semantically related sets. For this reason, WeightLifter optionally represents three-way trade-offs using barycentric coordinates in an equilateral triangle. Fig. 1e, for example, illustrates the three-way trade-off between the car *Price*, *Mileage*, and *Age*, which Jane considers as her most important criteria.

As far as possible, the visual encoding and handling of three-way trade-offs is consistent with those of two-way trade-offs. Specifically, at each vertex, the set of assigned criteria is represented by stacked bars which also supports drag and drop of criteria for configuration. Moreover, the Overall Trade-off Weight is shown as an additional stacked bar on top of the triangle if it is less than 100%.

As a potential problem, barycentric coordinates are far less well-known than sliders. To facilitate their understanding, we decided to visually encode the current weighting as the intersection of three lines (Fig. 1e). Each line extends from the position of the current weighting to the opposite side of its corresponding vertex, as the length of this part directly represents the relative weight of that vertex. Hue indicates the corresponding criteria of these lines. Users can change the current weighting by dragging the intersection point within the triangle.

4.4 Visualization of Rank Sensitivities

A key goal of WeightLifter is to convey how changes of the current weighting impact the decision (R2). For this reason, we highlight regions around the current weighting in blue in which the current solution does not change (see Fig. 3). In Fig. 1a, for example, increasing the weight of the criterion *Age* in the leftmost slider does not change which car model is considered the best choice. However, slightly increasing the weight of *Price* in the sixth slider or the triangle results in a switch of the optimal choice. In general, the distance of the current

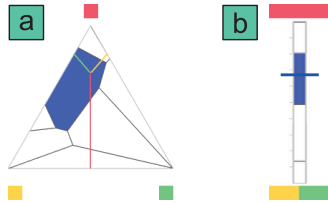


Fig. 3: The rank sensitivity for the current solution in (a) a three-way trade-off (b) a two-way trade-off.

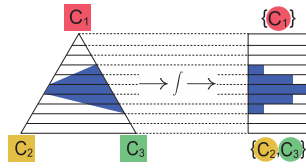


Fig. 4: Illustration of the computation and visualization of a global rank distribution along binned weights of C_1 .

weighting from the borders of the blue region represents the stability of the current solution regarding changes of weights. Situations in which the current weighting is far away from all borders indicate robust decisions while narrow bands reveal that the decision is highly sensitive to changes of weights.

In addition to the blue region around the current weighting, we compute all boundaries where the current solution switches and indicate them as black lines within sliders and within the triangle (see Fig. 3). These boundaries convey a segmentation across each trade-off with respect to the current weighting. For example, the boundaries reveal that the triangle in Fig. 3a can be segmented into seven regions, i.e., the blue region around the current solution plus six others. The trade-off along the slider in Fig. 3b intersects four regions. Each of these regions corresponds to a particular solution which is the best choice for any weighting inside the region. Each region could thus be seen as a visual representation of a solution in terms of the most favorable weights and may, for example, serve as a visual element for selecting that solution in weight space by clicking on it.

The number, size, and shape of the regions inside a particular slider or triangle change when the user modifies the current weighting outside that element. The reason for this is that each slider and the triangle represent a particular slice across the weight space that moves with changes of the current weighting. In Fig. 3a, the intersection point could be dragged rightwards to increase the weight of green at the cost of yellow. This moves the trade-off represented by the slider in Fig. 3b towards the right edge of the triangle, where it intersects only two regions.

While the segmentation is based on switches of the best solution, sliders and the triangle also highlight regions in light blue where the current solution is still within the rank context, i.e., it is ranking high, but not first. The extents of this rank context convey how fast the current solution drops out of decision-relevant ranks. This provides additional information about the stability of a decision, which may increase the effectiveness of sliders [15]. In the triangle in Fig. 1, for example, the rank context is narrower for increases of the weight of car *Price* (turquoise) than for significant decreases. We note that the boundaries of the rank context generally do not align with any boundaries between first-ranking solutions.

4.5 Visualization of Global Rank Distribution

The sensitivity visualization along trade-offs (Sec. 4.4) is insufficient for a global characterization of solutions (R3). As the main reason, each two-way or three-way trade-off represents only a small part of the weight space for problems involving more than three criteria. Therefore, the number of visible regions in sliders and the triangle is typically (much) smaller than the overall number of first ranking solutions within the entire weight space.

In order to support a global characterization of favorable weight space regions (R3), WeightLifter takes a different approach. We compute a segmentation of the entire weight space by sampling (see Sec. 7 for details). Each segment corresponds to all combinations of weights

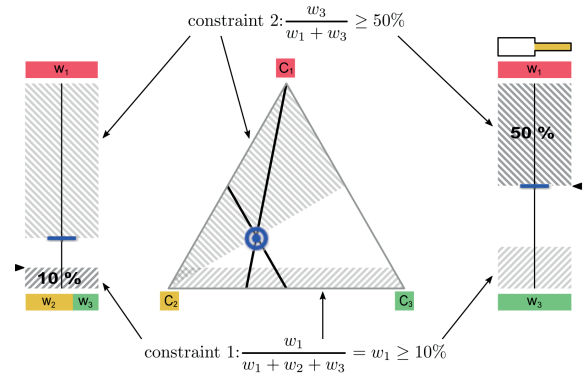


Fig. 5: Illustration of two constraints on different two-way trade-offs, each also implicitly constraining the other trade-off.

for which a particular solution is considered optimal, i.e., has minimal overall cost. For visualization, the idea is to project the segment of the current solution along each slider.

Fig. 4 illustrates a schematic example for a MCDM problem of three criteria. The barycentric triangle thus represents the entire weight space for this example. Assuming that the blue area corresponds to the segment of the current solution, the distribution of this segment can be visualized along two-way trade-offs as shown by the vertical histogram to the right. Each histogram bin corresponds to a particular part of the weight space. In Fig. 4, this sub-division is defined in equal steps along the trade-off between the red criterion and the combination of the green and yellow criteria. For example, the bottom-most bin in Fig. 4 comprises all combinations of weights, where the weight for the red criterion C_1 is very small. In general, each two-way trade-off can be used to define a sub-division of the weight space. WeightLifter displays corresponding histograms next to and aligned with each slider (see Fig. 1b). While Fig. 4 shows a rather coarse subdivision for illustration, Fig. 1 uses more fine-grained bins. We do not show information regarding global distribution for three-way trade-offs in order to avoid overloading the visualization.

Each blue histogram bar represents the proportion of the weight space of the corresponding bin where the current solution ranks first, i.e., the relative frequency within the bin. In particular, a full bar indicates that the current solution will not change regardless of the weights orthogonal to that trade-off. In Fig. 4, for example, the fifth bar from the bottom shows that the current solution will stay top-ranking within that bin for any proportion of weights between the yellow and the green criteria. Conversely, white bins indicate that the solution does not rank first for any weighting inside the corresponding intervals.

The vertical histograms support a global assessment of the distribution of the favorable region for a given solution in weight space. In addition to the distribution of the first rank, we optionally also show the distribution of the rank context in weight space, as illustrated in Fig. 1. In this figure, for example, the histogram for the second slider from the left shows that the current solution only ranks first if the weight for the criterion *Doors* (yellow) is within a narrow range of very small weights. The rank context shows that this solution does not even rank second or third for high weights for *Doors*.

4.6 Constraining the Weight Space

While the intuition of decision makers about preferences is often vague, it is rarely the case that they have no intuition at all. In terms of the weight space, not all combinations of weights correspond to meaningful preferences. Assuming that Jane is looking for a cheap car, for example, she will not be interested in cars that only rank first if the weight for *Price* is very low.

In order to enable filtering of decision-irrelevant weight combinations (R4), WeightLifter supports specifying *constraints* in the weight space. Upper or lower boundaries along two-way trade-offs can be interactively defined by dragging small triangles from the bottom or the top of the slider. Fig. 5 shows a schematic sketch of two types of constraints for a problem involving three criteria. On the left slider, the constraint excludes weightings where w_1 is smaller than 10%, or

equivalently, where the sum of w_2 and w_3 exceed 90%. It can be understood as an absolute lower limit on w_1 . This type of *absolute constraint* is defined in sliders which involve all criteria. In the triangle, its boundary is parallel to the bottom edge.

The constraint on the right slider in Fig. 5 defines that w_3 is greater or equal to w_1 . It could be understood as enforcing the proportion between subsets of criteria. This type of *proportional constraint* is defined in sliders which do not involve all criteria. In our example, it is independent of w_2 and its boundary extends through the vertex of C_2 in the weight space. In the guiding example, proportional constraints could help to express that the weight for *Price* should be larger than the weight for *Age*. A specific application of proportional constraints is ordering the criteria by relevance [36].

In general, WeightLifter supports constraints between criteria subsets A and B that can be expressed with respect to a percentage c , as:

$$\frac{\sum_{i=A_1}^{A_n} w_i}{\sum_{i=A_1}^{A_n} w_i + \sum_{i=B_1}^{B_n} w_i} \circ c$$

$\circ \in \{\geq, \leq, =\}$ $A, B \subset \{C_1, \dots, C_n\}$ $A \cap B = \emptyset$ $0\% \leq c \leq 100\%$

In case of absolute constraints, the denominator becomes 1 (i.e., 100% weight), as shown in Fig. 5. According to this definition, constraints represent regions of the weight space with linear boundaries. For visual representation, sliders and the triangle indicate parts inside constrained weight space regions as hatched areas (see Fig. 5 for a schematic sketch and Fig. 1d for an illustration of the real system). We visually distinguish between regions that are explicitly excluded by constraints defined on that slider (dark gray hatching), and regions that are implicitly impossible due to constraints along other trade-offs (light gray hatching). For example, both sliders in Fig. 5 represent straight lines in the triangle that intersect both constraints, i.e., the constraint specified on the own slider as well as the one specified on the other slider, respectively. Using this geometric interpretation, it becomes clear that the size of the constrained areas along sliders depends very much on the position of the trade-off within the weight space. For example, increasing w_3 (green) at the cost of w_2 (yellow) in Fig. 5 would shift both trade-off lines to the right within the triangle and thus reduce the size of the hatched areas on both sliders.

As a special type of constraint, the user can fix the current weighting on any slider by a dedicated lock button. In Fig. 6, for example, the weight for *Doors* in the leftmost slider is fixed to a low value, as Jane, at this point only investigating cars with 4 or 5 *Doors*, does not consider this criterion as decision-relevant anymore. Fixing weights can also be helpful if particular weights are externally given. Moreover, this effectively reduces the dimensionality of the weight space, which can be reasonable in case of a large number of criteria, e.g., for mitigating performance problems when sampling high-dimensional spaces (Sec. 7). In order to avoid that the navigable weight space is reduced to a single point for every trade-off, a fixed criterion is removed from the other trade-offs. The reduced overall trade-off weights are indicated on top of each trade-off (see Fig. 6).

In addition to their direct visual representation in sliders and the triangle, constraints affect the computation of all global characteristics of the weight space. For example, global rank distribution histograms, as described in Sec. 4.5, refer to the constrained rather than the entire weight space. In general, constraints enable to express a particular degree of fuzzyness regarding the user preferences, which may range from considering the entire weight space to looking at a single point.

5 SYSTEM INTEGRATION

As reflected by the requirements (Sec. 2), MCDM is a multi-faceted task. While WeightLifter addresses the requirements that concern the weight space (R1, R2, R3, R4, R7, R8), decision makers often prefer solutions (R9) and need to inspect and compare a small set of decision-relevant solutions in detail (R5, R6, and R7 regarding comparisons beyond weight-related aspects). To address all requirements, this section suggests integrating WeightLifter with other linked views on a conceptual level (see Fig. 6). Sec. 6 proposes a MCDM workflow that concretizes the interplay between the views. We found the subsequent views relevant in many contexts:

Ranked Solution Details: To provide precise values and to explain the effect of the current weighting on the ranking of the solutions (R6), our system offers a multi-column ranking visualization inspired by LineUp [16]. Textual columns show solution attributes, e.g., the name of the car model in Fig. 6a. An additional column shows the weighted sum of costs as a stacked bar chart (Fig. 6b) where each segment corresponds to the cost of one criterion while the overall bar length corresponds to the sum of all weighted costs. This encoding explicitly conveys for which criteria each solution deviates from the optimum, i.e., the smaller the bar, the better. Moreover, a column called *rank frequency* (Fig. 6c) shows the percentage of all possible weight combinations for which the particular solution ranks first (dark gray) or achieves ranks within the rank context (light gray). This information can be interpreted as a notion of importance. The solutions can be sorted by any column. Changes of the current weighting trigger an immediate re-sorting of the list. Our implementation also displays the current weighting as a stacked percentage plot above the ranked table (Fig. 6d), which indicates the cost function per criterion as a small icon that expands to a larger graph when being hovered by the mouse.

Criteria Value View: Decision makers often need to see the raw criteria values. We visualize this data in parallel coordinates which is a prevalent technique for visualizing MCDM problems [3]. All axes are scaled such that the current solution is a centric horizontal reference line [2].

Additional views may be necessary to show application-specific characteristics of solutions. Sec. 8.1 illustrates an example of adding a scatter plot for showing meta-information about the solutions.

For consistent integration, all views are linked in multiple ways:

Color coding and ordering of criteria: Criteria are consistently discriminated by hue for all views. The order of the criteria is consistent as well, e.g., for sliders in WeightLifter and parallel coordinate axes.

Current solution: All views highlight the current solution in blue and update on changes of the current solution.

Comparison solution: Hovering a solution in any view temporarily highlights it in purple in all views (see Fig. 6). This facilitates a pairwise comparison of solutions. In WeightLifter, each two-way trade-off adds a temporary slider to indicate the rank-sensitivity of the comparison solution, and extends the vertical histograms by showing the global rank distribution also for the comparison solution.

Set of decision-relevant solutions: In our system, a solution is considered decision-relevant if it is ranking first or within the rank-context for any combination of weights inside the constrained weight space and if it is not excluded by external filters. Such filters may be defined by interval brushes in the Criteria Value View (Fig. 6e) or by excluding individual solutions in the Ranked Solution Details. Filters thus support incorporating decision making strategies such as Elimination by Aspects [38] (R9). Non-decision-relevant solutions are shown as light gray context in the Criteria Value View. The Ranked Solution Details lists only decision-relevant solutions (R5) and summarizes the weighted sum of costs for all others as a histogram in a separate row at the bottom of the list (Fig. 6f).

6 WEIGHTSPACE-AWARE DECISION MAKING WORKFLOW

The process of MCDM involves multiple steps. In this section, we propose a workflow that coarsely distinguishes between an initial setup of the MCDM problem by constraining the solutions based on a-priori knowledge (Sec. 6.1), and an iterative exploration of the solutions for reducing the set of candidates (Sec. 6.2). The workflow builds upon the integration of WeightLifter with other views (Sec. 5) and is illustrated by our guiding car purchase example (see Fig. 6).

6.1 Setting Initial Constraints

A typical first step in decision making is to reduce the set of all solutions by setting filters on the criteria themselves (R9) and by narrowing the weight space to reflect realistic preferences (R4).

Constraining the Criteria: Jane starts by limiting the solutions to family suitable cars that have 4 or 5 *Doors* by brushing in the parallel

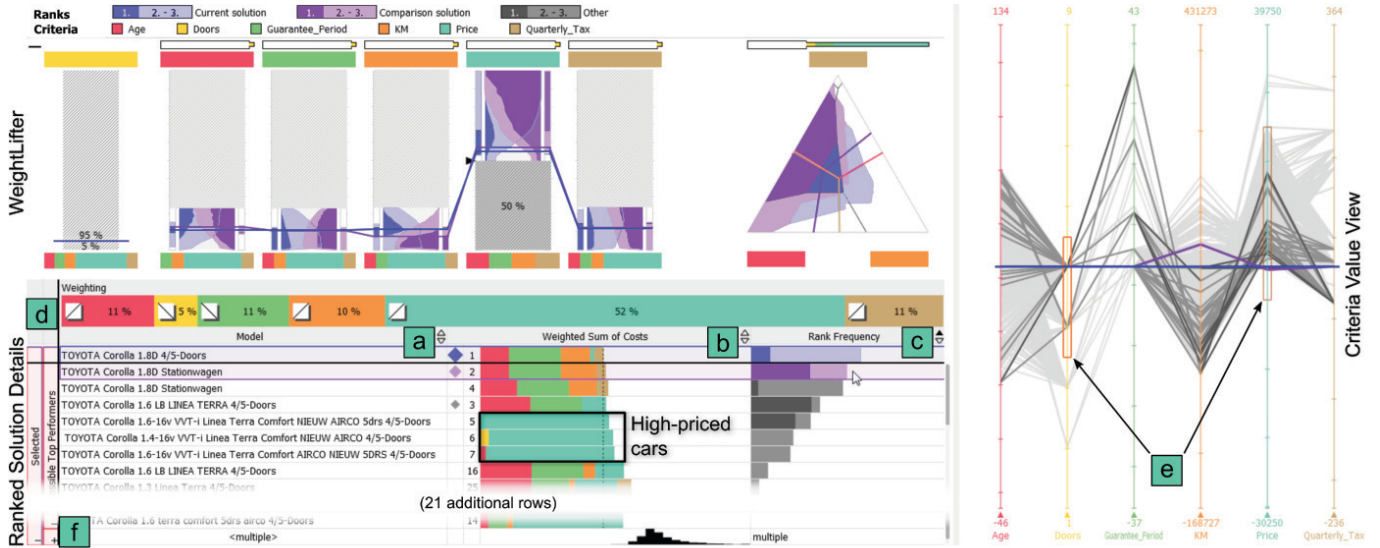


Fig. 6: Integration of WeightLifter with other views. The Ranked Solution Details involves multiple columns (a, b, c) and the current weighting (d) to list a reduced set of decision-relevant solutions. Solutions can be filtered, e.g., by brushes in a linked parallel coordinate plot (e). Excluded solutions are shown as context (f). All views highlight the current solution (blue) and a comparison solution (purple).

coordinate plot. Due to a limited budget, she likewise puts a hard limit of 25,000 EUR on *Price*. These filters reduce the set to 808 cars.

Constraining the Weight Space: Having restricted the number of *Doors*, Jane does not care much if her car has 4 or 5 *Doors* and fixes the corresponding weight to 5%. Despite the limit of 25,000 EUR, however, Jane still cares much about *Price* and reflects this by constraining the weight of *Price* to at least 50%. Jane also sets the rank context to include second and third-ranking cars. After these steps, 31 cars are considered decision-relevant.

6.2 Exploring and Reducing Candidate Solutions

Ordering by Global Rank Frequencies: In order to focus on the most probable candidates, Jane orders all cars in the Ranked Solution Details by their global rank frequencies (Fig. 6c). Interestingly, the car in the top row is much less frequently the first-ranking choice than the car in the second row, but is most often among the top-three cars and also has the smallest overall costs for the current weighting. While the seven cars with highest rank frequencies have similar overall costs, these costs originate from different criteria (Fig. 6b). The four cars in the top four rows are cheap yet old whereas the next three models incur their costs from their high *Price*.

Pair-wise Comparison of Solutions: Fig. 6 illustrates the comparison of the top-ranking two cars, i.e., the current solution (blue) and the comparison solution (purple) when Jane hovers the second row of the Ranked Solution Details. The local sensitivities and global rank distributions in WeightLifter make Jane aware of high sensitivities of the decision on even slight variations of her preferences. For example, the first rank of the current solution is highly sensitive to any variation in weights for *Age* (red) or *Mileage* (orange) as well as for increases of weight on *Guarantee Period* (green) or *Tax* (brown). The local sensitivity slider for *Price* shows that the two solutions will eventually switch their ranks when further increasing the weight on *Price*. The global rank distribution histograms reveal that the comparison solution is globally the best choice for very high weights on *Price* and is frequently top-ranking for very low weights on *Mileage*.

However, inspecting the underlying criteria values for *Price* and *Mileage* in the parallel coordinates shows that the difference in *Price* is very small while the difference in *Mileage* is more significant. Jane considers the significantly lower *Mileage* more important than the slightly lower *Price* and thus decides to manually reject the second-ranking comparison solution because the current solution seems the better option to her. After similar pair-wise comparisons, Jane ultimately narrows the set of candidates to three car models for which she arranges a test-drive.

7 IMPLEMENTATION

The WeightLifter technique has been implemented as part of *Visplore*, a system for visual exploration of multivariate datasets. *Visplore* is implemented in C++ and uses OpenGL for rendering. A multi-threading architecture [28] is used to maintain interactivity during computations.

For computing global rank frequencies and global rank distributions, we use an MCMC sampling of the weight space using the method of Tervonen et al. [36]. We chose this method because it explicitly considers user-defined constraints and slices of the weight space. This method efficiently samples within the constrained weight space rather than rejecting samples outside the constraints, which can be inefficient. For problems of up to six criteria, we compute the number of samples that would be needed for a regular grid with an Euclidean distance of 3% weight between samples. For an unconstrained weight space of five criteria, this corresponds to 66,045 samples which take 1.2 seconds to compute on a 3.2 GHz Haswell System. For six criteria, the number of samples increases to 501,942 which require 22.1 seconds. For even higher dimensional problems, we limit the number of samples to one million. For ten criteria, this typically still obtains an acceptable precision for global rank frequencies, but may introduce sampling artifacts in the form of global rank distribution spikes.

For sampling along two- and three-way trade-offs, we use an incremental sampling strategy to enable a more fine-grained sampling than for the global sampling. Starting from the extrema of the trade-off, we compare the rankings of the first and context ranks. If they differ, we split at the half-way point of each two-way trade-off. This recursive splitting continues until we achieve pixel-density sampling or the ranks are identical. This strategy exploits the convexity of the weight space. Any constraints on trade-offs define linear boundaries in the weight space (see Fig. 5) and preserve this convexity.

8 EVALUATION

To evaluate our approach on multiple levels [26], this section first describes a usage scenario of WeightLifter to a real-world problem in car engine design. We then report qualitative feedback collected during regular meetings with domain experts in automotive engineering over two years, including feedback from a one-month deployment and preliminary feedback from decision makers from other domains.

8.1 Usage Scenario: Powertrain Optimization

The high-level task of this usage scenario is the configuration of an automatic transmission. In this usage scenario, the parameter space comprises *Engine Speed* (abbr. *Speed*) and *Load Signal* (abbr. *Load*). *Speed* states the rotations of the crankshaft per minute and *Load* is the

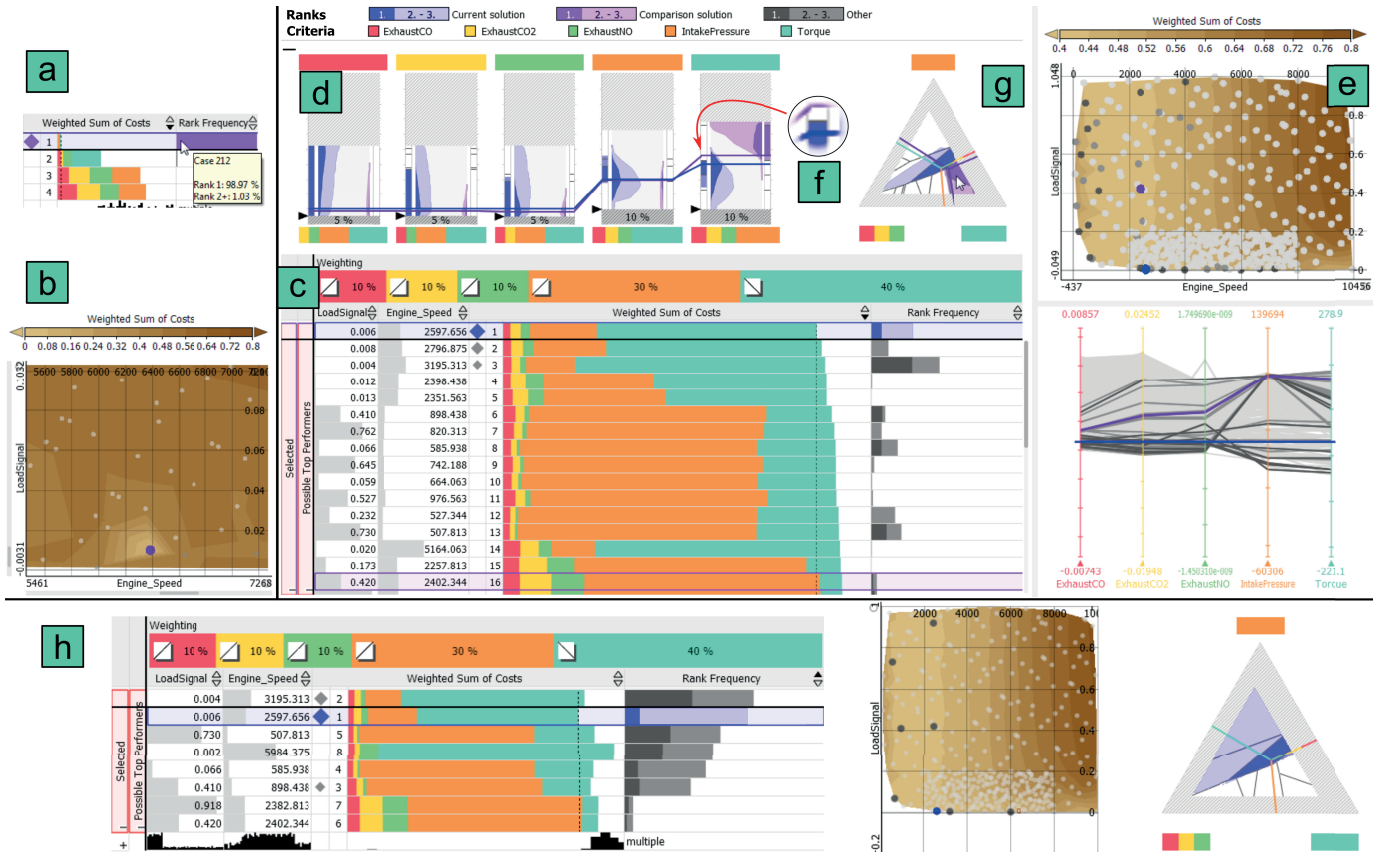


Fig. 7: A case study for powertrain optimization. A dominating solution (a) turns out as a non-converged simulation run (b). After setting an initial weighting (c) and weight space constraints (d), multiple parameter regions are revealed to rank first for various weight combinations (e). High sensitivities to the weights on *Pressure* and *Torque* (f) motivate grouping *Exhaust Emissions* for an examination as three-way trade-off (g). Reducing samples by importance and parameter space coverage, the engineer creates a representative set for future optimization (h).

percentage at which the gas pedal is pressed. An external tool provides simulation results for 400 sample positions in this parameter space. The optimization comprises five criteria: (1) maximize *Torque*, (2) minimize *Pressure* at the intake, and minimize the *Exhaust Emissions* for (3) *CO*, (4) *CO₂*, and (5) *NO*. The first goal of this scenario is to identify reasonable parameter regions which optimize the criteria. The second goal is to identify the weightings corresponding to a small set of representative solutions in these regions as input to an external gradient-based optimization.

For this application, an additional scatter plot shows *Speed vs Load* for all simulation runs. A brown background shows an interpolation of the overall cost v (defined in Sec. 4.1) as a function of *Speed* and *Load* for the current weighting. This scatter plot was part of the version evaluated by domain experts (see Sec. 8.2), who required this representation of the simulation parameter space to establish a mental model of the specific application problem.

After data import and problem setup, the Rank Frequency column reveals that a single solution ranks first for 99% of the weight space (Fig. 7a). However, the scatter plot reveals the point to be an outlier (Fig. 7b) and the parallel coordinates show that its *Exhaust Emissions* are precisely zero. The engineer considers it as a non-converged simulation run and excludes it manually.

The next step is to narrow down the search. Reflecting approximate intuition, the engineer defines initial weights as 40% *Torque*, 30% *Pressure*, and 10% for each of the *emission rates* (Fig. 7c). She also filters the weight space to ensure a minimum of 10% weight for *Torque* and *Pressure* as well as a minimum of 5% for each of the *emissions* (Fig. 7d). The rank context is set to include 2nd and 3rd ranking solutions, because they may be relevant for identifying interesting parameter space regions. In fact, the top-ranking solutions cover multiple parameter regions, mostly those with low *Speed* and almost zero *Load* (Fig. 7e). The current solution is also located at the edge of the

sampled parameter space.

The two-way trade-offs indicate high sensitivities of the current solution for lowering the weight of *Pressure* and increasing the weight of *Torque* (Fig. 7f). The engineer hovers solutions in the ranked list to assess the global distribution of ranks for the five criteria. This reveals that most solutions are globally more sensitive to changes in weights of *Torque* and *Pressure* while the sensitivities of the three *Exhaust Emissions* are lower and similar to each other.

Based on this information, the engineer wants to inspect the three-way trade-off between *Torque*, *Pressure*, and the group of *Exhaust Emissions*. Assigning the criteria to the vertices of the weight space (Fig. 7g). Hovering segments in the three-way trade-off efficiently relates weight regions to locations in the parameter space, and vice versa. An interesting finding, for example, is that slightly increasing the weight of *Torque* causes a sample in the inner part of the parameter space to rank first, as highlighted in purple. The ranked list and the parallel coordinates show that the compared solution significantly improves on *Torque* while being worse for all other criteria. Information like this guides the definition of additional simulation runs in corresponding parts of the parameter space as the first goal of this usage scenario.

For the second goal, the engineer orders the solutions in the list by rank frequency. She selects eight solutions based on their rank frequencies while ensuring that all top-ranking parts of the parameter space are represented by at least one sample. Filtering on this set updates the segmentation of the weight space (Fig. 7h). For each solution in this set, WeightLifter enables to identify a central weighting that is as far away as possible from any weight boundaries. The engineer confidently uses these weightings as input parameters of a subsequent automatized optimization.

8.2 Qualitative Feedback

Longitudinal feedback: Over two years, six domain experts in automotive engineering from a long-term partner company were involved in the design process of WeightLifter. Every two months, they provided feedback in joint workshops of approximately two hours each based on paper sketches and interactive prototypes.

A first result was a refined understanding of MCDM-related problems in this application context. It is a core task of the engineers to identify regions in a parameter space that simultaneously optimize multiple criteria. Numerical optimization tools are frequently used but require the specification of weights for the criteria. Previously, weights were guessed, which reduced confidence in the automatic optimization results. More recently, the domain experts started with a coarse sampling of the parameter space (see Sec. 8.1). While this helps to understand the sensitivities of the individual criteria, finding suitable weight factors was supported only indirectly and still incurred much effort. The engineers thus envisioned a tool for relating criteria weightings to parameter space regions.

Early in the process, we showed LineUp [16] to the engineers. While their feedback was positive overall, they wanted to make the dissatisfaction with criteria more explicit and advocated cost functions without an inherent upper bound. More importantly, the analysis of weight variations was considered inefficient.

Despite this input, however, it turned out that most engineers were initially not used to thinking in terms of a “weight space.” In their existing workflows, they solely relied on a point-wise exploration of weight spaces. The engineers agreed that an analysis of weight regions is very helpful. Still, it took them approx. three workshops and repeated explanation in the context of specific domain problems to develop a mental model of a weight space. While two-way trade-offs were considered simple to understand, barycentric coordinates were new to them and required time for familiarization. In this process, the linking with familiar views such as the parallel coordinates and the scatter plot of the simulation parameter space was crucial. Many comments during this phase referred to requests for inspecting precise values (e.g., for weights, parameters) and usability issues in previous prototypes, which were addressed for later versions.

Deployment: After incorporating the input, we deployed the system to four experts for final evaluation. After one month, we collected their feedback in separate interviews using the rose-bud-thorn method [22]. The engineers confirmed that WeightLifter significantly improves their intuition about weight spaces and it enables them to consider weight space regions simultaneously when identifying interesting parts of the parameter space. It also supports a systematic identification of weightings for subsequent optimization. Both aspects are considered significant gains of efficiency. Moreover, the new possibilities encouraged the engineers to include additional criteria in their decisions, which was considered a significant qualitative gain.

Most parts of WeightLifter were well-understood, i.e., the navigation in weight space based on trade-offs, and the representation of local weight-related sensitivities. Constraining the weight space was called the usual case, e.g., when optimizing for particular classes of cars. The engineers also emphasized that considering multiple top ranks (i.e., the rank context) is necessary, as solutions often have similar overall costs. The feedback to the global rank distribution was mixed. Two engineers were still unsure about their interpretation, while the other two considered them highly useful for identifying relevant subsets of criteria. As future improvements, the engineers wished to use WeightLifter to specify new simulation runs and to include quantitative indicators of weight-related sensitivities.

Other domains: Besides automotive engineering, we also collected preliminary qualitative feedback from decision makers in health policy making, energy transmission operation, and aviation infrastructure planning. We provided individual WeightLifter workshops of approximately one hour based on the guiding example. They all acknowledged the relevance of MCDM by Additive Weighting and confirmed that setting weights is a common problem. In energy operation, for example, the decision maker suggested to use WeightLifter for model

selection in data mining. He judged the three-way trade-off as a very suitable representation for deciding between model accuracy, model complexity, and training effort. In health policy making, it must be decided, e.g., whether the costs of certain medical treatments should be covered by the public health insurance. Such decisions involve stakeholders with conflicting priorities and are based on indicators about the population that are often afflicted by uncertainties. In this case, WeightLifter was found suitable to partly compensate for data uncertainty by showing weight-related sensitivities. Overall, all three decision makers considered WeightLifter as helpful for communicating decisions and mediating between stakeholders by making decisions more transparent.

9 DISCUSSION AND FUTURE WORK

WeightLifter makes Additive Weighting more easily accessible as a MCDM strategy. Additive Weighting is consistent and compensatory, i.e., a high value in one criterion can make up for a low value in another criterion. However, Additive Weighting demands much mental processing from the user to determine the trade-off weights [27]. The visualization of the weight space allows the decision maker to understand the effects of weights and reduces the mental overhead.

Concerning scalability, we found that WeightLifter is reasonably applicable to MCDM problems of up to 10 criteria. In addition to the number of discriminable colors [41] and available screen space, an important factor limiting the number of criteria is the challenge to globally sample a high-dimensional weight space in sufficient density (Sec. 7). For higher-dimensional MCDM problems, a prior analysis may indicate correlations between criteria and thus allow us to pre-filter criteria. Alternatively, the dimensionality may also be reduced by fixing the weights for particular criteria. Focusing only on high-ranking solutions usually reduces the decision candidates to a manageable set even in the case of thousands of potential solutions.

While WeightLifter focuses on the effect of weights on the solution, the cost functions also have an important impact. As for the weights, slight changes to cost functions may affect the ranking. We thus plan to extend the global analysis to incorporate variations in cost functions.

In the future, WeightLifter could track the interaction history to recall and communicate the decision making process. Furthermore, we plan to conduct a controlled user study for quantitatively comparing multiple MCDM systems (e.g., WeightLifter, LineUp [16], and Pareto Frontier visualizations) regarding particular decision-making tasks.

10 CONCLUSION

This paper introduced WeightLifter, a visualization technique for increasing the efficiency and transparency of Multi-Criteria Decision Making based on Additive Weighting. An automated analysis of weight spaces efficiently reduces an initial set of solutions to potential decision candidates. The dynamic definition of two- and three-way trade-offs enables a flexible analysis in terms of decision-relevant conflicts between criteria. Along each trade-off, a visual representation of the weight space conveys the stability of decisions to varying or uncertain preferences and characterizes solutions in terms of favorable preferences. We illustrated how WeightLifter can be linked to other views to support a dynamic ranking and comparison of solutions and we described a workflow for weight space-aware MCDM. WeightLifter received very positive feedback from users working in automotive engineering and is widely applicable to decision making problems in general contexts. We thus believe that WeightLifter can be beneficial in many application contexts by reducing uncertainty in the decision making process.

11 ACKNOWLEDGEMENTS

This work has been supported by the Austrian Funding Agency (FFG) within the scope of the COMET K1 program and the FFG-funded project 843550 (DEXHELPP). Thanks go to all project participants of AVL List GmbH, and to O. Rafelsberger, T. Mühlbacher, and C. Arbesser for valuable comments.

REFERENCES

- [1] S. Afzal, R. Maciejewski, and D. Ebert. Visual analytics decision support environment for epidemic modeling and response evaluation. In *IEEE Conference on Visual Analytics in Science and Technology (VAST)*, pages 191–200, 2011.
- [2] G. Andrienko and N. Andrienko. Constructing parallel coordinates plot for problem solving. In *Symposium on Smart Graphics*, pages 9–14, 2001.
- [3] M. Bagajewicz and E. Cabrera. Pareto Optimal Solutions Visualization Techniques for Multiobjective Design and Upgrade of Instrumentation Networks. *Industrial & Engineering Chemistry Research*, 42(21):5195–5203, 2003.
- [4] W. Berger, H. Piringer, P. Filzmoser, and E. Grllr. Uncertainty-Aware Exploration of Continuous Parameter Spaces Using Multivariate Prediction. *Computer Graphics Forum*, 30(3):911 – 920, 2011.
- [5] S. Bergner, M. Sedlmair, T. Moller, S. N. Abdolyousefi, and A. Saad. ParaGlide: Interactive Parameter Space Partitioning for Computer Simulations. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1499–1512, 2013.
- [6] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. *Computer Graphics Forum*, 31(3):1235–1244, 2012.
- [7] J. Brans and P. Vincke. A preference ranking organisation method: The PROMETHEE method for MCDM. *Management Science*, 31(6):647656, 1985.
- [8] S. Bruckner and T. Möller. Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.
- [9] J. Butler, J. Jia, and J. Dyer. Simulation techniques for the sensitivity analysis of multi-criteria decision models. *European Journal of Operational Research*, 103(3):531–546, 1997.
- [10] G. Carenini and J. Lloyd. ValueCharts: analyzing linear models expressing preferences and evaluations. In *Proceedings of the working conference on Advanced visual interfaces, AVI '04*, pages 150–157. ACM, 2004.
- [11] Y.-H. Chan, C. D. Correa, and K.-L. Ma. Flow-based scatterplots for sensitivity analysis. In *IEEE Conference on Visual Analytics in Science and Technology (VAST)*, pages 43–50. IEEE, 2010.
- [12] S. Chen, D. Amid, O. M. Shir, L. Limonad, D. Boaz, A. Anaby-Tavor, and T. Schreck. Self-organizing maps for multi-objective pareto frontiers. In *IEEE Pacific Visualization Symposium*, pages 153–160. IEEE, 2013.
- [13] D. Coffey, C.-L. Lin, A. Erdman, and D. Keefe. Design by Dragging: An Interface for Creative Forward and Inverse Design with Simulation Ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2783–2791, 2013.
- [14] C. di Sciascio, V. Sabol, and E. Veas. uRank: Visual analytics approach for search result exploration. In *IEEE Conference on Visual Analytics in Science and Technology (VAST)*, pages 217–218. IEEE, 2015.
- [15] S. G. Eick. Data visualization sliders. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, pages 119–120. ACM Press, 1994.
- [16] 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 (InfoVis '13)*, 19(12):2277–2286, 2013.
- [17] J. Horn et al. Multicriterion decision making. *Handbook of evolutionary computation*, 1:F1, 1997.
- [18] Z. Konyha, K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive Visual Analysis of Families of Function Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1373–1385, 2006.
- [19] P. Korhonen and J. Wallenius. Visualization in the Multiple Objective Decision-Making Framework. In J. Branke, K. Deb, K. Miettinen, and R. Sowski, editors, *Multiobjective Optimization*, volume 5252, pages 195–212. Springer Berlin Heidelberg, 2008.
- [20] M. M. Kksalan, J. Wallenius, and S. Zionts. *Multiple criteria decision making from early history to the 21st century*. World Scientific, 2011.
- [21] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. *Interactive Decision Maps*, volume 89 of *Applied Optimization*. Springer US, 2004.
- [22] LUMA Institute. Vision Statement: A Taxonomy of Innovation, 2014.
- [23] K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual steering-rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008.
- [24] C. A. Mattson, A. A. Mullur, and A. Messac. Smart Pareto filter: obtaining a minimal representation of multiobjective design space. *Engineering Optimization*, 36(6):721–740, 2004.
- [25] K. Miettinen and M. M. Mkel. Interactive multiobjective optimization system WWW-NIMBUS on the Internet. *Computers & Operations Research*, 27(7-8):709–723, 2000.
- [26] T. Munzner. A Nested Process Model for Visualization Design and Validation. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):921–928, 2009.
- [27] J. W. Payne, J. R. Bettman, and E. J. Johnson. *The Adaptive Decision Maker*. Cambridge University Press, 1 edition edition, 1993.
- [28] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.
- [29] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data. In *IEEE Conference on Data Mining Workshops*, pages 233–240, 2009.
- [30] A. Pretorius, M.-A. Bray, A. Carpenter, and R. Ruddle. Visualization of Parameter Space for Image Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2402–2411, 2011.
- [31] B. Roy. Classement et choix en prsence de points de vue multiples. *Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle*, 2(1):57–75, 1968.
- [32] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. John Wiley & Sons, Ltd, 2008.
- [33] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):2161–2170, 2014.
- [34] J. Sorger, T. Ortner, C. Luksch, M. Schwarzler, E. Grllr, and H. Piringer. LiteVis: Integrated Visualization for Simulation-Based Decision Support in Lighting Design. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):290–299, 2016.
- [35] T. Tervonen and J. R. Figueira. A survey on stochastic multicriteria acceptability analysis methods. *Journal of Multi-Criteria Decision Analysis*, 15(1-2):1–14, 2008.
- [36] T. Tervonen, G. van Valkenhoef, N. Batrk, and D. Postmus. Hit-And-Run enables efficient weight generation for simulation-based multiple criteria decision analysis. *European Journal of Operational Research*, 224(3):552–559, 2013.
- [37] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.
- [38] T. Torsney-Weir, M. Sedlmair, and T. Möller. Visualization for decision making under uncertainty. In *Workshop on Visualization for Decision Making under Uncertainty (VDMU)*, 2015.
- [39] J. J. van Wijk and R. van Liere. HyperSlice: Visualization of scalar functions of many variables. In *IEEE Conference on Visualization*, pages 119–125, 1993.
- [40] M. Velasquez and P. T. Hester. An analysis of multi-criteria decision making methods. *International Journal of Operations Research*, 10(2):56–66, 2013.
- [41] C. Ware. *Information Visualization: Perception for Design*. Interactive technologies. Morgan Kaufmann, third edition edition, 2013.