

Joint Optimisation of Energy Networks

Current version: Pre-release 0.62.9

An Energy Decision System for planning and operation.

A planning result can shape decades of investment. JOEN connects the question, assumptions, mathematics and evidence so you can explain the decision and test what would change it. It will be open source, free and publicly available under the commercially permissive Apache 2.0 licence, bringing investment planning, resource scheduling and dispatch into one Energy Decision System.

What would you like to explore?

Your modelling question	How JOEN helps	What you can explore
What should we build, and when?	Long Term (LT) capacity expansion and retirement, with financial assumptions and capacity requirements.	Investment pathways across generation, storage, transmission and flexible demand.
How should we use limited resources?	Medium Term (MT) scheduling shares energy requirements across shorter periods and values stored water.	How longer-term resource limits affect daily operation.
How will the system operate?	Short Term (ST) economic dispatch and unit commitment for supported chronological studies.	Generation, charging, pumping, power flows, unserved energy and marginal prices.
How do the stages fit together?	Stages share the same formulation and pass capacity decisions, storage levels and remaining obligations forward.	Consistent assumptions and a clear record of decisions between solves.

From a modelling question to a decision

Study step	What you do
1. Describe your system	Import an AEMO ISP or ESOO XML dataset, or create your own model data using editable tables.
2. Choose the detail	Set the time resolution, assets and constraints. Choose the stages and solver for your study.
3. Plan and operate	Run LT, MT or ST on its own, or connect them. Choose a single solve or a rolling horizon.
4. Explore the results	Review investment and dispatch results alongside solution quality and the assumptions passed between stages.

From a result to an investment case: choose capacity in LT, allocate water and energy in MT, and examine dispatch in ST. Follow the assumptions and requirements between stages, challenge a binding limit and compare alternatives. Build a decision you can explain.

Open source. Inspectable decisions.

An audit trail records a run; open source also lets you inspect its mathematics. JOEN's planned release connects inspectable code, model meaning and study evidence so modellers and supporting AI agents can understand, check and extend the model.

Open source turns transparency into practical control

Design pattern	What matters to the modeller	Open-source advantage
AI and audit trails around a closed engine	Robust engines offer deep capability. Logs record a run, but closed equations limit what users and agents can independently inspect.	Inspect published code, rules and plans; test support and check explanations against implementation.
Ownership and collaboration	Teams need to adapt and maintain methods as requirements change.	Reuse, modify and develop commercially under Apache 2.0. Maintain extensions and share improvements on your own timetable.

A technical foundation for understanding and speed

Mechanism	Implemented design	Value to the study
Ontology: concepts and connections	A shared map of model concepts and relationships. For example: a battery stores energy, connects to a network node and has charging and discharging variables. Executable declarations generate this catalogue.	Give modellers and agents a common vocabulary for finding capabilities and dependencies.
Semantics: what each item means	The precise meaning of model data and mathematics: units, time behaviour and roles. For example: battery state of charge is stored energy in MWh at a defined time boundary, rather than charging power in MW.	Interpret inputs, equations and results consistently; connect meaning to sparse matrix rows and columns.
Data lineage	Source selections, transformations, stage links and raw/effective adjustments remain attached to model evidence.	Trace an outcome to assumptions and numerical changes. Build trust through evidence.
Rules and recipes	Rules declare mathematics and requirements. Recipes compose them through a shared registry and dependency checks. Python packages can add formulations and semantic declarations.	Extend the model while reusing preparation, compilation, solving and publication. Validate the new mathematics.
Plans and sparse matrices	Intermediate plans describe variables, equations and results. The compiler maps them to compact numeric coordinates and emits bounded sparse batches.	Keep meaning inspectable while avoiding dense asset-by-time expansion and per-coefficient Python construction.

Symbolic and Theoretic Prover: joen-math

Capability	What it does and why it helps
Symbolic and logical verification	joen-math, DecisionSigma's internal software, links implementation claims to independent symbolic equations and ontology references. It applies symbolic and logical checks to registered claims, reports derivations, verification status and coverage, and checks selected sparse rows against mathematical contracts. This helps expose discrepancies between intended mathematics and implementation.
Beyond chatbots	Agents can combine code, semantics, lineage and symbolic evidence to investigate a result and distinguish checked claims from gaps. JOEN remains the execution authority; joen-math verifies declared claims rather than solving a second model.
Develop and test a change	Develop adapters, constraints, variables and formulations, then compare runs and check supported symbolic claims. Modellers direct the work and review the mathematics and evidence.

Trust through evidence: inspect the code, follow its meaning and check the mathematics. Symbolic verification reports checked claims and coverage gaps; it does not prove the entire model correct.

Start with your model.

Choose the detail you need.

Start with AEMO ISP or ESOO XML datasets, or compatible XML from commercial software such as PLEXOS. Build from scratch or connect your own data through a custom source adapter. All routes use the same JOEN formulation.

Bring in data or create a new model

Capability	What JOEN does	Why it helps
AEMO XML to CSV	Converts ISP, ESOO and compatible XML objects, memberships and properties to CSV. External time-series files are supplied separately.	Review and update model inputs in a familiar table format.
Source model conventions	Reads scenarios, effective dates, Timeslices, Data Files, units, defaults and supported RPN expressions.	Keep the source model's meaning when preparing a JOEN study.
Edit your model	Identifies assets by class and name, and relationships by their named endpoints. CSV edits follow the source dataset's conventions.	Update assets and properties without looking up source-system IDs.
Build from scratch	Create assets, relationships, values and custom constraints in workspace tables or JOEN's common dataset format.	Develop a new model directly in tables, without first creating an XML dataset.
Create your own source adapter	Develop and register a Python adapter that maps your data, units and relationships into JOEN's common records.	Connect your own formats while reusing chronology, formulation, solvers and results.
Check and prepare	Validates the data and prepares editable records for use in a solve. Reports conflicting inputs and features that are not supported.	Find input issues before running a large study.

Chronology Supported

Chronology	How it works	Status and use
Native	Uses one decision step for each original model interval.	Supported. Detailed dispatch, storage and operating constraints.
Chronological blocks	Groups intervals into ordered load blocks. Choose the grouping period and block count; demand energy is preserved within each period.	Supported. Smaller models that retain the order of operation.
Sampled	Uses selected representative periods rather than every original interval.	Under testing. Dataset and operating-constraint validation is ongoing.

Chronology settings apply across LT, MT and ST. Block settings control peaks and required state boundaries. Reduced-time results need checks before conclusions about every original interval. Sampled chronology is under testing.

Bring long-term choices into shorter-term decisions.

Explore investment choices in LT, allocate resources in MT and carry those decisions into ST operation. JOEN keeps track of the capacity, storage levels and requirements passed from one stage to the next.

Long Term: what to build and when

Capability	What you can model	Why it helps
Build and retire assets	Generators, lines, batteries, purchasers and Power2X, with continuous or whole-unit investment choices.	Compare portfolios across supply, networks, storage and flexible consumption.
Capacity over time	Commissioning dates, build and retirement limits, economic and technical life, and capacity requirements.	Represent how the system changes across the planning horizon.
Investment costs	Capital and fixed costs, asset financing rates (WACC), system discount rates and cash-flow timing.	Compare investments using consistent financial assumptions.
Value at the horizon end	Choose no residual credit, upfront capital less residual value, or full annuity liability. Residual value can use straight-line life or remaining-annuity present value.	Explore how end-of-horizon assumptions affect investment choices.
Capacity passed forward	Dated capacity records carry the LT decisions into later stages. Battery power and energy capacity remain separate.	Use the planned portfolio in a more detailed operating study.

Medium Term: how to allocate resources

Capability	What JOEN does	Why it helps
Energy and long-horizon custom constraints	Allocates longer-period requirements to shorter solves, tracking what has been used and what remains.	Keep longer-term obligations in view during detailed operation.
Storage levels	Passes battery state of charge and reservoir volumes to the next stage.	Start the next solve from the appropriate physical conditions.
Water values and targets	Uses valid upstream LP marginal values to value retained water or penalise shortfalls against a target, with the selected decomposition settings.	Account for water needed beyond a shorter solve horizon.

Short Term: how the system operates

Capability	What you can model	Why it helps
Economic dispatch	Generator output, battery charging/discharging, hydro generation/pumping and transmission flows, subject to network and resource limits.	Explore operating costs, unserved energy and available marginal prices.
Unit commitment	Supported chronological online/start decisions, startup costs, ramp limits, minimum up/down times and initial conditions.	Represent operating flexibility. Advanced profiles and some rolling commitment features remain limited.
Single or rolling operation	Run ST independently or use LT/MT capacity, storage levels and energy accounts. Rolling solves use separate lookahead and committed periods.	Carry physical conditions and remaining obligations forward as operation progresses.

LT decides capacity, MT allocates resources and ST schedules operation. Run stages independently or connect them through explicit capacity, state and account handoffs. Unit-commitment support depends on the chronology and solve configuration.

Diagnostics, observability and numerics

Follow progress, investigate models and control stage handoffs. Optional tools depend on configuration and solver support.

Automatic diagnostics

Check	What JOEN does
Input and model checks	Checks inputs, stage links, chronology and resources. Optional fast screening detects direct row/bound and integer-bound contradictions before solving; it is off by default.
Infeasibility analysis	When configured, infeasibility automatically triggers supported Farkas certificates, IIS conflict analysis or selected feasibility relaxation. Diagnostic limits apply; the original status remains.

Observability

What you can follow	What is available
Progress and solve quality	Events and run status show stages, active phases and completed/declared solve counts. Summaries retain objective, available violations, MIP bounds/gaps and termination reason.
Find the mathematics behind a model concept	Start with an asset, constraint or time period. Semantic links identify the relevant sparse rows and columns, helping you scope an investigation and trace the equations, variables and coefficients that represent it.
Inspect large models with purpose	Build custom diagnostics or connected AI tools that use these mappings to locate and extract the relevant LP formulation from GB-scale text files. For example, inspect a battery balance or trace a binding network constraint. Scoped evidence makes large models easier to explain, check and extend.

Numerics and algorithms

Control / algorithm	How it works	Use and limits
LP algorithms	Auto, barrier/IPM, primal or dual simplex, with stage-specific native solver options.	Backend-dependent. HiPO needs a compatible HiGHS build.
Degeneracy options	Optional deterministic-cost regularisation: missing-cost fallbacks and stable entity-ranked epsilon increments for selected variable families.	Off by default. Alters objective preferences and tie selection; adds no rows or columns.
Scaling and conditioning	Automatic all-small-row equilibration; objective scaling (none/auto/user); explicit continuous-column scaling. Exact semantic presolve removes eligible identities and redundancy.	Restores model units; reports range/structure. Benefits depend on the model.
Crossover	Backend option converts a barrier result to a basic LP solution, with available iteration and quality records.	Enable/disable where supported. A small barrier gap does not prove feasibility.
MILP then LP	Fixes usable MILP integer choices, then solves an LP for conditional marginal values.	Separate LP quality; original MILP status/gap retained.
Capacity and state reconciliation	Rebuilds capacity from build/retire decisions and expiry; corrects eligible state-bound residuals under explicit policies.	Raw/effective values retained. Uncapped correction is opt-in.
No-crossover polishing	Optional scaled minimum-change convex projection on connected continuous capacity/state values, using HiGHS or Gurobi.	Off by default; accepted no-crossover barrier only. No integer repair.
Account handoff and validation	Passes limits, used quantities and remaining obligations forward. Reconciliation uses sparse activity bounds; checks affected rows/bounds and retains restart evidence.	Policy-based; adjustments and violations remain visible. No feasibility guarantee.

The model applies a sparse data and matrix methodology. Initial benchmarks against AEMO ISP scenarios range from 20-30 minutes for model build, with around 100-120 GB peak memory during the build.

Technical Specification

Implemented property families for AEMO ISP, ESOO and compatible XML datasets, using source-table names. Related time-period variants are grouped below.

Generation, batteries, hydro and investment

Area	Implemented properties	Support notes
Generator capacity	Units; Max Capacity; Rating; Rating Factor; Min Load; Fixed Load; Min Stable Level; Min Stable Factor; Aux Incr; Min Load Penalty; Fixed Load Penalty; Min Stable Level Penalty.	Resolves availability and output bounds. Per-unit and whole-fleet settings are kept distinct.
Generator costs	VO&M Charge; Generating VO&M Charge; Fuel Price; Heat Rate; Heat Rate Incr; Mark-up; Bid-Cost Mark-up.	Heat Rate Incr takes precedence where supplied. Heat Rate Base is not included in operating cost.
Generator commitment	Initial Generation; Initial Units Generating; Initial Hours Up/Down; Max Ramp Up/Down; Min Up/Down Time; Start Cost; Run Up/Down Rate.	Conditional: supported discrete chronological single solves. Advanced profiles and rolling commitment cohorts are limited.
Generator energy	Min/Max Energy; Min/Max Energy Penalty; Min/Max Capacity Factor; Energy Scalar.	Applicable interval and hour/day/week/month/year variants. Selected time resolution must support the obligation.
Supply offers and pump bids	Offer Quantity/Price; Offer Price Incr/Scalar; Offer Quantity Scalar/Format; Pump Bid Quantity/Price; Pump Bid Price Incr/Scalar; Pump Bid Quantity Scalar.	Banded offers and bids. Nonzero Offer Base, Offer No Load Cost, Pump Bid Base and ordered-clearing requests are not supported.
Battery operation	Units; Max Power; Max Load; Capacity; Charge/Discharge Efficiency; Initial/Min/Max SoC; VO&M Charge; Charging/Discharging VO&M Charge.	Power and energy are separate. Generic operating cost can follow the selected direction policy.
Battery limits and bids	Min/Max Energy Charging; Min/Max Energy Discharging; related Penalty properties; Energy Target and Penalty; Max Cycles; Offer/Bid Quantity and Price with Incr/Scalar modifiers.	Time-period variants apply. Targets are closing states; cycles use equivalent throughput. Nonzero offer/bid base and no-load terms are excluded.
Hydro reservoirs	Initial/Min/Max Volume; Natural Inflow; Natural Inflow Scalar/Incr; Max Spill; Min/Max Release; Max Generator Release; Target, Target Custom and Target Penalty.	Hour/day/week/month/year targets where applicable. Physical volume bounds apply at selected closing states.
Hydro guidance	Balance Period; End Effects Method; Recycle Penalty; Recycle Ignore Look Ahead; Water Value; Decomposition Method; Decomposition Penalty a/b/c/x.	Source decomposition needs valid upstream LP storage marginals. Target timing must fit the chosen chronology.
Waterways and pumping	Waterway Min/Max Flow; Input/Output Scalar. Generator Efficiency Incr; Pump Efficiency; Pump Load; Pump Units; Pump VO&M Charge.	Only zero Traversal Time and zero storage Loss Rate are currently accepted. Nonzero values are rejected.
Investment properties	Build Cost; FO&M Charge; WACC; Economic Life; Technical Life; Min/Max Units Built and Built in Year; Min/Max Units Retired and Retired in Year; Retirement Cost.	Common planning framework across generators, batteries, lines and purchasers. Power2X includes its supported build and finance properties.

Min/Max and Up/Down abbreviate the named properties. Calendar variants apply only where defined for the asset and compatible with the chosen time resolution.

Technical Specification

Network, demand, conversion and policy properties. The loader reads 480 mappings across five resources, including limited and import/reporting entries; the count does not mean every mapped feature is fully supported.

Network, demand and policy properties

Area	Implemented properties	Support notes
Transmission lines	Units; Min/Max Flow; Loss Allocation; Loss Base/Back; Loss Incr/Back; Loss Incr2/Back; Offer Quantity/Price and Back variants; Offer Quantity Format; Wheeling Charge/Back; Fixed Charge.	Quadratic losses use linear segments. LP and supported integer/SOS2 choices are available. Interface aggregate expansion is not implemented.
MLFs and settlement	MLF Intercept; Flow Coefficient; asset Marginal Loss Factor; Line Marginal Loss Factor Back; Region MLF Adjusts Offer Price and MLF Adjusts Bid Price.	Physical-loss equations and settlement cost adjustments are separate. Equation/topology checks apply.
Demand and reliability	Node/Zone/Region Units and Load; Load Participation Factor; Price of Dump Energy; Node Allow Dump Energy; Region VoLL; DSP Bid Quantity/Price; Max Unserved Energy; Max Dump Energy.	Spatial energy limits include time-period variants where defined. Regional allocation preserves total demand.
Purchasers	Units; Max Load; Fixed Load; Bid Quantity; Bid Price; Min/Max Energy and Penalty; Min/Max Load Factor.	Period variants apply. Load Obligation is import-only; nonzero Bid Base is excluded.
Power2X	Units; Max Load; Min Stable Level; Efficiency; VO&M/FO&M Charge; Bid Quantity/Price; Min/Max Production and Penalty; Min/Max Capacity Factor; Production Rate; Water Consumption; Recovery Price Adder.	Single efficiency point; period variants and LT build/finance. Advanced curves and startup dynamics are limited.
Fuel and emissions	Fuel Units; Price; Transport Charge; Min/Max Offtake and Penalty. Emission Shadow Price; Max Production and Penalty; relationship Production Rate and Removal Rate.	Period variants apply. Fuel units must match heat rates. Emission Price is reporting-only; Shadow Price affects the objective.
Adequacy and outages	Firm Capacity; Firm Capacity Incr; Min/Max Capacity Reserves; Min/Max Capacity Reserve Margin; Capacity Shortage Price; Line Min/Max Rating; Forced Outage Rate; Outage Factor.	Class-specific coverage and selected outage derating. Nonzero Firm Capacity Unit Count is not supported.
Custom constraints	Sense; Penalty Price; RHS, RHS Day/Week/Month/Year/Custom; Scalar; Include in LT Plan, MT Schedule and ST Schedule; generation, sent-out, load, flow, production, pumping, offtake, release, unserved-energy and capacity/build/retirement coefficients.	Supported coefficient roles share temporal accounts. Installed/Generation Capacity Coefficient have stage-specific limits.

Custom Decision Variables and Reverse Polish Notation (RPN)

Feature	Dataset / expression support	Use and limits
Custom Decision Variables	Define variables in custom_variables.csv with a key, name, unit, lower/upper bounds and objective coefficient. Reference them in Custom Constraints. Advanced records support interval/hour/day/week/month/year periods and lagged definitions.	Continuous variables. Period/lag settings require source/canonical declarations; past references need opening history.
Reverse Polish Notation (RPN)	Fixed/affine postfix expressions reference generation, pumping, battery charge/discharge, line flow and custom variables, with supported lags.	Opt-in. Nonlinear decision expressions and unsupported control instructions are blocked by default.