MCEWR

Multi-Criteria Evaluation Model for Water Resources

Web Tutorial

The evaluator at mcewr.m-rami.org, step by step: opening and editing a problem, every method with its expected results, objective weights, comparison, sensitivity and rank-reversal – with screenshots of the website

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

The lessons, files, settings and expected results of this tutorial are those of the verification examples of the model, computed by the website itself. Companion documents: MCEWR_Methods_Explained.pdf (the methods taught with solved examples), MCEWR_Web_User_Guide.pdf (every control of the website explained), MCEWR_Quick_Start.pdf, MCEWR_Method_Reference.pdf (the mathematics).

Contents

1. Opening the evaluator 1
2. Opening and editing a problem (steps 1 to 4) 3
2.1 Step 1 Project: opening a problem 3
2.2 Step 2 Alternatives 3
2.3 Step 3 Criteria 4
2.4 Step 4 Decision matrix, and how editing works 5
3. Running an evaluation (steps 5 to 7) 7
4. Worked examples: every method, with expected results 9
4.1 Weighted Average Method (WAM) – NIJKAMP.MCE 9
4.2 PROMETHEE II – PROM_EX.MCE 10
4.3 Compromise Programming (CP) – COMP_HAN.MCE 11
4.4 ELECTRE II – ELEC_EX.MCE 13
4.5 Analytic Hierarchy Process (AHP) – AHP_EXM3.MCE 14
4.6 Expected Value (EV) – NIJKAMP.MCE 15
4.7 Regime Analysis (RA) – NIJKAMP.MCE 16
4.8 TOPSIS – NIJKAMP.MCE 18
4.9 VIKOR – NIJKAMP.MCE 19
4.10 ELECTRE III – ELEC_HAN.MCE 20
4.11 SPOTIS (CP with the provided limits, s = 1) – COMP_HAN.MCE 22
5. Step 5 Weights: given, equal, Best-Worst and objective weights 24
5.1 The Best-Worst Method 25
6. Comparing the methods 27
7. Sensitivity: one input at a time 29
7.1 The weights 29
7.2 The decision values 30
7.3 The settings of the method 30
8. Robustness: Monte Carlo on the data and the weights 32
9. Rank-reversal diagnostics 34
10. Reports, output files and saving 35
11. The website, your data and the server 36
12. Tips and troubleshooting 37

1. Opening the evaluator

The website runs the verified MCEWR engine: every number it shows is produced by the verified engine of MCEWR 2.0.0 (five audited methods; EV, RA, TOPSIS, VIKOR, ELECTRE III provisional; objective weights; analyses; desktop; web), running on the MCEWR server. Nothing is installed; open **https://mcewr.m-rami.org** in a browser and click **Evaluate** in the top bar. Three other ways in:

How	Where	Notes
The Evaluate page	`https://mcewr.m-rami.org/app.html`	the evaluator itself; bookmark it
From a worked example	*Worked examples* page → **Open in the evaluator**	opens the evaluator with that example already loaded (the address carries `?example=<file>`)
With one of the 23 examples	step *1 Project* → the *Open an example* list	the verification examples of the dissertation and the published articles (the three GIS control files are listed but need the desktop edition)

The page is a guided workflow of **eleven steps**, listed in the sidebar on the left in three groups: *Define the problem* (1 Project, 2 Alternatives, 3 Criteria, 4 Decision matrix), *Set up the evaluation* (5 Weights, 6 Method and settings) and *Results and analyses* (7 Results, 8 Compare, 9 Sensitivity, 10 Robustness, 11 Rank reversal). Click a step in the sidebar or use the **Next** / **Previous** buttons at the foot of every step; a tick marks the steps that are done. The toolbar at the top opens an example, a file or a new problem and holds the **Save** menu; the pill at its right says whether the engine is ready or working, and the status line at the bottom shows every message. If the pill says *Connecting…* for more than a few seconds, section 12 tells what to do.

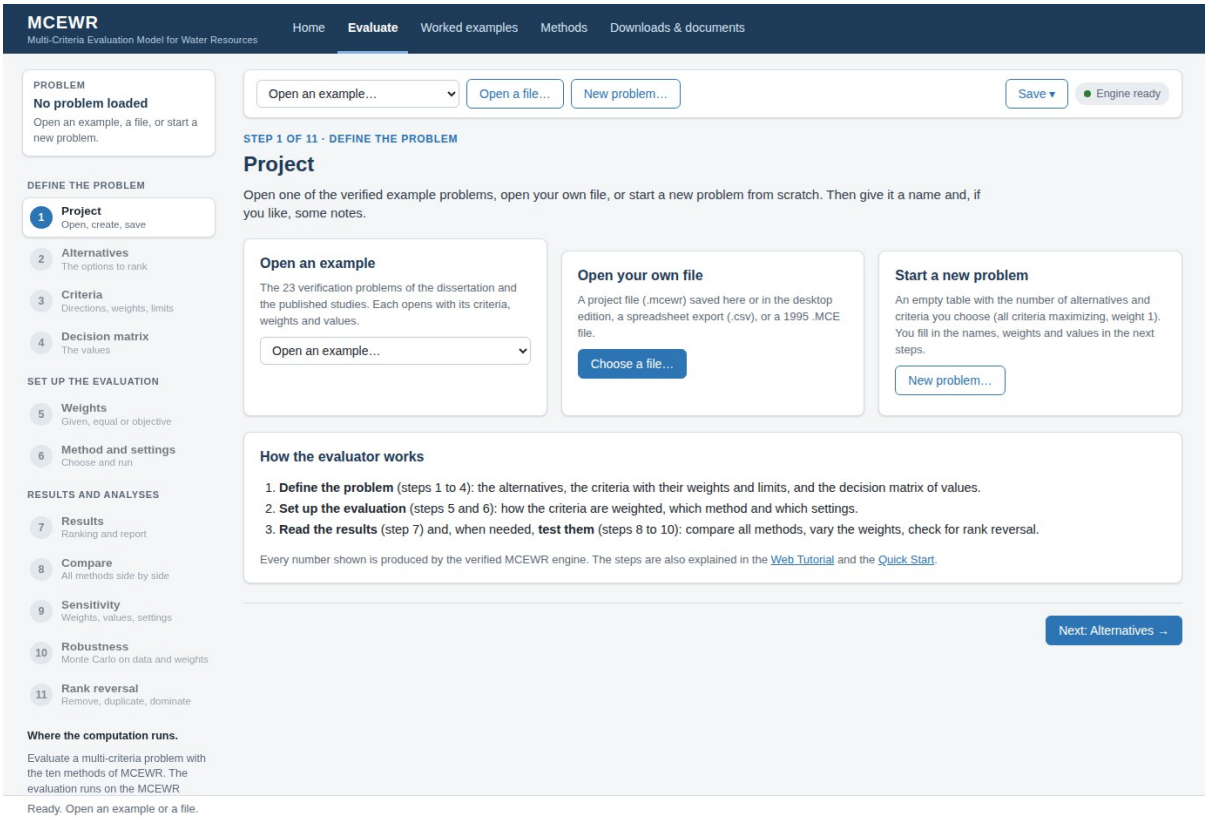


Figure 1.1 – The evaluator at start: step 1 Project, no problem loaded; the eleven steps in the sidebar.

A typical session: open a file (or an example, or a new empty problem), check and if needed edit the alternatives, the criteria and the values (steps 2 to 4), choose the weights (5) and the method with its settings (6), run, read the ranking (7), download a report; then ask the analyses (8 to 11) how robust the decision is. Everything you type is sent to the server only when a button asks for a computation, is computed, and is discarded; nothing is stored on the site.

2. Opening and editing a problem (steps 1 to 4)

2.1 Step 1 Project: opening a problem

Step 1. **Open an example**: the list (in the toolbar and in the first card of step 1) holds the example files with a one-line note each; choosing one loads it at once.

Step 2. **Open a file** accepts three formats – the 1995 text files (`.MCE`), project files (`.mcewr`, JSON, what *Save* writes) and CSV tables (`.csv`, Web User Guide section 9.3).

Step 3. **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) to fill in; the page then goes to step 2.

Step 4. Once a problem is open, the card **This problem** shows the counts (alternatives, main criteria, leaf criteria, levels) and lets you set the **name** used for the saved files and the reports, a **title** and **notes** – both kept in the project file. Warnings of the file reader, if any, appear under the card.

Choose `REDBLUFF.MCE` (the Red Bluff Diversion Dam study of the 2000 paper: 11 alternatives, 3 main criteria, 12 leaf criteria, 2 levels). The summary reads "REDBLUFF.MCE: 11 alternatives, 3 main criteria, 12 leaf criteria, 2 level(s)":

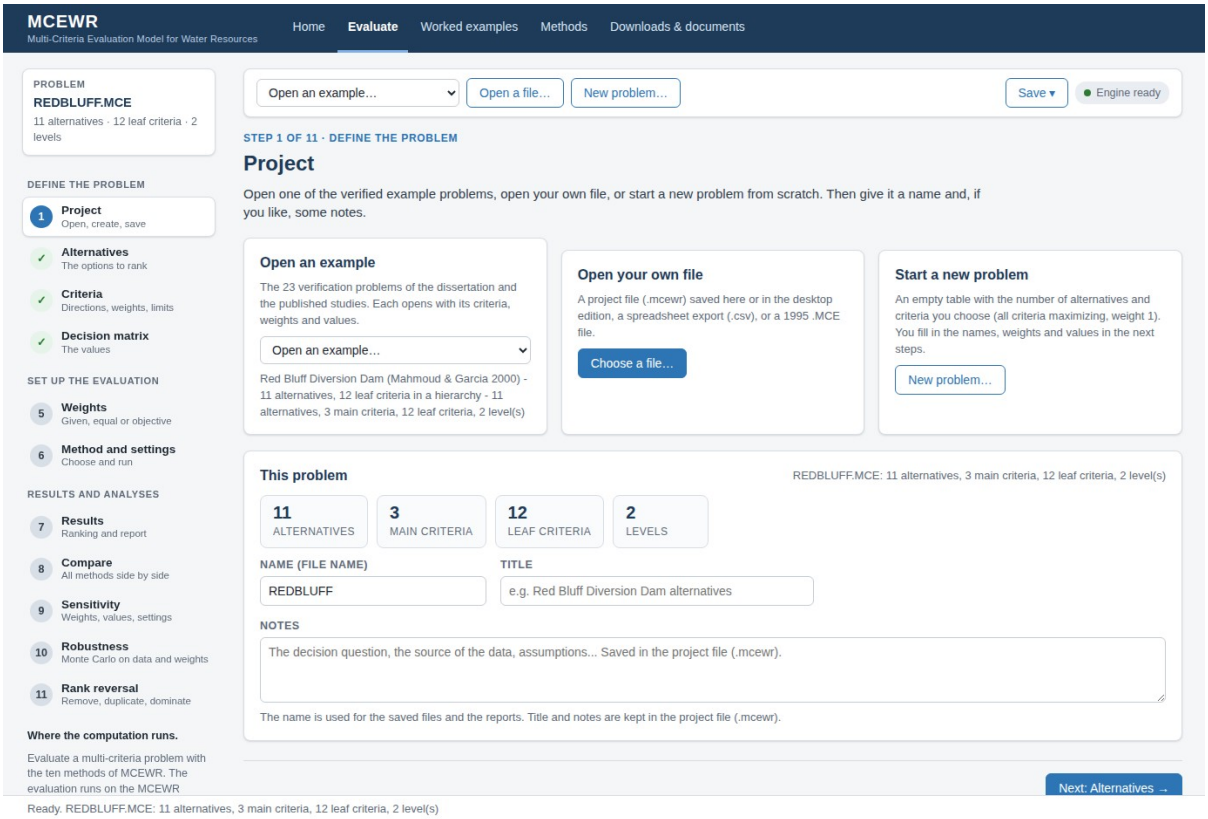


Figure 2.1 – Step 1 with REDBLUFF.MCE loaded: the counts, the name, the title and the notes.

2.2 Step 2 Alternatives

One row per alternative: its id, its **name** (an input cell), the **used** switch, and a **remove** button. **+ Add an alternative** appends a row (values 0 on every criterion) and puts the cursor in its name. A problem needs at least two alternatives; a file whose values are Saaty comparisons of the alternatives (the AHP examples) cannot receive new ones here – the page says so.

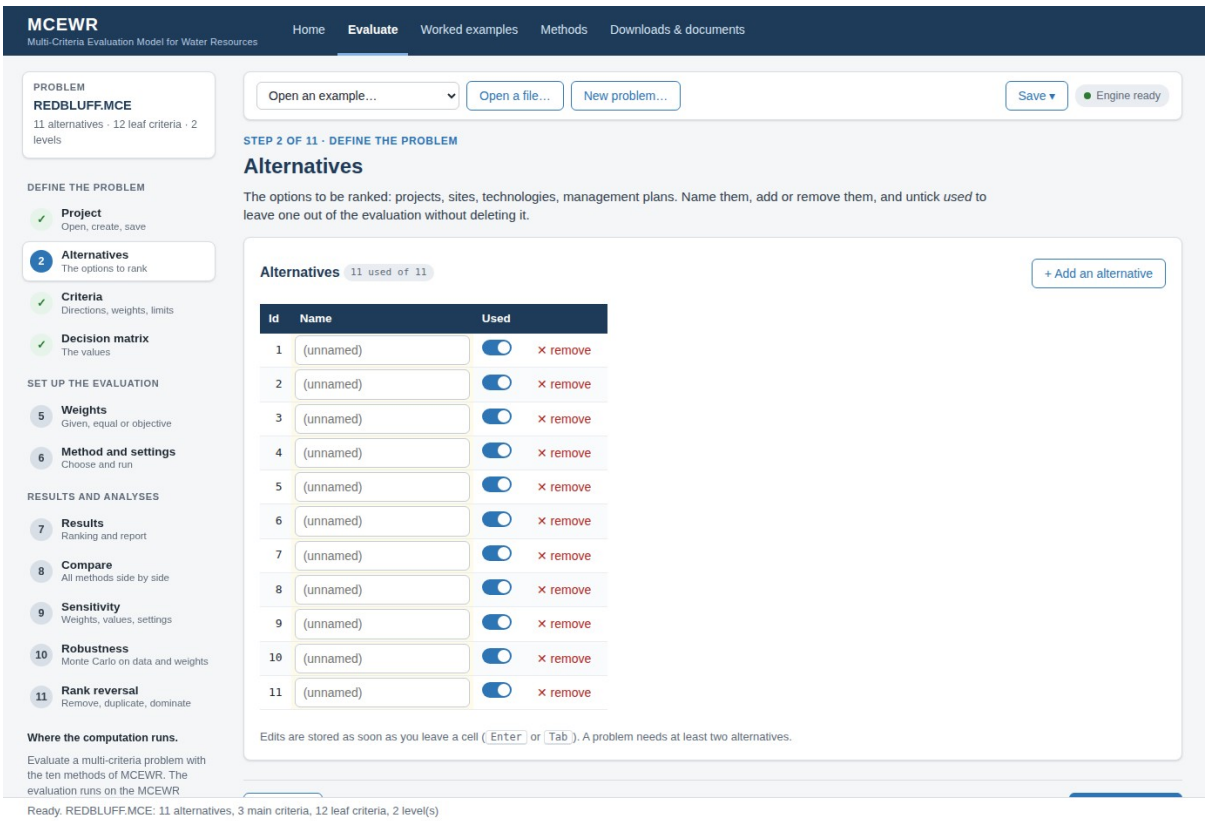


Figure 2.2 – Step 2: the eleven alternatives of the Red Bluff study (the 1995 files carry no names).

2.3 Step 3 Criteria

One row per criterion, top-down, sub-criteria indented under their parent (an aggregated criterion is shown in bold with a ▾ mark). The columns:

Column	Meaning	Editable
Id, Parent	the identifier of the criterion and of its parent (– for a main criterion)	no
Name	the name	yes
Type	`max ↑` (the larger the better) or `min ↓`	yes: a list
Weight, Share	the declared weight and its share of the group's total; 'Saaty' when the group's weights come from pair-wise comparisons	weight: yes (not for Saaty groups)
Units	the unit of the values	yes
Upper, Lower	the limit values of the criterion (the ideal and anti-ideal of CP with provided limits; the ELECTRE II discordance scale)	yes
Pf, p, q	the PROMETHEE preference function (1-6) and its thresholds, also the ELECTRE III preference and indifference thresholds – shown by the switch *Show thresholds*	yes
Veto	the ELECTRE III veto threshold (0 = none) – shown by the same switch	yes
Used	a switch: off excludes the criterion from the evaluation without deleting it	yes
+ sub, ✕	add a sub-criterion under this criterion (a leaf then becomes aggregated and loses its values), or remove the criterion with its sub-criteria	buttons

+ Add a main criterion appends a criterion (max, weight 1) to the top level.

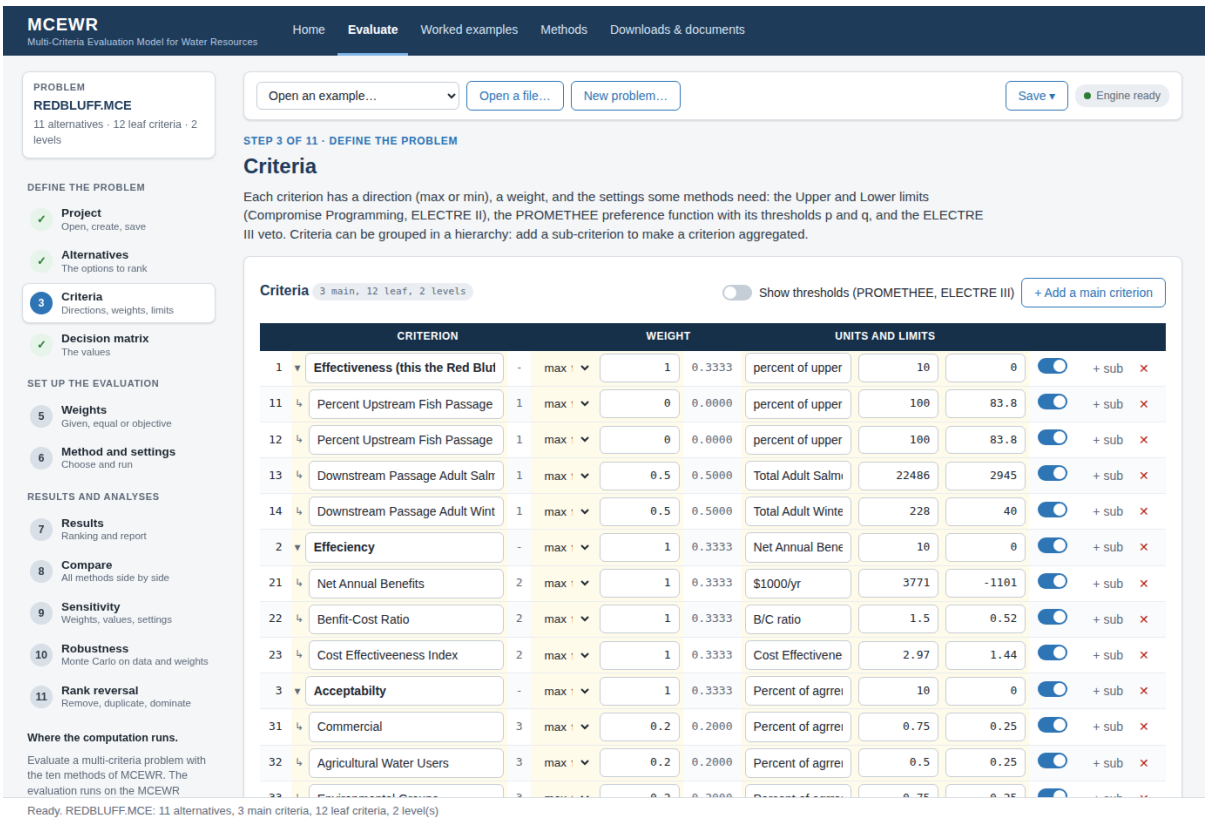


Figure 2.3 – Step 3: the criteria of the Red Bluff study, a two-level hierarchy.

The first rows of the Red Bluff criteria table read:

id	name	parent	type	weight	share	upper	lower	units
1	Effectiveness (this the Red Bluff		max	1	0.3333	10	0	percent of upperstream
11	Percent Upstream Fish Passage Effe	1	max	0	0.0000	100	83.8	percent of upperstream
12	Percent Upstream Fish Passage Effe	1	max	0	0.0000	100	83.8	percent of upperstream
13	Downstream Passage Adult Salmon Sa	1	max	0.5	0.5000	22486	2945	Total Adult Salmon Sav
14	Downstream Passage Adult Winter-Ru	1	max	0.5	0.5000	228	40	Total Adult Winter-Run

2.4 Step 4 Decision matrix, and how editing works

The decision matrix has one row per alternative (id, name) and one column per leaf criterion, headed by the criterion's id, its direction (↑ larger is better, ↓ smaller is better), its name and its units. **Every yellow cell is an input: click it, type the new value, and press Enter (the cursor moves down), Tab (it moves right) or click elsewhere** to confirm. The value is checked at once: a value that is not a number, a negative weight, or a weight of a Saaty group is refused with a message and the cell is marked red. An accepted edit flashes green and goes into the problem the page holds, so the next evaluation uses it – change a weight, run again, and watch the ranking move. A block of numbers copied from a spreadsheet can be **pasted** into a cell: it fills the cells to the right and below in one go. The same rules apply to the input cells of steps 2 and 3.

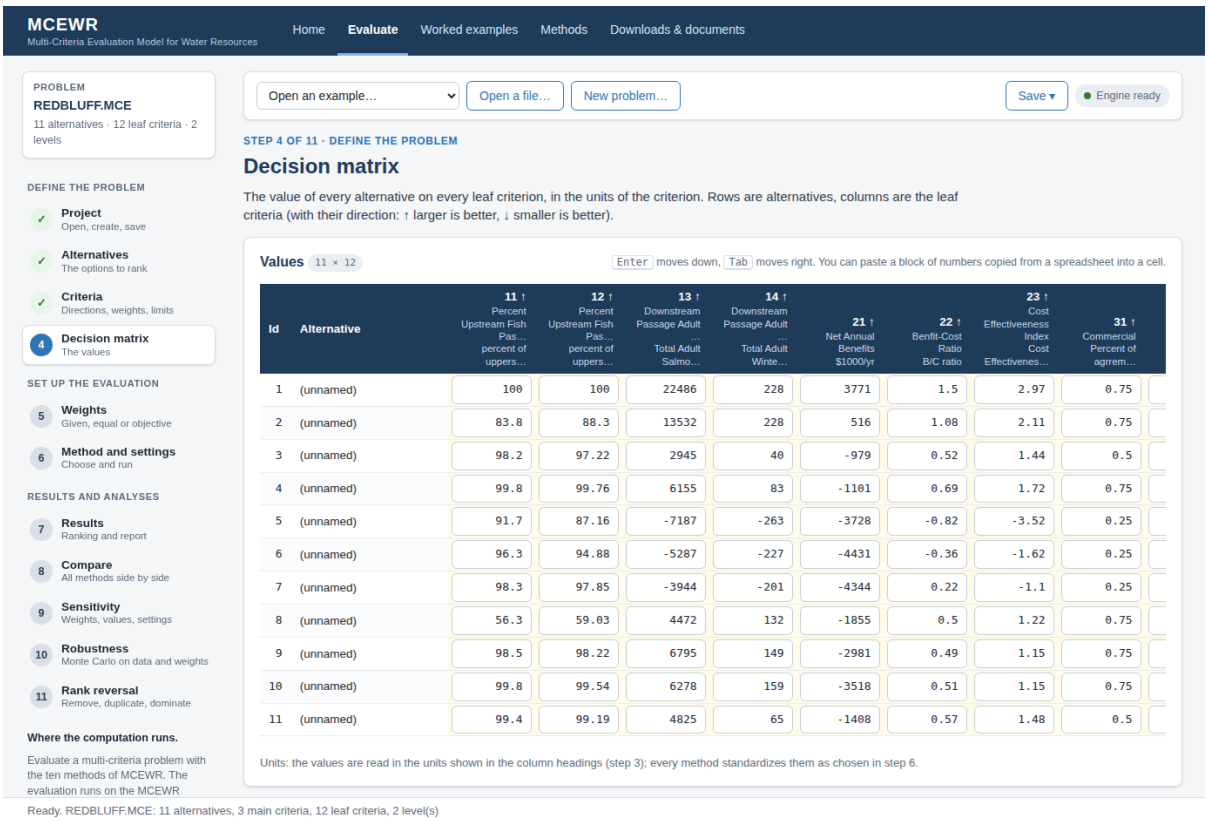


Figure 2.4 – Step 4: the decision matrix of the Red Bluff study; the column headings carry the direction, the name and the units.

The **Save** menu of the toolbar downloads the edited problem as a project file (`.mcewr` – the problem, the title and notes, the Best-Worst judgments and the settings of the last run), a CSV table or a 1995 `.MCE` file (the `.MCE` format cannot hold ELECTRE III vetoes; the page says so). Reopen the saved file with *Open a file* – on the website or in the desktop application; a project file reopens with its settings in place.

> **Note.** Saaty pair-wise comparison values (AHP examples) are shown as 'Saaty' and are not edited in the table; edit them in the file. Everything else – names, types, weights, limits, thresholds, values, the used switches, the rows themselves – is edited in place.

3. Running an evaluation (steps 5 to 7)

Step **5 Weights** chooses how the criteria are weighted (section 5 below); step **6 Method and settings** chooses the method and runs it; step **7 Results** shows the ranking.

Step 1. **Method**: one of the ten methods. The five methods of the 1995 program are audited against the dissertation and the legacy output files; the five added in version 2.0 are marked *(provisional)*: verified against closed forms, independent implementations and hand-computed examples, but a published worked example is still to be reproduced (Method Reference, chapters 8-13). The line under the list gives the family, the reference and whether the largest or the smallest value is best.

Step 2. **Standardization**: 0 leaves the data as given; 1 maps every criterion to 0-1 with (v - min)/(max - min) – the usual choice for data in different units; 2 divides by the sum; 3 by the maximum.

Step 3. **Weights** (step 5): given, equal, Best-Worst, or one of the three objective schemes; the switch **Divide the given weights by the sum of their group** is ticked by default – untick it to use the weights exactly as entered (needed to reproduce the published tables of some examples).

Step 4. **Options of the method**: the options of the selected method appear in their own card, taken from the program's method registry (for example `cp_power` and `cp_use_provided_limits` for CP, `vikor_v` for VIKOR, `ra_seed` for RA). PROMETHEE II offers the switch **promethee_partial** (the PROMETHEE I partial ranking).

Step 5. **Advanced**: the text-book switches (all off reproduces the 1995 program exactly; each switch applies one correction of the mathematical review – Web User Guide, section 5.5; Method Reference) and the ids of alternatives or criteria to **exclude** from this run.

Step 6. **Ready to run**: the card on the right lists what the run will use and flags what does not fit – for example Compromise Programming with provided limits while every limit is 0, PROMETHEE thresholds in data units while the values are standardized, or the Best-Worst Method without judgments.

Step 7. **Run the evaluation**. Step 7 opens: the **best alternative** in the headline, the Overall ranking table (rank badges, alternative, name, ranking value with a bar – a longer bar is better –, domination and a note; NF = a zero value with no domination; excluded alternatives are listed last), the **Report** buttons (PDF, HTML, Text), the notes of the engine, and two folding panels: the run **summary** (the settings, the ranking and the warnings in one text) and the **output files** (the two files of the method in the 1995 layout).

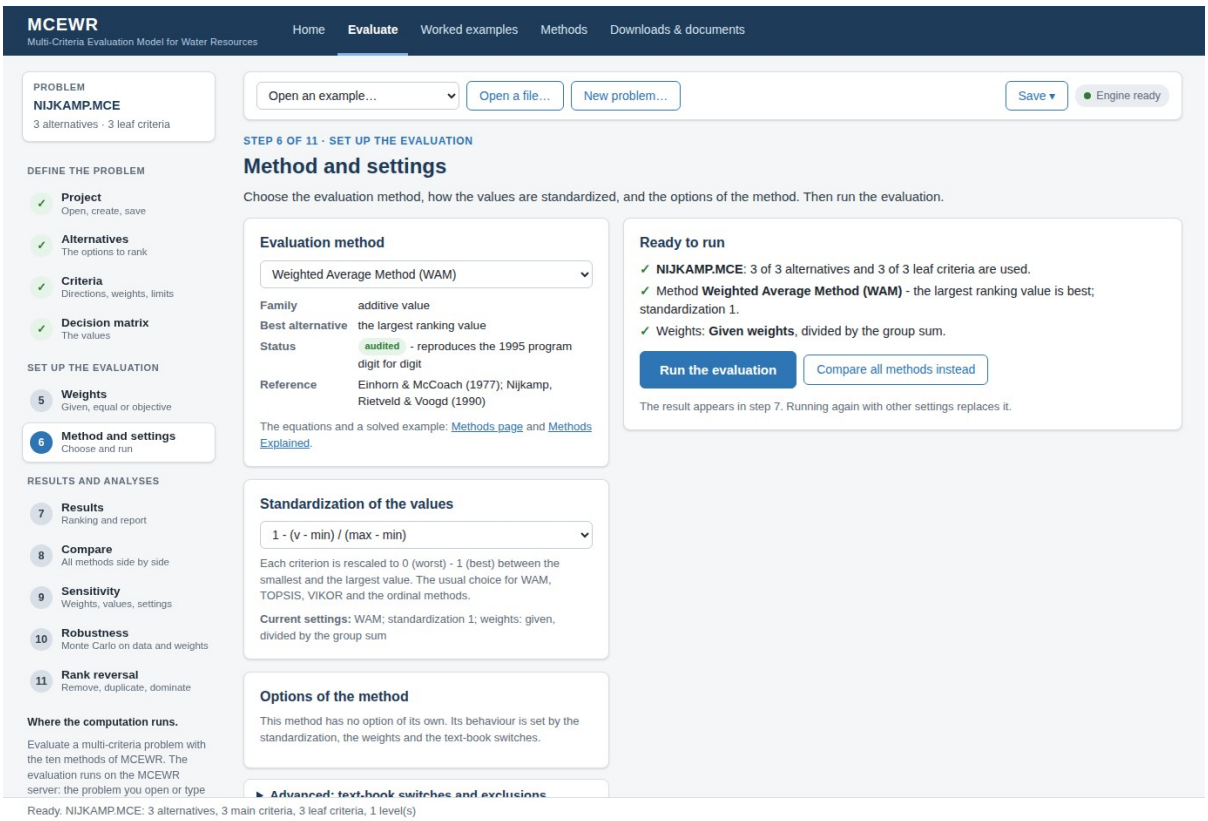


Figure 3.1 – Step 6: WAM on the Nijkamp example, standardization 1, everything else at its default; the Ready to run list on the right.

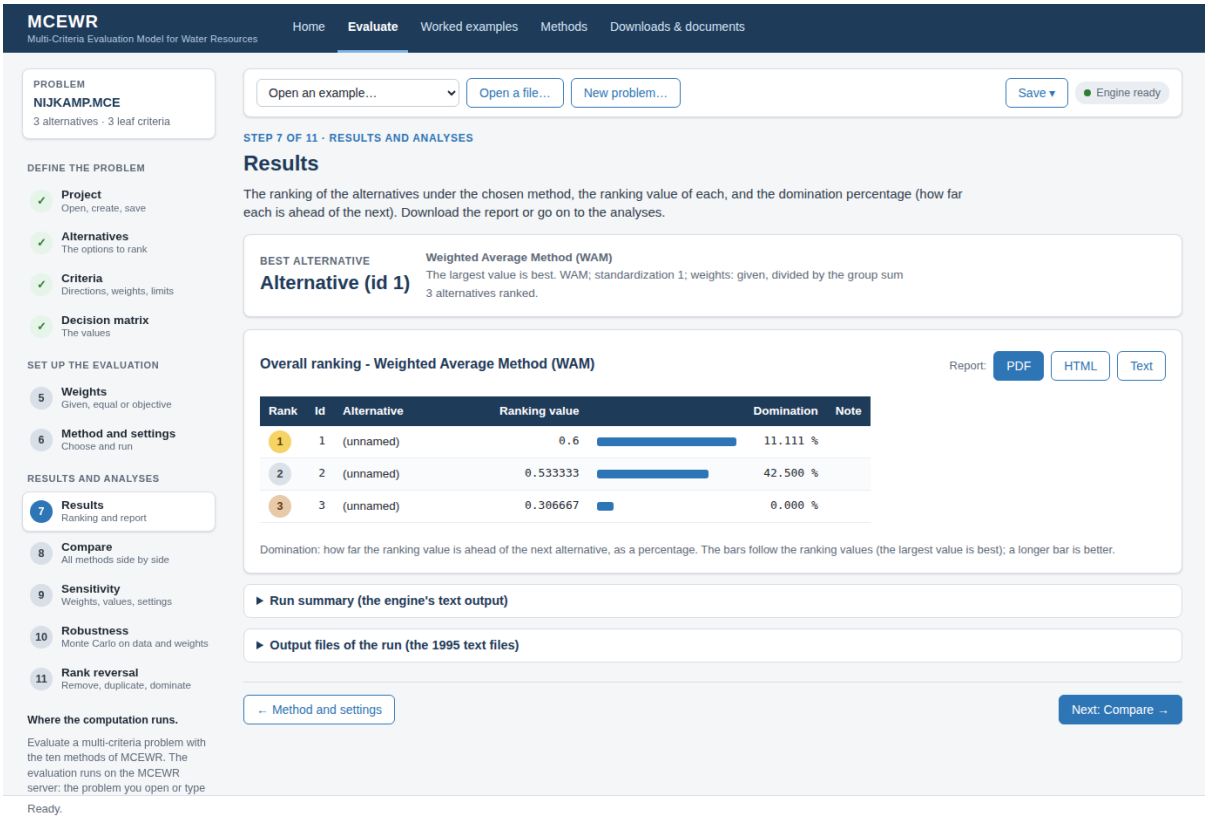


Figure 3.2 – Step 7: the best alternative, the Overall ranking with its bars, the report buttons.

4. Worked examples: every method, with expected results

Each subsection gives the file, the exact settings, the ranking to expect and what to look at. All files are in the *Open an example* list of step 1; the expected results were produced by the engine of the website and are the numbers of the verification examples of the Method Reference – identical to those of the desktop application.

4.1 Weighted Average Method (WAM) – NIJKAMP.MCE

Example: the Nijkamp example of the User Guide, section 6.1.

Step 1. Step **1 Project** → in the *Open an example* list choose `NIJKAMP.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **Weighted Average Method (WAM)**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **1 (v - min) / (max - min)**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the largest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	1	0.6	11.111 %	
2	2	0.533333	42.500 %	
3	3	0.306667	0.000 %	

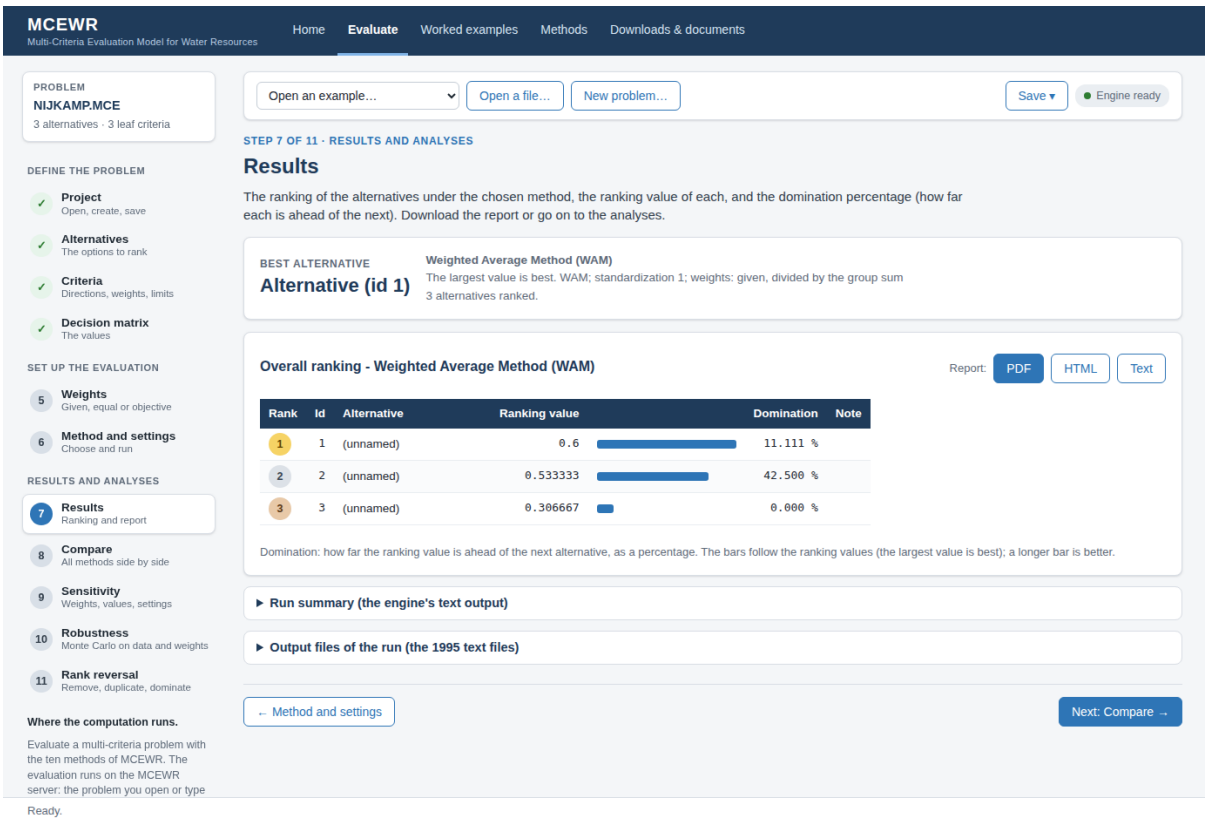


Figure 4.1 – Weighted Average Method (WAM) on NIJKAMP.MCE (step 7, Results).

What to notice

- The ranking value is the weighted sum of the standardized values; the largest is best. Alternative 1 leads with 0.6, alternative 2 follows with 0.5333 – the values of Nijkamp's textbook example.
- The Domination column is the percentage by which an alternative leads the next one in the list; the last alternative has none.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `wam.txt`, `sortwam.txt`.

4.2 PROMETHEE II – PROM_EX.MCE

Example: the six-alternative example of Brans, Vincke & Mareschal (1986).

Step 1. Step **1 Project** → in the *Open an example* list choose `PROM_EX.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Untick the switch **Divide the given weights by the sum of their group** (the weights of this example are used raw, as in the published table).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **PROMETHEE II**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **0 none: the data as given**.

Step 5. Method options: leave the defaults.

Step 6. Switch on **promethee_partial** (the PROMETHEE I partial ranking).

Step 7. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the largest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	5	1.4645	94.141 %	
2	2	0.0858107	216.550 %	
3	4	-0.100013	174.324 %	
4	6	-0.274359	63.151 %	
5	3	-0.447618	62.710 %	
6	1	-0.72832	0.000 %	

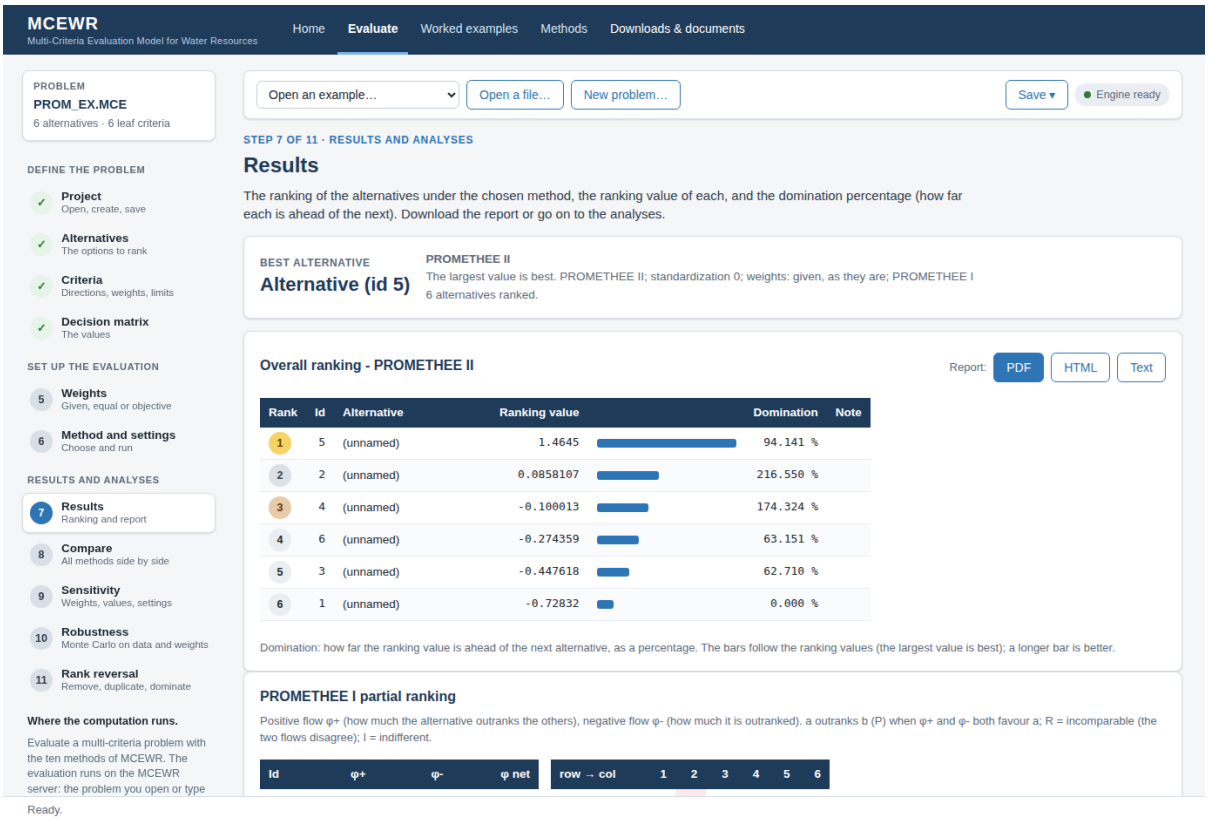


Figure 4.2 – PROMETHEE II on PROM_EX.MCE (step 7, Results).

What to notice

- The ranking value is the net flow φ; the largest is best. With **promethee_partial** on, step 7 adds a card with the positive and negative flows, the P / I / R relation of every pair and the incomparable pairs of the PROMETHEE I partial ranking (also in the summary and in `promethe.txt`).
- The weights of this example are used raw: the flows are those of the original article (Brans, Vincke & Mareschal 1986, Table 3).

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `promethe.txt`, `sortprom.txt`.

4.3 Compromise Programming (CP) – COMP_HAN.MCE

Example: the Han River example with s = 2 and the provided limits.

Step 1. Step **1 Project** → in the *Open an example* list choose `COMP_HAN.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Untick the switch **Divide the given weights by the sum of their group** (the weights of this example are used raw, as in the published table).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **Compromise Programming (CP)**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **0 none: the data as given**.

Step 5. Method options: `cp_power` = 2, `cp_use_provided_limits` = on.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the smallest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	4	1.29546	16.308 %	
2	8	1.50672	19.177 %	
3	9	1.79565	5.969 %	
4	3	1.90283	2.424 %	
5	5	1.94896	8.634 %	
6	7	2.11724	4.805 %	
7	6	2.21899	43.403 %	
8	2	3.1821	143.491 %	
9	1	7.74812	0.000 %	

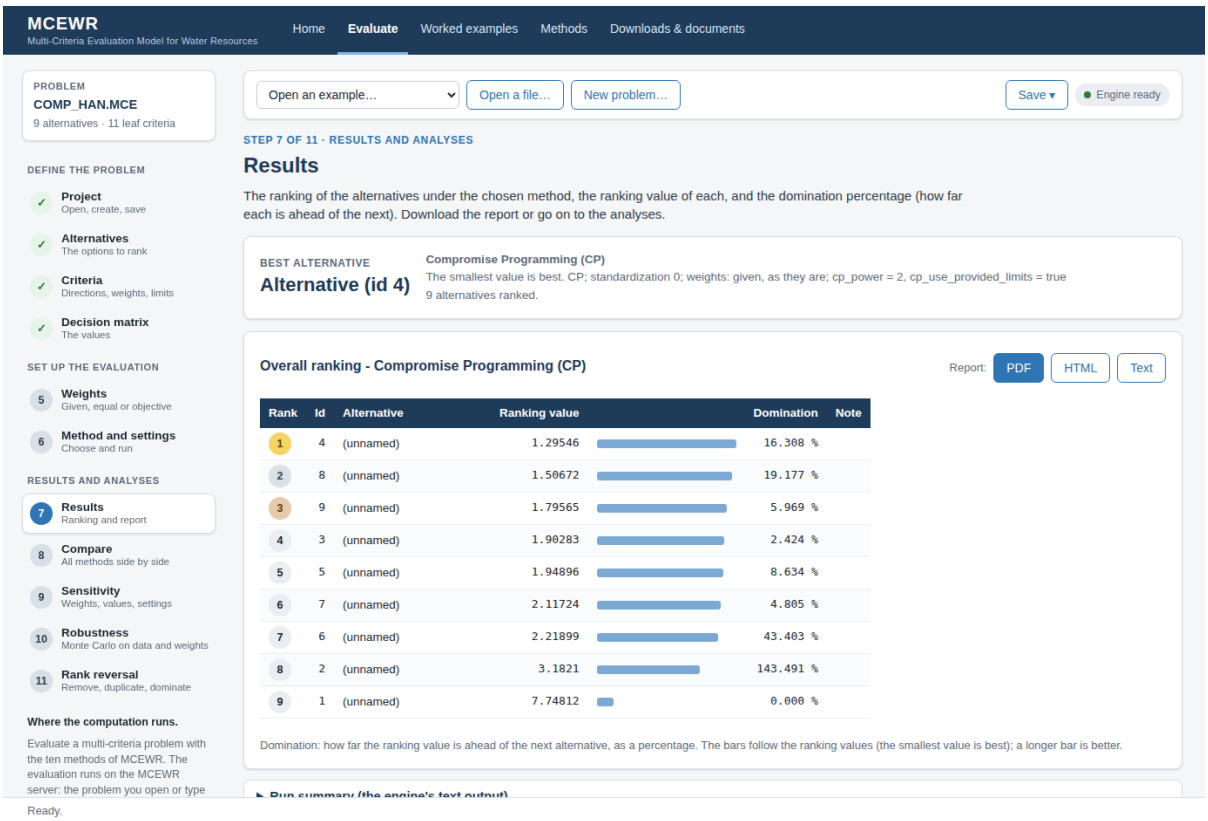


Figure 4.3 – Compromise Programming (CP) on COMP_HAN.MCE (step 7, Results).

What to notice

- The ranking value is the distance to the ideal; the **smallest** is best. `cp_power = 2` is the Euclidean distance; the *provided limits* are the Upper/Lower values of the file, which is what makes the result independent of the other alternatives.
- Set `cp_power = 1` afterwards and run again: that is the SPOTIS computation of section 4.11.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `compro.txt`, `sortcomp.txt`.

4.4 ELECTRE II – ELEC_EX.MCE

Example: the five-alternative ELECTRE II example of the dissertation.

Step 1. Step **1 Project** → in the *Open an example* list choose `ELEC_EX.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Untick the switch **Divide the given weights by the sum of their group** (the weights of this example are used raw, as in the published table).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **ELECTRE II**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **0 none: the data as given**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the smallest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	3	1	100.000 %	
2	1	2	25.000 %	
3	5	2.5	20.000 %	
4	2	3	33.333 %	
5	4	4	0.000 %	

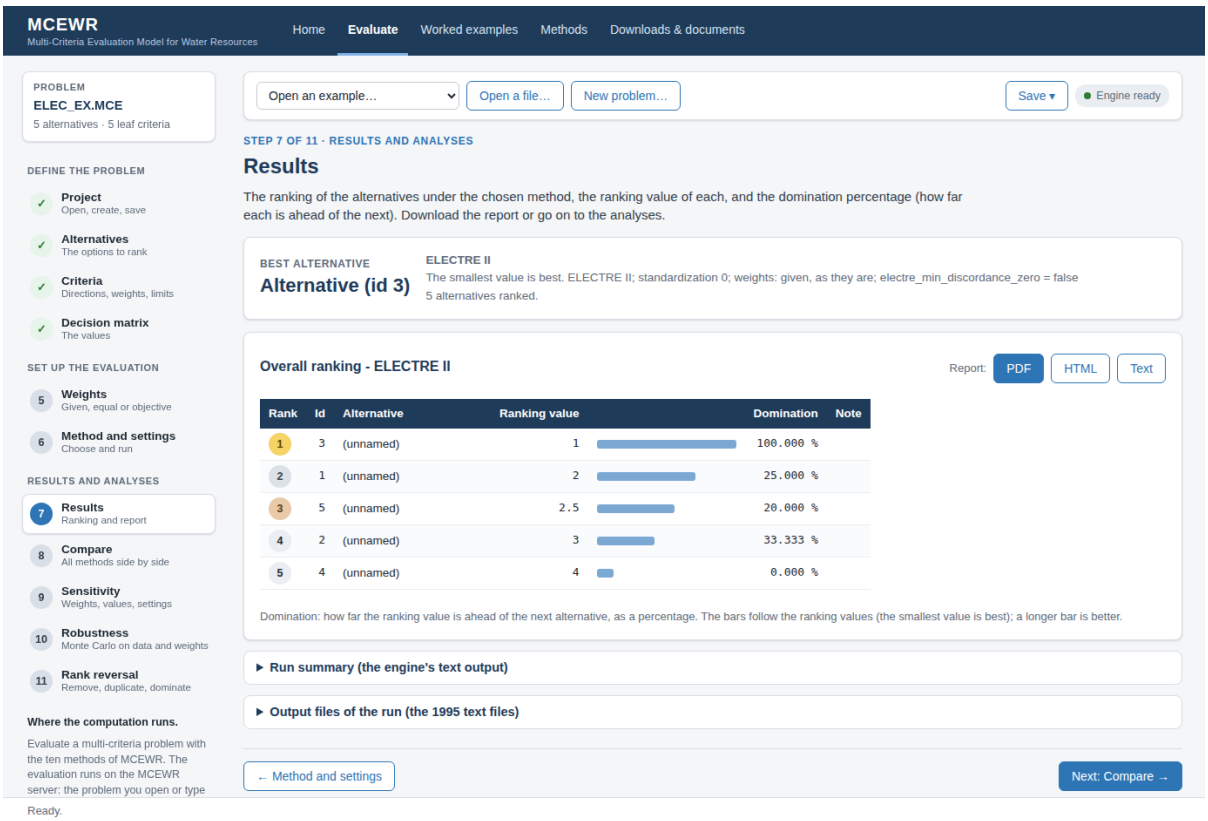


Figure 4.4 – ELECTRE II on ELEC_EX.MCE (step 7, Results).

What to notice

- The ranking value is the median of the forward and the reverse rank of the outranking graph; the **smallest** is best. Tied medians share a rank.
- If a file produces error 201 (a cycle of outranking relations) raise the concordance or lower the discordance thresholds of the file (the ELECTRE II thresholds are in the file's group line; Web User Guide, section 4.2) – or use ELECTRE III.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `electreii.txt`, `sortele2.txt`.

4.5 Analytic Hierarchy Process (AHP) – AHP_EXM3.MCE

Example: the Saaty pair-wise comparison example (Appendix 2 of the dissertation).

Step 1. Step **1 Project** → in the *Open an example* list choose `AHP_EXM3.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **Analytic Hierarchy Process (AHP)**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **0 none: the data as given**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the largest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	1	0.401928	14.297 %	
2	2	0.344462	26.375 %	
3	3	0.25361	0.000 %	

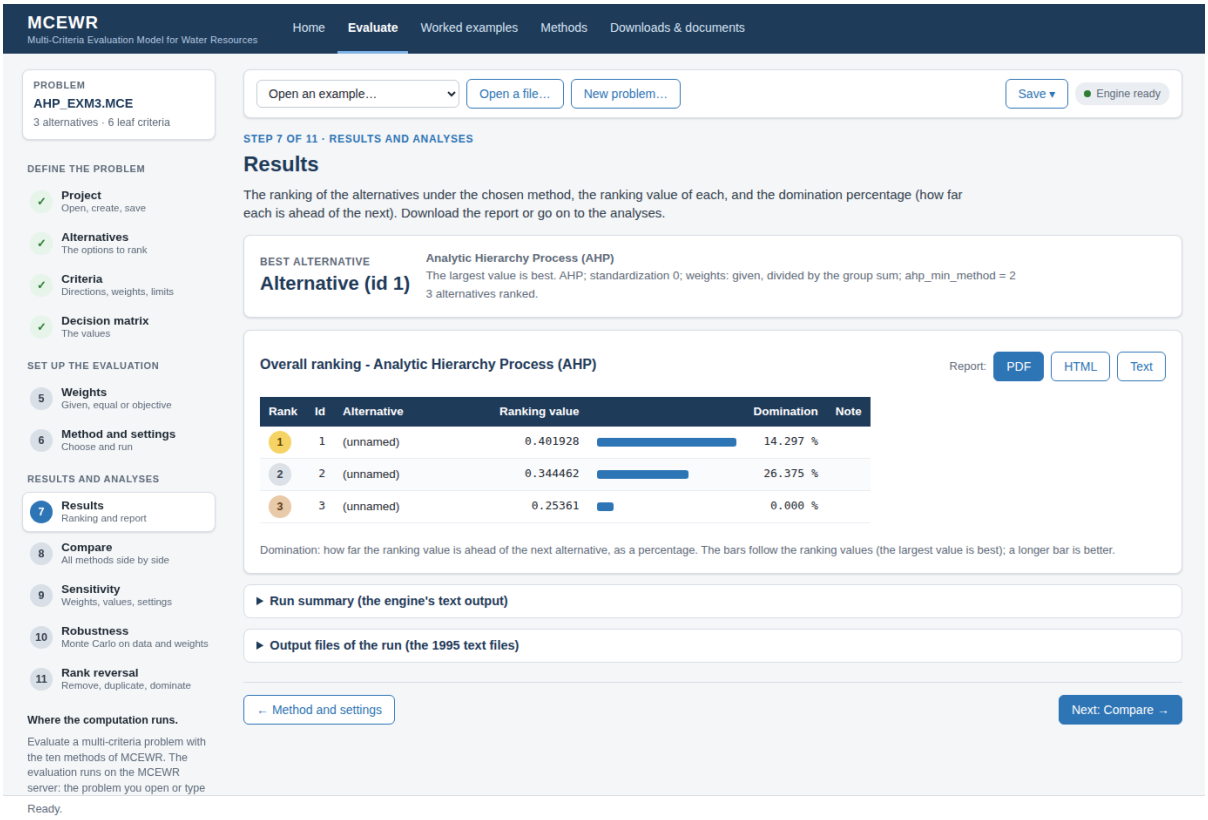


Figure 4.5 – Analytic Hierarchy Process (AHP) on AHP_EXM3.MCE (step 7, Results).

What to notice

- The values of this example are Saaty pair-wise comparisons, so steps 3 and 4 show 'Saaty' in the weight column of the group and in the decision matrix. The summary lists the consistency ratio of every comparison matrix (acceptable when CR ≤ 0.10).
- The ranking value is the global priority; the largest is best.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `ahp.txt`, `sortahp.txt`.

4.6 Expected Value (EV) – NIJKAMP.MCE

Example: the Nijkamp example, ordinal weights and scores.

Step 1. Step **1 Project** → in the *Open an example* list choose `NIJKAMP.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **Expected Value (EV)**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **1 (v - min) / (max - min)**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the largest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	2	0.583333	9.524 %	
2	1	0.527778	26.316 %	
3	3	0.388889	0.000 %	

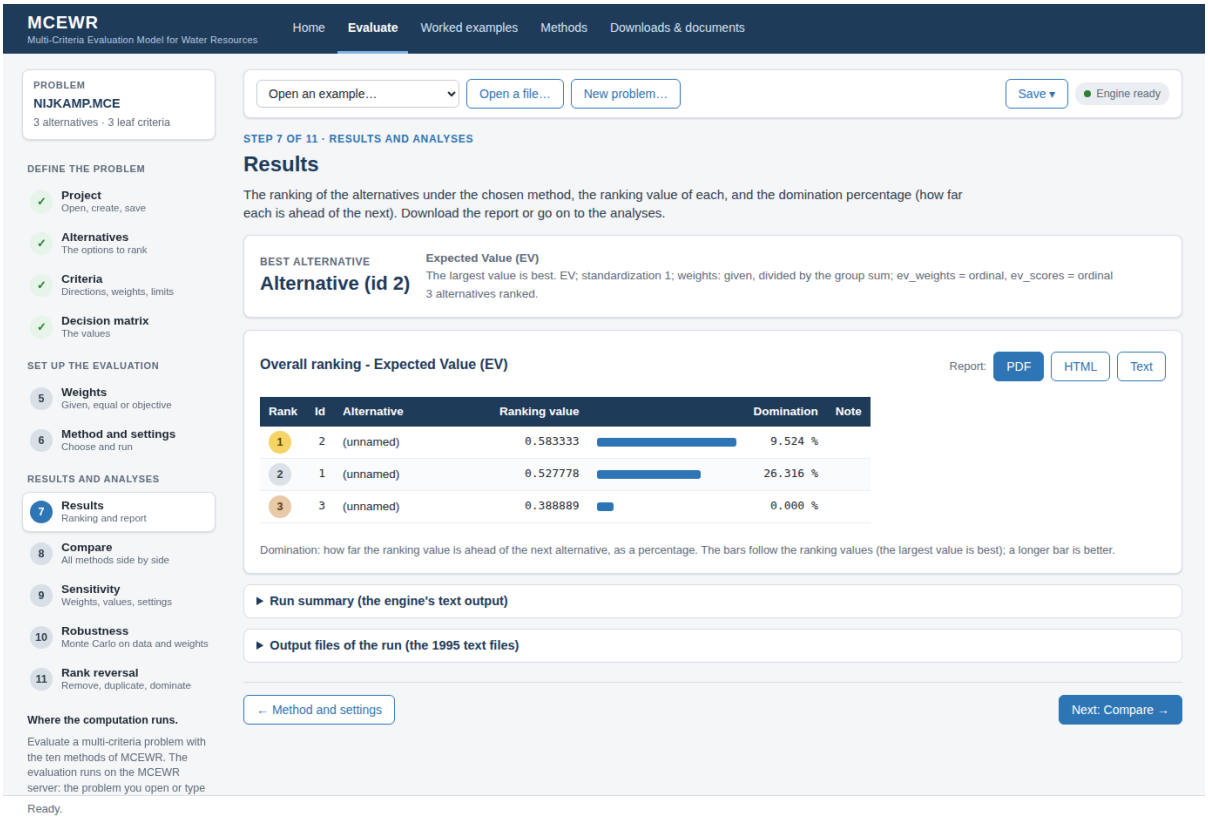


Figure 4.6 – Expected Value (EV) on NIJKAMP.MCE (step 7, Results).

What to notice

- EV replaces the weights by the expected values of their ranking (rank-order-centroid weights) and the scores by the expected values of the alternatives' ranking under each criterion; the largest is best. The summary carries the line *Status: PROVISIONAL*: the formulation is verified against closed forms and a hand-computed example, and a published worked example is still to be reproduced (Method Reference, chapter 8).
- Set `ev_weights` and `ev_scores` to `cardinal` and run again: the result is then exactly WAM.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `expval.txt`, `sortev.txt`.

4.7 Regime Analysis (RA) – NIJKAMP.MCE

Example: the Nijkamp example, regime probabilities over the ordinal weight set.

Step 1. Step **1 Project** → in the *Open an example* list choose `NIJKAMP.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **Regime Analysis (RA)**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **1 (v - min) / (max - min)**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the largest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	1	1	50.000 %	
2	2	0.5	100.000 %	
3	3	0	0.000 %	

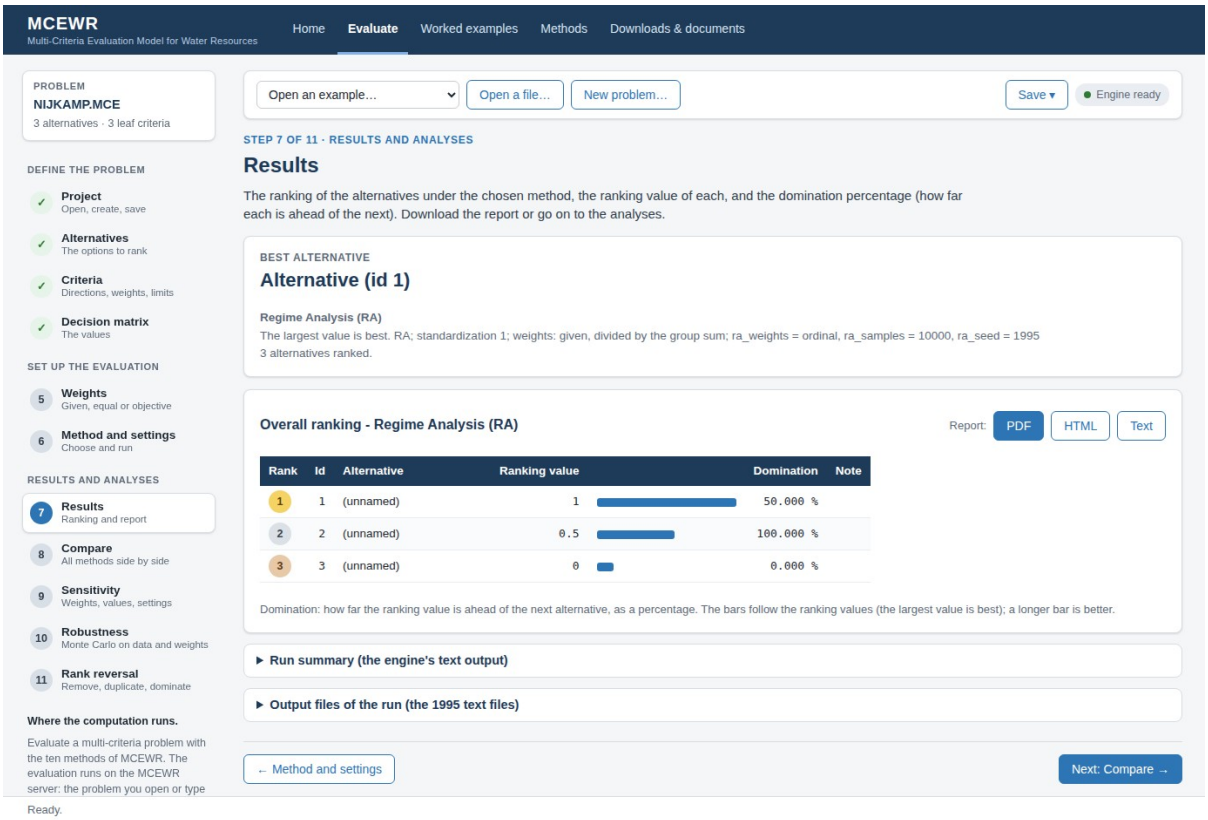


Figure 4.7 – Regime Analysis (RA) on NIJKAMP.MCE (step 7, Results).

What to notice

- The ranking value is the mean probability of beating another alternative over the set of weight vectors consistent with the ranking of the declared weights; the largest is best. On this example every pair is decided with certainty, hence the values 1, 0.5 and 0.
- The seed `ra_seed` is recorded in the detail file: the same seed reproduces every number; `ra_weights = cardinal` lets the declared weights decide each pair outright.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `regime.txt`, `sortra.txt`.

4.8 TOPSIS – NIJKAMP.MCE

Example: the Nijkamp example, closeness to the ideal and the anti-ideal.

Step 1. Step **1 Project** → in the *Open an example* list choose `NIJKAMP.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **TOPSIS**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **1 (v - min) / (max - min)**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the largest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	1	0.552959	3.032 %	
2	2	0.536191	37.081 %	
3	3	0.337365	0.000 %	

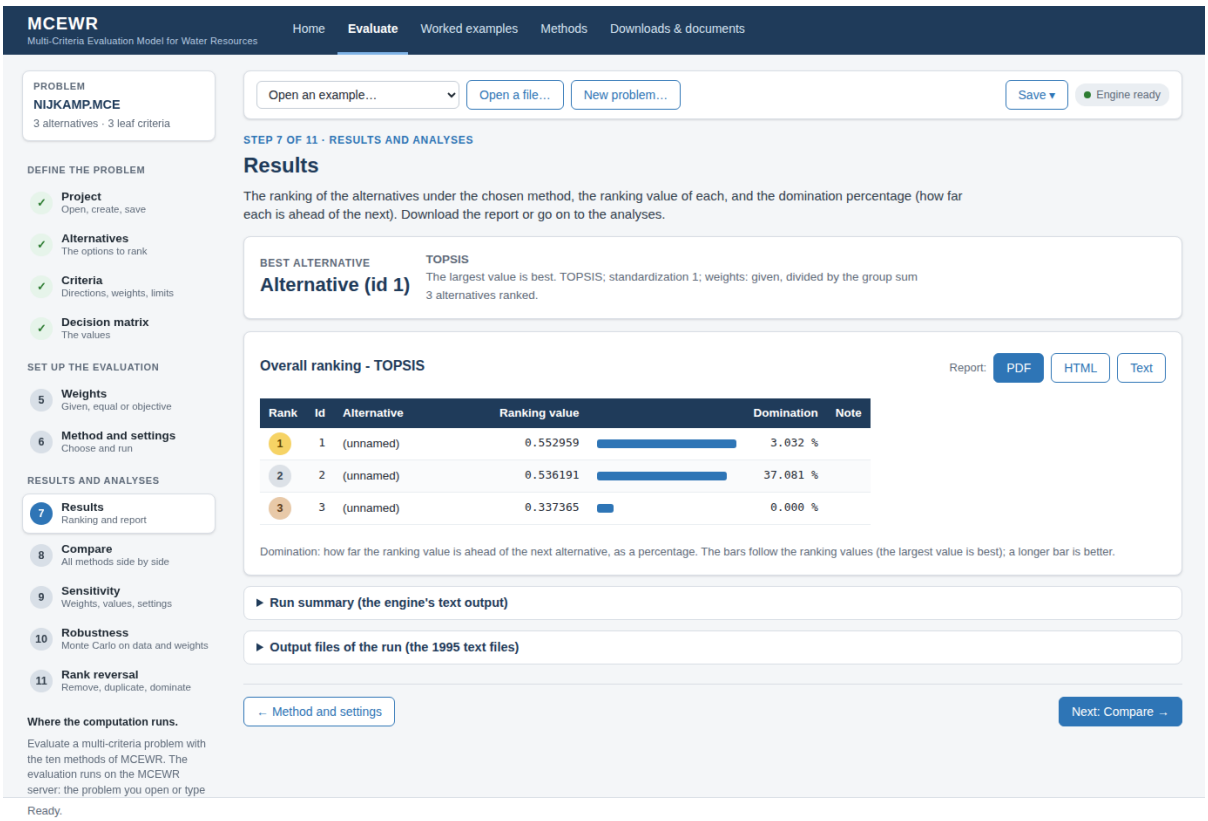


Figure 4.8 – TOPSIS on NIJKAMP.MCE (step 7, Results).

What to notice

- The ranking value is the closeness C = D− / (D+ + D−) to the ideal and the anti-ideal; the largest is best. Compare with WAM on the same file: the order is the same here, but TOPSIS can reverse when an alternative is added or removed (section 8).
- With standardization 0 the method is the text-book TOPSIS on the raw data; with another standardization it is TOPSIS on the standardized data.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `topsis.txt`, `sorttop.txt`.

4.9 VIKOR – NIJKAMP.MCE

Example: the Nijkamp example, compromise index Q with v = 0.5.

Step 1. Step **1 Project** → in the *Open an example* list choose `NIJKAMP.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **VIKOR**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **1 (v - min) / (max - min)**.

Step 5. Method options: `vikor_v` = 0.5.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the smallest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	2	0.113636	340.000 %	
2	1	0.5	100.000 %	
3	3	1	0.000 %	

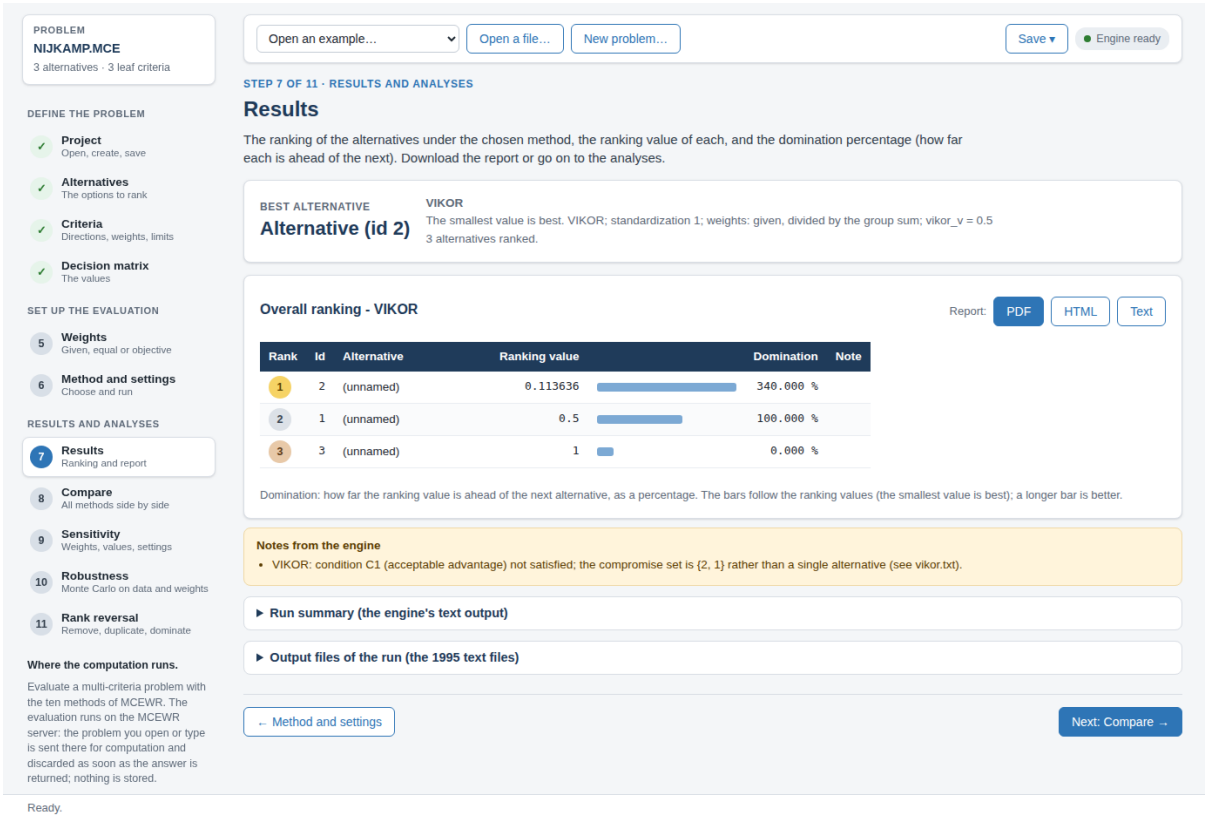


Figure 4.9 – VIKOR on NIJKAMP.MCE (step 7, Results).

What to notice

- The ranking value is the compromise index Q; the **smallest** is best. `vikor_v` = 0.5 weighs the group utility S and the individual regret R equally; try 1 and 0 to rank by S alone or by R alone.
- The warnings under the summary tell whether the two acceptance conditions of the compromise solution hold; when they do not, the set of compromise solutions is listed.

The run issues a warning, shown under the summary: “VIKOR: condition C1 (acceptable advantage) not satisfied; the compromise set is {2, 1} rather than a single alternative (see vikor.txt).”.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `vikor.txt`, `sortvik.txt`.

4.10 ELECTRE III – ELEC_HAN.MCE

Example: the Han River ELECTRE example with the thresholds of the file.

Step 1. Step **1 Project** → in the *Open an example* list choose `ELEC_HAN.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Leave the switch **Divide the given weights by the sum of their group** ticked (every group sums to 1).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **ELECTRE III**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **1 (v - min) / (max - min)**.

Step 5. Method options: leave the defaults.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the smallest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	8	1	50.000 %	
2	9	1.5	100.000 %	
3	4	3	16.667 %	
4	2	3.5	0.000 %	
4	3	3.5	0.000 %	
4	6	3.5	0.000 %	
4	7	3.5	14.286 %	
5	5	4	12.500 %	
6	1	4.5	0.000 %	

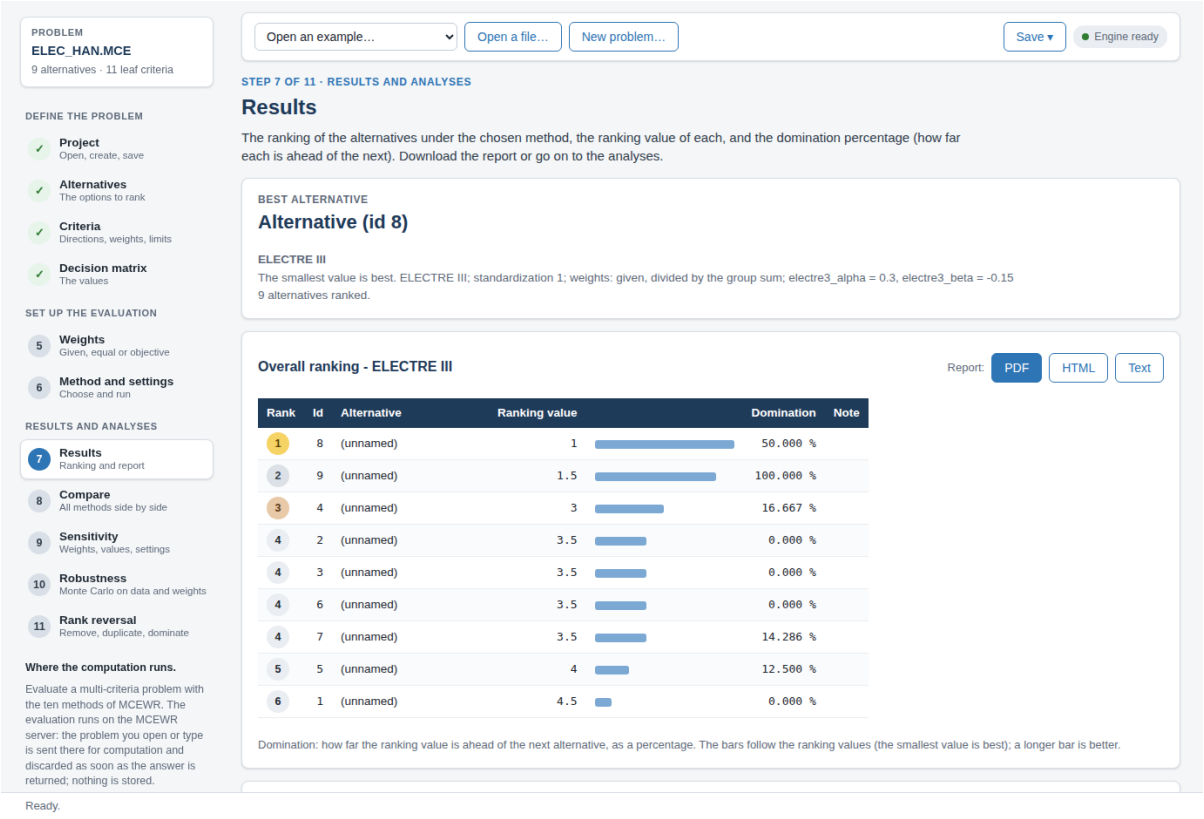


Figure 4.10 – ELECTRE III on ELEC_HAN.MCE (step 7, Results).

What to notice

- Step 7 adds the card **ELECTRE III distillations and partial preorder**: the two distillations, the position of every alternative in each, and the relation matrix of the final partial preorder (P = ahead in both, I = equal in both, R = incomparable) with the list of incomparable pairs. On this file the two distillations disagree on alternative 5, which is incomparable with 1, 2, 3, 4, 6 and 7; the complete ranking orders it by the mean position, the method itself leaves those pairs unordered.
- The ranking value is the median position of the descending and the ascending distillations; the **smallest** is best. The pseudo-criteria use the p and q of the file; a veto can be given per criterion in step 3 (switch on *Show thresholds*; column *Veto*, 0 = none) – it is kept by the project and CSV formats but not by the 1995 .MCE format.
- `electre3_alpha` and `electre3_beta` are the coefficients of the discrimination threshold $s(\lambda) = \alpha + \beta\lambda$ (defaults 0.30 and −0.15).

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `electre3.txt`, `sortele3.txt`.

4.11 SPOTIS (CP with the provided limits, s = 1) – COMP_HAN.MCE

Example: the Han River example as SPOTIS: CP with the provided limits and s = 1.

Step 1. Step **1 Project** → in the *Open an example* list choose `COMP_HAN.MCE`. Steps 2 to 4 fill (alternatives, criteria, decision matrix); look at the weights and the types in step 3 before evaluating.

Step 2. Step **5 Weights** → keep the card **Given weights**. Untick the switch **Divide the given weights by the sum of their group** (the weights of this example are used raw, as in the published table).

Step 3. Step **6 Method and settings** → in the *Evaluation method* list select **SPOTIS (CP with the provided limits, s = 1)**. The card states the family, the reference, the status and whether the largest or the smallest value is best.

Step 4. Standardization: choose **0 none: the data as given**.

Step 5. Method options: `cp_power` = 1, `cp_use_provided_limits` = on.

Step 6. Leave the text-book switches off (in *Advanced*), read the **Ready to run** list, and press **Run the evaluation**. The page moves to step **7 Results**: the best alternative, the ranking with its bars, the report buttons, the notes of the engine, the run summary and the output files.

Expected results (the Overall ranking table of step 7; the smallest ranking value is best):

Rank	Alternative	Ranking value	Domination	Note
1	8	2.24765	7.718 %	
2	9	2.42113	21.622 %	
3	4	2.94463	0.438 %	
4	5	2.95754	17.484 %	
5	7	3.47464	9.774 %	
6	3	3.81427	3.142 %	
7	6	3.93411	13.984 %	
8	2	4.48424	28.926 %	
9	1	5.78134	0.000 %	

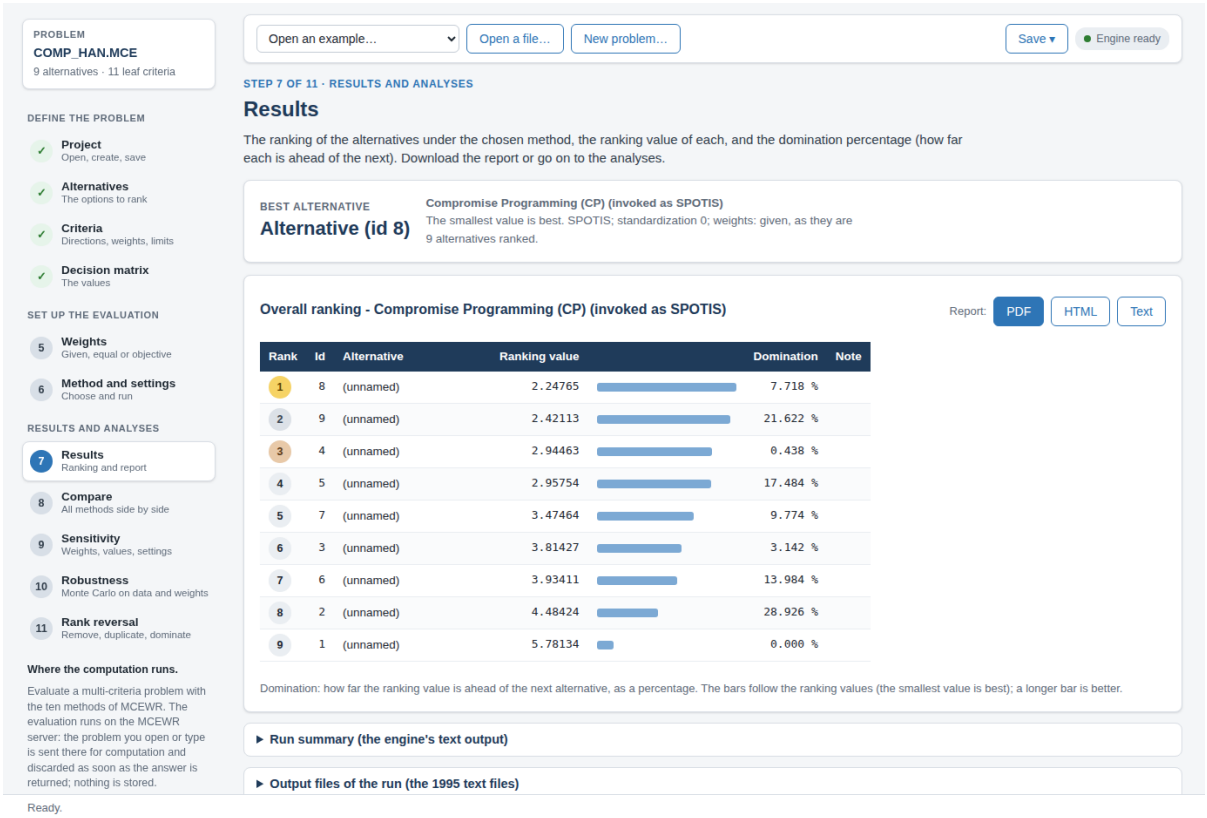


Figure 4.11 – SPOTIS (CP with the provided limits, s = 1) on COMP_HAN.MCE (step 7, Results).

What to notice

- Compromise Programming with the provided limits and s = 1 is, term for term, the SPOTIS method of 2020 (Method Reference, section 5.7): the smallest value is best, and the result does not depend on which other alternatives are present.
- This is the rank-reversal-free computation: section 8 shows 0 reversals for it on every test.

Output files (the *Output files of the run* panel at the foot of step 7; each opens the file in the 1995 layout): `compro.txt`, `sortcomp.txt`.

5. Step 5 Weights: given, equal, Best-Worst and objective weights

Step **5 Weights** chooses, with six cards, how the criteria are weighted in every run: **Given weights** (the weight column of step 3, divided by the sum of its group unless the switch is off), **Equal weights** (1/n in every group – the neutral starting point), the **Best-Worst Method** (section 5.1) and the three objective schemes derived from the data: **Entropy** (a criterion whose values differ much weighs more), **CRITIC** (much dispersion and little correlation with the other criteria) and **MEREC** (removing the criterion changes the overall performances much). Below the cards, the table **The weights each method gives** shows, for every group of sibling criteria, the weights of every scheme; the chosen scheme is highlighted, and the run uses it. Open `COMP_HAN.MCE` (the Han River example, 9 alternatives, 11 criteria) and read:

Group	Criterion	Type	Given	Entropy	CRITIC	MEREC
main criteria	1	max	0.2000	0.1225	0.0771	0.2054
main criteria	2	max	0.1500	0.1306	0.1505	0.0649
main criteria	3	max	0.0500	0.0184	0.0603	0.0464
main criteria	4	max	0.0500	0.0785	0.0709	0.0598
main criteria	5	max	0.0750	0.3090	0.0805	0.1579
main criteria	6	max	0.1750	0.1136	0.0696	0.1589
main criteria	7	max	0.0750	0.0615	0.1447	0.0922
main criteria	8	max	0.0500	0.0673	0.0638	0.0719
main criteria	9	max	0.0250	0.0094	0.0629	0.0310
main criteria	10	max	0.1000	0.0428	0.1454	0.0689
main criteria	11	max	0.0500	0.0465	0.0741	0.0426

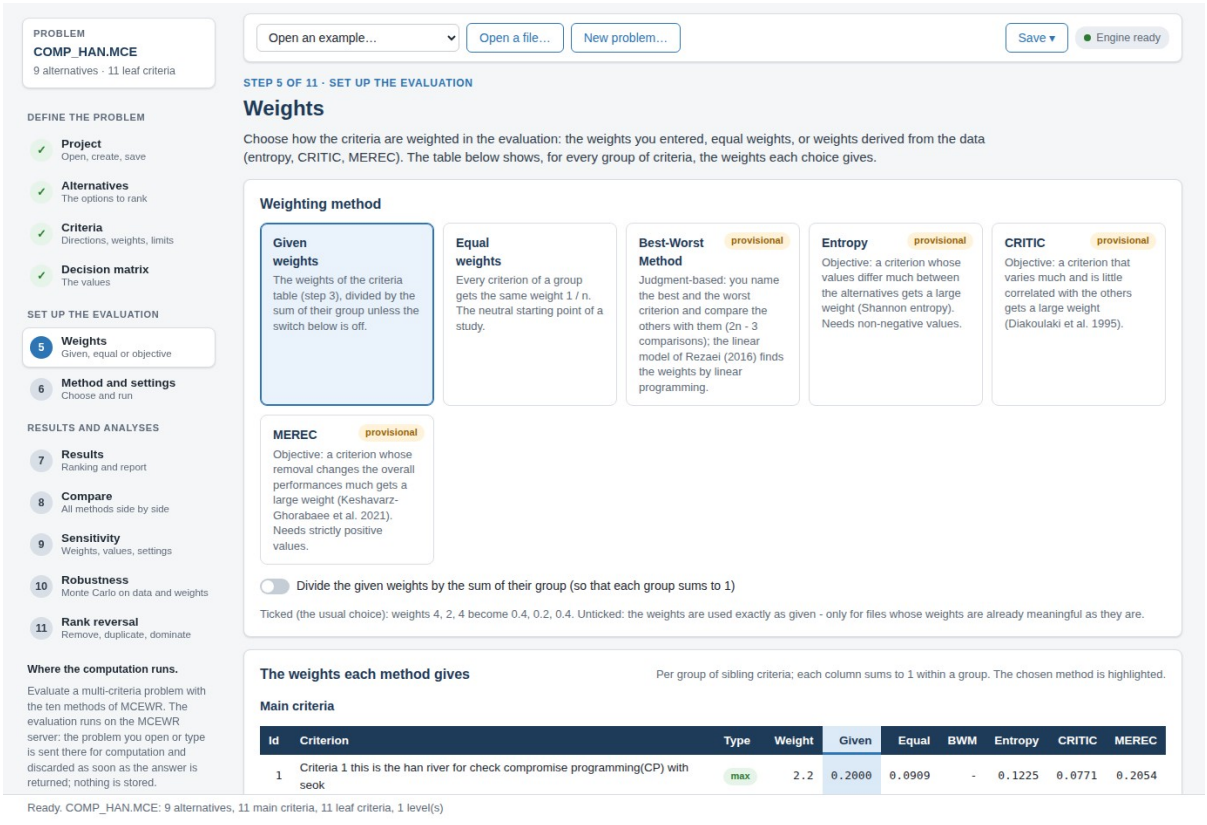


Figure 5.1 – Step 5 for the Han River example: the six cards and the table of the weights each scheme gives.

Entropy and MEREC need positive data (a note under the table says when a group cannot be computed); a group with aggregated or Saaty criteria keeps its given weights under the objective schemes.

5.1 The Best-Worst Method

The Best-Worst Method (Rezaei 2015; linear model 2016) derives the weights of a group of n criteria from 2n − 3 comparisons instead of a full pairwise matrix. Choose the card **Best-Worst Method**: a card **Best-Worst judgments** appears with one block per group of criteria. Open `NIJKAMP.MCE` and, for the main criteria:

Step 1. **Best (most important)**: criterion 1; **Worst (least important)**: criterion 2. The comparisons of the best with itself and of the worst with itself are fixed at 1.

Step 2. **Best over this – a(B,j)**: how much more important the best criterion is than each criterion, on the 1–9 scale (1 = equally, 3 = moderately, 5 = strongly, 7 = very strongly, 9 = extremely more important). Enter 4 for criterion 2 and 2 for criterion 3. The 4 on the worst criterion is the best-to-worst comparison; it is mirrored automatically into the other vector.

Step 3. **This over worst – a(j,W)**: how much more important each criterion is than the worst. Enter 2 for criterion 3.

Step 4. Press **Compute the weights**. The engine solves the linear program of Rezaei (2016) – minimize the largest violation ξ of the comparisons – and shows the weights in the BWM column of the table: **0.5714, 0.1429, 0.2857** (4 : 1 : 2). The line under the block gives ξ* and the consistency ratios.

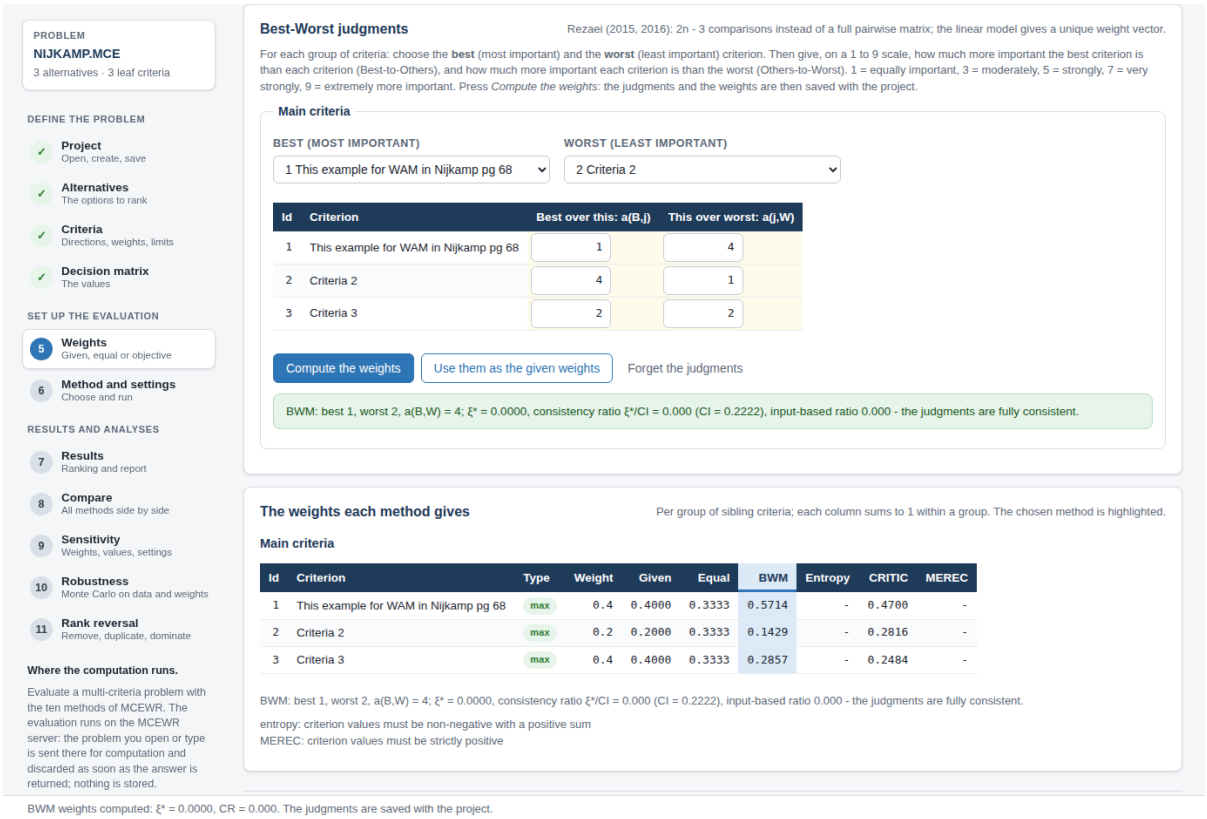


Figure 5.2 – The Best-Worst judgments of the Nijkamp example and the weights they give (the BWM column, highlighted).

These judgments are **fully consistent** – every product a(B,j)·a(j,W) equals a(B,W) = 4 – so ξ* = 0 and the weights are exactly a(j,W) divided by their sum. Real judgments rarely are: ξ* then grows with the contradictions, and two ratios help to judge them – the **consistency ratio** ξ*/CI (CI is the largest ξ* any system with this many criteria and this best-to-worst value can produce; 0 = perfect, 1 = worst possible) and the **input-based ratio** of Liang, Brunelli and Rezaei (2020), read from the comparisons alone. A ratio well above 0.1–0.2 is a reason to look at the judgments again. The judgments are saved in the project file and travel with it; **Use them as the given weights** copies the computed weights into the weight column of step 3; **Forget the judgments** removes them. A hierarchy has one block per group; groups without judgments keep their given weights, and the run says so in its notes. Run

WAM with standardization 1 and the BWM weights: alternative 2 leads with 0.6667, then 1 with 0.4286 and 3 with 0.3905 – the same ranking as with the weights 4, 1, 2 entered by hand.

6. Comparing the methods

Step 1. Open `COMP_HAN.MCE`; in step 6 choose standardization 1, and in step 5 leave the given weights (the settings of steps 5 and 6 apply to every method of the comparison).

Step 2. Open step **8 Compare**, keep the ten methods ticked (SPOTIS is unticked by default: it duplicates CP with the provided limits), and press **Run the comparison**. Each method runs on the server with the same settings; the tables fill within a second or two.

Step 3. Read the rank of every alternative under every method, the mean rank, the consensus ordering and the number of first places; below, the agreement between the methods (Spearman's rho of the rank vectors: 1 = identical, 0 = unrelated, −1 = opposite). **Report** (PDF or HTML) downloads the comparison report.

Expected: the comparison of nine methods; ELECTRE II is not evaluated on this file with these settings – its outranking graph has a cycle (error 201) – and the note under the tables says so; the comparison goes on with the other nine. The consensus ordering is 9, 8, 4, 5, 7, 3, 6, 2, 1.

Alternative	Name	WAM	PROMETHEE II	CP	AHP	EV	RA	TOPSIS	VIKOR	ELECTRE III	Mean	Cons.	1st
9		1	2	2	1	2	3	1	2	1	1.67	1	4
8		2	1	1	2	1	1	3	3	2.5	1.83	2	4
4		4	4	3	4	5	4	2	1	4.5	3.50	3	1
5		3	3	4	3	3	2	5	6	2.5	3.50	4	0
7		5	5	5	5	4	5	9	8	4.5	5.61	5	0
3		6	6	6	6	6	6	7	4	6	5.89	6	0
6		7	7	7	7	7	7	8	5	7	6.89	7	0
2		8	8	8	8	8	8	6	7	8	7.67	8	0
1		9	9	9	9	9	9	4	9	9	8.44	9	0

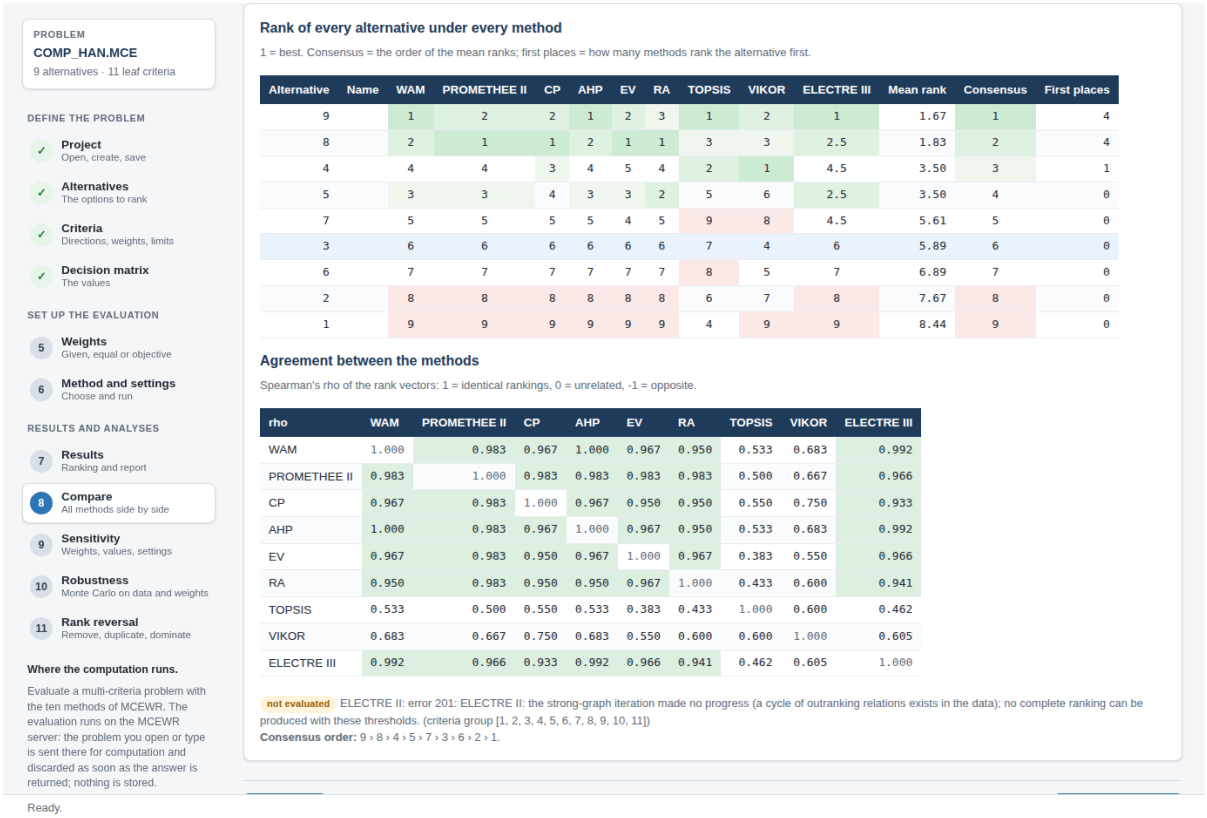


Figure 6.1 – Step 8: nine methods on the Han River example; the rank cells are shaded green for the first places and red for the last.

Read the rho matrix: the additive and distance methods (WAM, CP, AHP, EV) agree almost perfectly with each other; TOPSIS and VIKOR stand apart. Where methods disagree, the choice of method matters for this problem and the reasons – compensation versus outranking, distance to the ideal versus pairwise dominance – deserve attention (Method Reference, chapter 14).

7. Sensitivity: one input at a time

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

7.1 The weights

Step 1. Open `NIJKAMP.MCE`; in step 6 select **WAM** and standardization 1.

Step 2. Open step **9 Sensitivity**, part **Weights**. Group: *the main criteria*; steps of the sweep: 8. Press **Run the weight sensitivity**. The blue headline states the base winner and the most sensitive criterion.

Step 3. The table gives, per criterion, its base share of the weight, the winner in every interval of its share (the other criteria keep their proportions) and the nearest **critical share** – the smallest change of that weight that changes the decision.

Expected (the winner of the base run is alternative 1):

Criterion	Name	Base share	Winner by interval of the share	Critical share	Change
1	This example for WAM i	0.4000	0.000-0.438: 1; 0.438-1.000: 2	0.4375	9.4 %
2	Criteria 2	0.2000	0.000-0.143: 2; 0.143-1.000: 1	0.1429	28.6 %
3	Criteria 3	0.4000	0.000-0.333: 2; 0.333-1.000: 1	0.3333	16.7 %

These critical shares are exact: under WAM the value of an alternative is linear in the weights, and the winner changes when criterion 1 reaches 7/16 = 0.4375 of the weight, when criterion 3 falls to 1/3, and when criterion 2 falls to 1/7 – criterion 1 is the most sensitive (a 9.4 % change of its weight moves the decision).

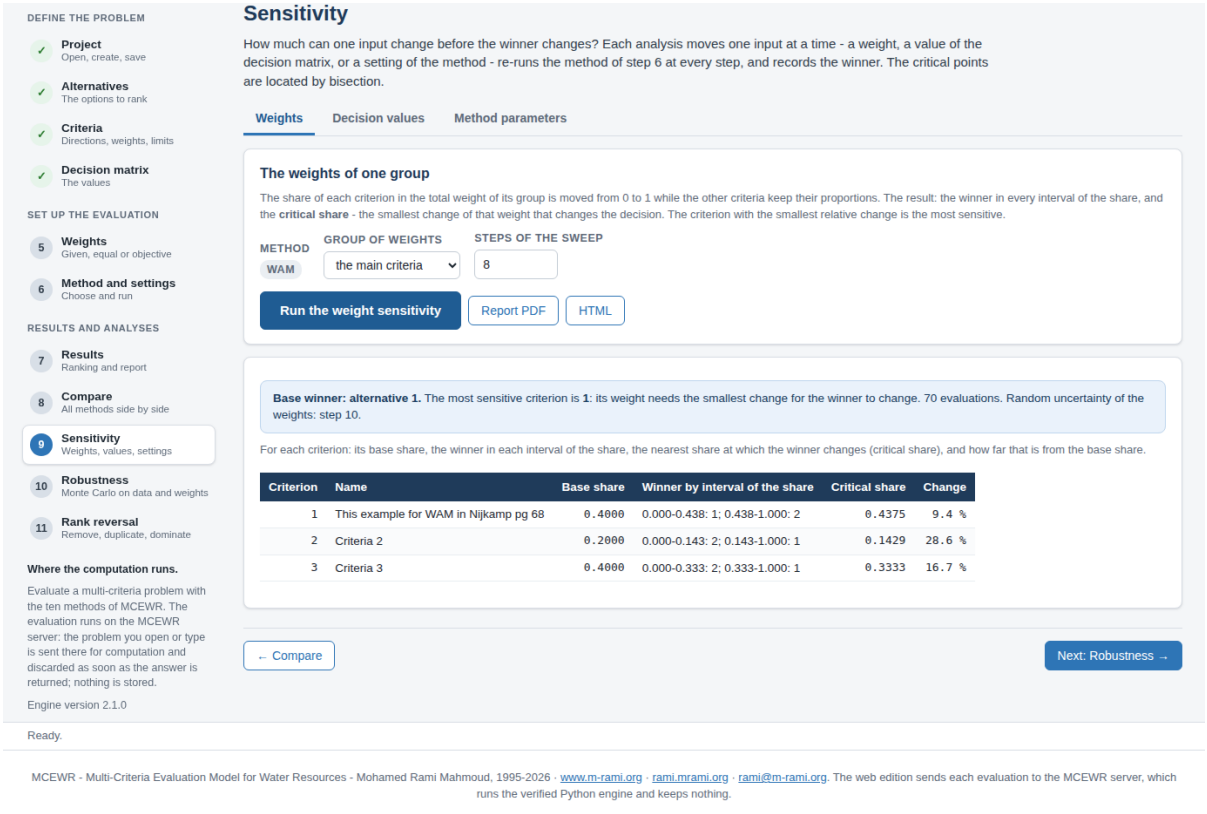


Figure 7.1 – Step 9, part Weights: WAM on the Nijkamp example.

7.2 The decision values

Step 1. Part **Decision values**. Alternatives to vary: *the base winner only*; steps per value: 20. Press **Run the value sensitivity**.

Step 2. Each value of alternative 1 is moved from *value − range* to *value + range* of its criterion (the range over the three alternatives), and the winner is recorded. 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 when a value is 0 and lets criteria in different units be compared.

Expected:

Alt.	Crit.	Base value	Winner by interval of the value	Critical value	Change
1	1	50000	0 to 1e+05: 1	-	-
1	2	500	-100 to 33.33: 2; 33.33 to 1100: 1	33.33	77.8 %
1	3	80	50 to 70: 2; 70 to 110: 1	70	33.3 %

Read: alternative 1 keeps first place whatever its value on criterion 1 (the sweep never changes the winner); on criterion 3 it loses first place when its value falls from 80 to **70** – a change of 10, one third of the range 30 – and that is the most sensitive value of the study; on criterion 2 the value must fall from 500 to 33.3 (77.8 % of the range) before alternative 2 wins. The critical value 70 can be checked by hand: at 70 the standardized values of criterion 3 become 1, 0.5 and 0, and the two weighted sums meet.

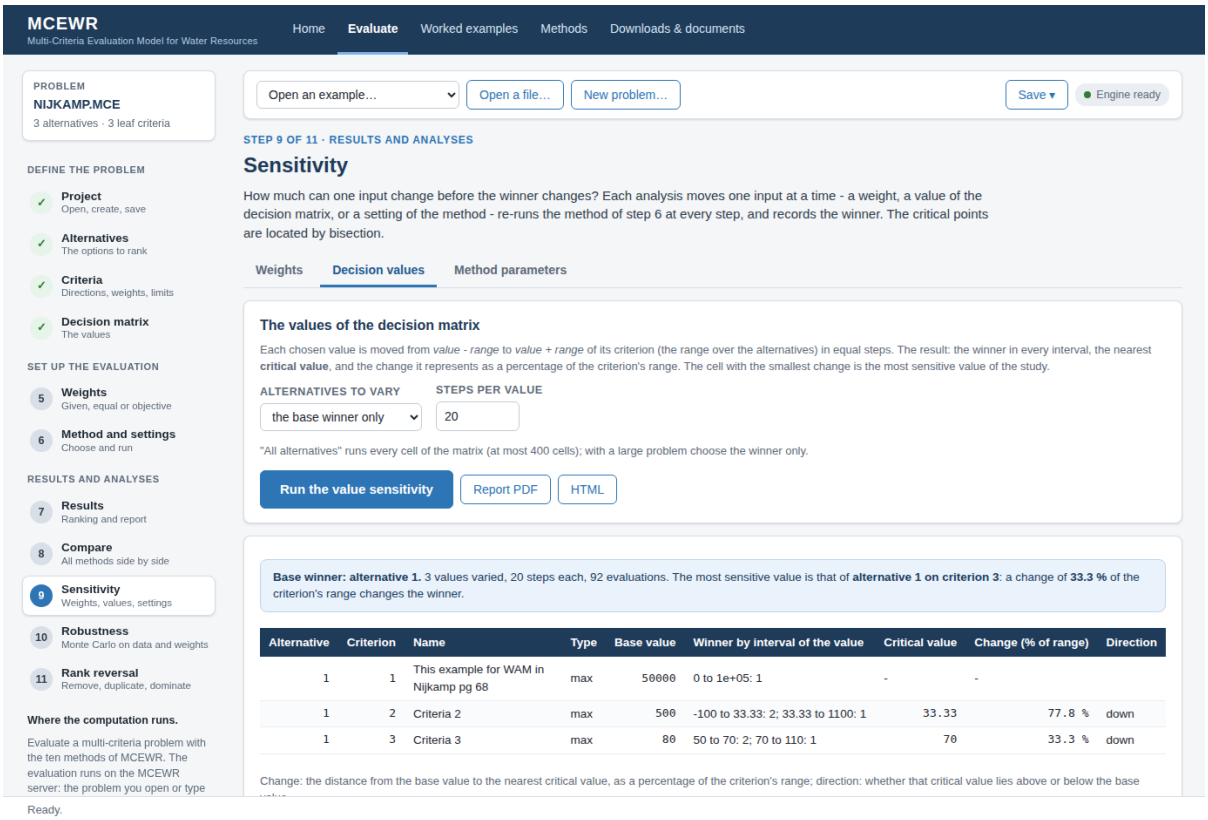


Figure 7.2 – Step 9, part Decision values: the three values of the winner, and the most sensitive one.

7.3 The settings of the method

Step 1. In step 6 select **VIKOR** (keep standardization 1). Open step 9, part **Method parameters**: the list offers the numeric settings of the method – the standardization code and, for VIKOR, the strategy weight `vikor_v`. Choose `vikor_v` and the values 0, 0.25, 0.5, 0.75, 1 (the page proposes them). Press **Run the parameter sensitivity**.

Step 2. The table gives, for every value, the winner, the whole order (ties joined by =) and Spearman's rho of the positions against the base run.

Expected (the base run is v = 0.5):

Value of vikor_v	Winner	Order	Rho vs base	Same positions
0	2	2, 1, 3	0.866	no
0.25	2	2, 1, 3	1.000	yes
0.5	2	2, 1, 3	1.000	yes
0.75	2	2, 1, 3	1.000	yes
1	1	1, 2, 3	0.500	no

Read: the VIKOR compromise favours alternative 2 for every strategy weight up to 0.75 and switches to alternative 1 at v = 1 (group utility only); at v = 0 (individual regret only) alternatives 1 and 3 tie. The same tool answers whether the standardization alone changes the ranking: with WAM and `normalize_method` 0, 1, 2, 3 the winner is alternative 2 without standardization and alternative 1 with any of the three.

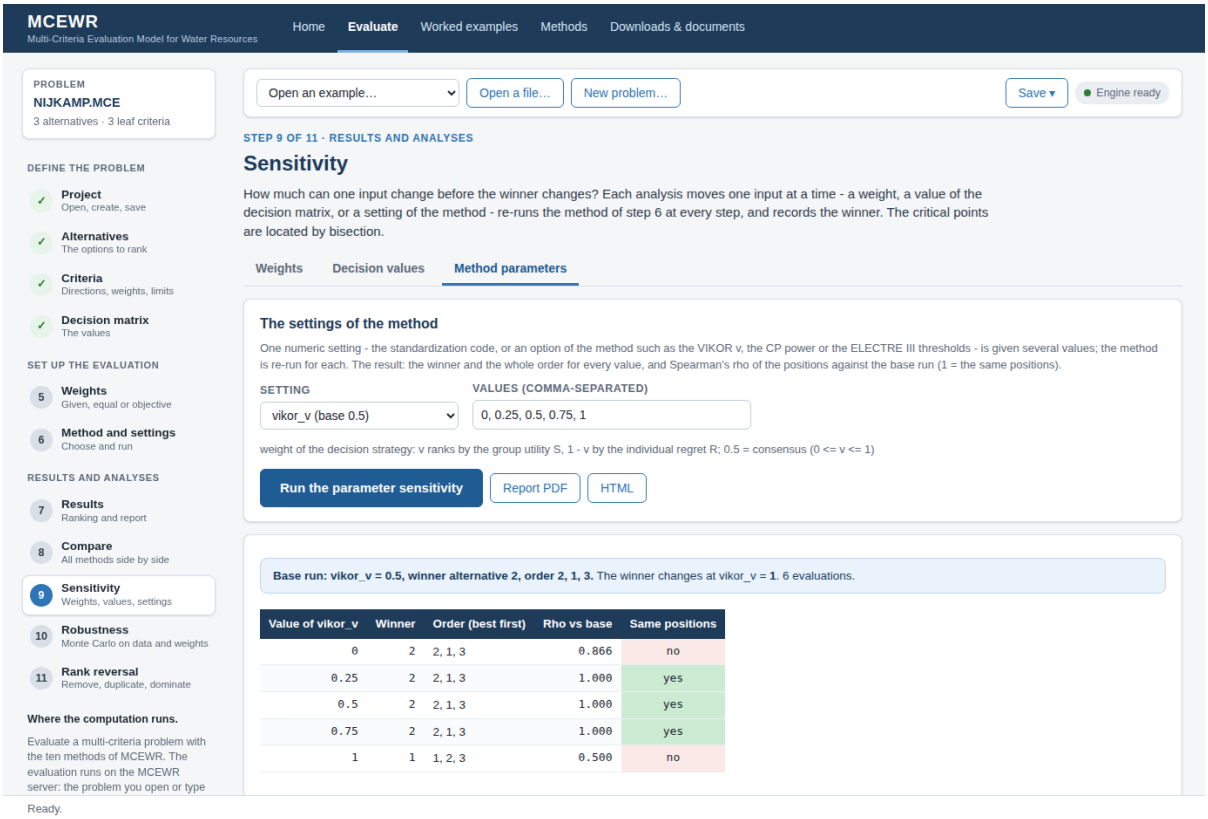


Figure 7.3 – Step 9, part Method parameters: the VIKOR strategy weight on the Nijkamp example.

8. Robustness: Monte Carlo on the data and the weights

Step **10 Robustness** perturbs all the inputs at once. Every value of the decision matrix, and if you wish every free weight, is multiplied by a random factor 1 + u; the method is re-run for every sample and the positions are counted. The result is the **rank acceptability** of every alternative – the fraction of the samples in which it takes each position – the fraction in which it keeps its base position, its mean rank, and the **winner stability**. u is drawn uniformly in [−p, +p], or from the triangular distribution on that interval with mode 0 when small errors are more likely than large ones; the seed makes the drawing repeatable.

Step 1. Open `NIJKAMP.MCE`; in step 6 select **WAM** and standardization 1. Open step **10 Robustness**: uncertainty of the values 0 %, of the weights 10 %, distribution uniform, samples 200, seed 1995. Press **Run the robustness analysis**.

Expected (the weights only; 99.5 % of the samples keep alternative 1 first):

Alternative	Base pos.	rank 1	rank 2	rank 3	Keeps base pos.	Mean rank
1	1	0.995	0.005	0.000	0.995	1.00
2	2	0.005	0.995	0.000	0.995	2.00
3	3	0.000	0.000	1.000	1.000	3.00

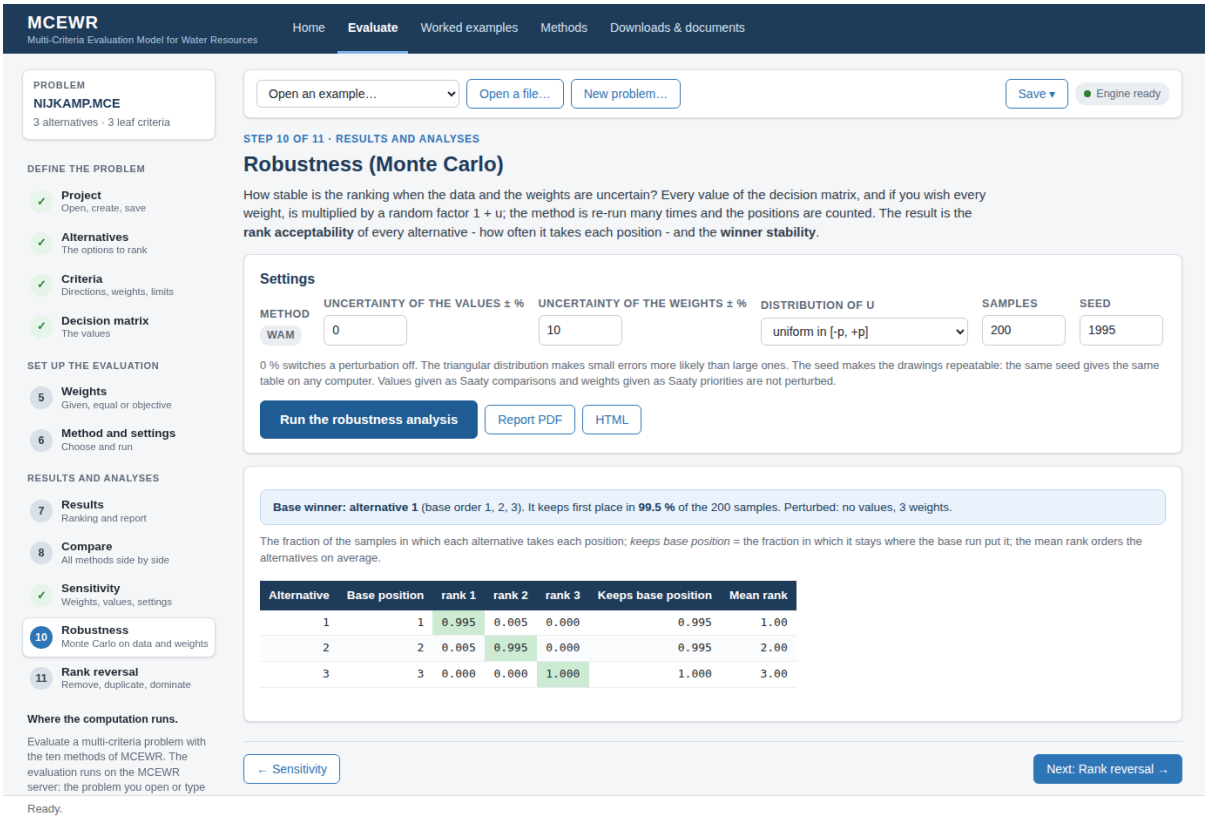


Figure 8.1 – Step 10: the weights of the Nijkamp example perturbed by ±10 %.

Step 1. Open `COMP_HAN.MCE`; in step 6 choose standardization 1 (WAM). In step 10: values 10 %, weights 10 %, triangular, 300 samples, seed 1995. Press **Run the robustness analysis**.

Expected (the first three rows; 85.7 % of the samples keep alternative 9 first):

Alternative	Base pos.	rank 1	rank 2	rank 3	rank 4	rank 5	rank 6	rank 7	rank 8	rank 9	Keeps	Mean
9	1	0.857	0.143	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.857	1.14
8	2	0.143	0.853	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.853	1.86

Alternative	Base pos.	rank 1	rank 2	rank 3	rank 4	rank 5	rank 6	rank 7	rank 8	rank 9	Keeps	Mean
5	3	0.000	0.000	0.900	0.100	0.000	0.000	0.000	0.000	0.000	0.900	3.10

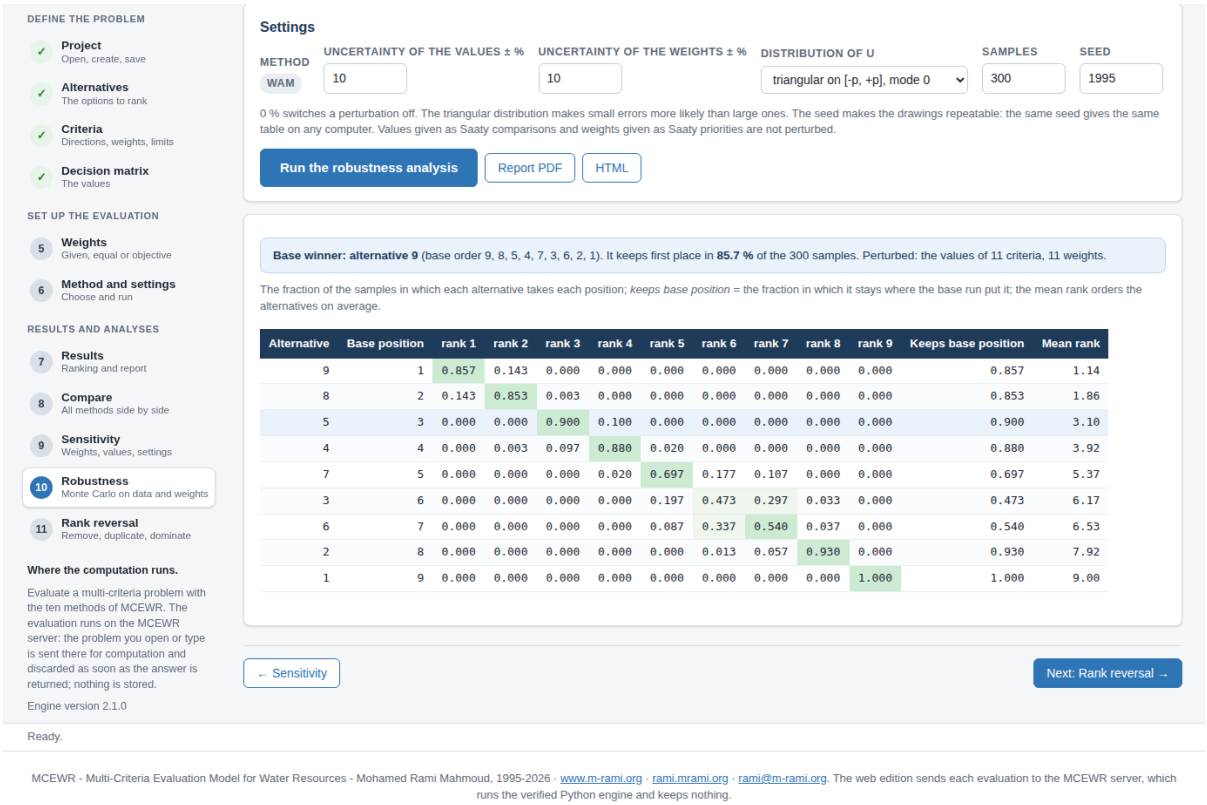


Figure 8.2 – Step 10: the Han River example with uncertain values and weights (triangular, ±10 %).

Read: with the values of the Han River data uncertain by ±10 %, alternative 9 stays first in about six samples out of seven and alternative 8 takes the other first places; the two are close. The same seed gives the same table on any computer, on the website and in the desktop application.

9. Rank-reversal diagnostics

Step 1. Open `COMP_HAN.MCE`; in step 6 choose standardization 1.

Step 2. Open step **11 Rank reversal**, tick WAM, PROMETHEE II, CP, AHP, TOPSIS and VIKOR, keep the three tests ticked, and press **Run the diagnostics**.

Step 3. The upper table gives, per method and test, the number of trials that changed the order of at least one pair of the original alternatives, and the number of reversed pairs; the lower table lists every instance.

Expected:

Method	remove	duplicate	dominated	Reversed pairs
WAM	0 / 9	0 / 9	0 / 1	0
PROMETHEE II	1 / 9	1 / 9	0 / 1	2
CP	0 / 9	0 / 9	0 / 1	0
AHP	0 / 9	0 / 9	0 / 1	0
TOPSIS	7 / 9	8 / 9	0 / 1	51
VIKOR	1 / 9	0 / 9	1 / 1	10



Figure 9.1 – Step 11: six methods on the Han River example; a green cell means no reversal.

Compromise Programming with the provided limits – the SPOTIS computation – measures every alternative against reference points fixed in advance and shows **0** on every test, on any file. TOPSIS computes its ideal and anti-ideal from the set of alternatives and reverses in 7 / 9 removals and 8 / 9 duplications on this file; PROMETHEE II and VIKOR reverse once each. The detail list names the pairs (a<>b: a was ahead of b in the base run).

10. Reports, output files and saving

Every **Report** button asks the server for the report of the result shown and downloads it in the chosen format: **PDF** (the same report the desktop application writes), **HTML** (opens in the browser, prints well) or **Text**. A run's report has the summary, the overall ranking, the criteria with their shares, the pay-off and standardized matrices, the rank and the ranking value of every alternative under every criterion, the Saaty consistency ratios (AHP), the PROMETHEE I relations when requested, the warnings, and the two output files reproduced verbatim. The comparison, sensitivity, robustness and reversal reports have their own sections (Method Reference, chapter 14). The Word and Excel reports exist in the desktop application only.

The **Output files of the run** panel of step 7 opens the two files of the run in the 1995 layout (`wam.txt` and `sortwam.txt` for WAM, and so on – Method Reference). Downloads go where the browser puts them (usually the *Downloads* folder); the file names carry the problem's name (step 1). The **Save** menu of the toolbar downloads the problem itself, with every edit, as `.mcewr`, `.csv` or `.MCE`.

> **Important.** The website keeps nothing between two visits: what you want to keep, download – the problem (Save as) and the reports. Reopening a saved `.mcewr` file restores the problem exactly, including the thresholds, the vetoes, the title and notes, the Best-Worst judgments and the settings of the last run.

11. The website, your data and the server

- **Browser**: any current Chrome, Edge, Firefox or Safari; a tablet works; a phone is cramped. No plug-in, no account.
- **Where the computation happens**: on the MCEWR server. When you press a *Run* button, the page sends the problem and the settings, the server runs the verified engine and answers within a second, the page shows the result. The problem is discarded as soon as the answer is returned; nothing is stored and the content of requests is not logged. The site says so on every page.
- **Limits** (a teaching service): at most 300 alternatives and 120 criteria, 5 000 samples of the perturbation, 200 steps of a sweep, and 120 calls per minute from one address. Beyond them the page shows a plain sentence, never an error page; larger studies belong to the desktop application.
- **The same numbers everywhere**: the website and the author's desktop edition run the same engine and are proved to agree, example by example, by the program's test suite. A result obtained on the site can be reproduced later from the saved `.mcewr` file and the settings printed in the summary.
- **A desktop edition** of the same engine, with Excel and Word reports and no size limits, exists for the author's own studies; it is not distributed through this site.

12. Tips and troubleshooting

- **The engine pill says *Connecting…* for more than ten seconds**, or *the MCEWR server did not answer*: your internet connection, or the server is being restarted. Wait a moment and press **Ctrl+F5** (a full reload).
- **Too many requests from this address**: more than 120 runs in one minute from one connection – a class behind one router counts as one address. Wait a minute.
- **The page looks old after an update of the site** (a missing button, a wrong version in the intro): the browser is showing a cached copy; **Ctrl+F5**.
- **Error 201** (ELECTRE II, a cycle of outranking relations): change the thresholds in the file's group line, or use ELECTRE III whose distillations always produce a ranking.
- **Error 245** with an option value: the value is outside its range (for example `vikor_v` must be between 0 and 1, `ra_samples` at least 100) – or the problem is larger than the server accepts.
- **Error 246** (objective weights): entropy and MEREC need positive data; CRITIC needs some dispersion. Use the given weights, or shift the data.
- **Error 245 with the Best-Worst Method**: a comparison is missing or outside 1–9, the best and the worst criterion are the same, or the best-to-worst value differs between the two vectors; the message names the criterion. A run with the BWM scheme and no judgments at all is refused: compute them in step 5 first.
- **Error 248** (an analysis): none of the selected methods could be evaluated on the file; the message says why for each.
- **Saaty examples**: the sensitivity analysis does not vary Saaty weights (they are priorities computed from the judgements), and the reversal tests that add an alternative are skipped for Saaty values.
- **Excluded alternatives** keep their row in the tables with the note 'excluded' and a ranking value of 0; use the *used* switch of step 2 or the *Exclude* boxes in *Advanced* of step 6.
- **Repeating a run exactly**: the summary lists every setting; the RA seed and the perturbation seed are recorded, so a run can be reproduced on any machine, on the site or on the desktop.
- **A file that will not open**: the message names the line and the reason (the same reader as the desktop). CSV files must have the three blocks `[criteria]`, `[alternatives]`, `[values]` (Web User Guide, section 9.3); Excel workbooks must be saved as CSV first.