MCEWR

Multi-Criteria Evaluation Model for Water Resources

Web User Guide

The reference of the website at mcewr.m-rami.org: every page, every control of the evaluator, every setting, message and limit

Model developed by Mohamed Rami Mahmoud, 1995-2026

Prof. Mohamed Rami Mahmoud, Ph.D., P.E.

Research Professor Emeritus, National Water Research Center, Cairo, Egypt

rami@m-rami.org · www.m-rami.org · rami.mrami.org

ORCID: 0000-0002-3393-987X · LinkedIn: linkedin.com/in/mohamed-rami-64947570

WAM · PROMETHEE II · CP · ELECTRE II · AHP · EV · RA · TOPSIS · VIKOR · ELECTRE III

Edition 2.1.0 - September 2026

Companion documents: MCEWR_Methods_Explained.pdf (the methods taught with solved examples), MCEWR_Web_Tutorial.pdf (the evaluator step by step, with screenshots and expected results), MCEWR_Quick_Start.pdf, MCEWR_Method_Reference.pdf (the mathematics of the ten methods and the analyses).

Contents

1. Introduction 1
2. The pages of the site 2
3. The evaluator: layout and status line 3
4. Defining the problem: steps 1 to 4 5
4.1 Step 1 Project: opening a problem 5
4.2 Step 2 Alternatives, step 3 Criteria 5
4.3 Step 4 Decision matrix 6
4.4 Saving 6
5. Running an evaluation: steps 6 and 7 8
5.1 Method 8
5.2 Standardization 9
5.3 Weights (chosen in step 5) 9
5.4 Method options 9
5.5 Text-book switches 10
5.6 Exclude 10
5.7 The results 11
6. Step 5 Weights: six ways to weight the criteria 13
6.1 The Best-Worst Method 13
6.2 The objective schemes 13
7. Step 8 Compare 15
8. Steps 9, 10 and 11: sensitivity, robustness and rank reversal 16
8.1 Step 9, part Weights: sensitivity of the ranking to the weights 16
8.2 Step 9, part Decision values: sensitivity to the data 16
8.3 Step 9, part Method parameters: sensitivity to the settings 17
8.4 Step 10 Robustness: Monte Carlo on the values and the weights 17
8.5 Step 11 Rank reversal 18
9. Reports, output files and input formats 20
9.1 Reports 20
9.2 Output files 20
9.3 Input formats 20
10. Messages, limits and privacy 22
10.1 Messages of the status line 22
10.2 Limits 22
10.3 Privacy 22
11. Frequently asked questions 24
12. Glossary 25

1. Introduction

This guide describes the website **https://mcewr.m-rami.org** – every page, every control of the evaluator, every setting and every message – for the people who use it: students, researchers and practitioners of multi-criteria evaluation in water resources. It is the reference; the *Web Tutorial* is the walk-through with screenshots and expected results, and the *Quick Start* is the summary. The mathematics of the methods is in the *Method Reference*; the methods are taught with solved examples in *Methods Explained*.

The website runs the verified MCEWR engine on the MCEWR server and shows the result in the browser. A desktop edition of the same engine exists for the author's own studies; the two give identical numbers on every example – the program's test suite proves it at every release. The desktop edition is not distributed through this site.

	Website	Desktop edition (the author's)
Installation	none – a browser	a program on the author's computers
Where the computation runs	on the MCEWR server; nothing stored	on your computer; files stay with you
Problem size	up to 300 alternatives × 120 criteria	no limit
Input	.MCE, .mcewr, .csv	the same, plus Excel workbooks and GIS rasters
Reports	PDF, HTML, text	the same, plus Word and Excel with charts
Methods and analyses	all ten methods; comparison, sensitivity of the weights, the values and the settings, Monte Carlo robustness, rank reversal; six weighting schemes including the Best-Worst Method	the same

Version 2.0.0 of the engine; the intro line of the Evaluate page shows the version the server is running, and the footer of every page says where the computation happens.

2. The pages of the site

Page	Address	Content
Home	`/`	what the model is, the ten methods with their status, the author and the publications, links to the pages below
Evaluate	`/app.html`	the evaluator (chapters 3-8); `/app.html?example=<file>` opens it with that example loaded
Worked examples	`/examples.html`	the 26 example files, each with the settings and the expected ranking computed by the engine, and a button *Open in the evaluator*
Methods	`/methods.html`	one entry per method: family, direction of preference, reference, status (audited or provisional), options
Downloads & documents	`/downloads.html`	the documents as PDF (the Quick Start, Methods Explained, the Web Tutorial, this guide and the Method Reference) and the example files

The *Worked examples* page is the fastest way to see what a method does: every example shows the exact settings and the ranking to expect; *Open in the evaluator* loads it so that the same numbers can be reproduced, then changed. The expected results on that page are recomputed by the engine each time the site is built, so they cannot disagree with the evaluator.

3. The evaluator: layout and status line

The Evaluate page is a guided workflow of **eleven steps**, listed in a sidebar on the left in three groups, with a toolbar at the top (open an example, a file or a new problem; the Save menu; the engine pill that says *ready* or *working*) and a **status line** at the bottom. One step is shown at a time; click a step in the sidebar or use the *Next* and *Previous* buttons at the foot of every step. A tick marks a step that is done; the problem, the settings and the results of every step stay in place when you switch.

Step	Purpose	Chapter
1 Project	open an example, a file or a new problem; name it, give it a title and notes; save	4
2 Alternatives	name, add, remove the alternatives; leave one out with its *used* switch	4
3 Criteria	directions, weights, units, limits and thresholds; add main and sub-criteria; remove	4
4 Decision matrix	the values of every alternative on every leaf criterion	4
5 Weights	given, equal, Best-Worst, entropy, CRITIC or MEREC weights; the table of the weights each gives	6
6 Method and settings	the method, the standardization, the options and switches; the readiness checks; run	5
7 Results	the best alternative, the ranking, the reports and the output files	5
8 Compare	run several methods with the same settings and compare the rankings	7
9 Sensitivity	one input at a time: the weights, the decision values, the settings of the method - where does the winner change?	8
10 Robustness	Monte Carlo: random uncertainty on the values and the weights; rank acceptability and winner stability	8
11 Rank reversal	does adding or removing an alternative change the order of the others?	8

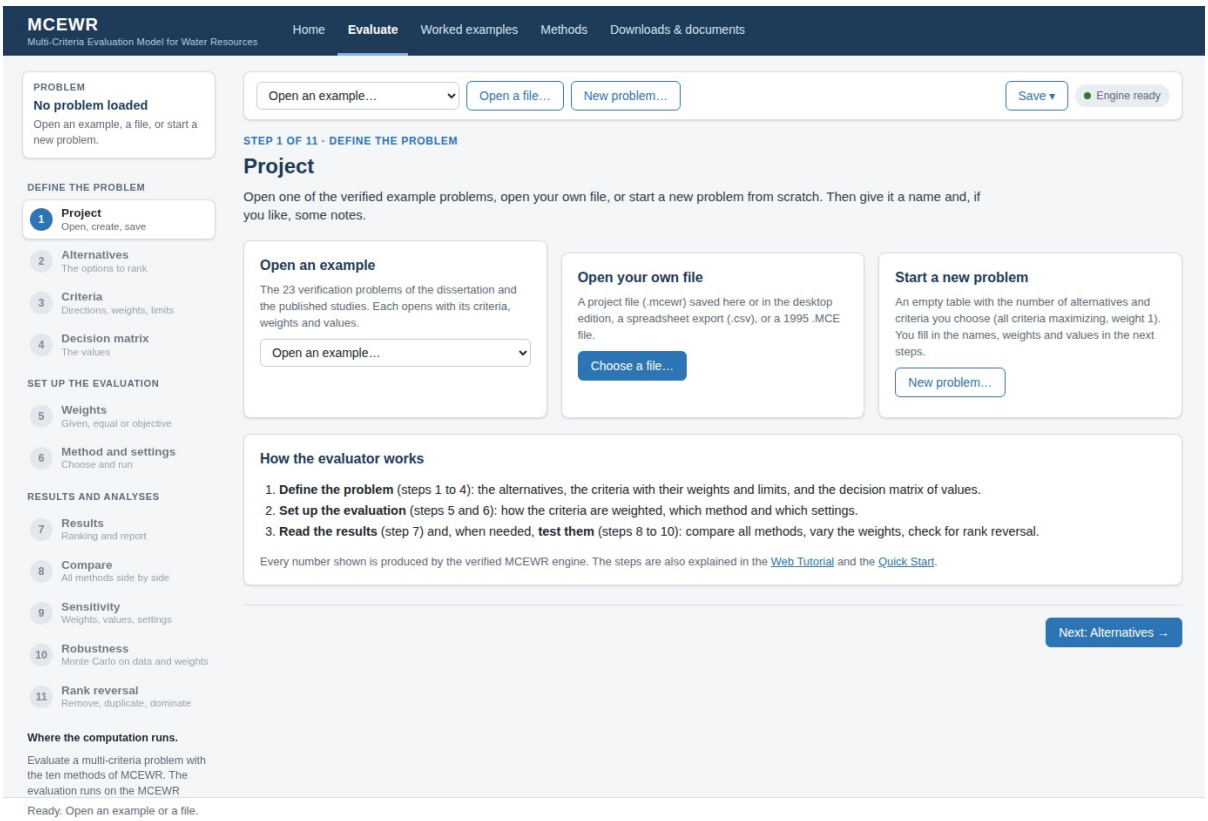


Figure 3.1 – The evaluator at start.

The status line reports what the page is doing and shows every message. Its normal states:

Status line	Meaning
Connecting to the MCEWR server…	the page is checking that the engine answers (a fraction of a second)
Ready.	waiting for you
Evaluating… / Comparing… / Sensitivity analysis… / Rank-reversal tests…	the server is computing; the buttons are disabled until it answers
a sentence in red	a refused edit, a method that could not run, a limit reached, or a connection problem – chapter 10 lists them

4. Defining the problem: steps 1 to 4

4.1 Step 1 Project: opening a problem

Control	What it does
Open an example (list)	loads one of the example files shipped with the model – the verification examples of the dissertation and of the published studies, each with a one-line note (Web Tutorial, chapter 4, gives their expected results); the three GIS control files are listed but need the desktop edition
Open a file	reads a file from your computer: `.MCE` (the 1995 format), `.mcewr` (project file, JSON) or `.csv` (three-block table, section 9.3). The file is sent to the server, read, and the problem comes back; the file itself is not kept
New problem…	asks for a name and the numbers of alternatives and criteria and creates an empty flat problem: all criteria maximizing, weight 1, values 0, PROMETHEE function 1
This problem (card)	the counts of the loaded problem (alternatives, main and leaf criteria, levels), the reader's warnings, and three fields kept in the project file: the *name* used for the saved files and the reports, a *title*, and *notes*
Save (menu, toolbar)	downloads the problem with every edit as `.mcewr`, `.csv` or `.MCE` (section 4.4)

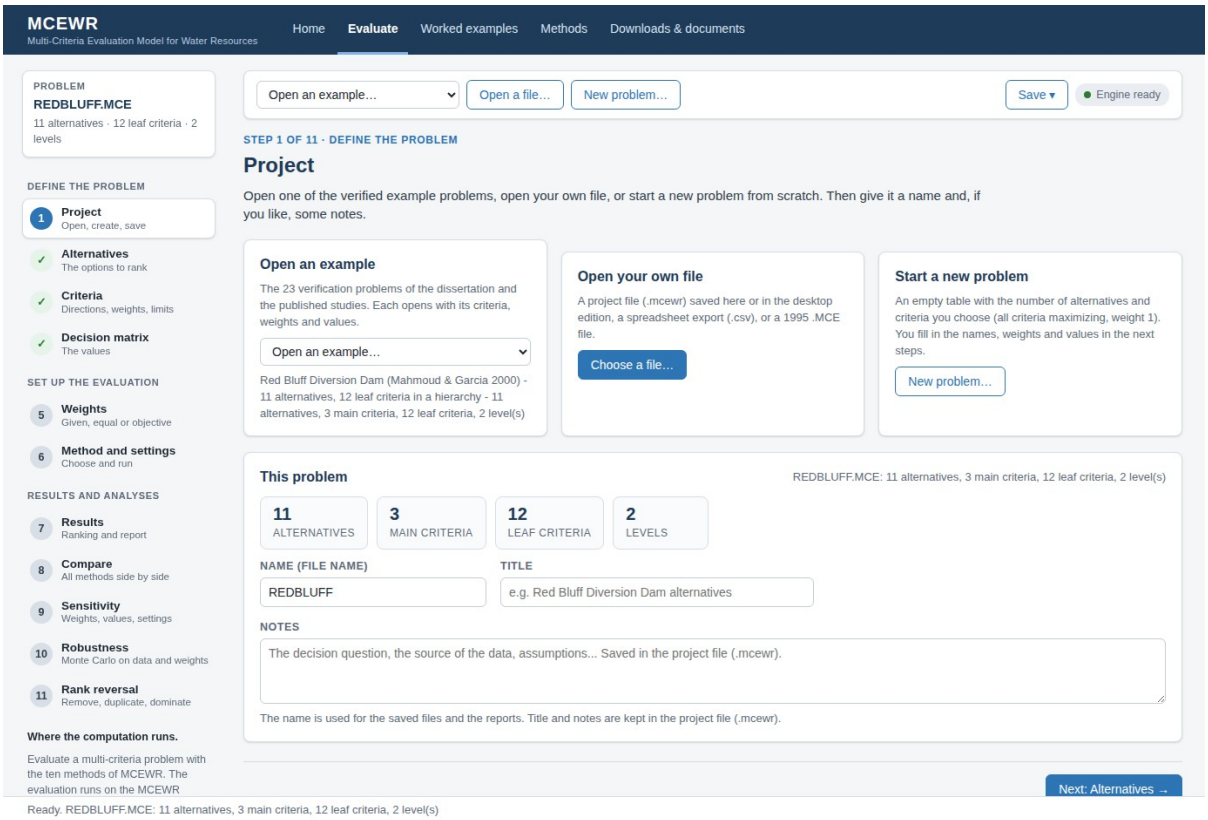


Figure 4.1 – Step 1 with REDBLUFF.MCE loaded.

4.2 Step 2 Alternatives, step 3 Criteria

Step 2 lists the alternatives with an input for the name, a *used* switch and a remove button; **+ Add an alternative** appends one with values 0. A problem needs at least two alternatives; a file whose values are Saaty comparisons of the alternatives cannot receive new ones on the page. Step 3 holds the criteria table below; **+ Add a main criterion** appends a top-level criterion, **+ sub** on a row adds a sub-criterion under it (a leaf becomes an aggregated criterion

and loses its own values), and ✕ removes a criterion with its sub-criteria. The switch **Show thresholds** reveals the PROMETHEE and ELECTRE III columns.

One row per criterion in the order of the file, sub-criteria after their parent. Yellow cells are editable.

Column	Meaning	Editable
id, parent	the identifier of the criterion and of its parent; a main criterion has no parent	no
name	the name shown in tables and reports	yes
type	`max` – the larger the better (a benefit) – or `min` – the smaller the better (a cost)	yes
weight	the declared weight; the model divides every group's weights by their sum unless the switch of step 5 is off	yes, not for Saaty groups
share	the weight's share of the group after normalization (computed)	no
upper, lower	the limit values: the ideal and anti-ideal of Compromise Programming with provided limits (and of SPOTIS), and the discordance scale of ELECTRE II	yes
units	the unit of the values, for the reports	yes
pf	the PROMETHEE preference function, 1 to 6 (usual, U-shape, V-shape, level, linear, Gaussian)	yes
p, q	the preference and indifference thresholds of PROMETHEE and ELECTRE III ($q \le p$)	yes
veto	the ELECTRE III veto threshold, 0 = none (kept by `.mcewr` and `.csv`, not by `.MCE`)	yes
used	`yes`, or `no` to leave the criterion out of the evaluation	yes

4.3 Step 4 Decision matrix

One row per alternative – id, name – and one column per leaf criterion, headed by its id, its direction (↑ or ↓), its name and its units. The values are the raw measurements in the criterion's unit; the standardization chosen in step 6 is applied at run time and never changes the table. AHP examples show *Saaty* where the values are pair-wise comparison matrices; those are read from the file and not edited on the page.

Editing (steps 2, 3 and 4). Click a yellow input cell, type, and press Enter (the cursor moves down), Tab (it moves right) or click elsewhere. The page sends the edit to the server, which checks it against the engine's rules: a value must be a number; a weight must not be negative; a Saaty weight cannot be edited. A refused edit is reported and the cell is marked red; an accepted one flashes green and is in force for the next run – there is no *apply* button. A block of numbers copied from a spreadsheet can be pasted into a cell of the decision matrix; it fills the cells to the right and below.

4.4 Saving

Format	Keeps	Use it for
`.mcewr` (project file, JSON)	everything: hierarchy, weights, limits, thresholds, vetoes, Saaty matrices, values, names, the *used* flags, the title and notes, the Best-Worst judgments and the settings of the last run	keeping your work; reopening it later with its settings; archiving a study
`.csv` (three blocks)	everything except Saaty matrices	editing in a spreadsheet program; sharing with people who do not use

Format	Keeps	Use it for
		MCEWR
`.MCE` (1995 format)	everything the 1995 program knew – not the ELECTRE III vetoes	compatibility with the original program and its published input files

Important. The website stores nothing. Closing the page loses the problem and the results unless they were downloaded. Downloads go where the browser puts them, usually the Downloads folder.

5. Running an evaluation: steps 6 and 7

Step 6 holds the settings and the *Run the evaluation* button; step 7 shows the results. A run takes a fraction of a second on the server. The card **Ready to run** of step 6 lists what the run will use and flags what does not fit the chosen method – limits that are all zero for Compromise Programming with provided limits, thresholds in data units while the values are standardized, a Best-Worst scheme without judgments.

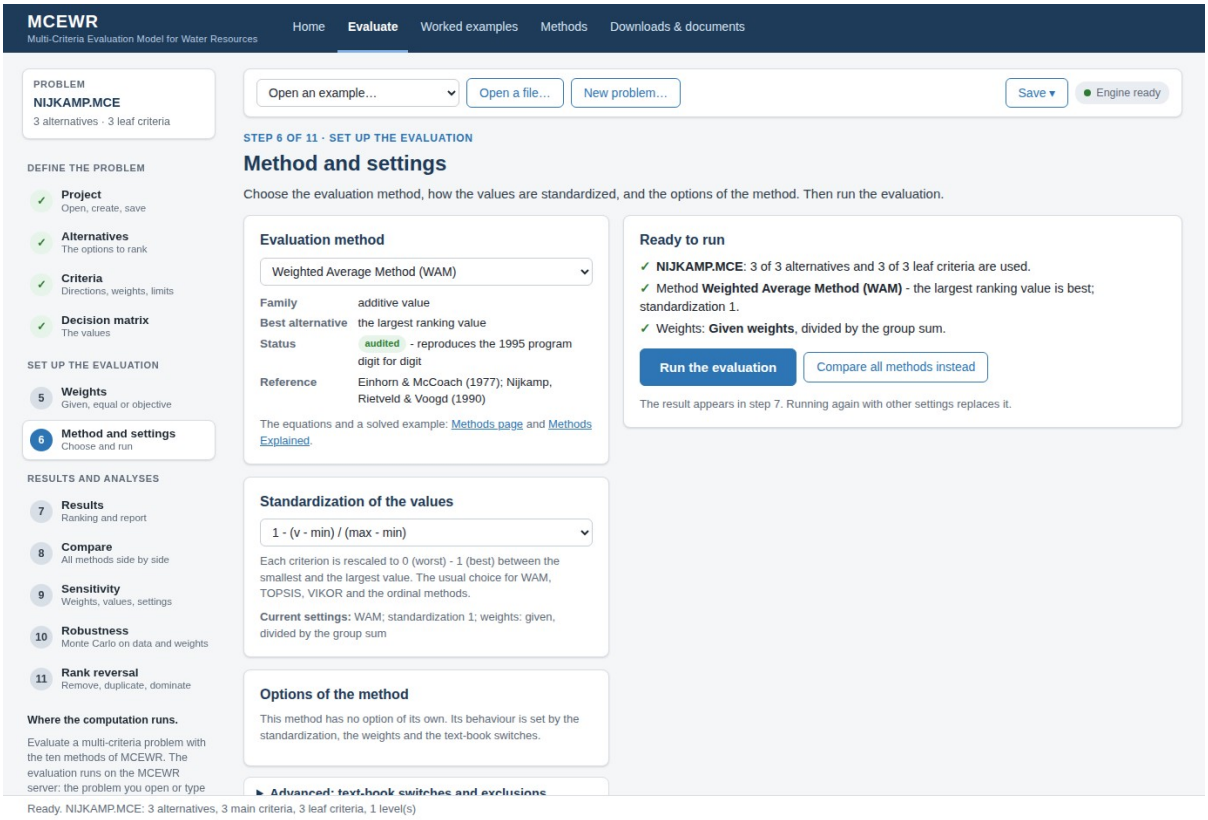


Figure 5.1 – Step 6: the method, the standardization, the options and the readiness checks.

5.1 Method

Method	Family	Best has the … value	Status
Weighted Average Method (WAM)	additive value	largest	audited
PROMETHEE II	outranking (PROMETHEE family)	largest	audited
Compromise Programming (CP)	distance to the ideal	smallest	audited
ELECTRE II	outranking (ELECTRE family)	smallest	audited
Analytic Hierarchy Process (AHP)	pairwise hierarchy	largest	audited
Expected Value (EV) (provisional)	ordinal / expected values	largest	provisional
Regime Analysis (RA) (provisional)	ordinal / outranking (regime method)	largest	provisional
TOPSIS (provisional)	distance to the ideal and the anti-ideal	largest	provisional
VIKOR (provisional)	compromise ranking (group utility and individual regret)	smallest	provisional
ELECTRE III (provisional)	outranking on pseudo-criteria (credibility and distillations)	smallest	provisional

Audited means reproduced exactly against the 1995 program's own output files and the dissertation's verification examples. **Provisional** means implemented from the primary literature and verified against closed forms, independent implementations and hand-computed examples, while a published worked example is still to be reproduced by the author; cite results of a provisional method as such. The line under the list repeats the family, the reference and the direction of preference of the selected method. **SPOTIS** is not a separate entry: Compromise Programming with *use the provided limits* and `cp_power` = 1 is the SPOTIS method term for term (Method Reference, section 5.7), and the summary of such a run says so.

5.2 Standardization

Code	Formula	When
0	none – the values as given	the values are already comparable, or the method standardizes itself (PROMETHEE and ELECTRE III work on raw differences with their thresholds; TOPSIS and VIKOR have their own scaling)
1	(v – min) / (max – min), per criterion	the usual choice for measurements in different units: every criterion runs from 0 (worst value present) to 1 (best); minimizing criteria are inverted
2	v / sum of the values	proportions of a total
3	v / max	fractions of the best value

5.3 Weights (chosen in step 5)

Card	Meaning
Given weights	the declared weights of step 3; every group is divided by its sum (the *share* column) while the switch is ticked
Equal weights	1/n in every group
Best-Worst Method	the weights computed from the best-to-others and others-to-worst judgments entered in step 5 (chapter 6)
Entropy / CRITIC / MEREC	the objective weights computed from the data replace the declared ones (chapter 6)
Divide the given weights by the sum of their group (switch)	off: the raw weights enter the formulas. Needed to reproduce some published tables (the tutorial says which); normally left on

5.4 Method options

Only the options of the selected method are shown; *(no option of its own)* otherwise. They are taken from the program's method registry, so it cannot differ from what the engine actually accepts:

Method	Option	Type	Default	Meaning
Compromise Programming (CP)	`cp_power`	int	1	the power s of the distance metric: 1 or 2
Compromise Programming (CP)	`cp_use_provided_limits`	bool	true	True = use the Upper and Lower limit values of the input file as the ideal and anti-ideal points (the result is then independent of which alternatives are present); False = recompute them from the data
ELECTRE II	`electre_min_discordance_zero`	bool	false	with unstandardized data: set the discordance scale s(i) to 0 instead of using the Upper limit value

Method	Option	Type	Default	Meaning
Analytic Hierarchy Process (AHP)	`ahp_min_method`	int	2	minimizing criteria: 1 = invert the standardized value, 2 = multiply the weight by -1
Expected Value (EV) (provisional)	`ev_weights`	str	ordinal	'ordinal' = expected (rank-order-centroid) weights from the ranking of the declared weights; 'cardinal' = the declared weights as they are
Expected Value (EV) (provisional)	`ev_scores`	str	ordinal	'ordinal' = expected scores (m+1-rank)/(m+1) from the ranking of the alternatives under each criterion; 'cardinal' = the standardized values as in WAM
Regime Analysis (RA) (provisional)	`ra_weights`	str	ordinal	'ordinal' = probabilities over the set of weights consistent with the ranking of the declared weights; 'cardinal' = the declared weights decide each pair outright
Regime Analysis (RA) (provisional)	`ra_samples`	int	10000	number of weight vectors drawn from the ordered simplex per group (ordinal weights only)
Regime Analysis (RA) (provisional)	`ra_seed`	int	1995	seed of the random drawings; recorded so that a run can be repeated exactly
VIKOR (provisional)	`vikor_v`	float	0.5	weight of the decision strategy: v ranks by the group utility S, 1 - v by the individual regret R; 0.5 = consensus (0 <= v <= 1)
ELECTRE III (provisional)	`electre3_alpha`	float	0.3	discrimination threshold s(lambda) = alpha + beta * lambda: alpha
ELECTRE III (provisional)	`electre3_beta`	float	-0.15	discrimination threshold s(lambda) = alpha + beta * lambda: beta

PROMETHEE II has one box instead of options: **also the PROMETHEE I partial ranking** adds the P / I / R relations of every pair and the list of incomparable pairs to the summary, the report and the detail file.

5.5 Text-book switches

All off, the engine reproduces the 1995 program exactly – the numbers of the dissertation and of the published articles. Each switch applies one of the five corrections identified in the mathematical review of the porting (Method Reference, chapter 15); the summary and the report record which switches were on:

Switch	Effect
`exact_thresholds`	exact thresholds (PROMETHEE p, q and CP limits rescaled with the data)
`promethee_n_minus_1`	PROMETHEE flows divided by n - 1
`cp_root`	CP: the s-th root of the distance
`parents_only`	aggregated criteria only are inverted at upper levels
`electre_symmetric`	ELECTRE II: symmetric discordance

5.6 Exclude

Ids of alternatives or of criteria to leave out of this run only, separated by commas – a quicker way than editing the *used* column when the exclusion is temporary. Excluded alternatives keep a row at the end of the ranking with the note *excluded*.

5.7 The results

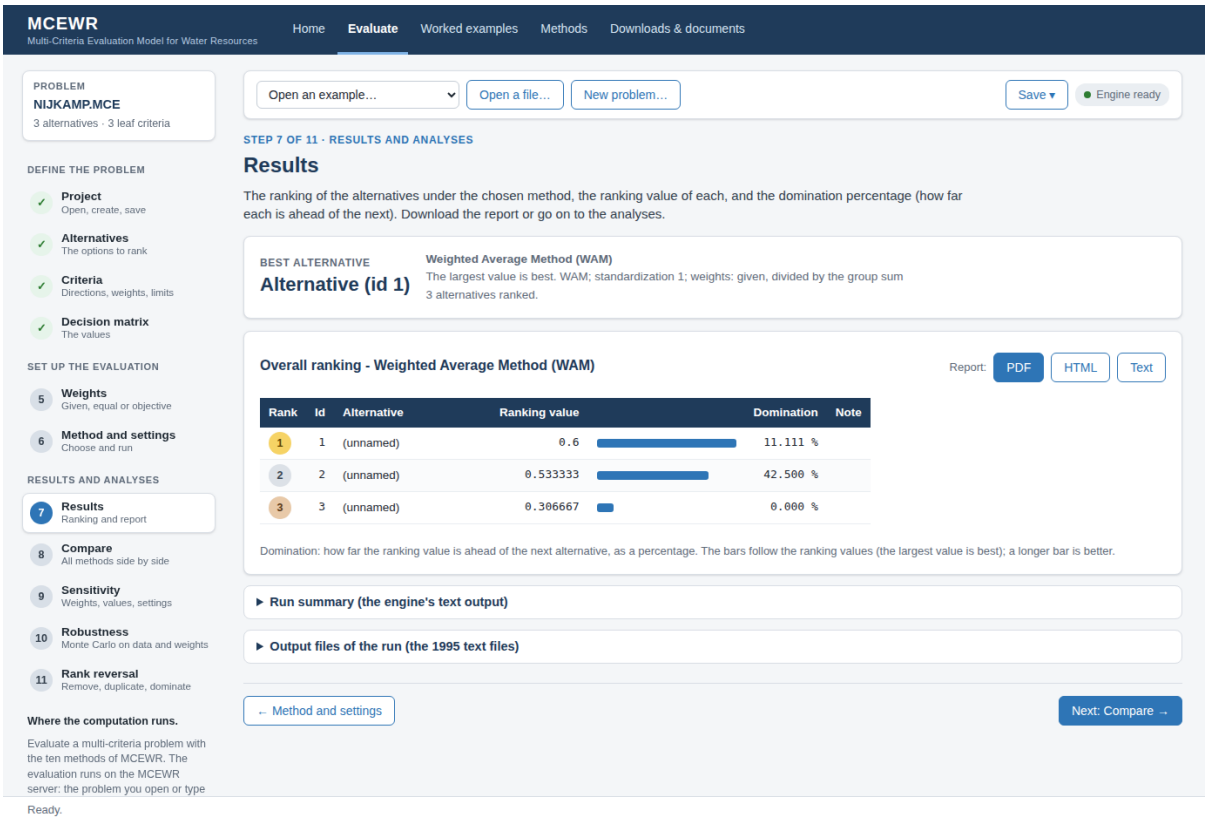


Figure 5.2 – Step 7: the results of a run.

Two methods add a card of their own under the ranking: PROMETHEE II with `promethee_partial` on shows the PROMETHEE I partial ranking (the positive and negative flows, the P / I / R relation of every pair, the incomparable pairs); ELECTRE III shows its two distillations, the position of every alternative in each, and the final partial preorder – the pairs the two distillations disagree on are **incomparable** (R), and the complete ranking of step 7 orders them by the mean position while the method itself leaves them unordered.

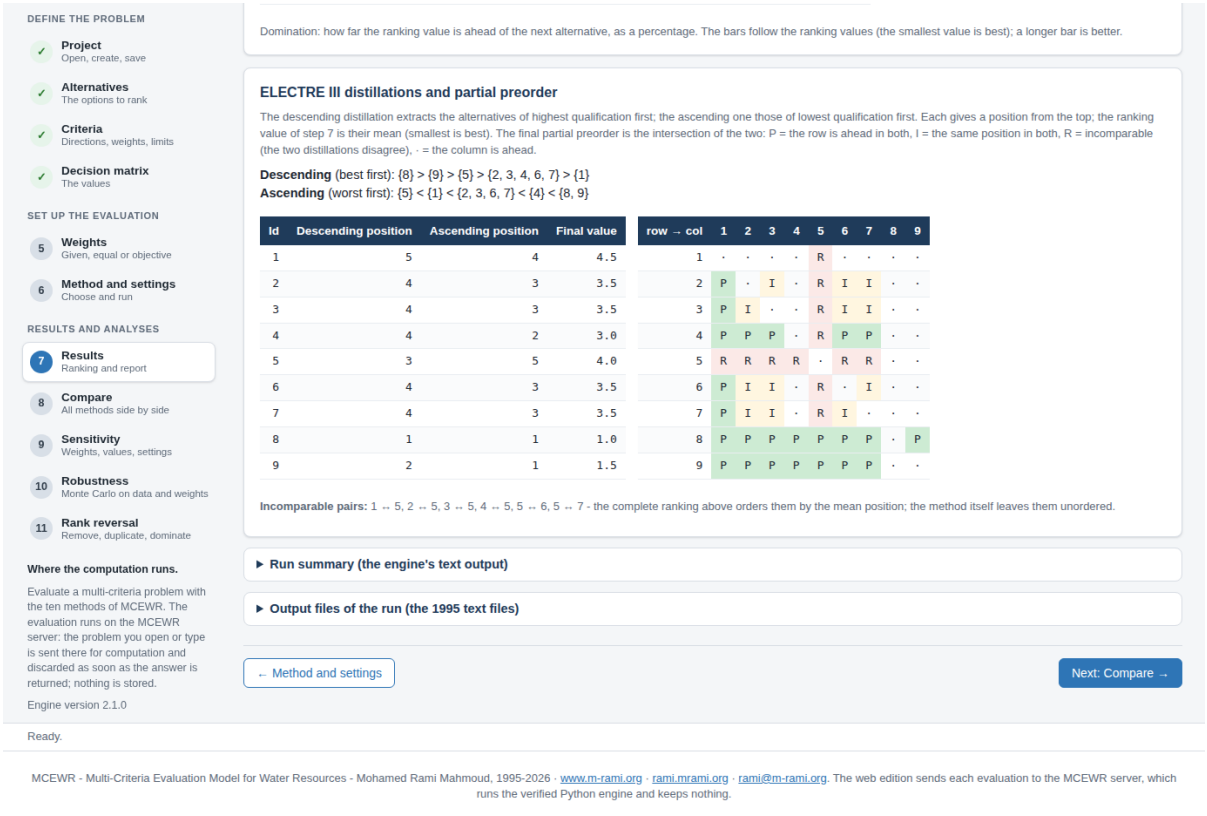


Figure 5.3 – Step 7 with ELECTRE III: the distillations and the partial preorder of the Han River example.

Element	Content
Title	the method and its direction of preference (*the largest value is best* / *the smallest value is best*)
Overall ranking	rank; alternative id; name; ranking value (the method's own number); domination – by how many percent the alternative leads the next one, 0 for the last; note – NF for a zero value with no domination, *excluded* for an excluded alternative
Report: PDF / HTML / Text	downloads the report of this run (chapter 9)
Output files	one button per output file of the method (for example `wam.txt` and `sortwam.txt`): the detail and the sorted-ranking files in the exact layout of the 1995 program
Summary	one text with the input, method, standardization, criteria and alternatives, text-book switches, the ranking and the output files
Warnings	anything the run wants you to know: a weight of zero, a threshold that never bites, a Saaty consistency ratio above 0.10, a VIKOR acceptance condition not met, an ELECTRE II cycle avoided

6. Step 5 Weights: six ways to weight the criteria

Step 5 chooses the weighting scheme of every run with six cards, and shows below them the table **The weights each method gives** – for every group of sibling criteria, the weights of every scheme, the chosen one highlighted. The given and equal weights and the Best-Worst Method are judgements of the decision maker; the three objective schemes are derived from the data alone. Objective weights serve two purposes: as a check – do the declared weights and the data point the same way? – and as a substitute when no expert weights exist; a group with aggregated or Saaty criteria keeps its given weights under them.

6.1 The Best-Worst Method

The Best-Worst Method (Rezaei 2015; linear model 2016) needs 2n − 3 comparisons for n criteria. Choose the card **Best-Worst Method**; the card **Best-Worst judgments** opens with one block per group of criteria. Name the **best** and the **worst** criterion of the group, then give on the 1–9 scale (1 equal, 3 moderately, 5 strongly, 7 very strongly, 9 extremely more important) the vector *Best over this* – how much more important the best is than each criterion – and the vector *This over worst* – how much more important each criterion is than the worst; the best-to-worst value is shared by the two vectors and mirrored automatically. **Compute the weights** solves the linear program of Rezaei (2016) on the server (minimize the largest violation ξ of the comparisons) and fills the BWM column; the line under the block reports ξ*, the consistency ratio ξ*/CI (0 = perfect, 1 = the most contradictory judgements possible) and the input-based ratio of Liang, Brunelli and Rezaei (2020). **Use them as the given weights** copies the weights into step 3; **Forget the judgments** removes them. The judgments are kept in the project file. A run with the BWM scheme uses them group by group; groups without judgments keep their given weights, and the run's notes say so.

6.2 The objective schemes

Scheme	A criterion weighs more when …	Needs
Entropy	its values differ much between the alternatives (low entropy of the normalized column)	positive values
CRITIC	its values are dispersed and little correlated with the other criteria (contrast and conflict)	some dispersion
MEREC	removing it changes the overall performances of the alternatives much	positive values

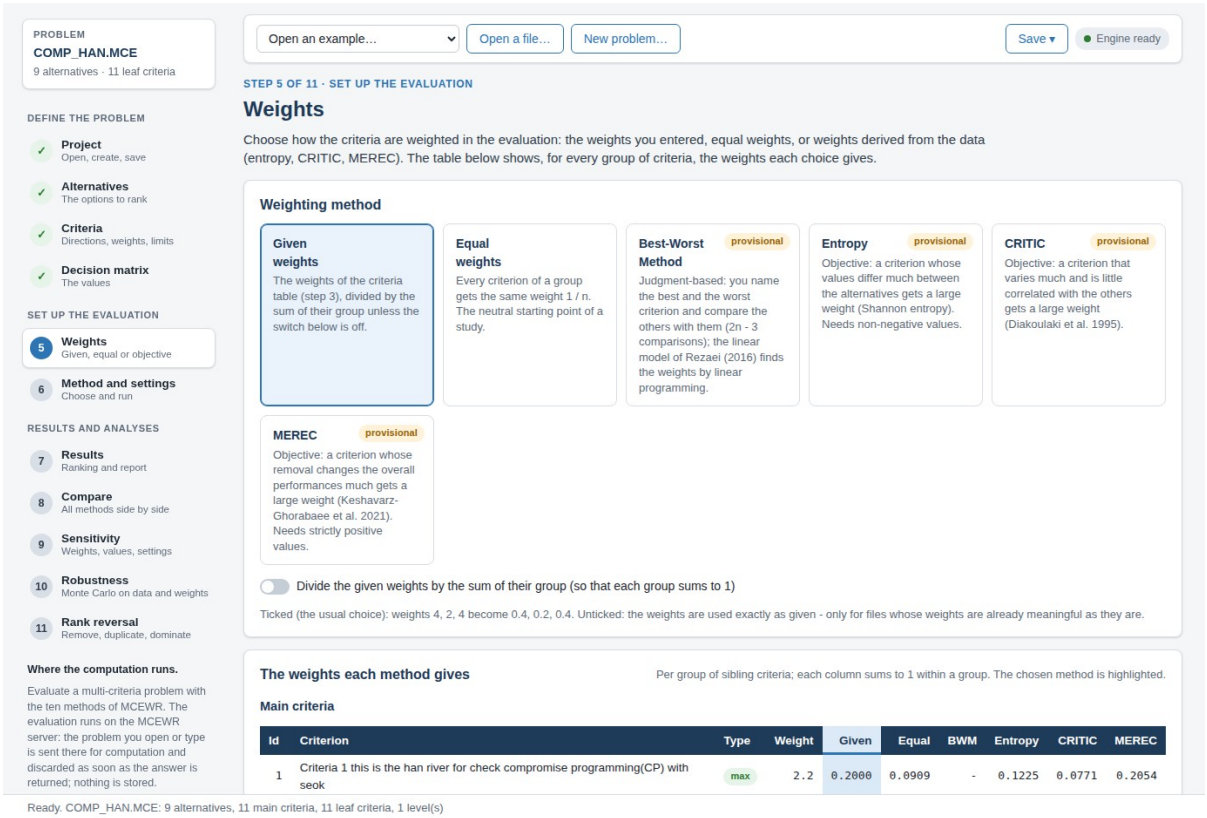


Figure 6.1 – Step 5 for the Han River example: the six cards and the table of the weights each scheme gives.

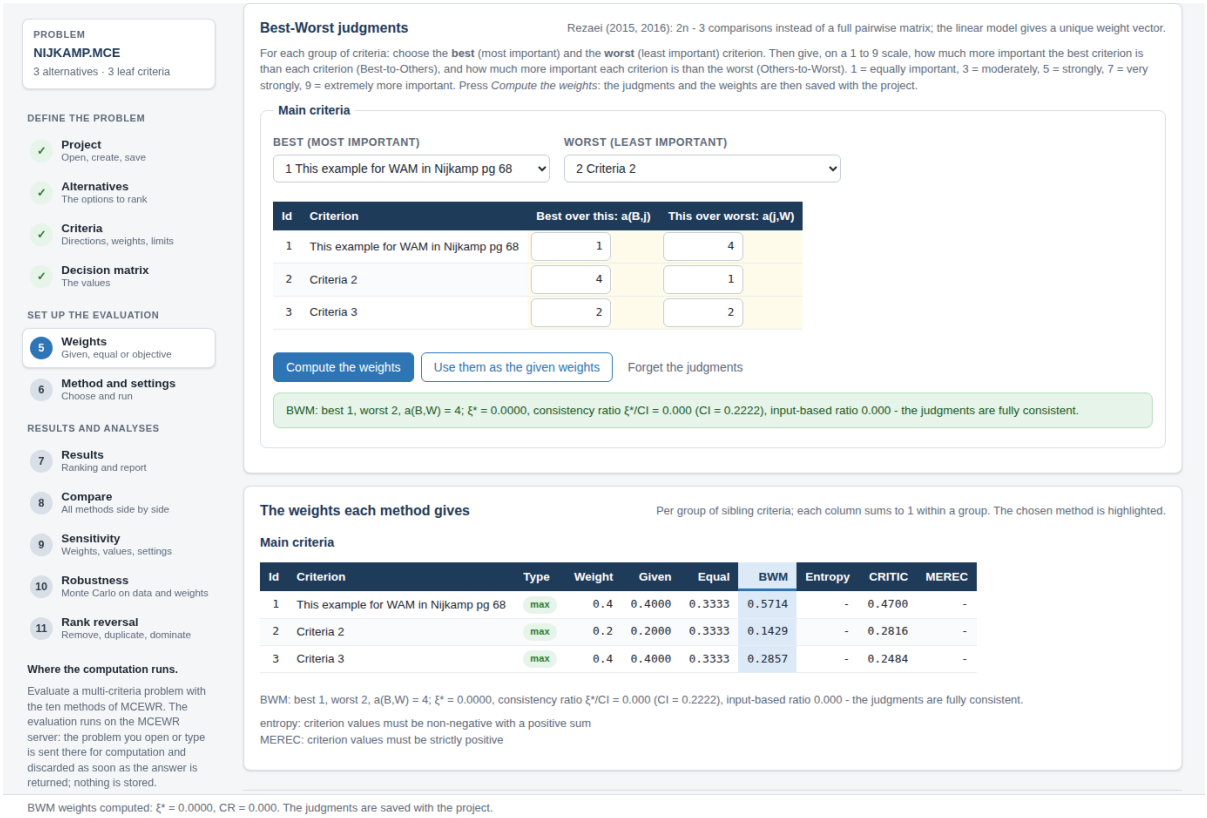


Figure 6.2 – The Best-Worst judgments of the Nijkamp example (best 1, worst 2) and the weights they give.

The chosen card is what every run, comparison and analysis uses; the scheme's name appears in the summary and the report. A note under a group's table says when a scheme cannot be computed for it (error 246 in the messages).

7. Step 8 Compare

The settings of steps 5 and 6 – standardization, weights, text-book switches, exclusions – apply to every method of the comparison; the methods' own options are those shown in step 6 for each. Tick the methods (all ten by default) and press **Run the comparison**.

Element	Content
Rank of every alternative under every method	one row per alternative: its rank under each method; then the mean rank, the consensus position (the ordering by mean rank) and the number of first places
Agreement between the methods	Spearman's rho of the rank vectors, method by method: 1 = identical order, 0 = unrelated, −1 = opposite
Notes	methods that could not run on this file with these settings, and why (for example ELECTRE II with a cycle in its outranking graph); the comparison goes on with the others
Report: PDF / HTML	the comparison report with all tables and the per-method rankings

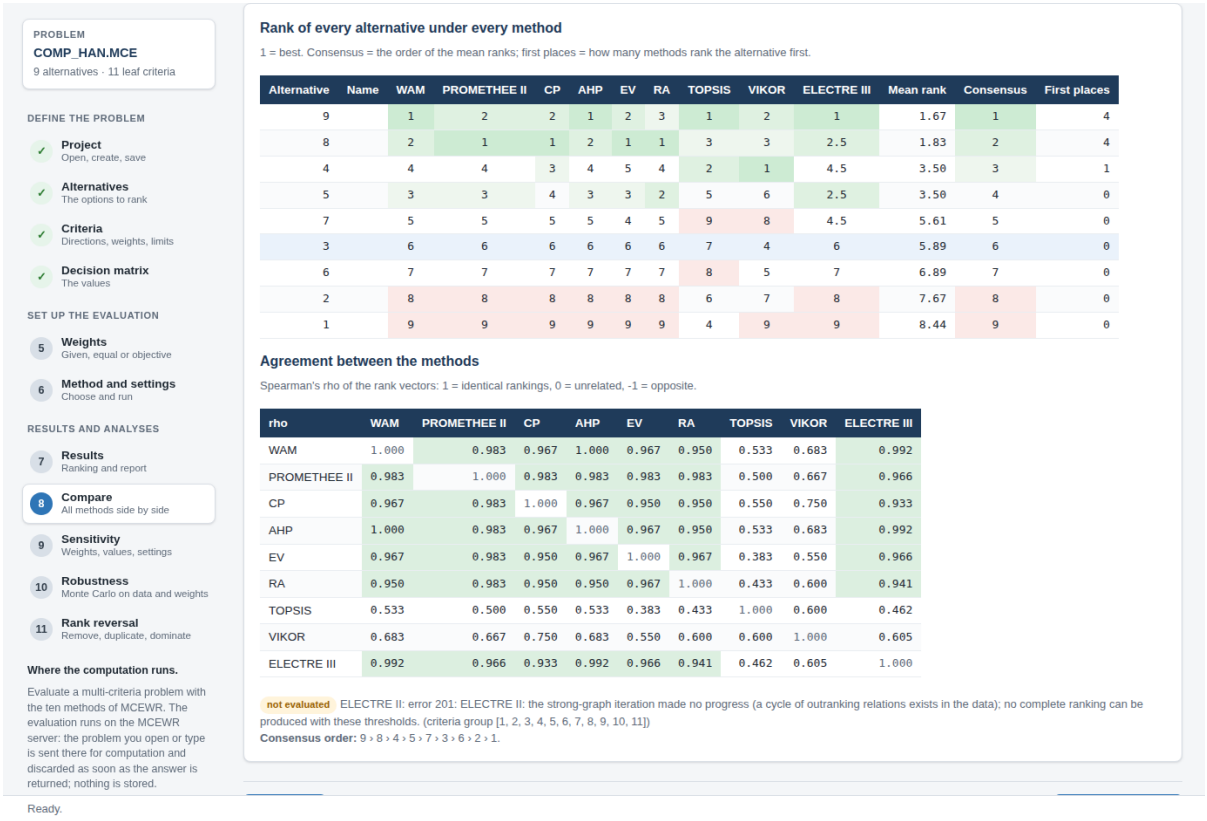


Figure 7.1 – Step 8: nine methods on the Han River example.

How to read it: an alternative first under every method is a robust choice; one whose rank spreads across methods depends on the philosophy of the method – compensation (WAM, AHP, EV), distance to an ideal (CP, TOPSIS, VIKOR) or outranking with thresholds (PROMETHEE, ELECTRE, RA). The rho matrix groups the methods that behave alike on this problem. This is the analysis of the Red Bluff paper of 2000, now one click.

8. Steps 9, 10 and 11: sensitivity, robustness and rank reversal

8.1 Step 9, part Weights: sensitivity of the ranking to the weights

Step 9 has three parts, chosen with the buttons at its top: the weights, the decision values and the settings of the method. Each moves one input at a time, re-runs the method and settings of steps 5 and 6 at every step, and locates by bisection the point where the winner changes. The part *Weights* asks:

Parameter	Meaning	Default
Group	the group of criteria whose weights are varied (the main criteria, or a parent's sub-criteria)	the main criteria
Steps of the sweep	how many values of each weight's share are tried between 0 and 1 (the other weights keep their proportions); a bisection then finds the exact critical share	20

Result	Content
One-at-a-time variation	per criterion: its base share; the winner in every interval of its share; the nearest critical share and the change of weight it represents – the smallest change of that weight that changes the decision
Headline	the base winner, the most sensitive criterion and the number of evaluations

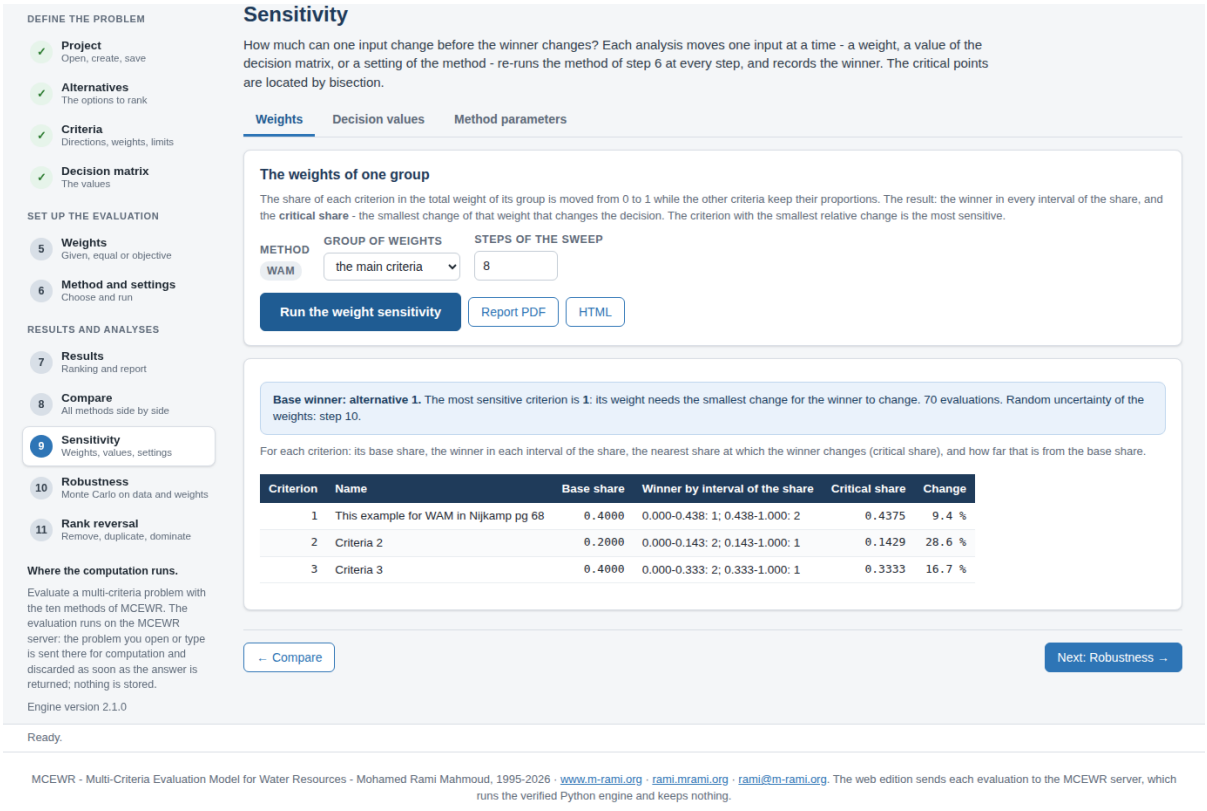


Figure 8.1 – Step 9, part Weights: WAM on the Nijkamp example.

8.2 Step 9, part Decision values: sensitivity to the data

The values of the chosen alternatives (by default the base winner only; or all alternatives, at most 400 cells) are moved one at a time from *value – range* to *value + range* of their criterion, where the range is that of the criterion over the alternatives. For every cell the table gives the base value, the winner in every interval, the

nearest **critical value** and the change it represents as a percentage of the criterion's range – a scale that exists even for a value of 0 and lets criteria in different units be compared – and whether the critical value lies above or below the base value. The headline names the most sensitive value of the study. Values given as Saaty comparisons are not varied.

8.3 Step 9, part Method parameters: sensitivity to the settings

One numeric setting – the standardization code (any method) or an option of the method (the VIKOR strategy weight, the Compromise Programming power, the ELECTRE III thresholds α and β) – is given a list of values (the page proposes one); the method is re-run for each. The table gives, per value, the winner, the whole order (ties joined by =) and Spearman's rho of the positions against the base run; a value the engine refuses, for instance a VIKOR v outside 0–1, is listed as not evaluated. The headline says at which values the winner changes.

8.4 Step 10 Robustness: Monte Carlo on the values and the weights

All the inputs are perturbed at once. Step 10 asks:

Parameter	Meaning	Default
Uncertainty of the values ± %	every value of the decision matrix is multiplied by 1 + u with |u| up to this fraction (0 = the values are not perturbed)	10
Uncertainty of the weights ± %	every free weight (not the Saaty priorities) is multiplied likewise (0 = the weights are not perturbed)	0
Distribution of u	uniform in [−p, +p], or triangular on [−p, +p] with mode 0 (small errors more likely than large ones)	uniform
Samples	how many random samples are drawn (at most 5 000 on the site)	300
Seed	the seed of the random drawing – the same seed gives the same table on any computer	1995

Result	Content
Rank acceptability	per alternative: its base position; the fraction of the samples that put it first, second, …; the fraction in which it keeps its base position; its mean rank
Headline	the base winner and order, the winner stability (the fraction of the samples that keep the base winner first), what was perturbed, and the number of samples the engine refused (for example negative values under an objective weight scheme)

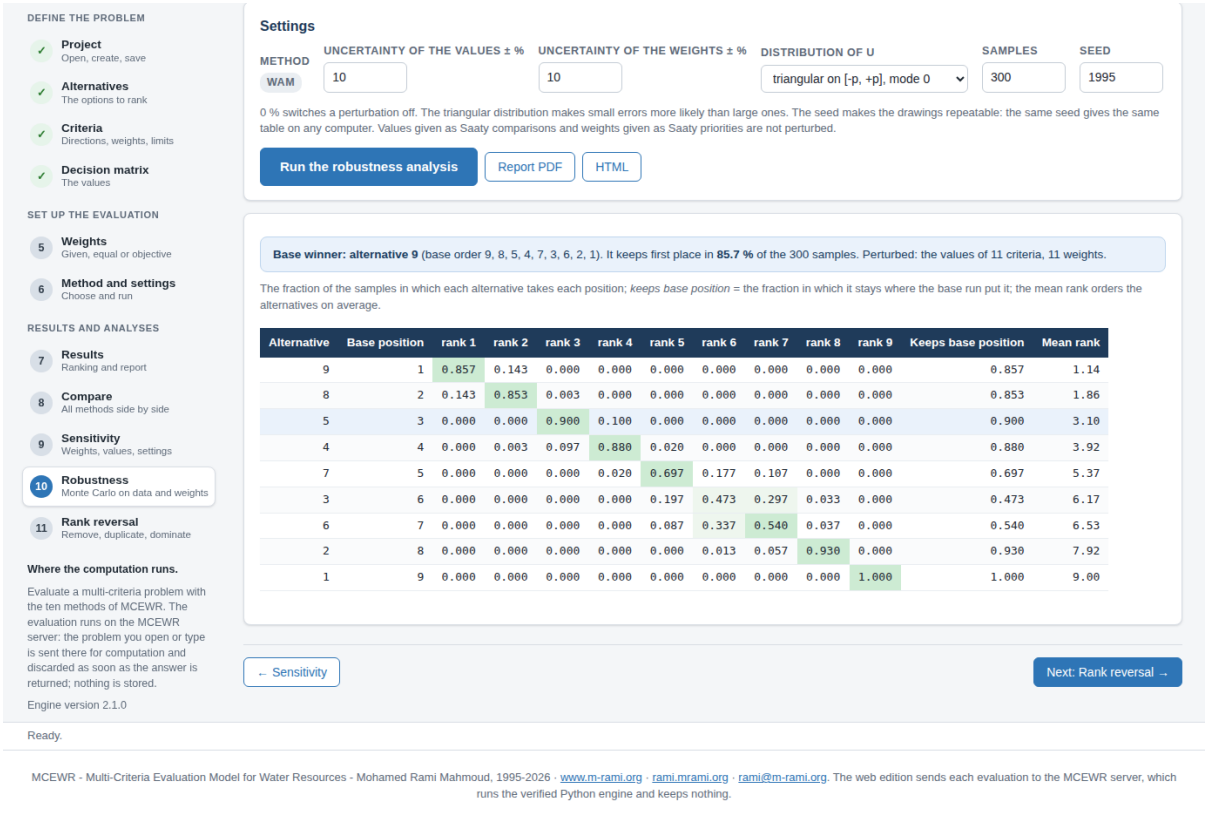


Figure 8.2 – Step 10: the Han River example with uncertain values and weights.

8.5 Step 11 Rank reversal

Rank reversal is the change of the order of two alternatives when a third one is added or removed – a known property of methods whose scale depends on the set of alternatives. Step 11 runs three tests on the methods you tick, with the settings of steps 5 and 6:

Test	What is done	A reversal means
remove	each alternative is removed in turn and the others re-ranked	two of the remaining alternatives swapped their order
duplicate	a copy of each alternative is added in turn	two of the original alternatives swapped their order
dominated	an alternative worse than every other on every criterion is added	the same – a dominated newcomer should change nothing

The upper table counts, per method and test, the trials with at least one reversal and the number of reversed pairs; the lower table lists every instance (a<>b: a was ahead of b in the base run). Compromise Programming with the provided limits – SPOTIS – never reverses, because its reference points are fixed in advance; TOPSIS, which computes its ideal from the set, is the method most often caught.



Figure 8.3 – Step 11: six methods on the Han River example.

9. Reports, output files and input formats

9.1 Reports

Every *Report* button asks the server to render the report of the result shown – from a fresh computation with the same settings, so it always matches the tables on the page – and downloads it:

Format	Content and use
PDF	the full report: summary, overall ranking, the criteria with their shares, the pay-off and standardized matrices, the rank and value of every alternative under every criterion, the Saaty consistency ratios (AHP), the PROMETHEE I relations when requested, the warnings, the two output files verbatim. Ready to attach to a report or a thesis
HTML	the same content as a web page: opens in the browser, prints well, copies into a document
Text	the same content as plain text, for archives and for diff tools

The comparison, sensitivity and reversal reports (PDF and HTML) carry their own tables. The Word and Excel reports with charts exist in the author's desktop edition only.

9.2 Output files

Each method writes two files in the layout of the 1995 program – the detail file (every intermediate table of the method) and the sorted-ranking file. The *Output files* buttons open them; their names are those of the original program (`wam.txt` / `sortwam.txt`, `promethe.txt` / `sortprom.txt`, `compro.txt` / `sortcomp.txt`, `electreii.txt` / `sortele2.txt`, `ahp.txt` / `sortahp.txt`, and the 2.0 methods likewise). The Method Reference explains the layout of each file.

9.3 Input formats

Format	What it is	Reference
`.MCE`	the text format of the 1995 program: a header, one block per group of criteria, the values. The 26 examples are in this format	Method Reference, appendix on the 1995 format
`.mcewr`	the project file: a readable JSON document with every setting and, optionally, the results of a run; what *Save as* writes and reads back	section 4.4 of this guide
`.csv`	three blocks – `[criteria]`, `[alternatives]`, `[values]` – with a header row each; written by any spreadsheet program	the example below

The CSV form of the Nijkamp example; the header row names every column the model understands, and only `id`, `name`, `type`, `weight` and the values are needed for the additive and distance methods:

```
[criteria]
id,parent,name,type,weight,upper,lower,units,promethee_function,p,q,p_star,p_0,p_minus,q_star,q_0
,saaty_weights,saaty_values
1,,This example for WAM in Nijkamp pg
68,1,0.4,5.0,5.0,agrement,1,10.0,10.0,0.7,0.6,0.55,0.55,0.3,0,0
2,,Criteria 2,1,0.2,5.0,5.0,agrement,1,30.0,0.0,0.7,0.6,0.55,0.55,0.3,0,0
3,,Criteria 3,1,0.4,5.0,5.0,agrement,1,30.0,0.0,0.7,0.6,0.55,0.55,0.3,0,0

[alternatives]
id,name
1,
2,
3,

[values]
alternative,1,2,3
1,50000.0,500.0,80.0
2,100000.0,-100.0,60.0
3,80000.0,100.0,50.0
```

10. Messages, limits and privacy

10.1 Messages of the status line

Message	Meaning and what to do
the MCEWR server did not answer (…)	the page could not reach the engine: your connection, or the server is restarting. Wait a moment and reload with **Ctrl+F5**
the server answered 404	the page is not at the address it was built for (it looks for the engine at `/api` next to itself). Use `https://mcewr.m-rami.org/app.html`
too many requests from this address; please wait a minute	more than 120 runs in one minute from one address – a class behind one router counts as one. Wait a minute
the server accepts at most 300 alternatives and 120 criteria	the problem is too large for the teaching service; contact the author for larger studies
the server accepts at most 5000 perturbation samples / 200 sweep steps	lower the parameter
Not a valid value: …	an edit or an option that the engine refuses (not a number, a negative weight, an option outside its range)
MCEWR ERROR 201	ELECTRE II: a cycle in the outranking graph with these thresholds; change them in the file's group line, or use ELECTRE III
MCEWR ERROR 245	an option or a setting outside its allowed values; the context line says which
MCEWR ERROR 246	an objective weighting scheme cannot be computed for a group (entropy and MEREC need positive data; CRITIC needs dispersion)
MCEWR ERROR 248	none of the selected methods could run in an analysis; the note says why for each
other MCEWR ERROR numbers	the errors of the input reader, with the line of the file; the same numbers as the 1995 program (Method Reference)

10.2 Limits

Limit	Value
alternatives × criteria per problem	300 × 120
size of one request (a problem with its settings)	2 MB – a 48 × 20 problem is about 60 kB
samples of the perturbation analysis	5 000
steps of a weight sweep	200
runs per minute from one address	120

The limits protect the shared server; they are generous for any classroom or research problem of the kind the model was made for (the Red Bluff study has 11 alternatives and 12 criteria, the Han River study 9 and 11). Studies beyond them need the author's desktop edition; write to rami@m-rami.org.

10.3 Privacy

When you press a *Run* button, the page sends the problem and the settings to the MCEWR server, which computes the answer in a private temporary folder and deletes it as soon as the answer is returned. Nothing is stored; the content of requests is not logged; there are no accounts and no cookies. The site states this on every

page. What you want to keep, download. If your data must not leave your computer at all, write to the author about the desktop edition.

11. Frequently asked questions

- **Are the results the same as in the desktop program and in the 1995 program?** Yes. The three editions run the same engine; the engine reproduces the 1995 program's output files to the last digit on every historical example, and the website is proved equal to the desktop edition on every example and every method at each release.
- **Can I use my Excel workbook?** Not directly: save the three blocks of section 9.3 as a CSV file from Excel (*Save as → CSV*) and open that on the site.
- **Why does the ranking value differ between methods?** Each method has its own number – a weighted sum, a net flow, a distance, a priority, a probability, an index – comparable within one run only. Quote the rank, and the method.
- **Larger is better or smaller is better?** The title of every result says it; five methods rank by the largest value (WAM, PROMETHEE, AHP, EV, RA, TOPSIS) and four by the smallest (CP, ELECTRE II, ELECTRE III, VIKOR). The reports print it above every table.
- **Two alternatives are 0.1 % apart – is one really better?** Not for practical purposes: say so, and look at the sensitivity analysis (step 9).
- **What does *provisional* mean for my results?** The method is implemented from the primary literature and passes independent checks, but the author has not yet reproduced a published worked example with it. Use it, compare it with the audited methods, and cite it as a provisional implementation.
- **Can I reproduce a result I obtained on the site?** Yes: download the problem (*Save as .mcewr*) and note the settings printed in the summary; the seeds of the random analyses are recorded there.
- **Can I get the program?** The desktop application is distributed by the author (rami@m-rami.org); the website exists so that nothing needs installing. The Python source is not available from the site.
- **Is there a size limit?** 300 alternatives × 120 criteria on the site; none in the desktop application.
- **Who do I ask?** rami@m-rami.org – questions, corrections, and worked examples that could promote a provisional method are welcome.

12. Glossary

Term	Meaning
alternative	one of the options to be ranked – a site, a design, a technology
criterion	a property by which the alternatives are judged; *leaf* criteria carry values, *aggregated* (parent) criteria carry the weighted result of their sub-criteria
weight, share	the declared importance of a criterion; its share is the weight divided by the group's total
pay-off matrix	the table of the alternatives' values under every leaf criterion
standardization	the transformation of every criterion's values to a common scale (section 5.2)
ranking value	the number a method computes for each alternative; the direction of preference says whether the largest or the smallest is best
domination	by how many percent an alternative leads the next one in the ranking
outranking	the relation *a is at least as good as b* judged criterion by criterion with thresholds (PROMETHEE, ELECTRE, RA), instead of a single score
ideal, anti-ideal	the best and the worst value of every criterion, either given (the limits) or taken from the alternatives; the distance methods (CP, TOPSIS, VIKOR) rank by distance to them
Saaty matrix	the pair-wise comparison matrix of AHP; its principal eigenvector gives the priorities, its consistency ratio tells whether the judgements are coherent
consensus	the ordering of the alternatives by their mean rank over the compared methods
Spearman's rho	the correlation of two rank vectors: 1 identical, 0 unrelated, −1 opposite
critical share	the share of the weight at which the winner changes
rank acceptability	the fraction of random weight perturbations that put an alternative at a given position
rank reversal	a change of the order of two alternatives caused by adding or removing another
audited / provisional	the status of a method's implementation (section 5.1)
text-book switches	the five optional corrections of the 1995 formulations (section 5.5)
SPOTIS	Compromise Programming with the provided limits and s = 1 – the rank-reversal-free computation