Sizing and Operations of Energy Systems Using GBOML

Meeting
ULiège, Belgium
Summer 2023

Bardhyl Miftari, Guillaume Derval and Damien Ernst

Table Of Content

Energy System Sizing and Operations

Examples

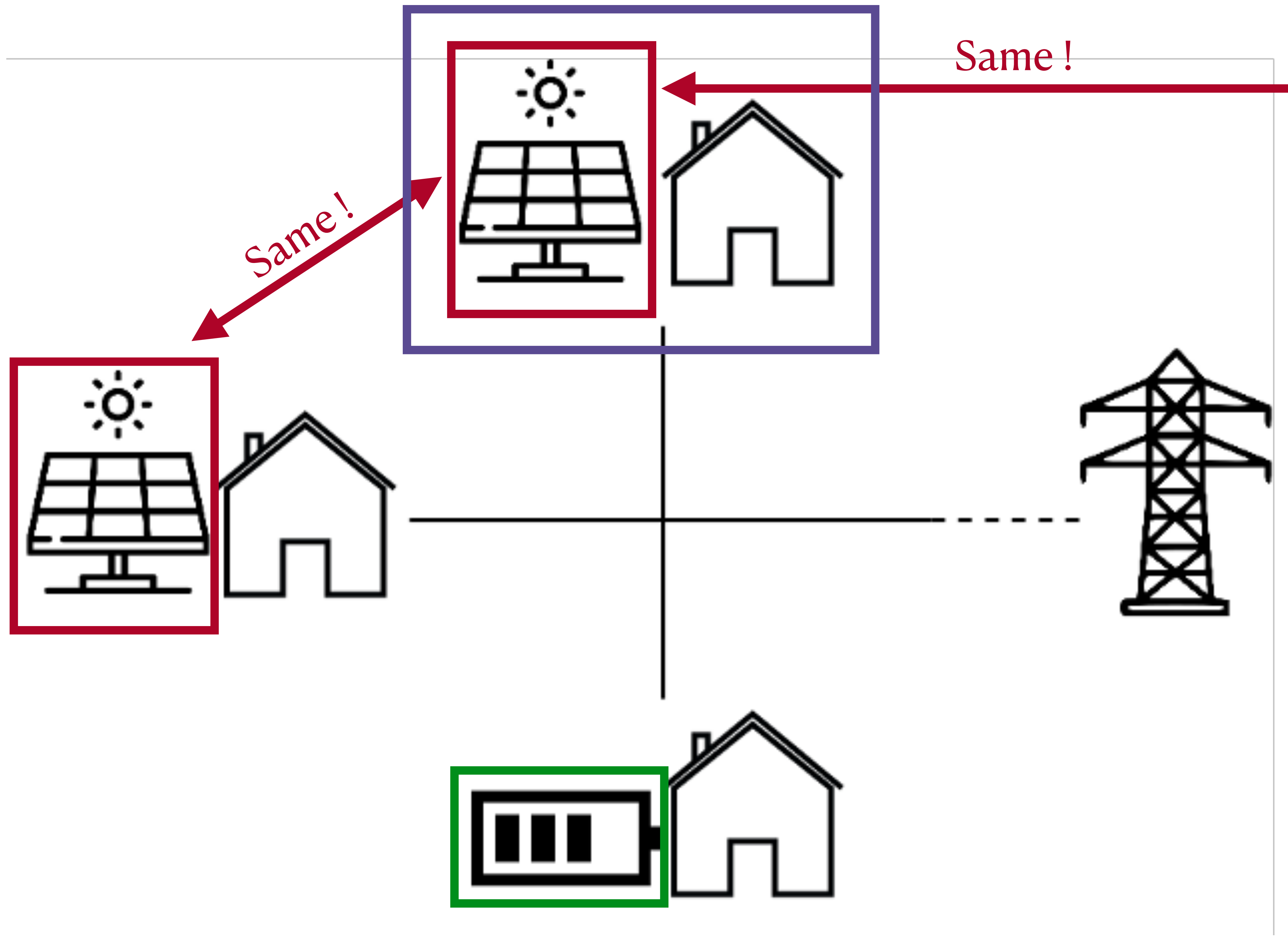


FIGURE 1: renewable energy community

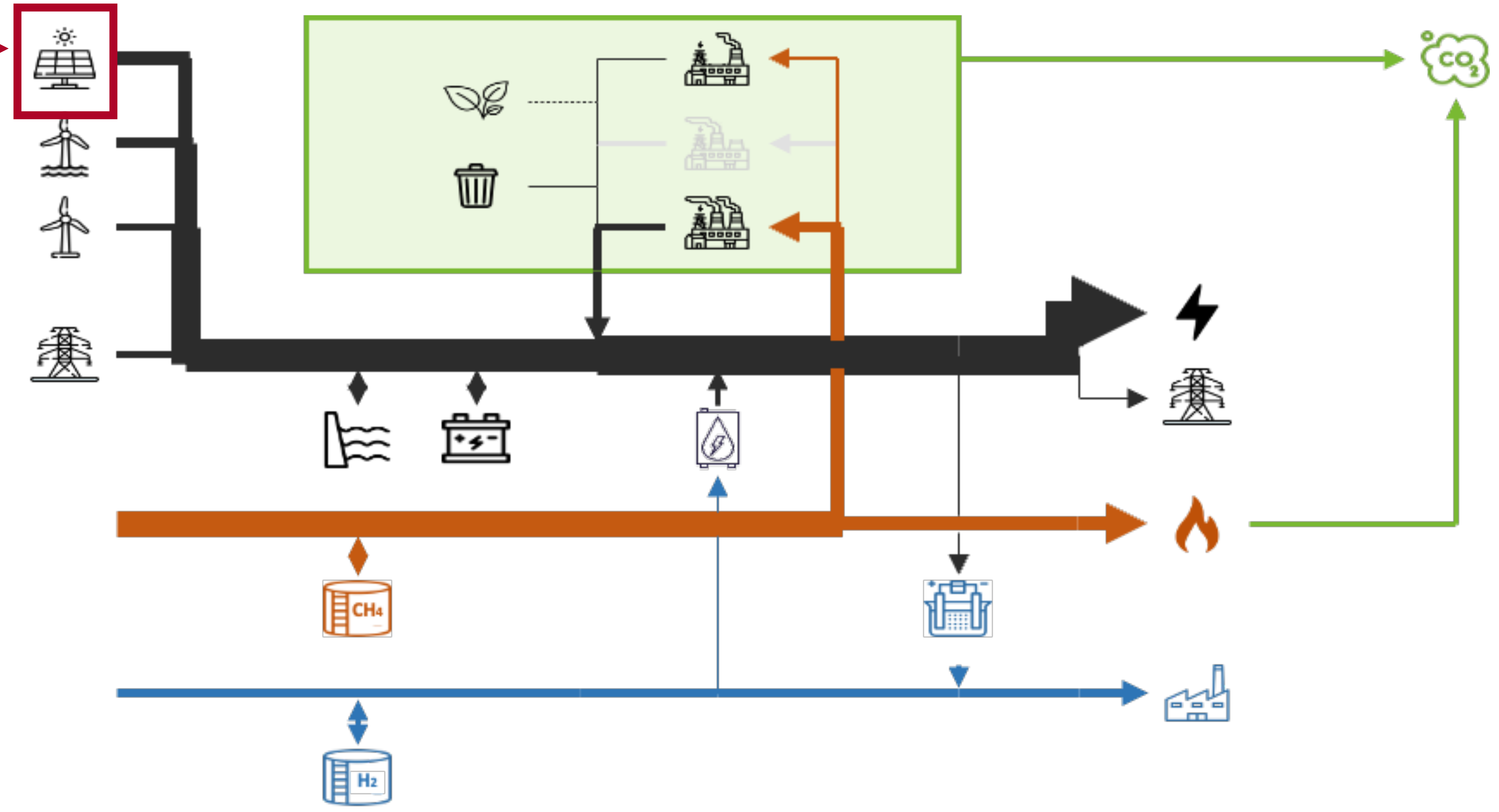


FIGURE 2: Belgian energy model

Energy System Sizing and Operations

Properties

- **Time is essential component**
  - Time-dependent systems
  - Optimized over a time period
- **Network of components**
  - Interconnection of independent components
  - Unique topologies

Energy System Sizing and Operations

The Basics

- **Energy System Modeling**
  - Modeling: Creating a (mathematical) representation of a physical system in order to enable its study
  - Energy System Modeling: Creating a representation of an energy system to answer a certain question or achieve a certain goal
- **Energy System Sizing**
  - Finding the optimal energy system size in order to achieve a certain goal
    e.g. What are the battery capacities needed in a given system ?
- **Energy System Operations**
  - Finding the optimal operations to perform in order to achieve a certain goal
    e.g. When do I charge or discharge my battery ?
- **Overall Objective**
  e.g. Minimizing the overall cost (investments and operation) or the environmental footprint
- **Energy systems sizing and operations**
  - One depends on the other

Energy System Sizing and Operations

An Example

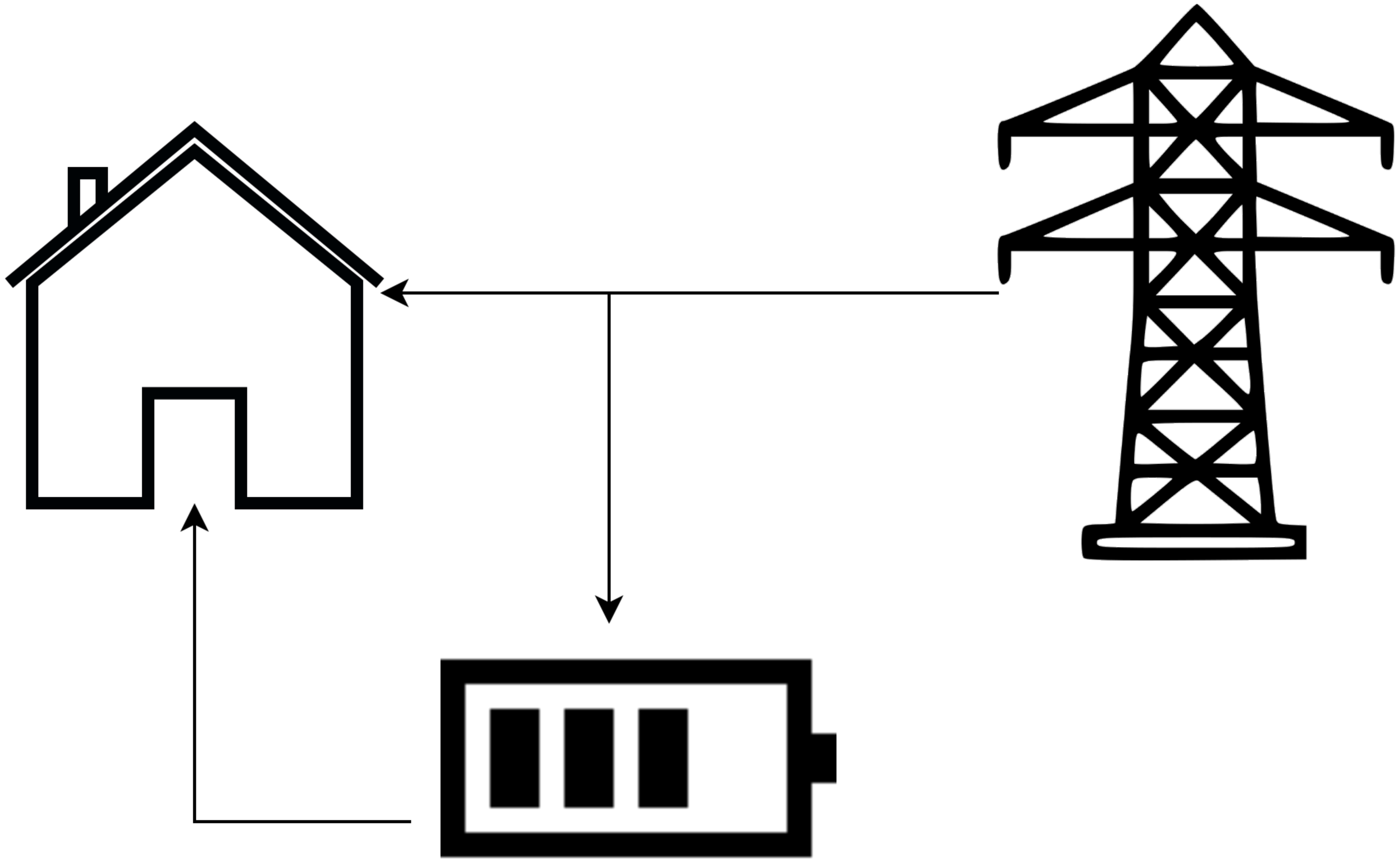


FIGURE 3 : Installing the optimal battery capacity given a known demand and a known hourly price of electricity and operating it.

Energy System Sizing and Operations

Finding a Solution

- Heuristics or iterative methods
  - Genetic algorithms
- Mathematical optimization
  - Expressed as optimizing a function over a feasible set

$$\min f(\mathbf{x}) \quad \text{s.t. } \mathbf{x} \in \mathcal{X}$$

  - The function f and the expression of the set $\mathcal{X}$ determines the optimization type (quadratic, non-linear, mixed integer, [...] programming)

Mixed-Integer Linear Programming [1]

The Basics

- Problem formulation:
    - Linear objective function
    - Feasible set is expressed as linear constraints

$$\min \ \mathbf{c}^T\mathbf{x}$$
$$\text{s.t.} \ \mathbf{A}\mathbf{x} \leq \mathbf{b}$$

- Enables to deal with relatively large models
- Non-linearities can be approximated with linear-piecewise functions

1 Abbreviation : MILP

Mixed-Integer Linear Programming

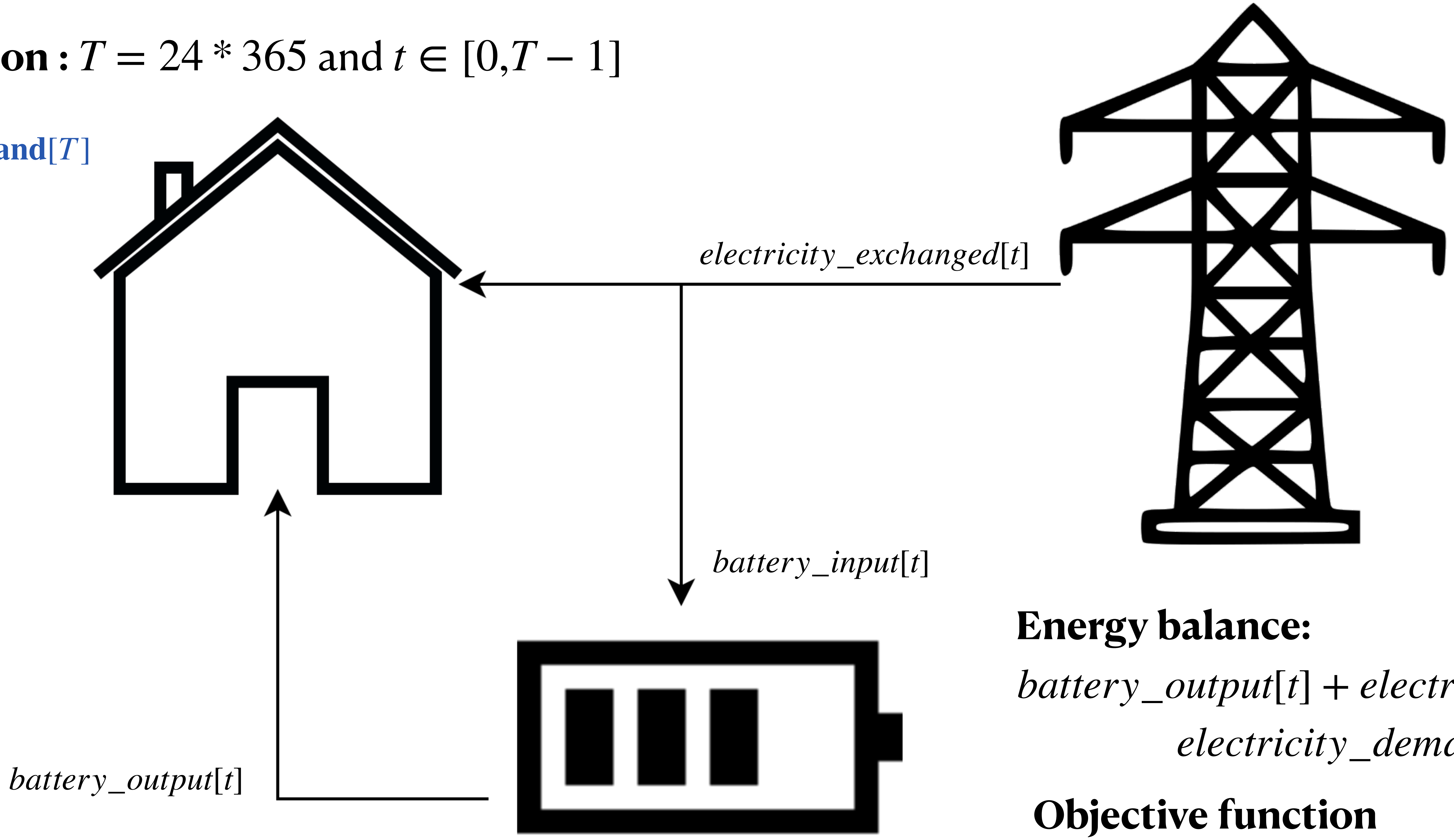
An MILP Example

Optimization horizon : $T = 24 * 365$ and $t \in [0, T-1]$

Known: **electricity_demand**$[T]$

Known: **electricity_price**$[T]$

electricity_exchanged$[T]$



state_of_charge$[T]$

$battery\_capacity$

Known: $battery\_price$

battery_output$[T]$

battery_input$[T]$

Energy balance:

$$battery\_output[t] + electricity\_exchanged[t] == electricity\_demand[t] + battery\_input[t]$$

Objective function

$$\min electricity\_exchanged[t] * electricity\_price[t] + battery\_capacity * battery\_price$$

Mixed-Integer Linear Programming

An MILP Example

Optimization horizon : $T = 24 * 365$ and $t \in [0, T-1]$

Known: $\mathbf{electricity\_demand}[T]$

Known: $\mathbf{electricity\_price}[T]$

Known: $battery\_price$

$\mathbf{state\_of\_charge}[T]$

$\mathbf{electricity\_exchanged}[T]$

$\mathbf{battery\_output}[T]$

$\mathbf{battery\_input}[T]$

$battery\_capacity$

$state\_of\_charge[t] \geq 0$

$electricity\_exchanged[t] \geq 0$

$battery\_output[t] \geq 0$

$battery\_input[t] \geq 0$

$battery\_capacity \geq 0$

Energy balance:

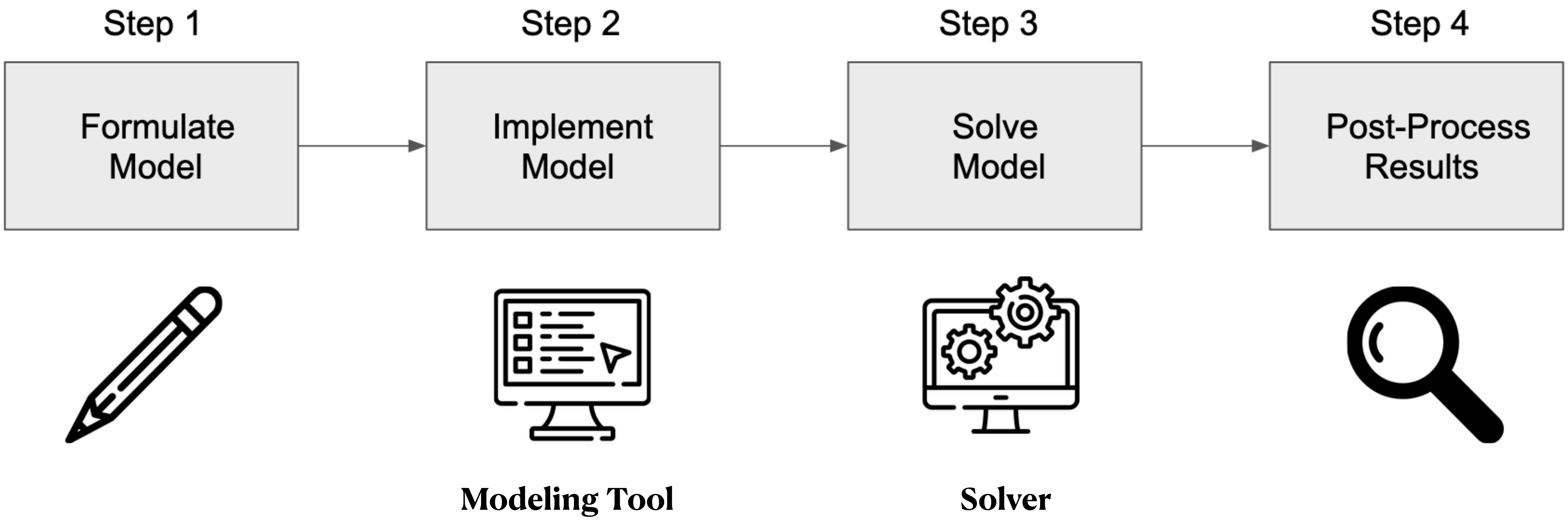
$$battery\_output[t] + electricity\_exchanged[t] == electricity\_demand[t] + battery\_input[t]$$

Objective function

$$\min: \quad electricity\_exchanged[t] * electricity\_price[t] + battery\_capacity * battery\_price$$

Mixed-Integer Linear Programming

Workflow



Solvers

An Overview

- Commercial solvers

  [1] [2] [3]

- Open-source solvers

  [4] [5] [6]

- Meta-solvers
  - DSP [7]

Modeling Tools

Basics

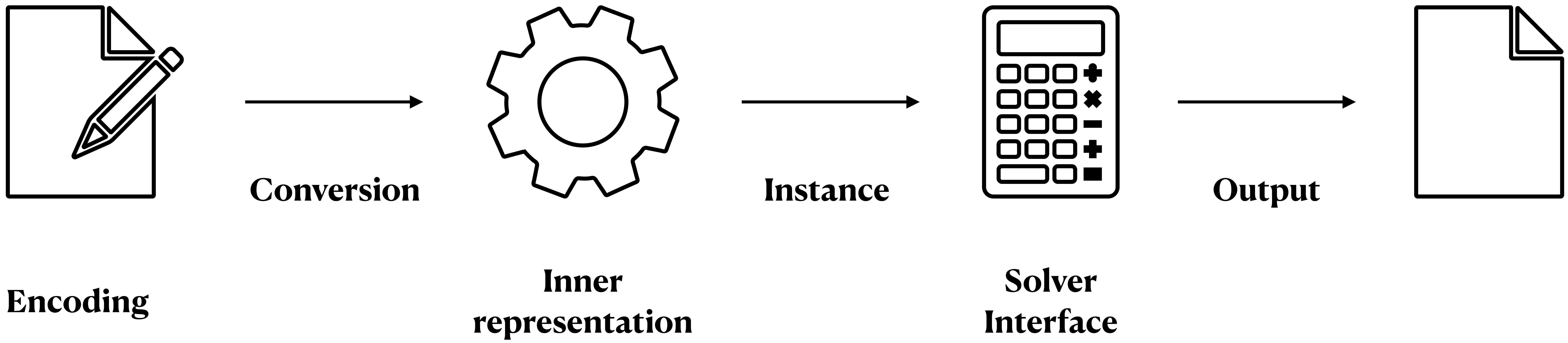


Figure 4: Modelling tools workflow

Modeling Tools

AMLs

- **A**lgebraic **M**odeling **L**anguages (**AMLs**)
  - Formulation close to mathematical notation
  - Very expressive (e.g. can represent any mixed-integer nonlinear program)
  - Often interface with multiple solvers
  - Examples:

[8] [9] [10] [11] [12]

Modeling Tools

OOMEs

- **O**bject-**O**riented **M**odeling **E**nvironments (**OOME**s)
  - Focus on one particular application (e.g. energy system sizing and operations)
  - Usually make use of predefined components that are “imported”
  - Typically have advanced data processing capabilities tailored to the application
  - Often open-source
  - Examples:

PyPSA [13]

Calliope [14]

PLEXOS [15]

BALMOREL Energy system model [16]

solph ['sɒlv] open energy modelling framework [17]

Dispa-SET Power system modelling [18]

Modeling Tools

Drawbacks of AMLs and OOMEs

- AMLs:
  - Fail to expose block structure
  - Do not enable reuse or do not have import-like capabilities
- OOMEs:
  - Lack the expressiveness of AMLs
  - Often cumbersome to add new components
  - Often rely on AMLs and inherit their shortcomings

Going Further

GBOML

- The **G**raph-**B**ased **O**ptimization **M**odeling **L**anguage (**GBOML**)[19-20] combines the strengths of AMLs and OOMEs
  - Open-Source and Stand-alone
  - Can represent any MILP
  - Exploits structure in various ways
  - Syntax close to the mathematical notation
  - Time-indexed models can be encoded easily
  - Allows component definition, re-use and component assembling
  - Interfaces with various solvers

Modeling Tools

GBOML

- Software developed in Python:
  - Few dependencies (PLY, NumPy, SciPy)
  - Provides two methods to encode models (text file and Python API)
  - Interfaces with several Solvers (Cplex, Gurobi, Xpress, HiGHS, CLP/CBC, DSP)
  - Produces plain .csv and structured .json outputs
- Fully documented - Clear issue handling

Going Further

GBOML

- In GBOML, structure is exploited at all levels:

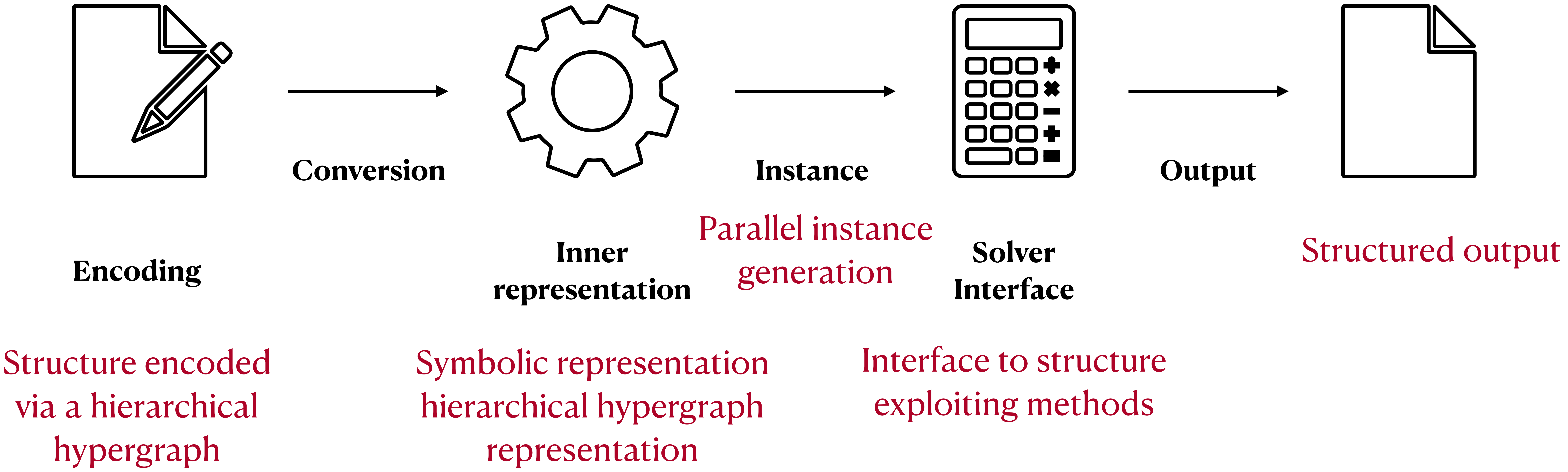


Figure 5: GBOML structure exploiting workflow

Modeling Tools

GBOML Hierarchical Hypergraph

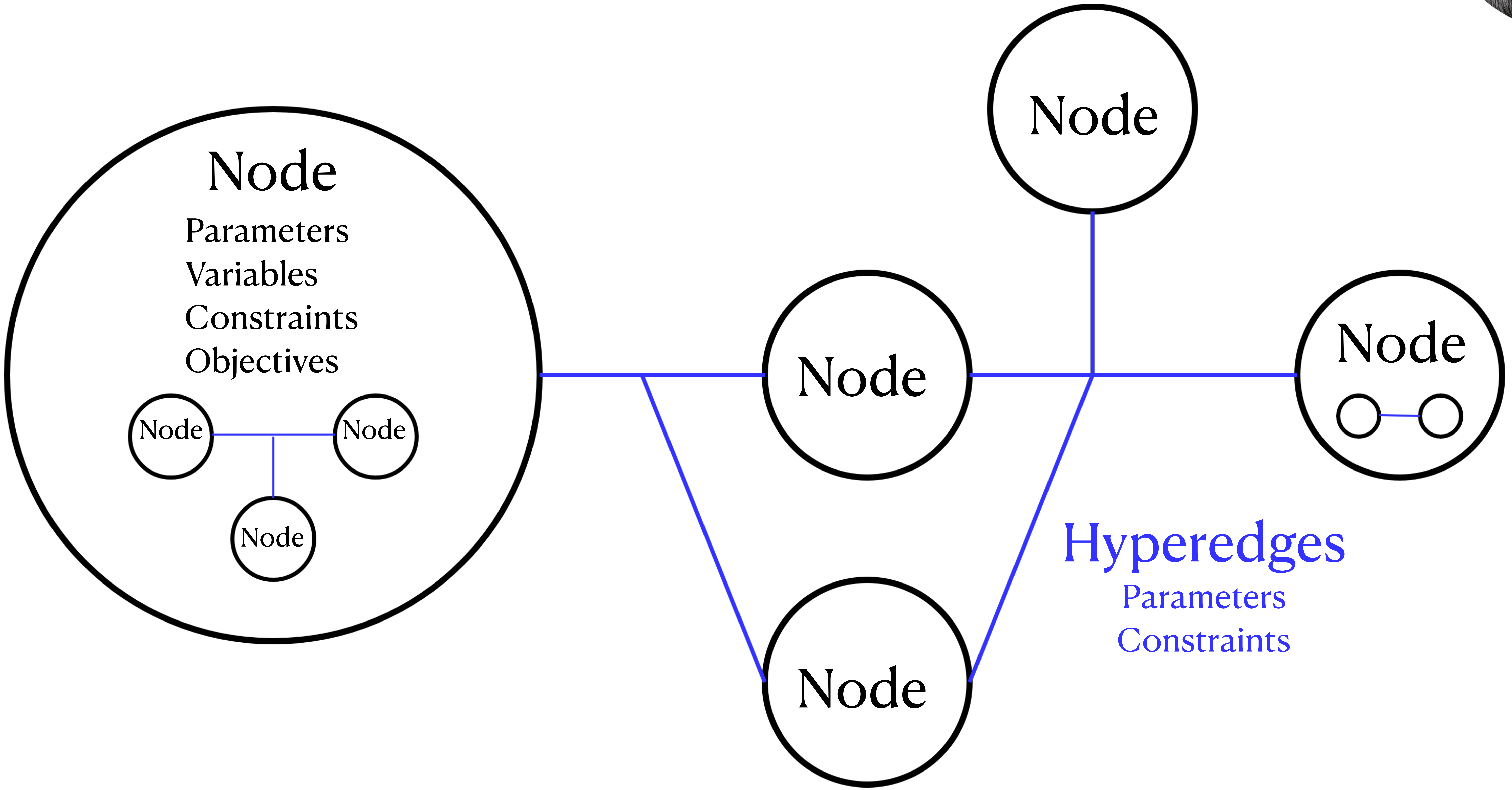


FIGURE 6 : Representation of one particular hierarchical hypergraph made-up of 5 nodes and 2 hyperedges. The node most to the left and to the right both contain a hypergraph themselves.

Modeling Tools

GBOML Hierarchical Hypergraph

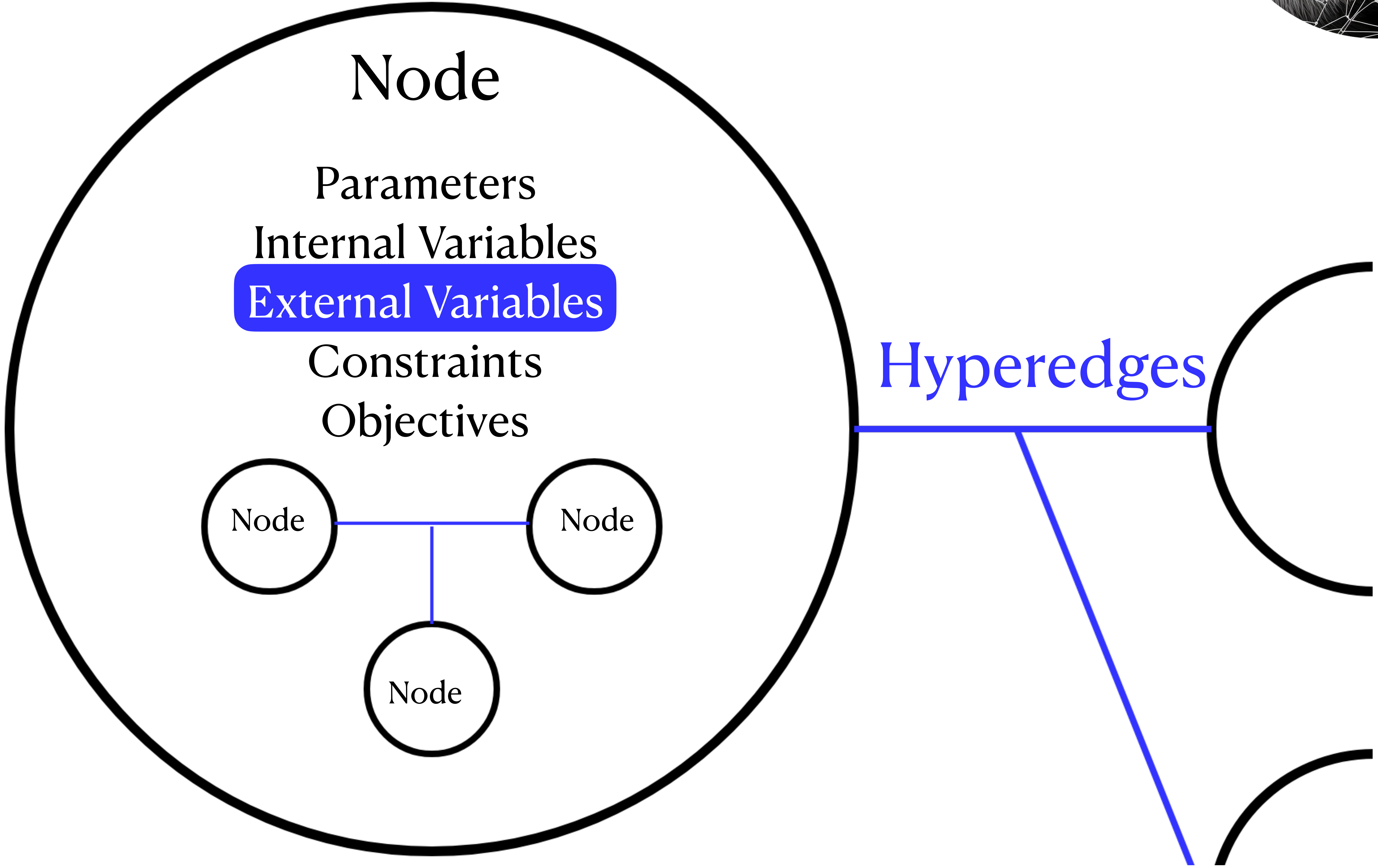


FIGURE 7 : Representation of one node made-up of parameters, internal/external variables, constraints, objectives and a hypergraph. The hyperedges connect only the external variables of different nodes.

Modeling Tools

GBOML Language

```
#TIMEHORIZON
 T = <value>;
```

```
#NODE <node_name>
 #PARAMETERS
  <param_def>
 #VARIABLES
  <var_def>
 #CONSTRAINTS
  <constr_def>
 #OBJECTIVES
  <obj_def>
```

```
#HYPEREDGE <edge_name>
 #PARAMETERS
  <param_def>
 #CONSTRAINTS
  <constr_def>
```

Modeling Tools

GBOML Language

```
#TIMEHORIZON
T = <value>;
```

```
#NODE <node_name>
 #PARAMETERS
  <param_def>
 #VARIABLES
  <var_def>
 #CONSTRAINTS
  <constr_def>
 #OBJECTIVES
  <obj_def>
```

```
#HYPEREDGE <edge_name>
 #PARAMETERS
  <param_def>
 #CONSTRAINTS
  <constr_def>
```

Modeling Tools

GBOML Language

```
#TIMEHORIZON
T = <value>;
```

```
#NODE <node_name>
 #PARAMETERS
  <param_def>
 #VARIABLES
  <var_def>
 #CONSTRAINTS
  <constr_def>
 #OBJECTIVES
  <obj_def>
```

```
#HYPEREDGE <edge_name>
 #PARAMETERS
  <param_def>
 #CONSTRAINTS
  <constr_def>
```

Modeling Tools

An Example in GBOML: Battery System

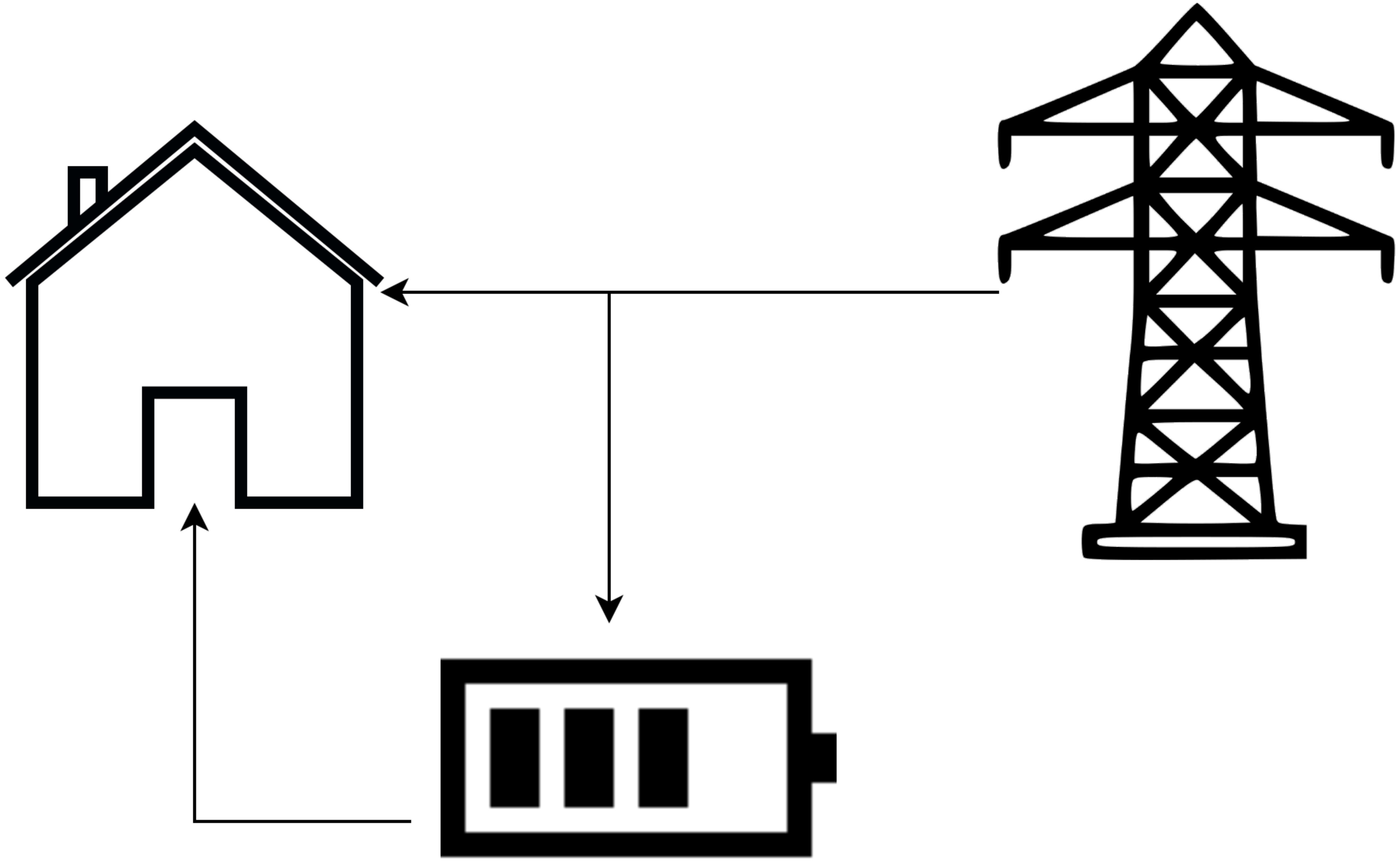


FIGURE 8 : Installing the optimal battery capacity given a known demand and a known hourly price of electricity and operating it.

Modeling Tools

An Example in GBOML: Battery System

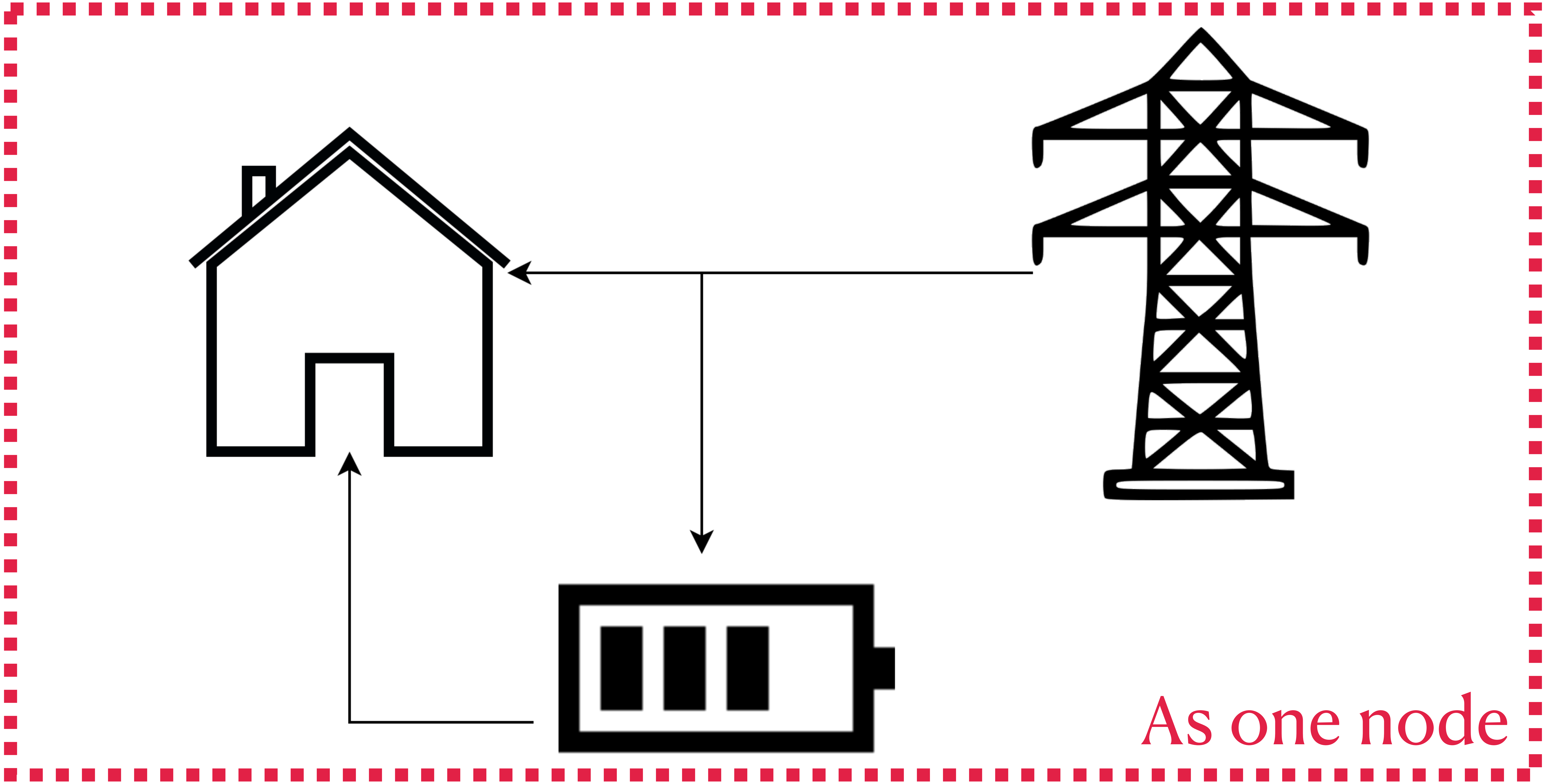


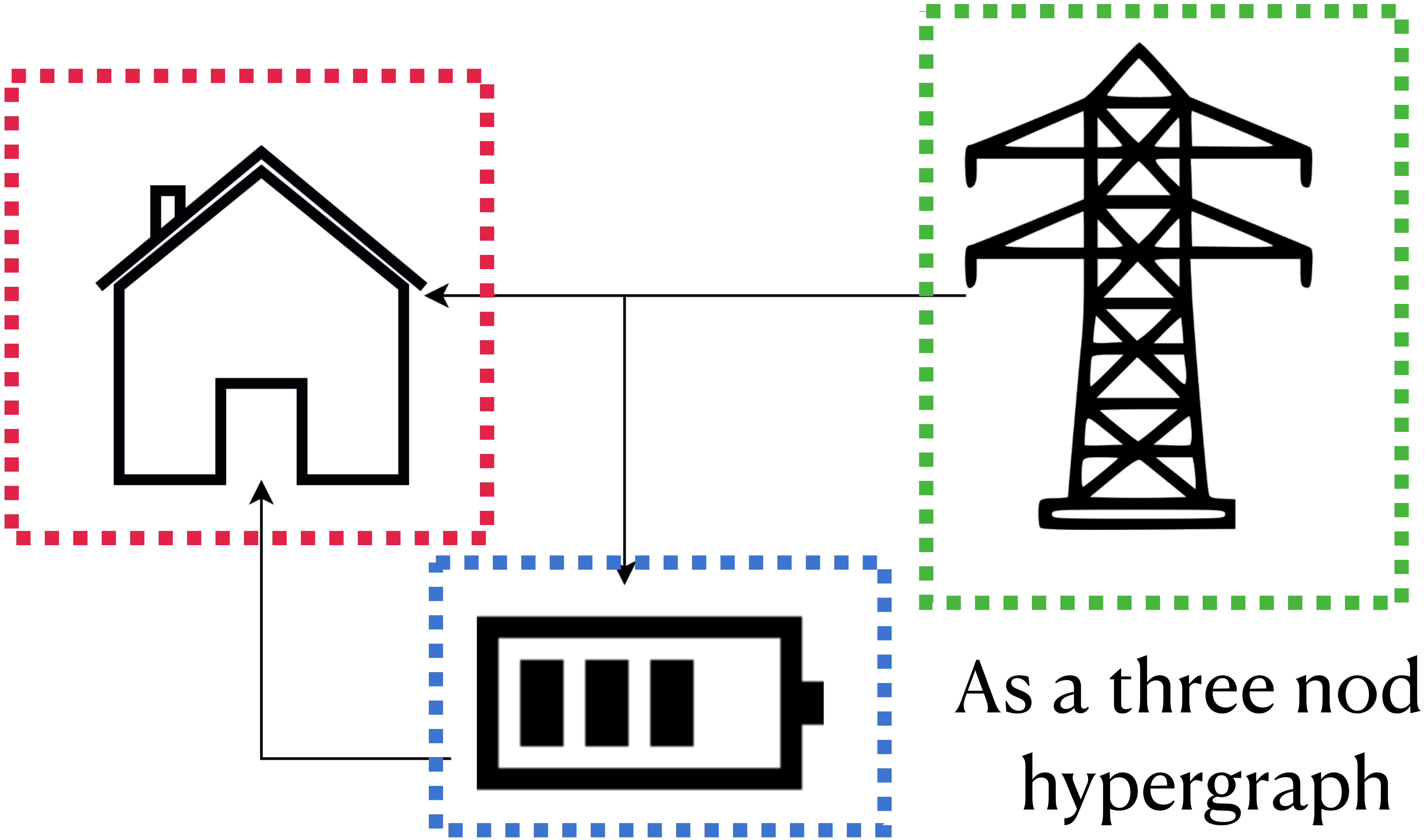
FIGURE 8 : Installing the optimal battery capacity given a known demand and a known hourly price of electricity and operating it.

```
#TIMEHORIZON T = 24*365;

#NODE Bat_House_Grid
#PARAMETERS
  elec_demand = import «demand.csv»;
  elec_price = import «elec_price.csv»;
  bat_price = 120;
#VARIABLES
  internal: electricity_exchanged[T];
  internal: battery_output[T];
  internal: battery_input[T];
  internal: state_of_charge[T];
  internal: battery_capacity;
#CONSTRAINTS
  electricity_exchanged[t] >= 0;
  battery_output[t] >= 0;
  state_of_charge[t] >= 0;
  battery_capacity >= 0;
  battery_capacity >= state_of_charge[t];
  battery_input[t] <= battery_capacity;
  battery_output[t] <= battery_capacity;
  state_of_charge[0] == state_of_charge[T-1];
  state_of_charge[t+1] == state_of_charge[t]+battery_input[t]-battery_output[t];
  battery_output[t]+electricity_exchanged[t] == elec_demand[t]+battery_input[t];
#OBJECTIVES
  min: electricity_exchanged[t]*elec_price[t];
  min: battery_capacity*bat_price;
```

Modeling Tools

An Example in GBOML : Battery



As a three node hypergraph

FIGURE 8 : Installing the optimal battery capacity given a known demand and a known hourly price of electricity and operating it.

```
#TIMEHORIZON T = 24*365;
```

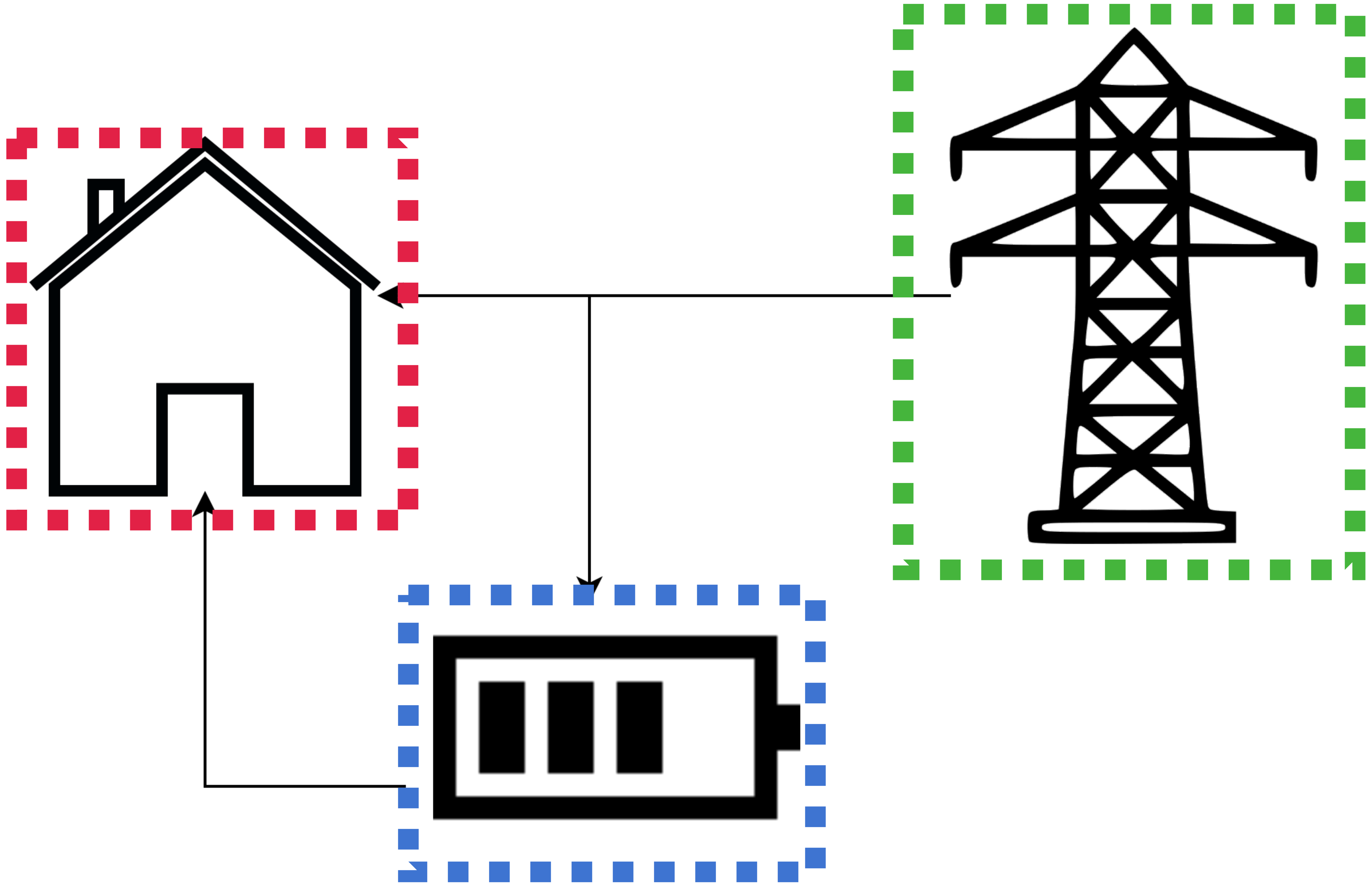
```
#NODE Battery
#PARAMETERS
  bat_price = 120;
#VARIABLES
  external: battery_output[T];
  external: battery_input[T];
  internal: state_of_charge[T];
  internal: battery_capacity;
#CONSTRAINTS
  battery_output[t] >= 0;
  state_of_charge[t] >= 0;
  battery_capacity >= 0;
  battery_capacity >= state_of_charge[t];
  battery_input[t] <= battery_capacity;
  battery_output[t] <= battery_capacity;
  state_of_charge[0] == state_of_charge[T-1];
  state_of_charge[t+1] == state_of_charge[t]
                          + battery_input[t]
                          - battery_output[t];

#OBJECTIVES
  min: battery_capacity*bat_price;
```

```
#NODE Grid
#PARAMETERS
  elec_price = import «elec_price.csv»;
#VARIABLES
  external: electricity_exchanged[T];
#CONSTRAINTS
  electricity_exchanged[t]>=0;
#OBJECTIVES
  min: electricity_exchanged[t]*elec_price[t];
```

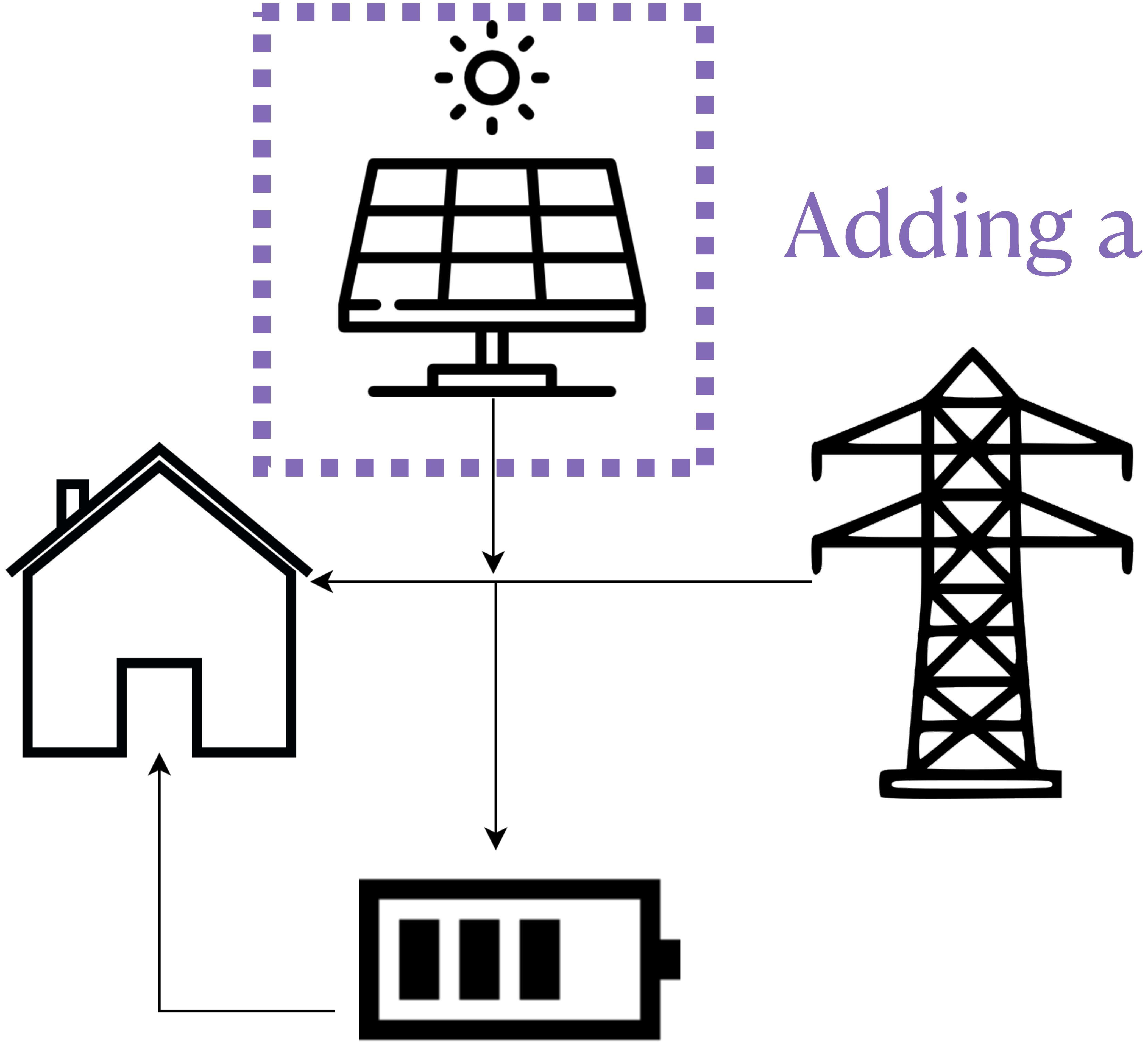
```
#NODE House
#PARAMETERS
  elec_demand = import «demand.csv»;
#VARIABLES
  external:demand[T];
#CONSTRAINTS
  demand[t] == elec_demand[t];
```

```
#HYPEREDGE Interconnection
#CONSTRAINTS
  Battery.battery_output[t]+Grid.electricity_exchanged[t]
  == House.demand[t]+Battery.battery_input[t];
```



Modeling Tools

An Example in GBOML: Battery - PV Panels



Adding a new node

FIGURE 9 : Installing the optimal battery capacity and PV capacity given a known demand and a known hourly price of electricity and operating it.

```
#TIMEHORIZON T = 24*365;

#NODE Battery = import Battery from ”house_bat_grid_3_node.txt”;

#NODE Grid = import Grid from ”house_bat_grid_3_node.txt”;

#NODE House = import House from ”house_bat_grid_3_node.txt”;

#NODE PV_panels
  #PARAMETERS
    cost = 110;
    irradiance = import ”irradiance.csv”;
  #VARIABLES
    external: electricity_prod[T];
    internal: capacity;
  #CONSTRAINTS
    electricity_prod[t] == irradiance[t]*capacity;
  #OBJECTIVES
    min: capacity*cost;
#HYPEREDGE Interconnection
  #CONSTRAINTS
    Battery.battery_output[t]+Grid.electricity_exchanged[t]+PV_panels.electricity_prod[t]
    == House.demand[t]+Battery.battery_input[t];
```

Modeling Tools

An Example in GBOML: Renewable Energy Community

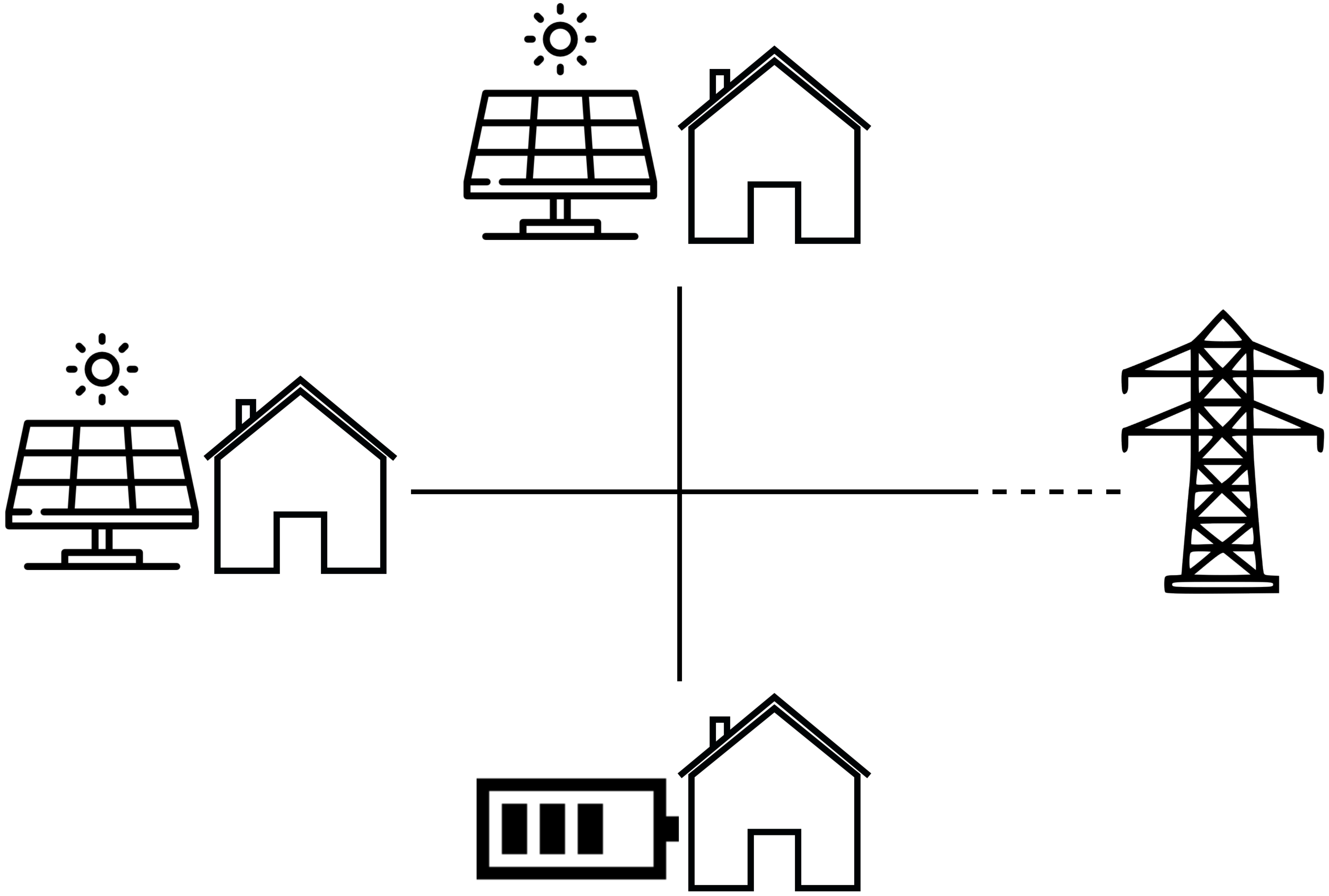
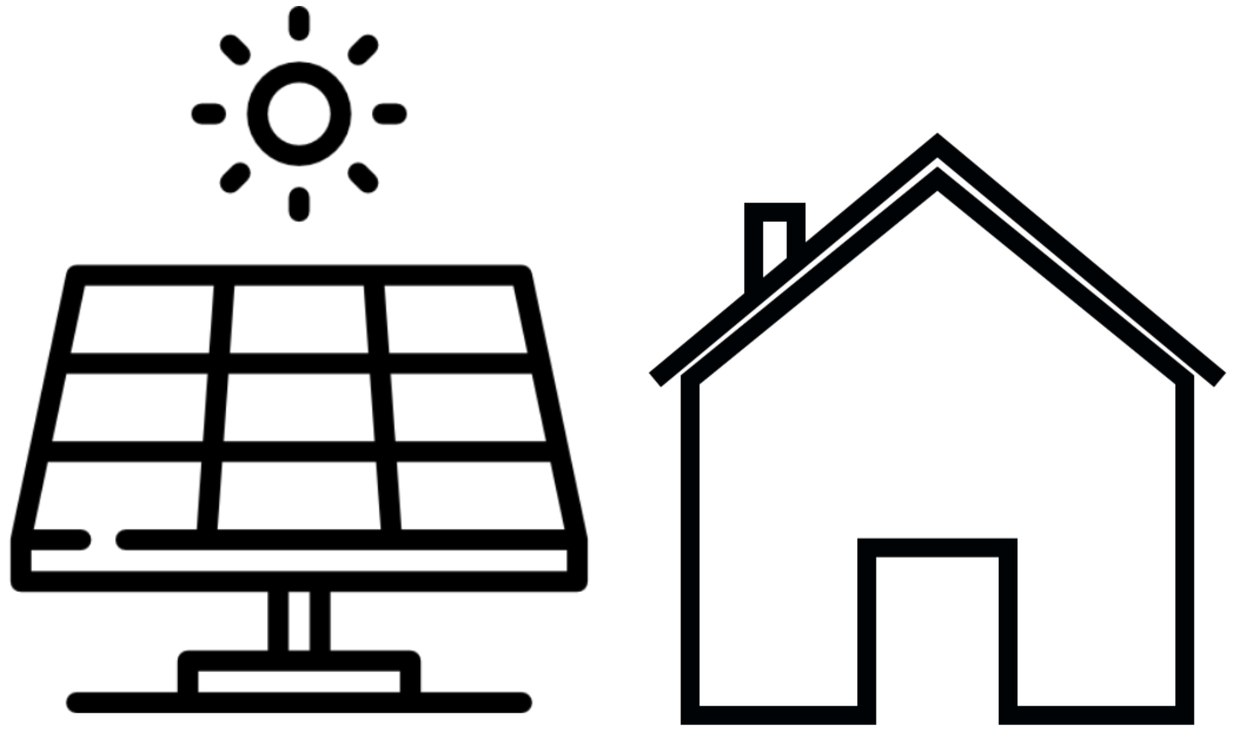


FIGURE 10 : Installing the optimal battery capacity and PV capacity in a renewable energy community

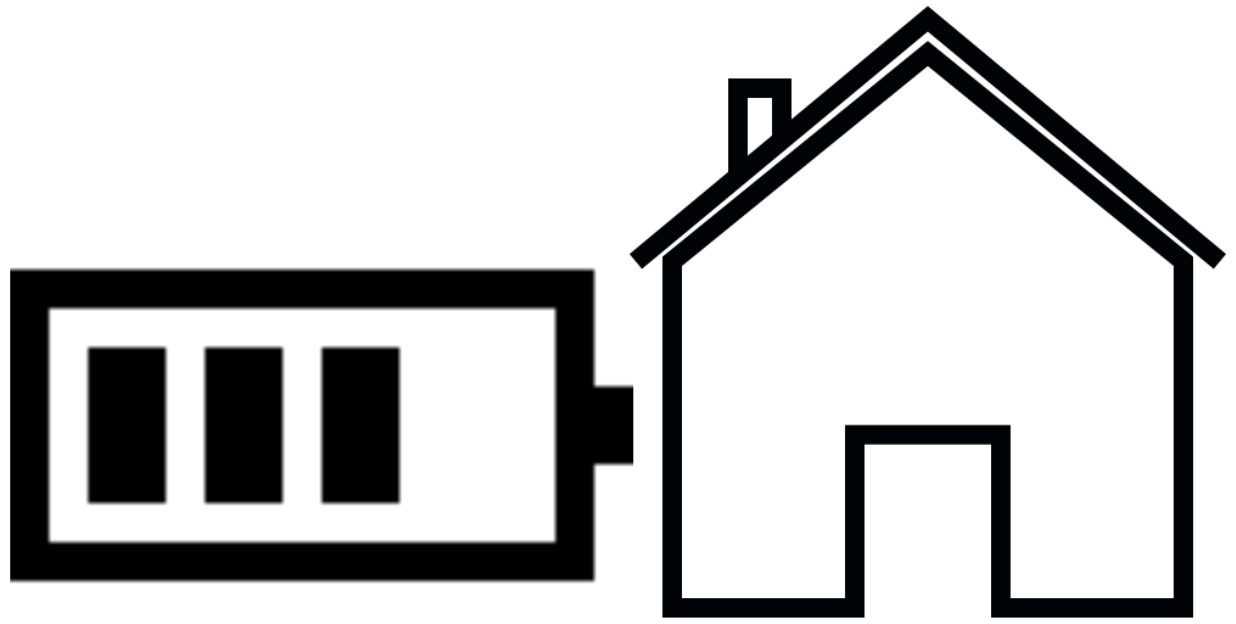


```
#NODE Prosumer
  #PARAMETERS
    elec_demand = import ”elec.csv”;
    cost = 110;
    irradiance = import ”irradiance.csv”

  #NODE House = import Grid from ”house_bat_grid_3_node.txt“ with
    elec_demand = Prosumer.elec_demand;

  #NODE PV = import PV_panels from ”house_bat_grid_pv.txt” with
    cost = Prosumer.cost;
    irradiance = Prosumer.cost;

  #VARIABLES
    external : pv_prod[T] <- PV.electricity_prod[T];
    external : demand[T] <-  House.demand[T];
```



```
#NODE Bat_consumer
  #PARAMETERS
    cost_bat = 110;
    elec_demand = import ”elec_demand.csv”;

  #NODE House = import House from ”house_bat_grid_3_node.txt” with
  elec_demand = Bat_consumer.elec_demand;

  #NODE Battery = import Battery from ”house_bat_grid_pv.txt” with
    bat_price = Prosumer.cost_bat;

  #VARIABLES
    internal : bat_input[T] <- Battery.battery_input[T];
    external : bat_output[T] <- Battery.battery_output[T];
    internal : energy_demand[T] <-  House.demand[T];
    external : demand[T];

  #CONSTRAINTS
    demand[t] == bat_input[t] + energy_demand[t];
```

Modeling Tools

An Example in GBOML: Renewable Energy Community

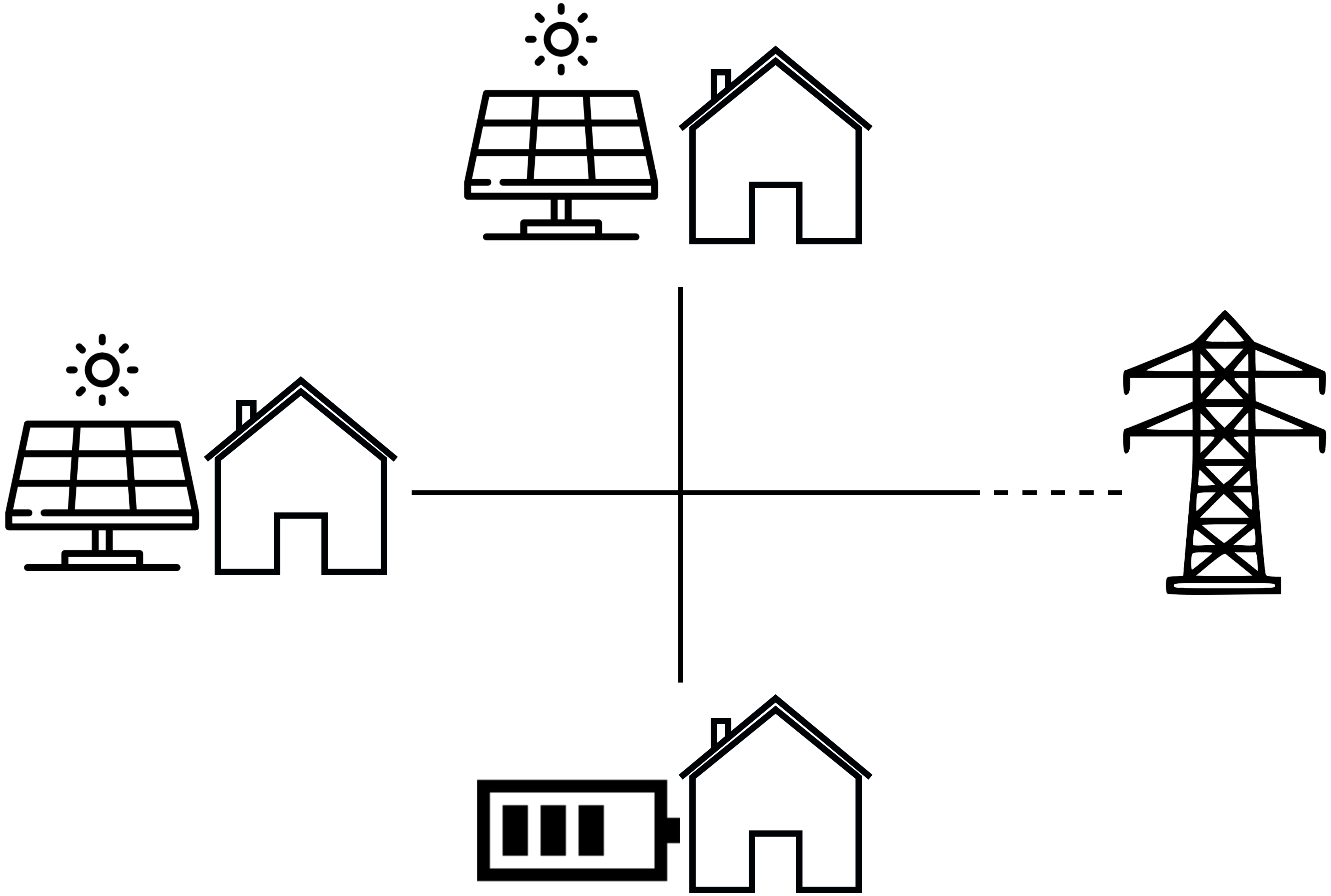


FIGURE 10 : Installing the optimal battery capacity and PV capacity in a renewable energy community

```
#TIMEHORIZON T = 24*365;

#NODE Bat_consumer = import Bat-consumer from ”bat_consumer.txt”;

#NODE Prosumer1 = import Prosumer from ”prosumer.txt”;

#NODE Prosumer2 = import Prosumer from ”prosumer.txt”;

#NODE Grid = import Grid from ”house_bat_grid_3_node.txt”;

#HYPEREDGE Interconnection
  #CONSTRAINTS
    Grid.electricity_exchange[t]
    + Bat_consumer.bat_output[t]
    + Prosumer1.pv_prod[t]
    + Prosumer1.pv_prod[t] == Prosumer1.demand[t]
                              + Prosumer2.demand[t]
                              + Bat_consumer.demand[t];
```

Modeling Tools

GBOML Output

```json
{
    "version": "0.1.3",
    "model": {
        "horizon": 10,
        "number_nodes": 1,
        "global_parameters": {},
        "nodes": {
            "H": {
                "number_parameters": 1,
                "number_variables": 1,
                "number_constraints": 1,
                "number_expanded_constraints": 10,
                "number_objectives": 1,
                "number_expanded_objectives": 10,
                "parameters": {
                    "b": [
                        4
                    ]
                },
                "variables": [
                    "x"
                ]
            }
        },
        "hyperedges": {}
    },
    "solver": {
        "name": "linprog",
        "status": true
    },
    "solution": {
        "status": "optimal",
```

	A	B	C	D	E	F	G	H	I
1	DISTRIBUTION.operating_cost	0.34500000000000003	0.32000000000000006	0.305	0.29500000000000004	0.28500000000000003	0.27	0.24	0.2250000000000000000
2	DISTRIBUTION.power_import	6.9	6.400000000000001	6.1	5.9	5.700000000000001	5.4	4.8	4.5
3	DISTRIBUTION.unnamed_objective	2817.4100000000003							
4	DEMAND.consumption	6.9	6.400000000000001	6.1	5.9	5.700000000000001	5.4	4.8	4.5
5	BATTERY.capacity	-0.0							
6	BATTERY.investment_cost	0.0							
7	BATTERY.energy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	BATTERY.charge	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	BATTERY.discharge	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	BATTERY.unnamed_objective	0.0							
11	SOLAR_PV.capacity	-0.0							
12	SOLAR_PV.investment_cost	0.0							
13	SOLAR_PV.electricity	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	SOLAR_PV.investment	0.0							

Modeling Tools

GBOML Performance[29]

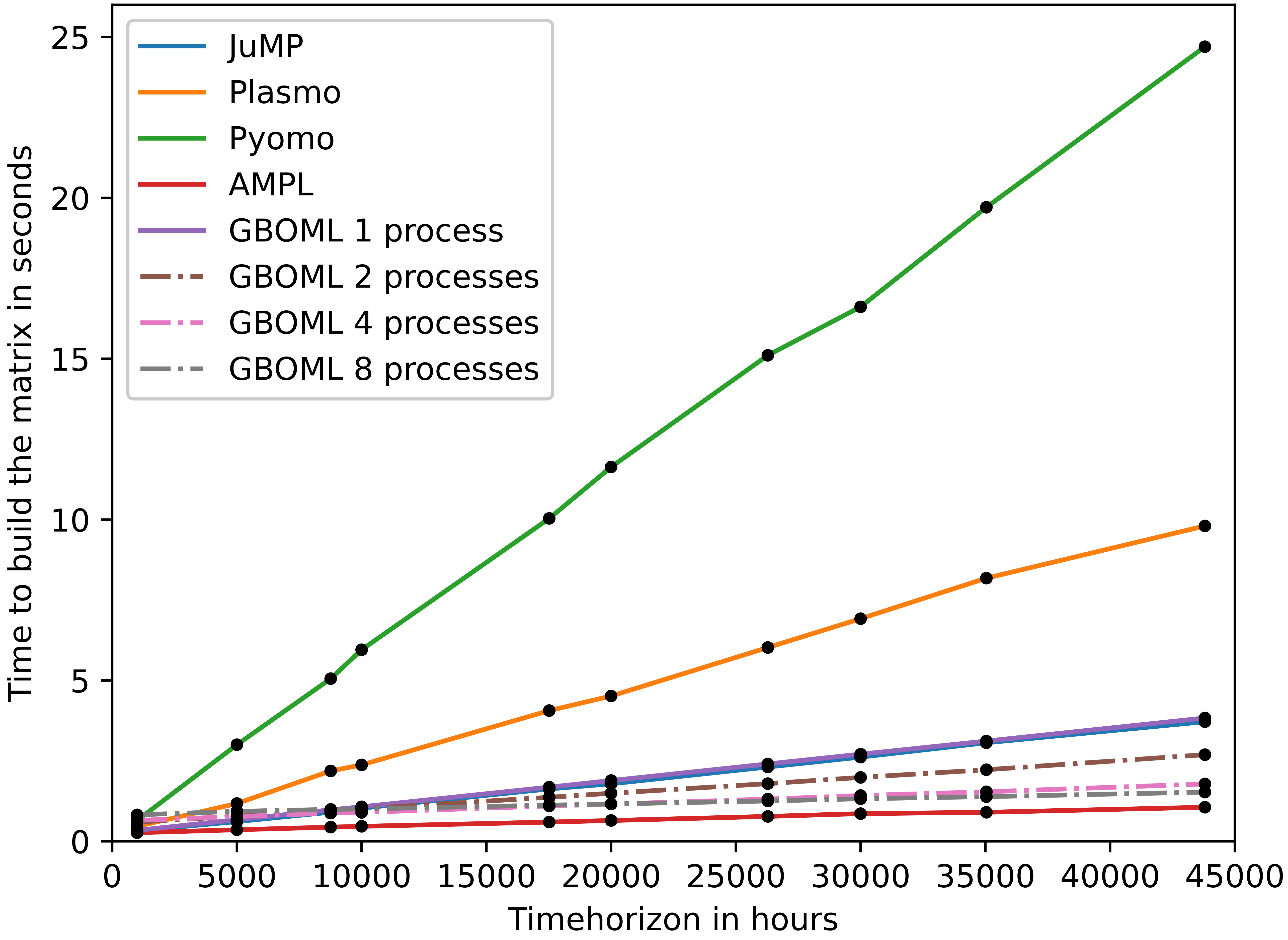


FIGURE 11 : Time taken to generate the matrices in different modeling tools for a growing time horizon for the remote hub [21]

Demo

http://tiny.cc/gboml_demo

Conclusion

GBOML

- Explained the sizing and operations of energy system
- Overview of the resolution process
- Introduced GBOML, a modeling tool for supply chain management and energy system sizing and operations
  - Easy to use and install
  - Allows model combination and re-use
  - Enables structure encoding
  - Fast
  - Interfaces with structure exploiting algorithms
- Illustrated several examples

Acknowledgments

- We would like to thank
  - SPF Economie (Federal government of Belgium)[22] for their financial support through the INTEGRATION project
  - The Walloon Region for their financial support through the INTEGCER project on renewable energy communities

References

[1] Gurobi Optimization, LLC. All Rights Reserved. https://www.gurobi.com/

[2] FICO® Xpress Optimization. https://www.fico.com/en/products/fico-xpress-optimization

[3] IBM ILOG CPLEX Optimizer. https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer

[4] SCIP, Solving Constraint Integer Programs. https://www.scipopt.org/

[5] HiGHS - high performance software for linear optimization. https://highs.dev/

[6] CBC/CLP from COIN-OR Foundation, Inc..https://www.coin-or.org/

[7] DSP, Argonne National Laboratory. https://github.com/Argonne-National-Laboratory/DSP

[8] The General Algebraic Modeling Language, GAMS. https://www.gams.com/

[9] JuMP https://jump.dev/JuMP.jl/stable/

[10] A Mathematical Programming Language, AMPL. https://ampl.com/

[11] Pulp. https://github.com/coin-or/pulp

[12] Pyomo. http://www.pyomo.org/

[13] PyPSA, Python for Power System Analysis. https://pypsa.org/

[14] Calliope. Calliope: a multi-scale energy systems modelling framework. https://calliope.readthedocs.io/en/stable/#

[15] Plexos, The Energy Analytics and Decision Platform for all Systems. https://www.energyexemplar.com/plexos

[16] Balmorel, http://www.balmorel.com/

[17] oemof.solph, https://github.com/oemof/oemof-solph

[18] The Dispa-SET model. http://www.dispaset.eu/en/latest/

[19] Bardhyl Miftari et al., ”GBOML: Graph-Based Optimization Modeling Language”, https://joss.theoj.org/papers/10.21105/joss.04158, 2022

[20] Bardhyl Miftari et al., “GBOML repository”, https://gitlab.uliege.be/smart_grids/public/gboml, 2021-23

[21] Bardhyl Miftari et al., ”GBOML: a Structure-exploiting Optimization Modeling Language in Python”, https://orbi.uliege.be/handle/2268/296930, 2022

[22] SPF Economie, Region Wallonne Belgique. https://economie.fgov.be/fr