Reproducibility package: rule-based vs fuzzy vs PSO-tuned EMS for a standalone PV–wind–battery–diesel microgrid

This repository accompanies the manuscript "Comparative Analysis of Rule-Based, Fuzzy Logic and PSO-Tuned

Energy Management Strategies for a Standalone Hybrid Renewable Microgrid: A Simulation Study".

It allows every table and figure of the paper to be regenerated. No measured field data are used:

all weather, load and generator-outage series are produced by the stochastic generator in `hres.py`.

Contents

| File | Purpose |

|---|---|

| `hres.py` | Plant models (PV, wind, battery, diesel), stochastic weather/load/outage generator, rule-based / fuzzy / PSO-tuned threshold controllers, PSO, battery aging (rainflow, Arrhenius), NPC/LCOE economics |

| `run_experiments.py` | Main experiments: training, 100-year baseline test, stress scenarios, economic sensitivities, 12-point sizing sweep, computational-cost timing -> `results.pkl` |

| `exp_ext.py` | Extension (does not modify `hres.py`): parameterised Mamdani FLC with six tunable design parameters, fast table-based fuzzy inference used only inside PSO fitness evaluations, generic PSO with seeded particle |

| `exp_flc.py` | Experiment A (equal-effort comparison): PSO tuning of the FLC with the same budget as the threshold policy -> `exp_flc.pkl` |

| `exp_train.py` | Experiment B (training-set sensitivity): threshold policy tuned with 1, 2, 5, 10 training years, 5 replicates each -> `exp_train.pkl` (checkpoints to `exp_train_partial.pkl`, resumable) |

| `exp_stats.py` | Paired statistics (bootstrap CI, Wilcoxon) |

| `make_figures.py` | Generates the figures from `results.pkl` |

| `build_docx.py`, `dx.py`, `refs.py` | Manuscript builder (not needed for the numerical results) |

| `results.pkl`, `exp_flc.pkl`, `exp_train.pkl` | Raw outputs behind every table and figure |

How to reproduce

1. Python 3 with NumPy, SciPy, Matplotlib (and python-docx for the manuscript builder). Record exact versions with `pip freeze > requirements.txt`.

2. `python run_experiments.py` (about 11 min on one CPU core; regenerates `results.pkl`)

3. `python exp_flc.py` (about 19 min; needs `results.pkl`; use a copy named `results_orig.pkl` as in the script header)

4. `python exp_train.py` (about 35 min; same requirement)

5. `python make_figures.py`

All random seeds are fixed. A re-run of `run_experiments.py` reproduced the stored `results.pkl` exactly (maximum absolute difference 0 over all test, stress, sizing-sweep and sensitivity outputs).

Timing figures (microseconds per decision, PSO seconds) depend on the machine and will differ.

Seed schedule (weather/load years)

| Use | Seeds |

|---|---|

| PSO training (two years) | 11–12 |

| Baseline out-of-sample test (100 years) | 1000–1099 |

| Stress scenarios (30 years each) | 2000–2029 |

| Economic sensitivities | 3000–3029 |

| Sizing sweep (20 years per cell) | 4000–4019 |

| Training-set study (Experiment B), replicate r, size n | 6000 + 100 r + k, k = 0..n-1 |

PSO: 24 particles, 50 iterations, five independent runs (swarm seeds 100–104); the run with the lowest training fitness defines the tuned policy. Sizing sweep: 16 particles, 30 iterations.

Tuned FLC parameters (Experiment A): SoC-membership shift, normalisation reference power, charge gain, base discharge fraction, discharge floor, generator charging allowance; bounds and defaults are in `exp_ext.py`.

Where parameters are documented

Assumed monthly climatology: manuscript Table 1. Plant configuration and model parameters: Table 2. 27-rule fuzzy rule base: Table 3. PSO settings, expert thresholds and tuned parameters: Table 4. Economic and battery-aging parameters: Table 5. Verification tests: Table 6. Equal-effort comparison: Table 12. Training-set sensitivity: Table 13.

Scope and limitations

The data are synthetic and calibrated to assumed monthly climatology; results describe a representative synthetic site.

The generator can be replaced by measured hourly series with a one-line change. See Sections 4.10–4.12 of the manuscript.