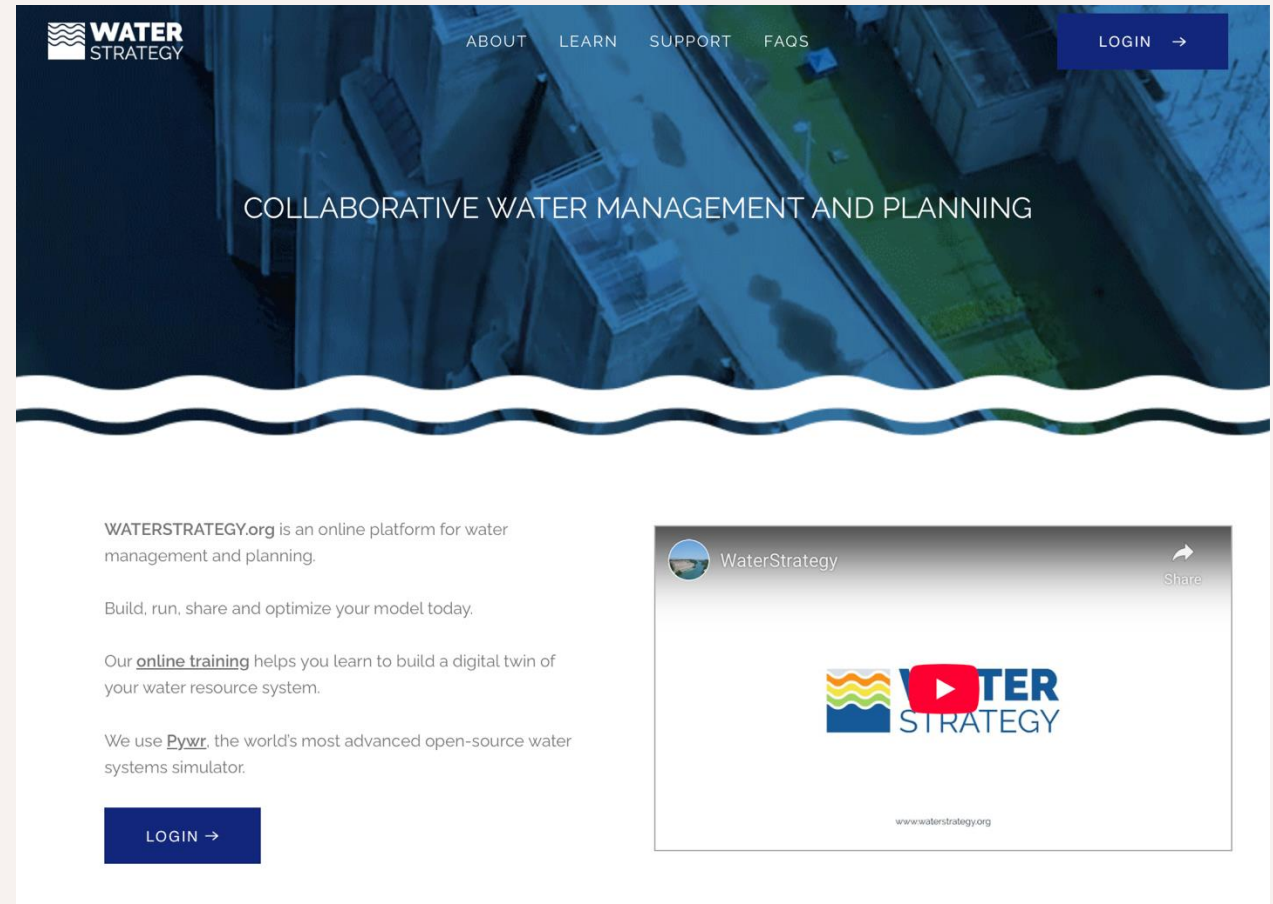


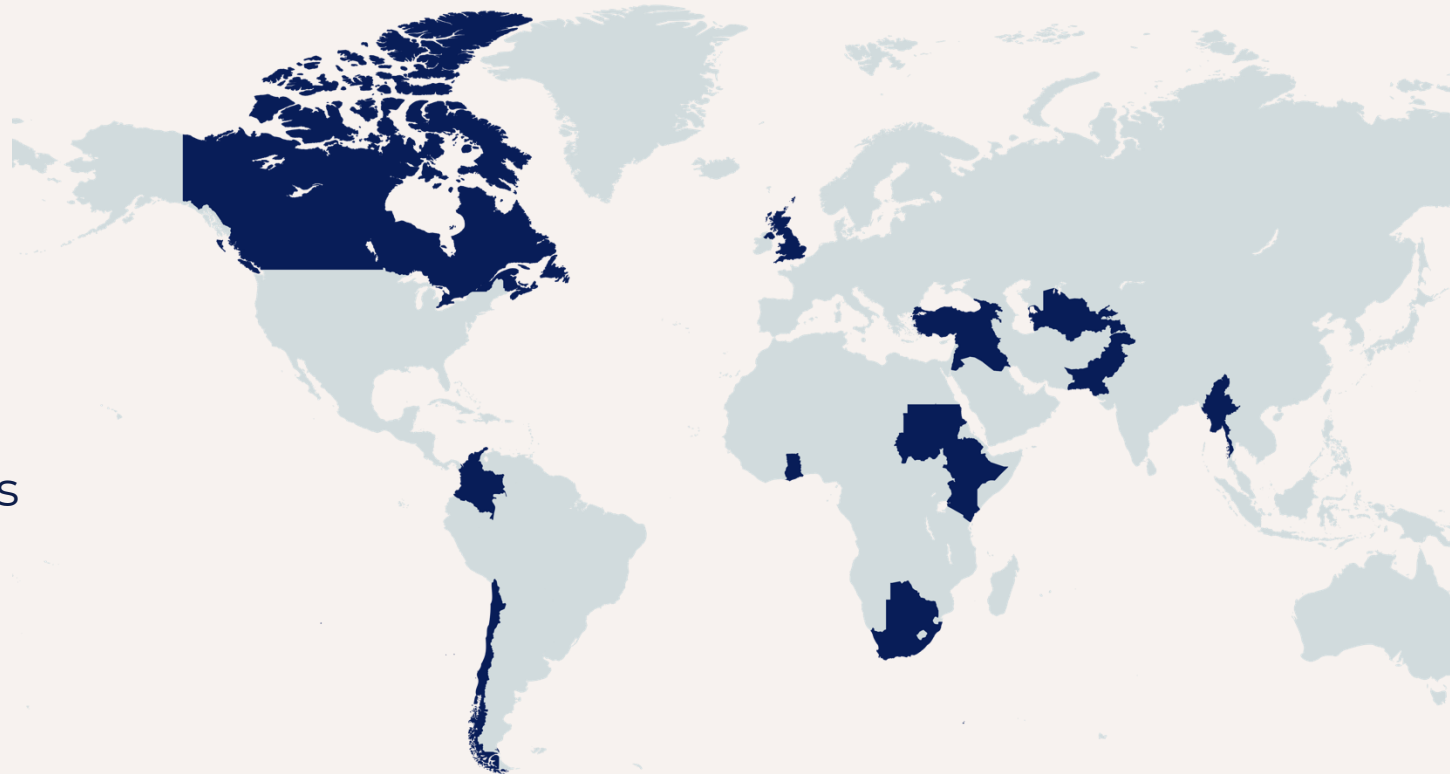
WaterStrategy.org



- WaterStrategy is an online modelling and decision-making platform
- Allows creating and running models on the cloud using Pywr and other model “engines”
- Web-based interface does not require installation of any software packages
- Supports many objective optimisation as well as least-cost planning (e.g. EBSD)



- 8 years in development
- £1.1m invested
- Applied in over 20 countries
- Used in over 34 consultancy projects worldwide
- Licenced by 4 English water companies and the Environment Agency
- Over 1400 users globally

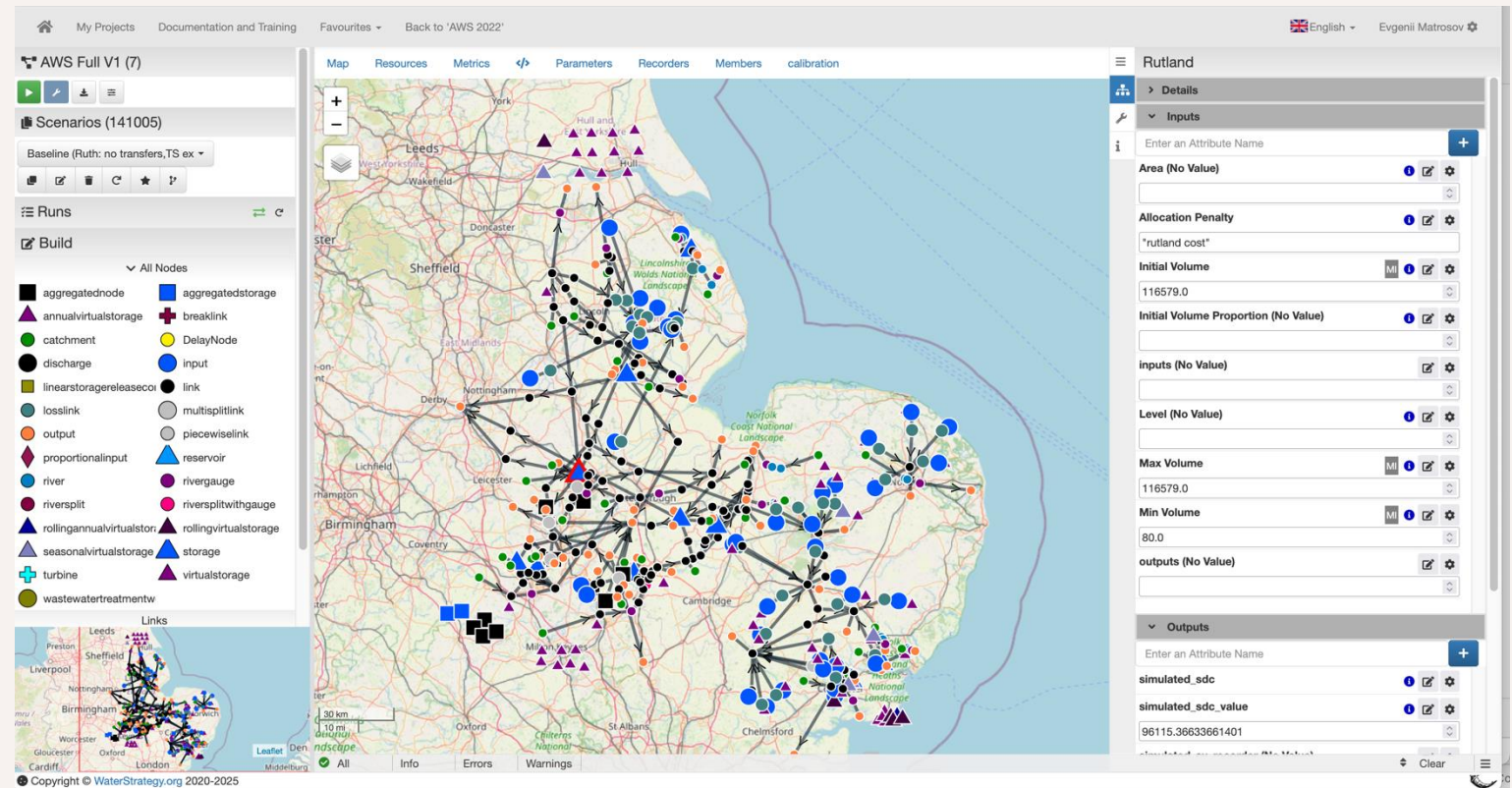


Current clients:



Pywr Interface

- Allows **building** and **running** Pywr models on the regional, water company and catchment scale
- Existing** models can be imported from Pywr JSON
- Pywr JSON files can be **exported** from interface
- Ability to upload H5, Excel and CSV datafiles
- Allows for downloading Pywr output files



Pywr Interface



Pywr **parameters** and **recorders** defined in bespoke tabs

The screenshot displays the Pywr web interface. The top navigation bar includes tabs for Map, Resources, Metrics, Parameters (active), Recorders, Members, and calibration. On the left sidebar, under 'Parameters-Type Categories', the 'pywr_parameter' section is expanded, showing a list of parameters. 'AWS_demand_saving_level' is highlighted with a blue dashed border. The main content area shows the title 'AWS_demand_saving -- AWS Full V1 (7) (Baseline (Ruth: no transfers, TS export and GW))'. Below this, there are tabs for Outputs, JSON (active), Metadata, and Options. The JSON tab contains a code editor with the following JSON structure:

```
1 {  
2   "type": "indexedarrayparameter",  
3   "index_parameter": "AWS_demand_saving_level",  
4   "params": [  
5     1,  
6     "WRE_DAC_saving",  
7     "WRE_TUBs_saving",  
8     "WRE_NEUBs_saving",  
9     "WRE_EDO_saving"  
10  ]  
11 }
```

Pywr Interface



Supports writing **custom parameters** and **recorders** using Python

Favourites ▾ Back to 'AWS 2022'

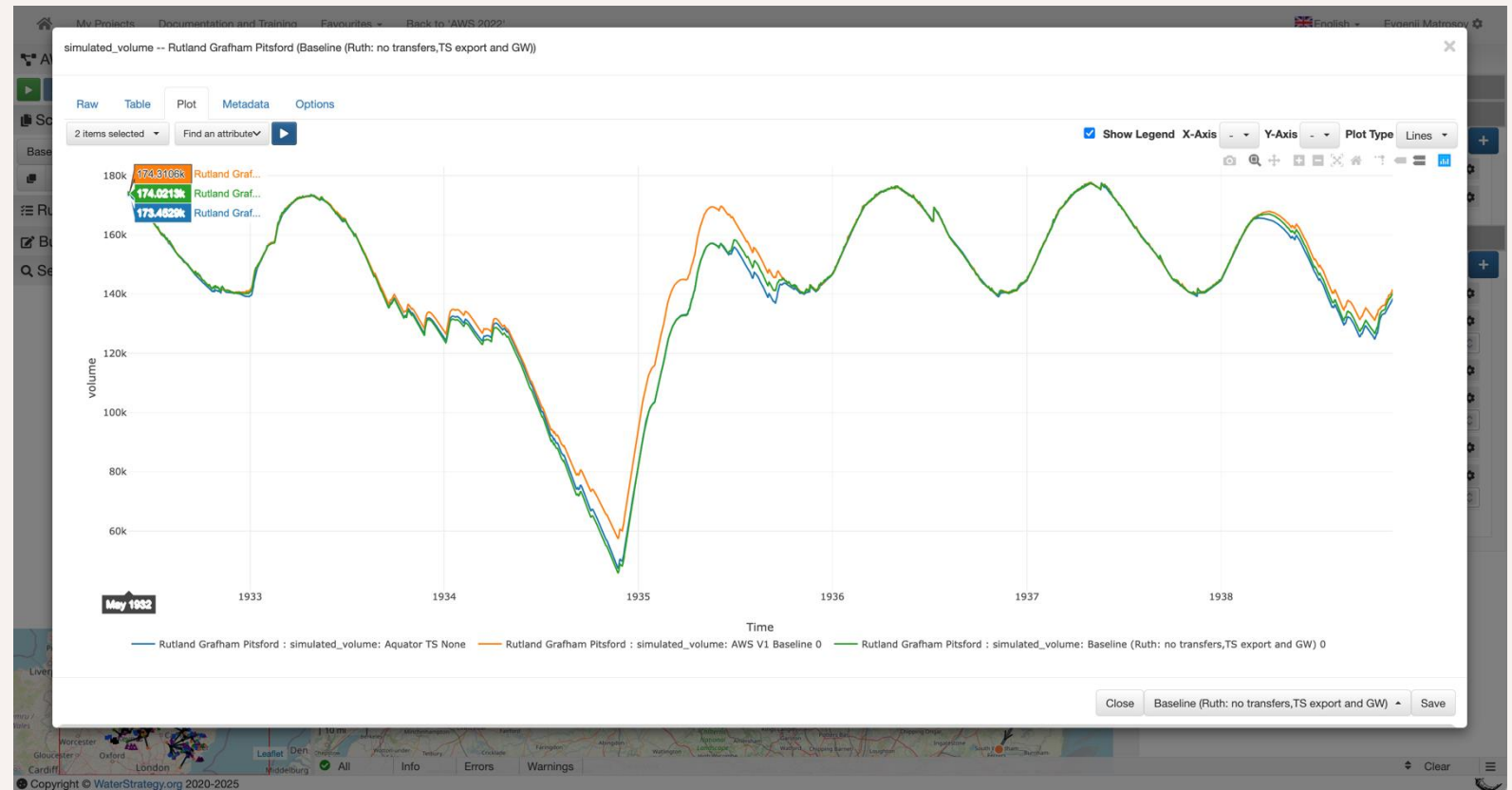
Map Resources Metrics `</>` Parameters Recorders Members calibration

Add a Rule + DayThreshold Add a rule type ▾ Python ▾

No Type	
DayThreshold	1 <code>class DayThreshold(AbstractThresholdParameter):</code>
TWAS_Trent_Torksey_transfer	2 <code> """ Returns one of two values depending on the day of week of the current timestep.</code>
TWAS_Short_Ferry_transfer	3 <code> """</code>
SLRWithamBostonAbstraction	4 <code> def _value_to_compare(self, timestep, scenario_index):</code>
NorthMuskhamAbstraction_150	5 <code> return timestep.datetime.weekday()</code>
Middle_Level_at_St_Germans_F...	6
Ely_Ouse_at_Denver	7 <code> @classmethod</code>
Bedford_Ouse_at_Earith_to_Fen...	8 <code> def load(cls, model, data):</code>
Ouse_Washes_Release	9 <code> threshold = load_parameter(model, data.pop("threshold"))</code>
	10 <code> values = data.pop("values", None)</code>
	11 <code> predicate = data.pop("predicate", None)</code>
	12 <code> return cls(model, threshold, values=values, predicate=predicate, **data)</code>
	13
	14 <code>DayThreshold.register()</code>

Pywr Interface

- Allows for defining, running and **visualising** multiple scenarios
- Scenario comparison tools to **analyse** system performance
- In-built tools to **run** Deployable Output analyses

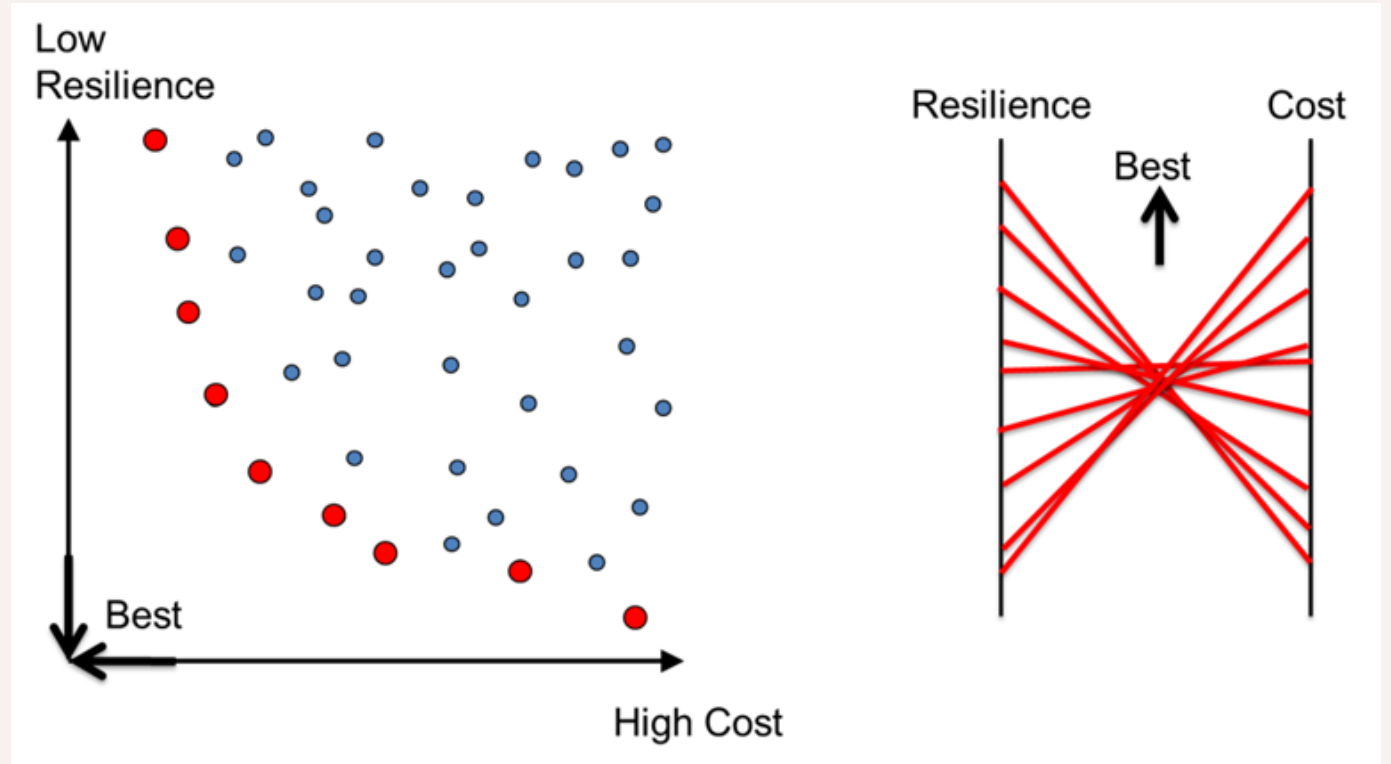


Many-Objective Evolutionary Optimisation



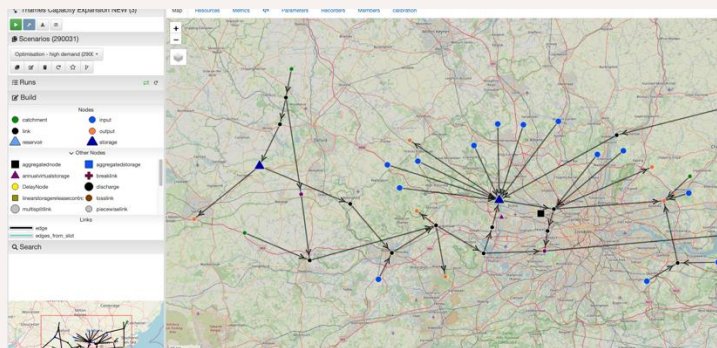
Cloud-based many-objective optimisation workflow

- WaterStrategy allows for formulating, running and visualising the outputs of many objective optimisations on the cloud
- Optimisations can be run on 1 or many computational cores without the need for high performance clusters

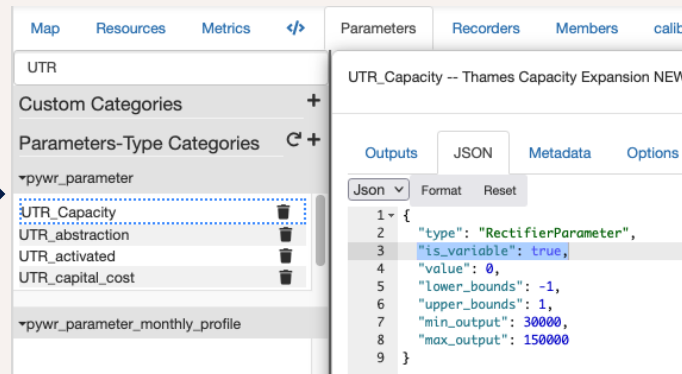


Example performance in two metrics visualised in a scatter plot (left) and parallel axis plot (right)

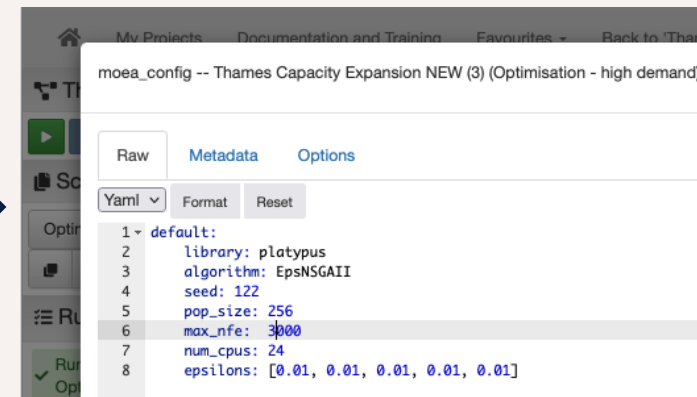
Model **built** using or uploaded into interface



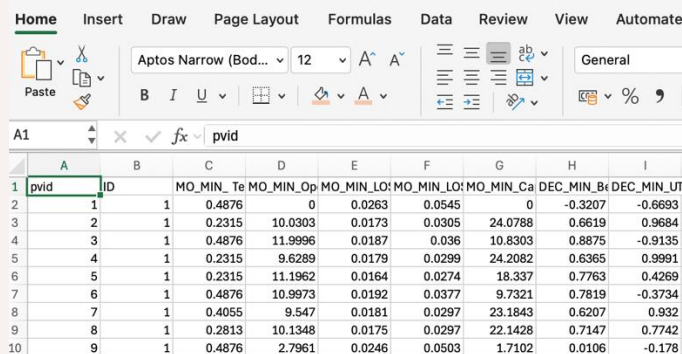
Decision variables, objectives, constraints **defined**



Optimisation **configured**

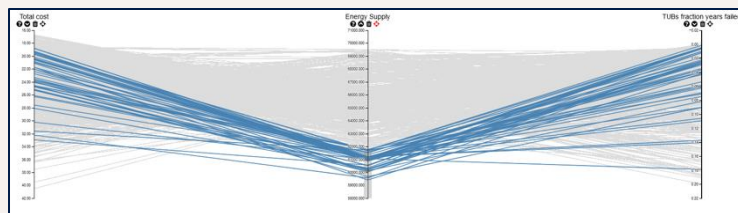


Optimisation outputs can be **downloaded** for further processing

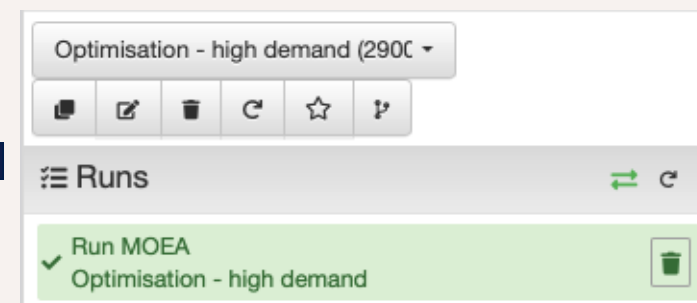


	A	B	C	D	E	F	G	H	I
1	pvid	ID	MO_MIN_Te	MO_MIN_Op	MO_MIN_LO	MO_MIN_LO	MO_MIN_Ca	DEC_MIN_B	DEC_MIN_UT
2	1	1	0.4876	0	0.0263	0.0545	0	-0.3207	-0.6693
3	2	1	0.2315	10.0303	0.0173	0.0305	24.0788	0.6619	0.9684
4	3	1	0.4876	11.9996	0.0187	0.036	10.8303	0.8875	-0.9135
5	4	1	0.2315	9.6289	0.0179	0.0299	24.2082	0.6365	0.9991
6	5	1	0.2315	11.1962	0.0164	0.0274	18.337	0.7763	0.4269
7	6	1	0.4876	10.9973	0.0192	0.0377	9.7321	0.7819	-0.3734
8	7	1	0.4055	9.547	0.0181	0.0297	23.1843	0.6207	0.932
9	8	1	0.2813	10.1348	0.0175	0.0297	22.1428	0.7147	0.7742
10	9	1	0.4876	2.7961	0.0246	0.0503	1.7102	0.0106	-0.178

Interactive parallel plot (Polyvis) made **available online**



Optimisation **launched** on the cloud



Optimisation Configuration

- Decision variables, objectives and constraints defined in editors
- Optimisation configuration defined in a configuration parameter
- Uses open-source MOEA libraries
- Number of computational cores defined by the user
- Optimisation runs on the cloud



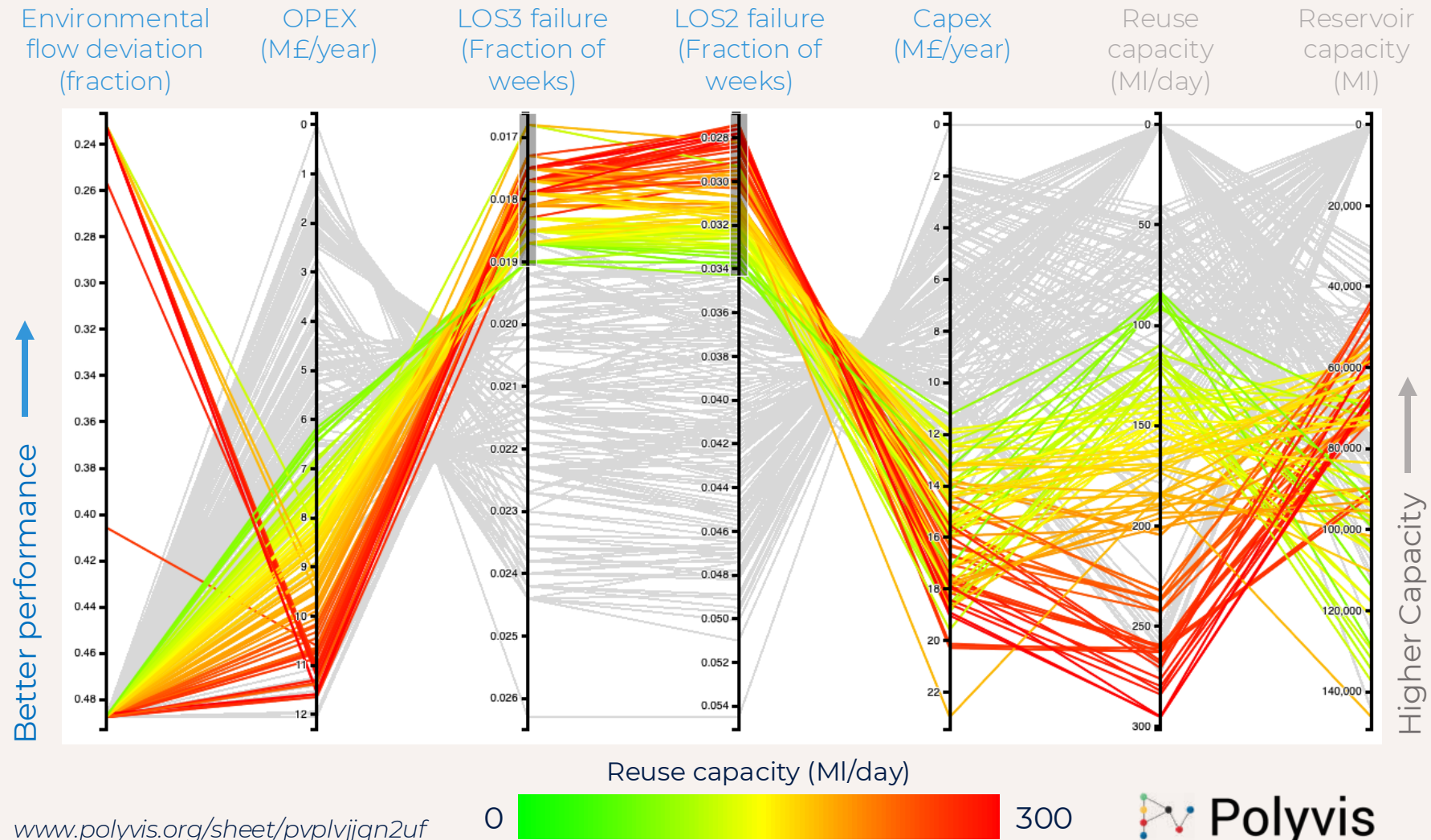
The screenshot displays two configuration editors side-by-side. The top editor is for JSON, with tabs for 'JSON', 'Metadata', and 'Options'. The 'JSON' tab is active, showing a configuration object with fields like 'type', 'agg_func', 'recorders', and 'is_objective'. The 'is_objective' field is highlighted with a red box. The bottom editor is for YAML, with tabs for 'Raw', 'Metadata', and 'Options'. The 'Raw' tab is active, showing a configuration object with fields like 'library', 'algorithm', 'seed', 'pop_size', 'max_nfe', 'num_cpus', and 'epsilons'.

```
JSON
Metadata Options
Json Format Reset
1 {
2   "type": "aggregatedrecorder",
3   "agg_func": "product",
4   "recorders": [
5     "Level_2_failure_param_recorder",
6     "one_over_Nw_param_recorder"
7   ]
8   "is_objective": "minimise"
9 }
```

```
Raw Metadata Options
Yaml Format Reset
1 default:
2   library: platypus
3   algorithm: EpsNSGAI
4   seed: 122
5   pop_size: 256
6   max_nfe: 1000
7   num_cpus: 32
8   epsilons: [0.01, 0.01, 0.01, 0.01, 0.01]
```

Polyvis output to visualise decisions and objectives

- Link to results in parallel plot format available online after optimisation complete (Polyvis)
- Optimisation results can be downloaded for further analyses offline
- Portfolios solutions can be run in Pywr



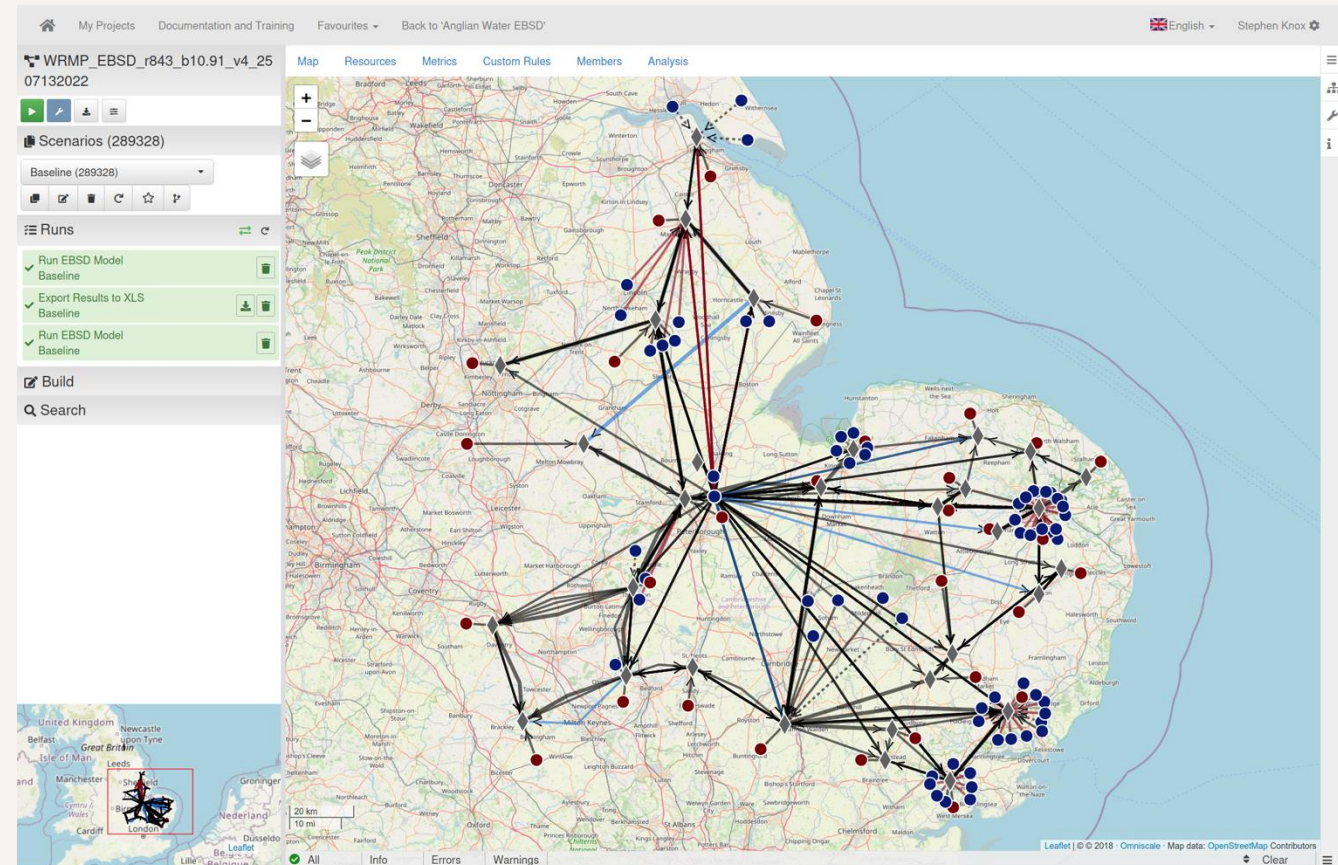
Economics of Balancing Supply and Demand (EBSD)



Economics of Balancing Supply and Demand (EBSD)



- WaterStrategy supports EBSD modelling
- Least-cost optimisation tool for investment planning
- Selects options (transfers and new supply) to meet supply-demand balance goals, at the least cost
- Schedules option activation
- Exclusivities, dependencies etc.
- Used by 3 UK water companies for their WRMP plans

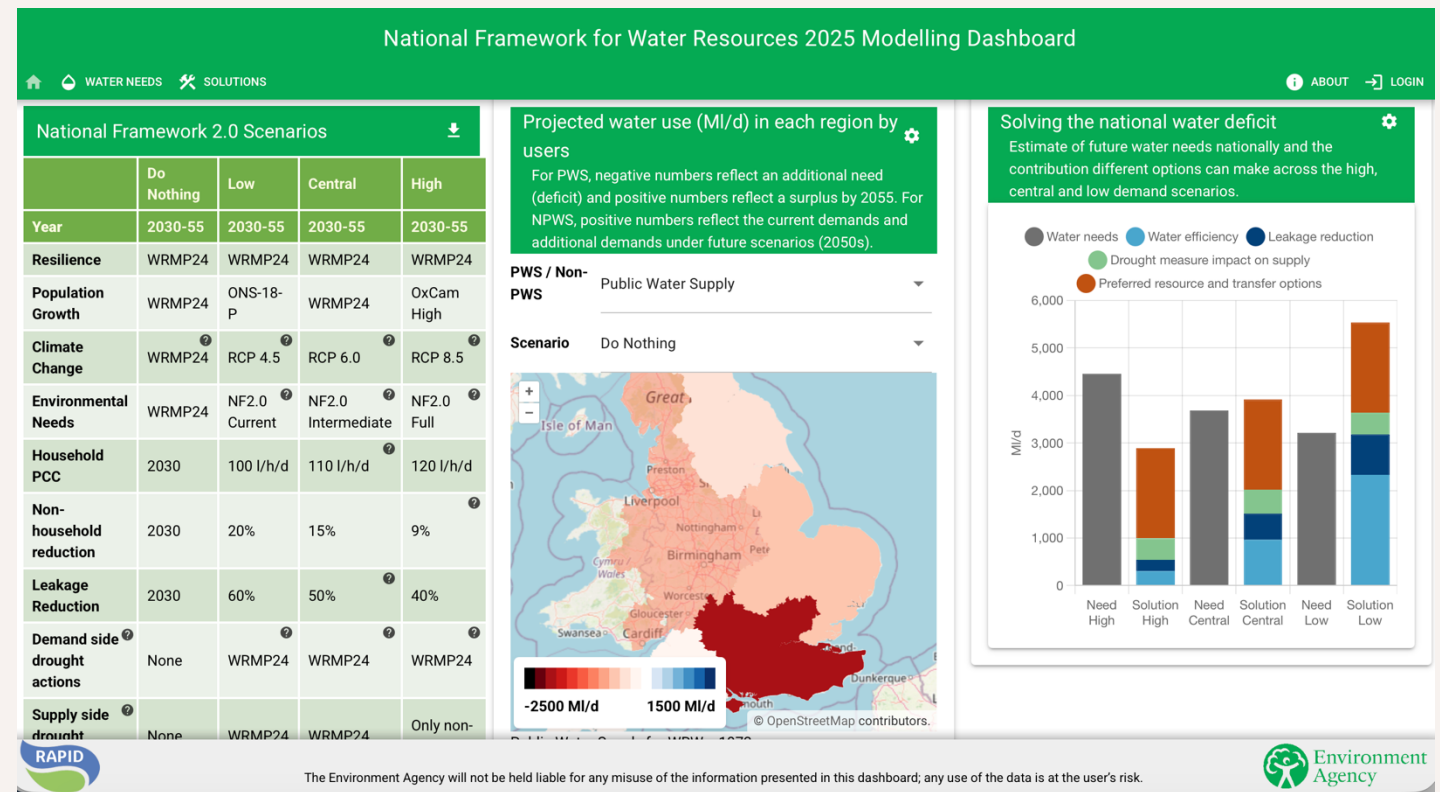


Custom Dashboards



Bespoke Solutions

- We maintain several publicly accessible platforms, and we can set up bespoke versions of these platforms for clients.
- We have worked with the Environment Agency to develop the web-based interactive modelling dashboard supporting the National Framework for Water Resources 2025
- Dashboards support public and login-based functionality



Next Features to be Released



Adaptive Planning Extension

Integrated AI-driven planning



**Soon to be implemented → Adaptive Planning Engine
→ Visualisation Explorer**

- Uses AI to explore thousands of possible futures and identify flexible investment options.
- Builds on our research in adaptive planning and multi-criteria optimisation.
- Enables planners to visualise trade-offs, risks, and adaptation pathways through interactive dashboards and maps.

AI-Driven Intelligent Modelling Assistant

Enhancing water modelling



Soon to be implemented → Integrated AI assistant for model creation

- AI assistant helps users build and check complex water system models.
- Learns from a growing library of real-world calibrated models.
- Suggests system structures, allocation rules, and validation checks.
- Explains model results and trade-offs in plain language to support better decisions.

Evgenii Matrosov
COO & Consulting Lead
T: +44 7595 472938
E: ematrosov@nexsys-analytics.com

Martin Findlay
Principal Consultant
T: +44 7522 860894
E: mfindlay@nexsys-analytics.com

