SoftwareX 35 (2026) 102920

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Modular SRF weight elicitation tool: Open-source software for tailored criteria weighting

River Huang [a,*], Miłosz Kadziński [b], José Rui Figueira [c], Peter Burgherr [a,d]

[a] Laboratory for Energy Systems Analysis, PSI Centers for Nuclear Engineering & Sciences and Energy & Environmental Sciences, Forschungsstrasse 111, Villigen PSI, 5232, Switzerland
[b] Institute of Computing Science, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland
[c] CEGIST, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal
[d] Department of Mechanical and Process Engineering, Chair of Energy Systems Analysis, ETH Zurich, Clausiusstrasse 33, Zurich, 8092, Switzerland

ARTICLE INFO

Keywords:
Multiple criteria decision aiding
Decision making under uncertainty
Criteria weighting
Deck-of-cards procedure

ABSTRACT

We present the Modular Simos-Roy-Figueira (SRF) Weight Elicitation Tool, an open-source, browser-based application for criteria-weight elicitation in Multiple Criteria Decision Aiding (MCDA). The state-of-the-art SRF software remains limited by fragmented support, obsolete implementations, or confinement to a single variant. In contrast, the proposed tool unifies multiple SRF-based procedures into a single maintained, reusable environment. We bring together classical SRF, SRF-II, Robust SRF, Assessment Through Prioritization (WAP), imprecise SRF, belief-degree imprecise SRF, and hesitant fuzzy linguistic SRF (HFL-SRF). A modular questionnaire assembles compatible procedural, informational, normative, and analytical components into the corresponding optimization model, while the interface provides deck-of-cards input, feasibility diagnostics, repair suggestions, example configurations, and exportable results. The application integrates a JavaScript frontend with a Flask backend and linear or mixed-integer models solved with open-source solvers. It returns normalized or representative weights, feasible weight ranges, robustness outputs, and graphical summaries. The software lowers the implementation barrier for advanced deck-of-cards weighting and supports transparent, reusable, and customizable preference modeling in research, teaching, and applied decision support.

Metadata

Code metadata.

Nr.	Code metadata description	Metadata
C1	Current code version	0.9.6
C2	Permanent link to code/repository used for this code version	https://github.com/cx7388/modular-srf
C3	Permanent link to reproducible capsule	https://github.com/cx7388/modular-srf/releases/tag/v0.9.6
C4	Legal code license	MIT License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python, JavaScript, HTML, CSS, Dockerfile; Flask, PuLP, HiGHS, NumPy, pandas, SciPy, scikit-learn, Plotly.js, XLSX.js
C7	Compilation requirements, operating environments & dependencies	Python 3.8+; Windows, macOS, or Linux; modern web browser; dependencies listed in `requirements.txt`
C8	If available, link to developer documentation/manual	https://github.com/cx7388/modular-srf/blob/main/USER_MANUAL.md
C9	Support email for questions	river.huang@psi.ch

* Corresponding author.
Email addresses: river.huang@psi.ch (R. Huang), milosz.kadzinski@cs.put.poznan.pl (M. Kadziński), figueira@tecnico.ulisboa.pt (J.R. Figueira), peter.burgherr@psi.ch (P. Burgherr).

https://doi.org/10.1016/j.softx.2026.102920
Received 9 June 2026; Received in revised form 26 July 2026; Accepted 3 August 2026
Available online 6 August 2026

1. Motivation and significance

In Multiple Criteria Decision Aiding (MCDA), weights express the relative importance assigned to criteria or the trade-offs accepted by the Decision Maker (DM) within a given aggregation model [9]. Their interpretation depends on the modeling framework in which they are embedded [2]. Since recommendations may alter substantially when weights change, weight elicitation remains one of the most consequential steps of a decision process.

A recent review identified a few dozen weighting methods [2], grouped into MCDA-inspired, monetary, data-driven, and distance-to-target families. Software support is also diverse, ranging from pairwise-comparison or scoring tools to broader decision-aiding environments [21]. HELDA, FITradeoff, MakeDecision.it, and MCDMaker support precise and imprecise weights, while AHP-OS, Apollo-Live, HELDA, and MAMCA support group weight collection. Open platforms such as *diviz* [22], the `MCDA` R package [6], and `pymcdm` [19,28] embed weighting procedures in broader workflows. Dedicated weighting libraries also exist; for example, `crispyn` implements objective and subjective weighting methods [3,4].

Within subjective weighting procedures, the Simos-Roy-Figueira (SRF) family is notable because it allows DMs to express relative importance through a deck-of-cards protocol rather than direct numerical assignment [13]. It has been applied in energy and environmental planning, engineering, project prioritization, infrastructure, organizational evaluation, public procurement, and healthcare-related decision support [29]. This relevance has stimulated robustness-oriented variants [29], alternative protocols such as SRF-II [1] and WAP [30], and a modular SRF framework [16].

Yet software support for SRF remains scattered. The original SRF program [13] and earlier *diviz* workflows [22] are no longer maintained. WAP was implemented in RAVI [30], and DecSpace includes SRF among broader decision-support methods [11,12]. These implementations are either variant-specific or embedded in larger platforms. Table 1 positions the proposed software relative to representative tools. General-purpose environments such as HELDA, Flexible and Interactive Tradeoff (FITradeoff), *diviz*, and `pymcdm` support broader MCDA workflows, but they do not provide a maintained, browser-based implementation of multiple SRF variants with modular model assembly and SRF-specific diagnostics. This creates a practical gap for analysts who need modern SRF variants without re-implementing optimization models or maintaining separate scripts.

To address this gap, we present the Modular SRF Weight Elicitation Tool, an open-source browser-based package for SRF-based criteria weighting, available at https://mcda-srf.psi.ch/. The software supports several SRF-related methods and a modular mode for configuring tailored procedures from compatible components. Building on the modular SRF framework [16], the tool transforms the framework's modeling logic into an executable workflow: questionnaire answers and card arrangements are translated into solvable formulations, supported by input validation, methodological guidance, graphical result views, repository examples, and import/export functions.

2. Software description

2.1. Software architecture

The Modular SRF Weight Elicitation Tool is a lightweight web application that separates user interaction, application management, model generation, and solver-based computation. The browser layer supports preference input and result inspection; the application layer handles validation, request handling, and data exchange; the model-generation layer translates the selected settings into a mathematical program; and the solver layer computes weights and variability analyses. Fig. 1 summarizes these components. This architecture supports predefined SRF-family methods and modular flat-criteria configurations within a common environment.

Technically, the browser sends the selected method, card arrangement, and method-specific inputs as a JSON request to the Flask endpoint. The backend normalizes the deck into rank groups, resolves modular questionnaire answers into component choices, and constructs the mathematical program with equality, rank-intensity, ratio, zero-reference, normalization, and optional normative constraints. Most configurations produce Linear Programming (LP) models; conditional logical gaps and inconsistency-restoration binaries require Mixed-Integer Linear Programming (MILP) models. Runtime JSON files store only the current calculation outputs, so no database is required.

2.2. Software functionalities

At a functional level, the software provides a unified environment for capturing card-based preference information, instantiating the corresponding optimization model, checking feasibility, computing weights, and returning outputs adapted to the selected elicitation setting. The current release supports eight user-facing configurations. The structural component is fixed to a flat criteria structure, while the remaining components are combined in different ways to reproduce established SRF-family methods or to define a tailored flat-criteria configuration in the sense of the modular SRF framework [16].

The predefined methods differ in both the elicitation protocol and the type of information they process. Standard SRF follows the classical deck-of-cards procedure with precise blank-card gaps and a precise global contrast ratio Z, returning a normalized weight vector [13]. SRF-II preserves the card-based logic but introduces a zero-importance reference parameter [1]. Robust SRF starts from the standard SRF setting but extends it with analytical tools for exploring the feasible region of weights and assessing their stability [29]. Assessment Through Prioritization (WAP) replaces the single global-ratio logic with local ratios between adjacent ranks [30]. The imprecise, belief-degree imprecise, and Hesitant Fuzzy Linguistic (HFL)-SRF configurations preserve the deck-of-cards spirit of the SRF family but alter the informational component by allowing interval-valued [10], belief-based [33], or hesitant fuzzy linguistic [31] representations of the elicited preference information. Finally, modular SRF exposes these design choices through a questionnaire, so that the user can assemble a supported configuration without manually formulating the underlying optimization model [16].

Table 1
Positioning of the modular SRF weight elicitation Tool relative to representative existing MCDA software.

Tool	Main orientation	SRF-specific support	Implication for the present contribution
HELDA	Web MCDA environment with elicitation facilities [21]	Not SRF-centered	Broader tool, not a dedicated multi-variant SRF implementation.
FITradeoff	Interactive tradeoff software [21]	No deck-of-cards SRF workflow	Uses a different elicitation paradigm.
diviz	MCDA workflow platform [22]	Historical SRF workflows; not maintained as a dedicated SRF tool	Shows prior SRF software interest, while a maintained focused tool is still needed.
`pymcdm`	Python MCDA library [19,28]	Code-level library; no SRF graphical interface	Less suited to guided elicitation with non-technical DMs.
Proposed tool	Dedicated browser-based SRF software	Multiple variants, modular configuration, diagnostics, robustness views, and import/export	Connects SRF advances to reusable applied software.

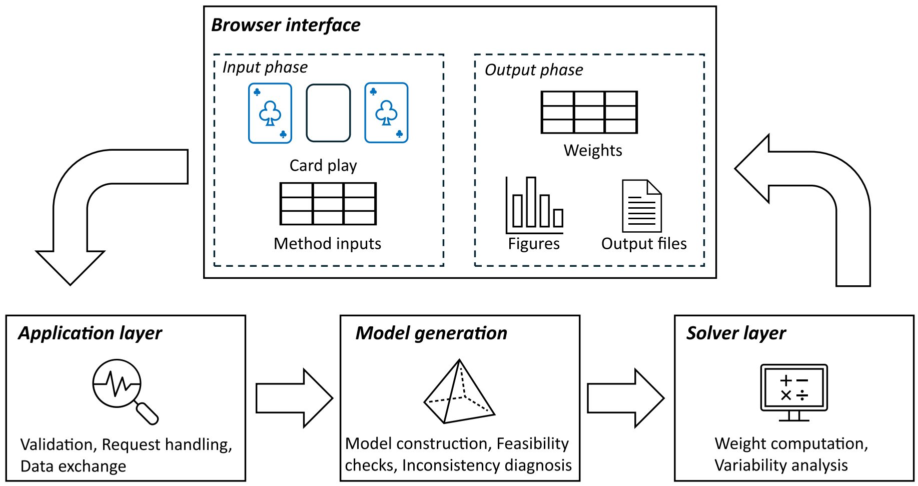


Fig. 1. Logical architecture of the modular SRF weight elicitation tool.

Table 2
Implemented SRF configurations as supported compositions of flat-criteria modular components.

Configuration	Preference information	Main outputs
SRF [13]	Criterion ordering, blank-card gaps, global contrast ratio z	Normalized weight vector
SRF-II [1]	SRF information plus a zero-reference parameter	Normalized weights with variability-oriented diagnostics*
Robust SRF [29]	SRF information with exploration of the feasible weight region	Normalized weights with variability-oriented diagnostics*
WAP [30]	Rank ordering with local ratio information between successive ranks	Normalized weights with variability-oriented diagnostics*
Imprecise SRF [10]	Interval-valued gaps and interval-valued global contrast information	Normalized weights with variability-oriented diagnostics*
Belief-degree imprecise SRF [33]	Belief distributions over admissible gap and ratio values	Normalized weights with variability-oriented diagnostics*
HFL-SRF [31]	Hesitant fuzzy linguistic evaluations for rank gaps and global contrast	Normalized weights with variability-oriented diagnostics*
Modular SRF [16]	Questionnaire-based selection of compatible flat-criteria components	Outputs determined by the selected configuration

* Variability-oriented diagnostics include normalized weights, minimum and maximum weights, Monte Carlo simulation, Average Stability Index (ASI), distribution views, and Principal component analysis (PCA) views.

Accordingly, the software does not treat the implemented methods as isolated algorithms but rather as distinct compositions of common modeling components. The procedural component determines the elicitation mechanism, such as standard SRF, SRF-II, or WAP. The informational component specifies how the DM's judgments are encoded, e.g., through precise inputs, interval-valued inputs, belief-degree distributions, or hesitant fuzzy linguistic terms, and may also include optional normative requirements such as anti-dictatorship or minimum-weight constraints. The analytical component determines the form of the requested outputs, ranging from a single normalized weight vector to variability-oriented summaries and visual diagnostics. Table 2 summarizes the implemented configurations, the corresponding preference information, and the outputs returned by the software.

Across these configurations, the software offers four main families of capabilities:

- *Preference capture and model instantiation*, starting from the card arrangement and configuration-specific inputs;
- *Feasibility checking and inconsistency handling*, including minimal consistency restoration suggestions when the elicited information is incompatible;
- *Weight computation and variability analysis*, ranging from a single normalized vector to representative weights, feasible bounds, Monte Carlo samples, ASI, and visual summaries such as distributions and PCA views;
- *Practical reuse features*, including eight downloadable example configurations, JSON configuration import/export, criteria-weight export in CSV, JSON, and XLSX formats, and optional normative constraints such as anti-dictatorship and minimum-weight requirements.

Result exports are designed for both direct reporting and downstream analysis. CSV files contain one row per criterion for tabular reuse, while the versioned JSON result payload records the criterion weights together with the selected method, decimal precision, ordered columns, units, criterion count, and available totals. XLSX workbooks include the displayed criteria-weight table and export metadata, plus sampling and extreme-scenario sheets when a variability-oriented calculation produces them.

2.3. Model generation, solver settings, and reproducibility

The model-generation workflow keeps predefined and modular configurations reproducible. Predefined method names and modular questionnaire answers are resolved into the same internal component representation, so equivalent configurations use the same model-building routines.

Optimization is performed through a PuLP-based free-solver layer [23]. HiGHS is used when available and CBC otherwise [17,26]; optional environment variables can select the backend, thread count, or time limit. The software uses the selected solver's default numerical tolerances. For stochastic variability analysis, the request stores the sample count, with a default of 200 and an interface cap of 20,000. A deterministic pseudo-random seed is derived from the effective inputs, model structure, constraints, and sample budget, making repeated sampling, ASI, and PCA outputs reproducible.

2.4. Input validation and inconsistency handling

Invalidity and inconsistency are handled at two distinct levels. First, the software checks whether the submitted information is sufficiently

Table 3
Representative validation and performance checks executed through the application calculation route. Timings are indicative single-machine runtimes and are reported to characterize computational scale rather than benchmark hardware performance.

Check	Configuration	Criteria	Sampling setting	Validation focus
Portfolio example used in this paper	Classical SRF	6	Not applicable	Reproduction of the predefined SRF workflow and deterministic weight calculation; below 0.1 s.
Portfolio example used in this paper	Modular zero-criterion with imprecise distances	6	2000 samples	Modular questionnaire resolution, feasible-region analysis, sampling outputs, and graphical payload generation; about 0.7 s.
Published original SRF example	Classical SRF	12	Not applicable	JSON import, card-layout reconstruction, and deterministic feasibility-based calculation; below 0.1 s.
Published Robust SRF example	Robust SRF	12	2000 samples	Min–max bounds, extreme-scenario analysis, sampling outputs, ASI, and export payload generation; about 0.4 s.
Published WAP example	WAP	5	2000 samples	Local-ratio constraint generation, feasible weights, and robustness diagnostics; about 0.3 s.
Published imprecise SRF example	Imprecise SRF	7	2000 samples	Interval-compatible constraint generation, feasible-region diagnostics, and output serialization; about 0.3–0.6 s.

complete and internally valid for constructing the mathematical formulation. The frontend and backend verify the minimum number of criteria, numeric validity, admissible domains, non-negativity requirements, probability inputs, and the ordering of interval bounds. Violations at this stage prevent model construction and are reported directly to the user, for example, “Invalid interval for rank gap after rank 2: $e_{\min} > e_{\max}$.”; “Invalid e_0 interval: $e_{\min,0} > e_{\max,0}$.”; “Minimum-weight requirement must be within $[0, 100]$.” These errors concern the validity of the input specification and therefore do not invoke the optimization-based inconsistency-restoration procedure.

Second, preference inconsistency arises when the individual inputs are valid but their joint interpretation does not admit a feasible weight vector. In this case, the backend constructs an inconsistency-restoration model that keeps the structural and normative requirements fixed, introduces relaxable indicators and residual variables for potentially conflicting judgments, links them through big-M constraints, and prioritizes first the number of modified judgments and then the magnitude of the required changes [15,16]. The resulting minimal-cardinality adjustment sets are translated into actionable recommendations concerning blank-card gaps, global or local ratios, interval bounds, and zero-reference values. Depending on the submitted configuration, the software may report messages such as “Increase blank cards between Rank 2 and Rank 3 from 1 to about 2.”; “Increase z ratio from 3.0 to about 4.0.”; “Increase e_0 value (SRF-II) from 0 to about 1.”; or “Decrease the upper bound of the local ratio between Rank 3 and Rank 2 from 4.0 to about 3.5.” The reported values are determined by the smallest identified restoration for the specific elicitation setting. If no actionable restoration can be generated, the original infeasibility message is returned.

2.5. *Testing, validation, and computational performance*

Testing and validation were performed through a combination of manual verification, executable example configurations, and route-level checks of the application workflow. The validation was designed to verify that imported JSON configurations are parsed correctly, card arrangements and method-specific parameters are translated into the intended mathematical formulation, feasible weights are computed consistently, variability outputs are generated when requested, and infeasible inputs trigger the appropriate diagnostic routines. The detailed portfolio example results are reported only in the illustrative example below; here, the focus is on the validation strategy and computational behavior of the software.

The executable checks cover both predefined and modular configurations, including classical SRF, modular SRF, robust SRF, WAP, and imprecise SRF cases. The repository examples were executed through the same Flask calculation route used by the graphical interface, rather than through isolated backend calls. This ensures that the tests cover the complete calculation pipeline from imported configuration to returned output payload. The checks confirm that the software can reproduce reference configurations, compute feasible weight vectors or feasible weight regions, generate variability-oriented diagnostics, and handle infeasible specifications through validation and restoration logic. Table 3 summarizes the representative checks and their computational scope.

Computational cost is driven mainly by the number of criteria, ranks, analytical outputs, and samples. Single-vector SRF runs require one small LP. Variability-oriented methods additionally solve $2n$ optimization problems for min–max bounds, may solve vertex-exploration LPs, and generate sample clouds for distribution and PCA views. Increasing the sample count, therefore, improves empirical plots but increases runtime approximately linearly during sampling.

Numerical and stochastic stability were assessed with the portfolio modular configuration under its dynamic-unit continuous formulation and its fixed-unit mixed-integer formulation. Solver comparisons used HiGHS 1.8.1 and CBC 2.10.3, the two supported free backends, and Gurobi 12.0 as an external reference. Feasibility, optimality, and integrality tolerances were set to 10^{-6} and 10^{-9} where applicable. Sampling sensitivity was evaluated at 200, 2000, and 20,000 draws, and seed sensitivity used seeds 101, 202, 303, 404, and 505. Differences were measured as the maximum absolute change over criteria in center weights and feasible bounds, expressed in percentage points, together with the change in ASI. Table 4 reports the results.

The solver and tolerance comparisons show no material numerical sensitivity for the tested formulations. The optimization-derived feasible bounds were independent of the sampling settings, whereas sampling-based center weights and ASI showed Monte Carlo variation. Increasing the budget reduced the observed between-seed center-weight range, and the 2000-draw setting used in the illustrative analysis limited the maximum range to 0.6 percentage points across the five tested seeds.

3. Illustrative example

We illustrate the software with a portfolio-evaluation problem involving eleven S&P 500 sector exchange-traded funds assessed on six criteria: Wealth, Volatility, Maximum Drawdown, Mean Squared Error of forecasts (MSE), Mean Absolute Error of forecasts (MAE), and Sign-Success Ratio of forecasts (SSR) [32]. The original study used standard SRF; here, the same case demonstrates both the predefined SRF configuration and a tailored modular configuration.

Fig. 2 shows the deck arrangement. Wealth is most important; Volatility and Maximum Drawdown are tied at the next rank, followed by SSR, while MSE and MAE are tied at the least important rank. Blank cards express differences in importance intensity.

Fig. 3 serves as a visual tutorial for the card-based elicitation workflow using the illustrated example. The user first selects the intended SRF configuration and checks the corresponding reference information in area 1. Area 2 then provides configuration-specific guidance, explaining the preference information required by the selected method. The elicitation itself starts from area 3, where criterion cards and blank cards are

Table 4
Stability assessment for the portfolio modular configuration. Gurobi is included only as an external reference; pp denotes percentage points.

Factor	Comparison	Observed stability
Solver, continuous	HiGHS, CBC, and Gurobi; 2000 draws	Center weights, feasible bounds, and ASI were identical across solvers.
Solver, mixed-integer	HiGHS, CBC, and Gurobi; 200 draws	Maximum unrounded center-weight difference 2.3×10^{-8} pp; reported centers, bounds, and ASI identical.
Solver tolerances	10^{-6} versus 10^{-9} for each solver and both formulations; 200 draws	Center weights, feasible bounds, and ASI were unchanged.
Sample count	200, 2000, and 20,000 draws with the input-derived seed	Relative to 20,000 draws, maximum center-weight differences were 0.9 pp at 200 and 0.1 pp at 2000; ASI was 0.924, 0.920, and 0.921, respectively. Bounds were unchanged.
Random seed	Five fixed seeds at each sample count	Maximum center-weight ranges decreased from 1.8 pp at 200 to 0.6 pp at 2000 and 0.2 pp at 20,000; ASI ranges were 0.918–0.924 and 0.919–0.921, while all five 20,000-draw runs gave 0.920.
Repeatability	Three repeated runs with unchanged effective inputs; 2000 draws	Center weights, feasible bounds, and ASI were identical across repetitions.
Infeasibility	Classical portfolio case with $z = 8$ and a 10% minimum criterion weight	All three solvers returned the same one-judgment minimal adjustment set: decrease z to approximately 3.

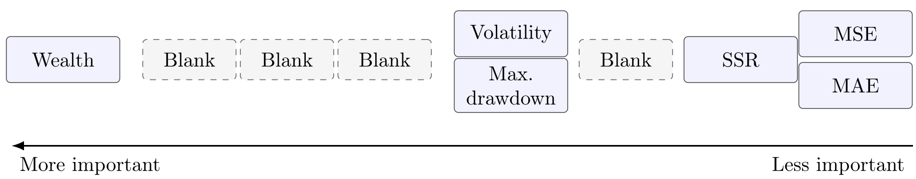


Fig. 2. Deck arrangement used for the illustrative portfolio example. Criteria appear from left (most important) to right (least important). Criteria in the same vertical block are judged ex aequo, and dashed grey blocks denote blank cards.

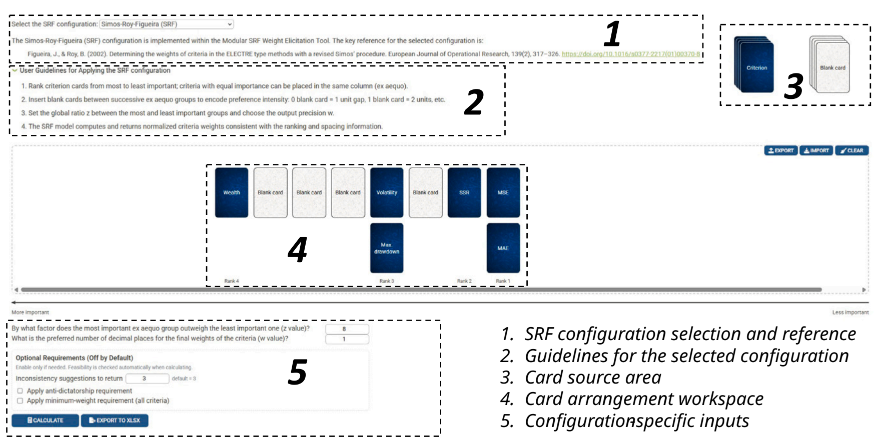


Fig. 3. Main interface of the modular SRF weight elicitation tool for the predefined SRF run of the example. The highlighted areas correspond to method selection and reference, methodological guidance, card source area, card arrangement workspace, and configuration-specific input fields.

available. The user drags these cards into the workspace in area 4: criterion cards placed in the same vertical group represent tied criteria, while blank cards inserted between successive groups indicate the perceived intensity gap between ranks. After the card arrangement is completed, the user provides the configuration-specific inputs in area 5, such as the global contrast ratio, zero-reference parameter, interval settings, sampling options, or additional constraints, depending on the selected method. The software then translates the completed card layout and input parameters into the corresponding calculation workflow and returns the expected outputs, such as normalized weights, feasible bounds, diagnostic messages, and variability visualizations when applicable.

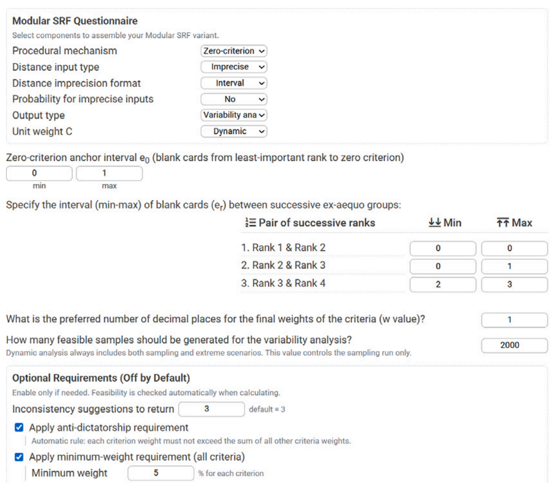
For the predefined run, the user selects standard SRF, arranges the cards as in Fig. 2, and provides $Z = 8$ and one decimal place for the final weights.

The software then returns the normalized weights in Fig. 4: 40% for Wealth, 20% for Volatility and Maximum Drawdown, 10% for SSR, and 5% for MSE and MAE. This reproduces the classical SRF interpretation of the deck.

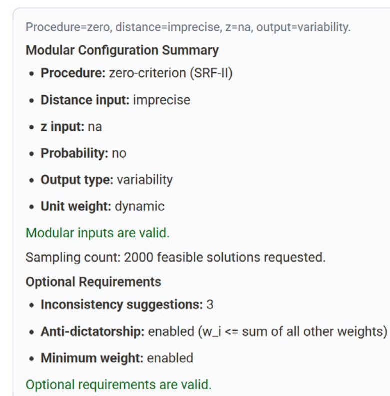
We then use the same deck arrangement to illustrate a tailored modular SRF configuration following the setting reported in [16]. In this run, the procedural mechanism is set to the zero-criterion formulation, the distance input type is set to imprecise, probability information for imprecise inputs is disabled, the requested output type is variability analysis, the unit weight remains dynamic, and sampling is selected as the variability method. The zero-criterion anchor interval is specified as $e_0 \in [0, 1]$, while the interval-valued blank-card distances between successive ranks are given by $[0, 0]$, $[0, 1]$, and $[2, 3]$. In addition, two optional normative requirements are activated: anti-dictatorship and a

Rank [r]	Criteria	Weights [%]
1	MAE	5.0
1	MSE	5.0
2	SSR	10.0
3	Max. drawdown	20.0
3	Volatility	20.0
4	Wealth	40.0
Sum	6	100.0

Fig. 4. Results obtained with the predefined SRF configuration.



(a) Questionnaire settings for the modular SRF configuration.



(b) Automatically generated summary of the selected modular configuration.

Average Stability Index (ASI) 0.9190 What is this?

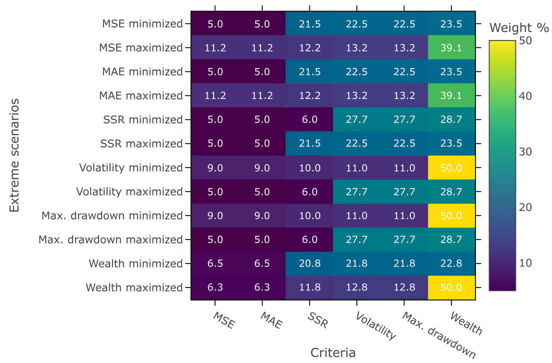
Rank [r]	Criteria	Center weight [k_center]	Min weight [k_min]	Max weight [k_max]
1	MAE	6.9	5.0	11.2
1	MSE	6.9	5.0	11.2
2	SSR	12.1	6.0	21.5
3	Max. drawdown	19.3	11.0	27.7
3	Volatility	19.3	11.0	27.7
4	Wealth	35.5	22.8	50.0
Sum	6	100.0		

(c) Representative weights and feasible bounds returned by the modular SRF configuration.

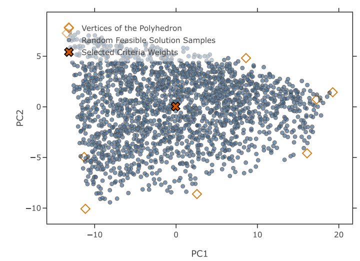
Fig. 5. Setup and results of the tailored modular SRF configuration.

minimum weight requirement of 5% for each criterion. Under this modular setting, the software returns not only a representative weight vector, but also information on the admissible variability of the weights obtained from two complementary analyses: an extreme-scenario analysis and a Monte Carlo sampling analysis with 2000 user-defined runs. The corresponding setup and summary of the selected modular composition are shown in Fig. 5(a) and (b). The resulting representative solution and criterion-wise feasible bounds are reported in Fig. 5(c). The center weights are 35.5% for Wealth, 19.3% for Volatility and Maximum Drawdown, 12.1% for SSR, and 6.9% for MSE and MAE. The corresponding feasible minimum and maximum values are also returned for each criterion, together with an ASI equal to 0.9190. Compared with the predefined SRF result, the modular configuration preserves the same overall ordering of criteria while providing a richer characterization of the admissible weights under imprecise distance information and additional normative constraints.

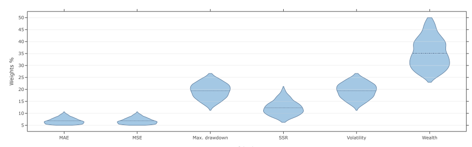
The variability analysis is reported through four complementary visual views in Fig. 6. The extreme-scenario analysis is summarized by the heatmap in Fig. 6(a). This view reports the weight vectors obtained when each criterion is alternately minimized and maximized over the feasible region. It shows, for example, that Wealth reaches 50.0% when it is maximized and decreases to 22.8% when it is minimized, while Volatility and



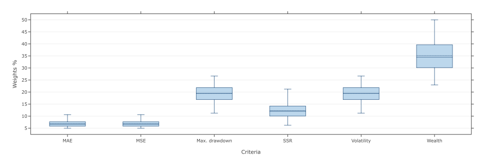
(a) Heatmap of criterion-wise extreme scenarios.



(b) PCA projection of sampled feasible solutions.



(c) Violin plot of sampled feasible weights.



(d) Box plot of sampled feasible weights.

Fig. 6. Variability-oriented visual outputs returned by the modular SRF configuration. Panel (a) summarizes the extreme-scenario analysis. Panels (b)–(d) report the sampling-based analysis through the PCA projection, violin plot, and box plot.

Maximum Drawdown vary between 11.0% and 27.7%. MSE and MAE remain within a narrower interval, from 5.0% to 11.2%. In this way, the heatmap provides a compact overview of the criterion-specific extreme scenarios.

The sampling-based analysis is visualized through the violin plot, box plot, and PCA view in Fig. 6(c), (d), and (b), respectively. The violin and box plots summarize the empirical distributions of the sampled feasible weights and show that Wealth remains the dominant criterion but also exhibits the widest admissible spread, whereas MSE and MAE remain the least important criteria with comparatively narrow distributions. The PCA projection provides a two-dimensional representation of the sampled feasible solutions together with the vertices of the feasible polyhedron and the selected representative solution. Taken together, these views show how the software extends a classical SRF elicitation into a richer modular analysis of admissible weight variability.

3.1. Examples provided in the repository

The public repository includes eight importable JSON examples stored as `srf-elicitation-config` snapshots. These cover the predefined and modular workflows used in this paper, plus Original SRF, SRF-II, Robust SRF, WAP, Imprecise SRF, and Belief-degree Imprecise SRF configurations.

All eight files are also available from an *Examples* menu beside the interface's Import and Export controls. Each selection downloads the original repository JSON file as an attachment through a Flask route restricted to the eight listed filenames. Users can then import, inspect, modify, and recalculate a configuration without locating files externally. This shortens the path to a first successful run and supports reproducibility, teaching, benchmarking, and methodological comparison.

4. Impact

Direct numerical weight assignment can be difficult for DMs, whereas the SRF deck-of-cards procedure expresses relative importance through ranks and blank-card gaps [13]. This interaction is relevant where weights act as importance coefficients [29], including ELECTRE [14] and PROMETHEE [5]. Published applications span environmental and infrastructure assessment [20], energy planning [24], healthcare facilities [8], cultural heritage [1,7], public procurement [25], retrofit decisions [18], and engineering [27]. These studies establish the cross-domain relevance of the underlying elicitation family; the present software provides a common implementation rather than requiring a separate model for each variant.

The practical contribution is supported by observable software coverage and verification results. First, one interface implements eight user-facing configurations spanning precise, robust, local-ratio, interval-valued, belief-degree, linguistic, and modular elicitation. Second, the application exposes eight downloadable configurations, and route-level checks covered standard, modular, robust, WAP, and imprecise workflows; the classical portfolio result was also reproduced by manual calculation. Third, deterministic sampling and downloadable source configurations support repeatable analysis. Finally, CSV and versioned JSON exports provide machine-readable criterion records for spreadsheets, scripts, and other decision-support workflows, while XLSX preserves richer variability tables.

The modular questionnaire further converts compatible procedural, informational, normative, and analytical choices into an optimization model [16]. Together with browser access, open-source solvers, validation messages, and inconsistency-repair suggestions, this reduces the need for analysts to implement and debug separate formulations. The evidence reported here demonstrates functional breadth, reproducibility, and interactive computational performance. It does not yet measure user adoption or decision quality across every application domain; those outcomes require field deployments and formal usability studies.

5. Conclusions

This paper presents the Modular SRF Weight Elicitation Tool, an open-source browser application that operationalizes eight predefined or modular configurations in a common deck-of-cards workflow. Its software contribution is the end-to-end translation of user judgments into linear or mixed-integer models, supported by validation, inconsistency-repair suggestions, variability analysis, graphical outputs, in-app example downloads, and criteria-weight export in CSV, JSON, and XLSX formats.

Correctness and reproducibility were examined through a hand-calculated portfolio case, executable published configurations, route-level checks across five workflow families, deterministic sampling, and representative runtime measurements. The results show that the tool can reproduce established weights while also exposing feasible ranges and uncertainty-oriented outputs that would otherwise require method-specific implementations. The eight downloadable configurations and machine-readable exports make these workflows easier to reproduce, adapt, teach, and connect to subsequent analyses.

5.1. Limitations and future development

The current implementation supports flat criteria only, is primarily designed for single-DM elicitation, and covers a selected subset of compatible flat-criteria components from the modular SRF framework. The reported checks establish correctness for representative configurations rather than exhaustive coverage of every admissible input combination. A formal continuous-integration test suite, large-scale and concurrent-use benchmarks, and structured usability studies would further strengthen evidence of reliability and practical use.

The software is maintained in a public, version-controlled repository that includes tagged releases and user documentation. Its layered architecture separates the interface, application logic, model generation, and solver integration, allowing individual components to be updated as software dependencies, browser technologies, or solver backends evolve without requiring a redesign of the entire application. Reliance on widely used open-source libraries further supports maintainability and portability. Future development will prioritize compatibility with updated dependencies, expansion of the supported SRF configurations, and computational improvements for larger variability-oriented analyses.

CRediT authorship contribution statement

River Huang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Miłosz Kadziński:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Conceptualization. **José Rui Figueira:** Writing – review & editing, Conceptualization. **Peter Burgherr:** Writing – review & editing, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work of River Huang and Peter Burgherr, published in this article, was carried out with the support of the Swiss Federal Office of Energy SFOE as part of the SWEET project SURE. The authors bear sole responsibility for the conclusions and results presented in this publication. Miłosz Kadziński was supported by the Polish Ministry of Science and Higher Education (grant no. 0311/SBAD). José Rui Figueira is financed by Portuguese funds through the FCT – Foundation for Science and Technology under project UID/97/2025 (CEGIST). We gratefully acknowledge Sultan Kobeyev for his early contributions to this work.

References

[1] Abastante F, Corrente S, Greco S, Lami IM, Mecca B. The introduction of the SRF-II method to compare hypothesis of adaptive reuse for an iconic historical building. Oper Res 2022;22(3):2397–436.
[2] Askham C, Arendt R, Bachmann TM, Dias LC, Amadei A, Scherer L, Qian H, Rupcic L, Steen B, Santos J, Motoshita M, do Carmo BBT, Bjørn A, Yokoi R, Laurent A, Sala S, Miebs G, Koffler C, Cinelli M. Review of weighting methods for life cycle impact assessment under GLAM. Int J Life Cycle Assess 2025;30(12):2691–724.
[3] Bączkiewicz A, Wątróbski J. Crispyn—a Python library for determining criteria significance with objective weighting methods. Softwarex 2022;19:101166.
[4] Bączkiewicz A, Wątróbski J, Sałabun W. Version [1.1]- [crispyn—a Python library for determining criteria significance with objective weighting methods]. Softwarex 2023;24:101541.
[5] Behzadian M, Kazemzadeh R, Albadvi A, Aghdasi M. PROMETHEE: a comprehensive literature review on methodologies and applications. Eur J Oper Res 2010;200(1):198–215.
[6] Bigaret S, Hodgett RE, Meyer P, Mironova T, Olteanu A-L. Supporting the multi-criteria decision aiding process: R and the MCDA package. EURO J Decis Process 2017;5(1–4):169–94.
[7] Bottero M, D'Alpaos C, Oppio A. Ranking of adaptive reuse strategies for abandoned industrial heritage in vulnerable contexts: a multiple criteria decision aiding approach. Sustainability 2019;11(3):785.
[8] Brambilla A, Lindahl G, Dell'Ovo M, Capolongo S. Validation of a multiple criteria tool for healthcare facilities quality evaluation. Facilities 2021;39(5/6):434–47.
[9] Cinelli M, Kadziński M, Gonzalez M, Słowiński R. How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy. Omega 2020;96:102261.
[10] Corrente S, Figueira JR, Greco S, Słowiński R. A robust ranking method extending ELECTRE III to hierarchy of interacting criteria, imprecise weights and stochastic analysis. Omega 2017;73:1–17.
[11] Costa AS, Figueira JR, Borbinha J. A multiple criteria socio-technical approach for the Portuguese army special forces recruitment. 4OR 2022;20(2):289–331.
[12] Costa AS, Rodrigues R, Xiang A, Figueira JR, Borbinha J. Supporting the use of decision aiding methods by non-specialists. In: International symposium on business modeling and software design. Springer; 2019. p. 81–94.
[13] Figueira J, Roy B. Determining the weights of criteria in the ELECTRE type methods with a revised simos' procedure. Eur J Oper Res 2002;139(2):317–26.
[14] Govindan K, Jepsen MB. ELECTRE: a comprehensive literature review on methodologies and applications. Eur J Oper Res 2016;250(1):1–29.
[15] Huang R, Corrente S, Siskos E, Burgherr P. The multi-level cards method: weights elicitation, consistency checking and restoration for hierarchical criteria. Omega 2025;103438.
[16] Huang R, Kadziński M, Figueira JR, Corrente S, Siskos E, Burgherr P. A modular Simos-Roy-Figueira framework for tailored weight elicitation in multi-criteria decision aiding. Expert Syst Appl 2026;311:131315.
[17] Huangfu Q, Hall JJ. Parallelizing the dual revised simplex method. Math Program Comput 2018;10(1):119–42.
[18] Kadziński M, Rocchi L, Miebs G, Grohmann D, Menconi ME, Paolotti L. Multiple criteria assessment of insulating materials with a group decision framework incorporating outranking preference model and characteristic class profiles. Group Decis Negot 2018;27(1):33–59.
[19] Kizielewicz B, Shekhovtsov A, Sałabun W. pymcdm—the universal library for solving multi-criteria decision-making problems. Softwarex 2023;22:101368.
[20] Marzouk MM, Nouh A, El-Said M. Developing green bridge rating system using simos' procedure. HBRC J 2014;10(2):176–82.
[21] Mesa Estrada LS, Haase M, Baumann M, Cinelli M. Decision support for energy system transformation – a systematic analysis of MADM software for sustainability assessment. Energy Strategy Rev 2026;63:102016.
[22] Meyer P, Bigaret S. diviz: a software for modeling, processing and sharing algorithmic workflows in MCDA. Intell Decis Technol 2012;6(4):283–96.
[23] Mitchell S, OSullivan M, Dunning I. Pulp: a linear programming toolkit for python. The University of Auckland, Auckland, New Zealand, 65(25) 2011.
[24] Neofytou H, Sarafidis Y, Gkonis N, Mirasgedis S, Askounis D. Energy efficiency contribution to sustainable development: a multicriteria approach in Greece. Energy Sources B: Econ Plan Policy 2020;15:572–604.
[25] Pictet J, Bollinger D. Extended use of the cards procedure as a simple elicitation technique for MAVT. application to public procurement in Switzerland. Eur J Oper Res 2008;185(3):1300–7.
[26] Saltzman MJ. Coin-or: an open-source library for optimization. In: Programming languages and systems in computational economics and finance. Springer; 2002. p. 3–32.
[27] Shanian A, Milani AS, Carson C, Abeyaratne RC. A new application of ELECTRE III and revised simos' procedure for group material selection under weighting uncertainty. Knowl-Based Syst 2008;21(7):709–20.
[28] Shekhovtsov A, Kizielewicz B, Sałabun W. Version - [pymcdm – the universal library for solving multi-criteria decision-making problems]. Softwarex 2025;24:102051.
[29] Siskos E, Tsotsolas N. Elicitation of criteria importance weights through the simos method: a robustness concern. Eur J Oper Res 2015;246(2):543–53.
[30] Tsotsolas N, Spyridakos A, Siskos E, Salmon I. Criteria weights assessment through prioritizations (WAP) using linear programming techniques and visualizations. Oper Res 2019;19(1):135–50.
[31] Wu H, Ren P, Xu Z. Promoting the physician–patient consensus with a hesitant fuzzy linguistic consensus method based on betweenness relation. Appl Soft Comput 2022;124:108979.
[32] Xidonas P, Thomakos D, Samitas A. On the integration of multiple criteria decision aiding and forecasting: does it create value in portfolio selection? Eur J Oper Res 2025;321(2):516–28.
[33] Zhang Z, Liao H. An evidential reasoning-based stochastic multi-attribute acceptability analysis method for uncertain and heterogeneous multi-attribute reverse auction. J Oper Res Soc 2023;74(1):239–57.