Energy Efficiency and Risk Management in Public Buildings

Project Number 260041

SUPPORTING ACTION

EnRiMa

Energy Efficiency and Risk Management in Public Buildings

Deliverable D4.3: Stochastic Optimization Prototype

Start date of the project: 01/10/2010
Duration: 42 months
Organisation name of lead contractor for this deliverable: URJC
Revision: 3.0-0 (March 29, 2013)

Project funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COOPERATION

European Commission

Contents

List of Figures

List of Acronyms

API Application Programming Interface

CHP Combined Heat and Power

DB Database

DoW Description of Work

DSS Decision Support System

DV Decision Variable

DW Data Warehouse

EnRiMa Energy Efficiency and Risk Management in Public Buildings

HTTP Hypertext Transfer Protocol

GUI Graphical User Interface

PP Type of deliverable "restricted to other participants including the Commission Services"

PV Photovoltaic

REST Representational State Transfer

SG Scenario Generator

SM Solver Manager

SMS Symbolic Model Specification

ToU Time of Use

WP Work Package

XML Extended Markup Language

Glossary

Alias A second name for a set, usually used in order to distinguish between different parameter or variable indices within an equation, or even within the same variable or parameter. For example, for energy input and energy output.

Conditional set A special type of subset whose content depends on other sets. Example: Input (or output) energy for each technology. When the conditional set only contains one element, it can be seen as a function of other sets' elements.

Constant A fixed scalar value that does not depend on any set index.

Constraint An equation that restricts the possible values of variables.

DSS Kernel The DSS component that is responsible for providing functionality needed to manage the system data and to run the scenario generator tool and solver.

Equation A mathematical relationship between parameters and variables. For simplicity, we gather both mathematical concepts of equation (equality relationship) and inequation (inequality relationship) under the common notion "equation".

Expression Combination of variable and parameter symbols with their set symbols as super- or subindices.

Multidimensional set A way of implementing conditional sets in the optimizer. Multidimensional sets provide mappings between elements of different sets. Thus, one of the dimensions is a subset of a given set and the rest of the dimensions are the values of other sets that are related to the former. There may be one-to-one (e.g., parent nodes) and many-to-many (e.g., output types of energy for a technology) mappings.

Objective A function of variables and parameters that the decision maker wants to optimize (minimize or maximize).

Optimizer Software application capable to deal with optimizations problems. Optimizers usually can be configured to use different *solvers*. Besides the optimization itself, these applications can perform other related tasks, such as modeling, data analysis, or data visualization. Examples: GAMS [6], AMPL [5], Matlab [13], Pyomo [7].

Parameter A known or uncertain characteristic inherent to a set element or a combination of set elements, which is fixed and cannot be changed by the decision maker. Examples: Generation capacity of a given model of PV generator, the energy demand during a specific time span.

Set Collection of elements of a given class that are related and combined with other sets elements to parameters and decision variables. Example: technologies, types of energy.

Solver Computational algorithm that solves an optimization problem, i.e., receives input data and returns a problem solution. Examples: CPLEX [11], lp_solve [1].

Solver Manager (SM) The module of the EnRiMa DSS that runs the Optimizer using the input provided by the Interface and prepares the solution returned by the Optimizer for the Interface.

Solver Manager Core Script The main script within the Solver Manager. It is started by the SM Interface, prepares input and output data, manages files, and interacts with the Optimizer.

Solver Manager Interface The component of the Solver Manager that is responsible for the communication between the core of the Solver Manager and the DSS Kernel.

Stochastic Optimization For the scope of this document, *Stochastic Optimization* has the same meaning as *Solver Manager*, for consistency with the deliverable names in the DoW. Stochastic optimization model, stochastic optimization problem, and similar compound forms refer to the inherent mathematical concepts.

Subset Elements in a set that are similar with respect to some characteristic. Example: Generation technologies, storage technologies.

Symbol The representation for a set, variable, parameter, or equation within a SMS.

Symbolic Model Specification The mathematical representation (composed of variables, parameters, and relations between them) of the stochastic model of all relevant energy subsystems and their interactions.

Variable A variable, or decision variable, is a characteristic of a set element or combination of set elements, which is unknown and that the decision maker can change, given that the constraints are satisfied. Example: number of PV panels to be installed at a given year. For convenience, *calculated variables* can be defined through an equation. For example, the available capacity of a given technology during a given year is calculated from other variables and parameters.

Executive Summary

The Energy Efficiency and Risk Management in Public Buildings (EnRiMa) project, whose overall objective is to develop a decision-support system (DSS) for operators of energy-efficient buildings and spaces of public use, has achieved its third milestone (Prototype version of EnRiMa DSS) after 30 months of productive and encouraging work. This document is a description of one of the deliverables enabling such a milestone. Deliverable D4.3 (Stochastic Optimization Prototype), whose type of deliverable is "prototype," consists of several software components and their documentation. The aim of this document is to provide less technical, comprehensive additional documentation for that software. Moreover, the deliverable and its content is put into context with the rest of the project, as well as related to both previous and future work. This self-contained Executive Summary condenses the contents of the whole document. Sections 1 to 4 explain some details (yet not technical) of the software, while in-depth details can be found, in the appendices and the software documentation.

The Stochastic Optimization Prototype implements the Symbolic Model Specification (SMS), which defines the mathematical representation of the stochastic model of all relevant energy subsystems and their interactions. This SMS was formerly defined in D4.2 and has been improved since its delivery until arriving at the version implemented in the prototype, thanks to the joint work among several project partners while testing and building examples. Likewise, it will evolve even further during the final version's development and testing.

The Stochastic Optimization appears in the project's Description of Work (DoW) mainly in Task 4.5 –Stochastic optimisation algorithms and solvers. Nonetheless, this limited initial vision has evolved into a so-called **Solver Manager**, which includes not only solution matters but also an accurate management of input and output data as well as communication capabilities through an **Interface** for a better integration with the rest of the DSS modules.

The Solver Manager Interface allows us to separate communication tasks and other interaction features from the core features of the Solver Manager. This modularity leads to a flexible implementation, in which future changes to the different components of the DSS (SMS, database (DB), solver, etc.) will not seriously affect the rest of them. Thus, the Solver Manager Interface retrieves data from the DB (which, in turn, come from the Scenario Generator (SG) and the Graphical User Interface (GUI)) and prepares the instance for the Solver Manager core script. Therefore, a change to the DB structure will affect only the Solver Manager Interface and not the inner components of the Solver Manager. Similarly, small changes to the SMS, such as symbol modifications, will affect only the Solver Manager Interface and not the DB. Additionally, the Solver Manager Interface will be in charge of controlling processes, instance versions, and other tasks outside the core tasks.

At the heart of the Solver Manager, the optimization environment revolves around two elements:

- the SMS, i.e., the optimization model;
- the Optimization Instance, which contains the data for the problem to be solved.

The SMS has been reshaped in order to provide the solver with a "by-node" notation (vs. the "by-scenario" initially foreseen in D4.2), which is more compact. Some improvements detected since D4.2 delivery have been also included. The complete, new SMS is included in this document as an appendix. The Optimization Instance is composed of the actual elements that populate the sets in the SMS and the parameter values needed for the equations in the model. Rules to build this instance have been designed jointly with the DB, GUI, and SG developers. After preparing the instance, the Solver Manager Interface calls the core script of the Solver Manager. This script prepares the SMS and the instance data for the Optimizer (GAMS for the prototype) and calls the Optimizer to solve the problem. Note that the latter also represents an isolated component that may be changed without affecting the rest of the modules (for example, changing the solver algorithm). Once the Optimizer has finished, the solution (decision variables' values and objective function value) is ready for the Solver Manager Interface to deliver it to the DB and eventually to the decision maker.

1. Introduction

Deliverable D4.3 of the EnRiMa project consists of a prototype of the Stochastic Optimization. The deliverable contains the prototype source code, the binary files, a manual explaining these, and the present document. This document outlines the prototype development, its functionality, and integration with the other modules of the EnRiMa DSS.

The Stochastic Optimization is the workhorse of the DSS. The prototype implements the evolved Symbolic Model Specification from the initial one developed for Deliverable D4.2 (Symbolic Model Specification) [10], after the joint work between the partners involved in the mathematical modeling of the energy systems. Furthermore, the initially envisaged concept of a *solver* has been extended to a so-called “Solver Manager” containing the actual solver as one of its components. This entails more control and flexibility. The integration with the rest of the modules of the EnRiMa DSS has been ensured through the development of an interface that communicates the optimization tasks with the DSS Kernel and, eventually, with the GUI.

The present prototype will be developed further, leading to the final version of the stochastic optimization module (Deliverable D4.6), which is due by the end of the project. The prototype implements the Strategic Module, while the Operational Module will be implemented in the final version. A detailed explanation of both modules can be found in deliverables D2.2 [9] and D4.2 [10]. The main difference between both modules is the type of decisions they deal with. Operational decisions are those involving the dispatch of installed technologies in the short term, whereas strategic decisions concern, in the long term, to which technologies to install and/or decommission, or which energy trading contracts to select. Figure 6 in Appendix C shows the relationship between both modules. We have focused on the strategic part due to the fact that the lower-level operational version (see Deliverable D2.2 [9]) is already running directly in the GUI, and, once the strategic module is working, the implementation of the operational part within the Stochastic Optimization will likely be quite fast.

A straightforward, self-contained description of each part of the Solver Manager is given in the following sections. Thorough details of the inner components are in the appendices. Hence, the remaining document is structured as follows:

- ◇ Section 2 describes the Solver Manager as a whole and how it integrates with the rest of the DSS.
- ◇ Section 3 describes the interface part of the Solver Manager, which communicates with the DSS Kernel.
- ◇ Section 4 describes the optimization environment, including the implemented SMS, the problem instance generation, the call to the optimizer, and the returned solution.
- ◇ Appendix A contains the complete SMS including data structures.
- ◇ Appendix B shows a numerical example for an optimization instance entirely solved by the Solver Manager.
- ◇ Appendix C outlines how the complete operational module is foreseen to be included in the solver manager framework.

2. Solver Manager Description

The Solver Manager is the module within the DSS Engine that provides the solution of the stochastic optimization problems. This module is independent and works in coordination with the rest of the DSS Engine modules, viz., the Scenario Generation tool and the DSS Kernel. As a low-level tool (that is, without direct interaction with the end user), it takes the SMS (model definition) and the instance data (sets and parameter values) provided by other modules (e.g., database or GUI), and, after preparing the data, applies the appropriate algorithms and computations. Then, a solution is returned. This solution is used by other modules, mainly to store and to present the results to the user (eventually, the decision maker).

The components of the Solver Manager are represented in Figure 1. The SM Interface communicates with the DSS Kernel, fetching input data and providing the results once the problem is solved. The SM Interface has been written in the programming language Python [16]. It is responsible for fetching the SMS and the instance data from the data services of the DSS Kernel, including the scenario generator tool output and the GUI information model. Once the solution is ready, it is delivered to the DSS Kernel in order to be available for the GUI. The SM Core Script is started by the SM Interface. It prepares input data, interacts with the optimizer, prepares output data, and manages files. The SM Core Script has been written in the R statistical software and programming language [14].

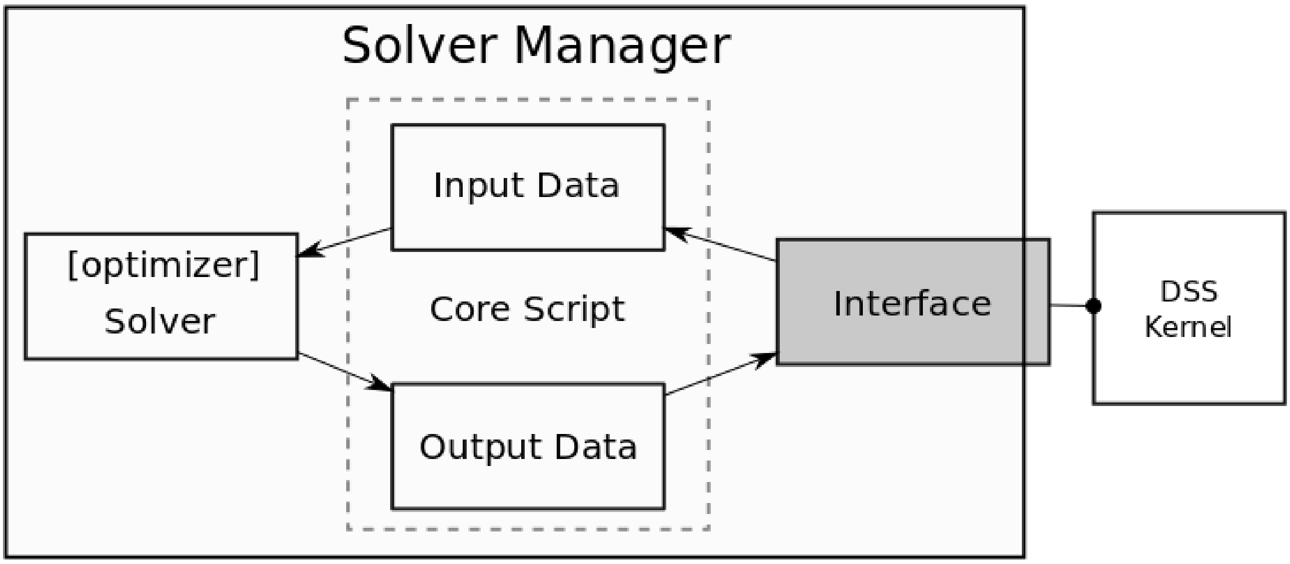


Figure 1: Solver Manager Scheme

We use the term *Optimizer* rather than *Solver*, which was used in previous deliverables, in order to clarify the role of the components. In fact, the Solver Manager is flexible enough to work with different optimization environments and optimizers, thanks to the data-driven SMS that has been developed. Thus, the prototype will perform a call to the GAMS optimization software [6]. The final version might handle further optimizers. The Optimizer, in turn, could use different solvers for the same problem.

The process flow is as follows: the data provided by the Interface are prepared and translated to the format required by the optimizer. Then, a call to the optimizer is performed, requesting the appropriate output. Finally, the output is processed and sent to the DSS Kernel through the SM Interface. The following sections contain a more detailed explanation of each step.

3. Solver Manager Interface Description

In this section, we describe the interface that allows the Solver Manager to interact with other data services in the DSS Kernel. Thus, this interface separates communication tasks and other interaction features (such as global configuration of the solver and data persistence) from the core features of the module described in the previous section. This solution offers clear advantages in terms of software modularity, maintenance, and development.

The interface has been written in Python, and it is currently able to exchange information with the rest of the system through a common MySQL database installed at Stockholm University premises and described in deliverable *D4.4 DSS Kernel Prototype Implementation*. This database also provides a common resource for data persistence purposes. Figure 2 illustrates the main interactions between the SM Interface and other components.

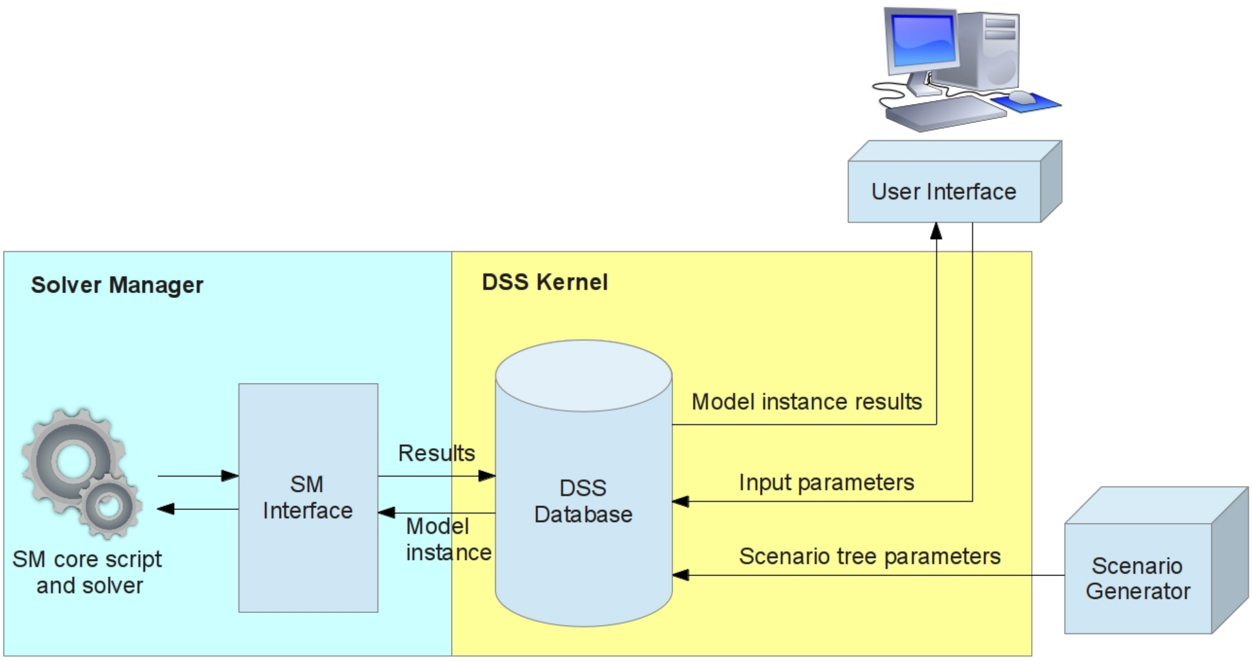


Figure 2: General schema illustrating the main interactions between the SM Interface and other system components.

The SM Interface is responsible for the following tasks within the scope of the Solver Manager:

- Retrieve all necessary input data from other DSS components in order to build a new model instance, according to the requirements of the Solver Manager core component. Mainly, this involves:
  - Information about scenario trees retrieved from the scenario generator.
  - Input parameter values introduced by the system user through the user interface.
- Create a new model instance and set up any other internal configuration details required to start a new analysis in the Solver Manager. Key information and parameters are currently stored in a local XML file. Subsequent versions will also store this information in the common database, so that it becomes accessible to other components, if needed.
- Call the SM Core Script, which executes the optimization of the new model instance.
- Provide the results for this model instance to the GUI, so that they can be displayed to the end user. This information is also stored in the common MySQL database for data persistence.

Figure 3 shows a schema of the overall structure and organization of this component. We remark that, at this stage, this design corresponds to a first prototype of the SM Interface. Therefore, it may be subject to further changes or adjustments, following recommendations from end users or modifications required to better interact with other software components.

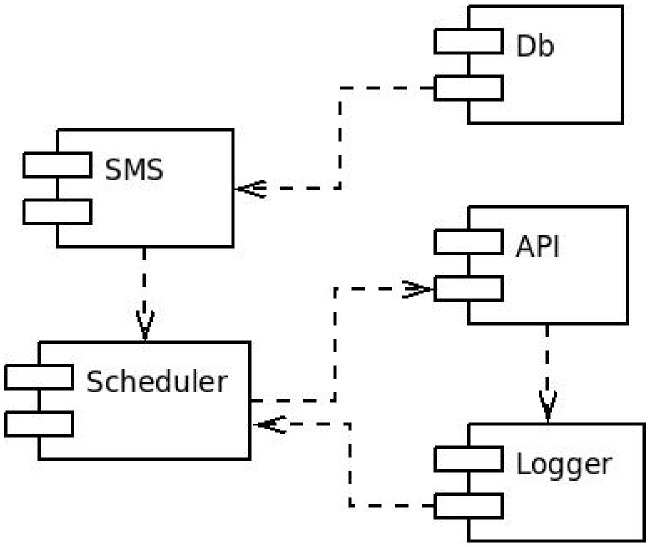


Figure 3: Internal organization of the SM Interface.

The functionalities included in the internal components are:

◇ SMS: Models the objects required to build a new Solver Manager model in the underlying R class.

◇ Scheduler: Manages and monitors the execution of independent models to be solved by the underlying R core application (SM Core Script).

◇ Db: Provides common utility methods for common interactions with the local database.

◇ API (to be developed): It will provide a simple REST interface [15] to interact remotely with the SMS interface, using standard HTTP petitions.

◇ Logger (to be developed): This component will be responsible for creating detailed execution logs for the SM Interface, improving the traceability of its actions to facilitate detection and tracking the source of any bugs or issues, as well as for accountability of all activities performed by the program.

Finally, Figure 4 illustrates the schema of the internal database used for data persistence. The following is a brief description of the tables concerning the Solver Manager:

◇ Tables `SMS_set`, `SMS_par` and `SMS_var` store information about the individual components defining the optimization model to be applied on a set of case instances. Further elements of this model are internally defined by the SM Core Script and do not need to be stored at the database level (such as additional constants, aliases, or equations).

◇ Tables `SMS_pars_sets` and `SMS_vars_sets` specify information about sets referenced in individual parameters and variables in the abstract model.

◇ Table `SMS_model` is the central table to store data about abstract models defined for optimization problems. The associated tables `SMS_model_sets`, `SMS_model_pars` and `SMS_model_vars` store information about the actual sets, parameters, and values included in each model.

◇ Table `SMS_mod_instances` is the pivotal table to store data about every case instance defined by end-users. Each model instance must be associated with a certain abstract model. Likewise, the associated tables `SMS_instance_sets`, `SMS_instance_subsets`, `SMS_instance_mdsets` and `SMS_instance_pars` store the actual values for each set, subset, multidimensional set, and parameter defined in the case instance.

◇ Finally, table `SMS_output_vars` stores the variable values resulting from the solution of the stochastic optimization problem. These result values can then be imported by other modules.

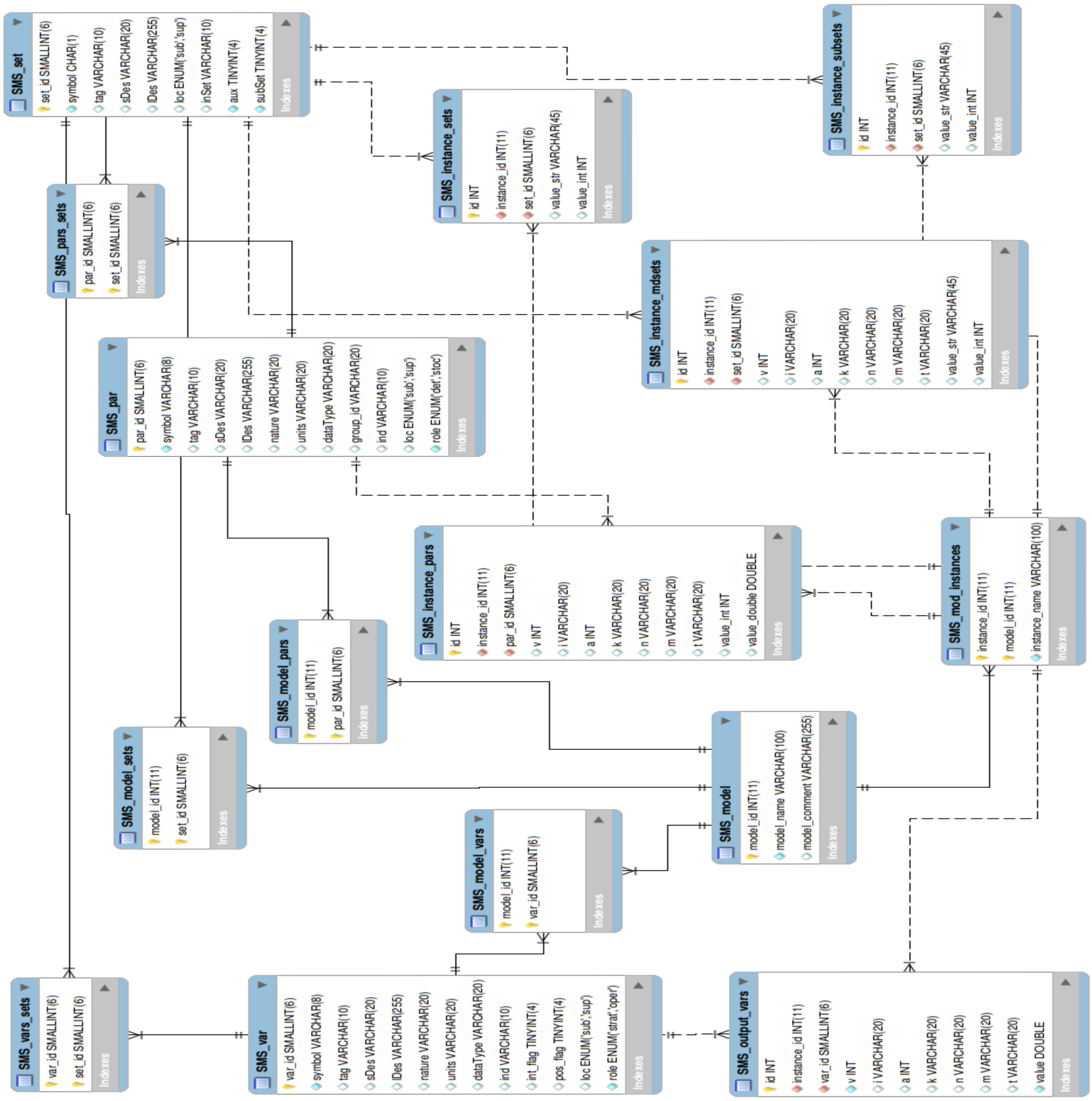


Figure 4: Database schema used by the SM Interface.

4. Optimization Environment

4.1. Symbolic Model Specification

Deliverable D4.2 constituted the initial Symbolic Model Specification (SMS), which defines the mathematical representation of the stochastic model of all relevant energy subsystems and their interactions. Even though most of the initially defined SMS remains, some changes have been made during the recent work in WP4 developing the prototype. Some of these changes are minor ones, such as symbol changes (e.g., former parameter OR) in order to overcome problems with the optimizers.

The rest of the changes represent mostly a simplification of the SMS. The relevant ones are:

- A distinction between variables for generation technologies and energy-absorbing technologies is not necessary. Instead, one only variable (x) is used for technologies, and the different types of technologies are defined through subsets.
- During a coordination meeting, it was decided that a *by-node* notation should be adopted, instead of the *by-scenario* notation foreseen in D4.2, Section 5. Therefore, instead of adding an index for the scenarios, the index for the periods (p) had been substituted by a node index (v), and new data structures for the nodes were introduced (parent node, period, and probability).
- The new *by-node* notation allowed the simplification of the equations that were in charge of computing the building capacity throughout the decision time span[1]. In addition, a new variable ($x_i^{v,a}$) that calculates the number of available devices was added. These changes also affect the way costs are calculated in the objective function equation (23).
- In order to manage risk better, instead of using forward markets in the strategic model, a new binary variable (h) has been added to decide between different contract options (e.g., ToU tariffs). Further parameters and equations were also needed.
- The investment costs and the maintenance costs are now different parameters, which clarifies the notation.
- The risk terms have not been included in the prototype in order to facilitate the implementation and tests of the new formulation of the model. This will be tackled during the next phase of the project.

The SMS is not suitable to be changed by the end user, and therefore it is treated as an inner component of the Solver Manager. Nevertheless, it is replicated in the DB so as to visualize the description of the entities in the model (parameters, sets, and variables) via the GUI, for the decision maker's better understanding of the optimization model.

The optimization instance generated by the prototype uses this new SMS. Nevertheless, it is subject to changes in case it is needed during the tests and further development of the Solver Manager after the delivery. Appendix A contains the whole SMS implemented in the prototype for reference.

4.2. Optimization Instance

An optimization instance is composed of the real data needed to solve the optimization problem. On the one hand, the actual set elements within each set in the SMS are to be determined. On the other hand, the numerical values for each parameter in the SMS must be known before calling the optimizer. The SM Interface is in charge of retrieving the data for the instance from the database, as decribed in section 3. Next, we briefly describe how the instance is built within the Solver Manager. Appendix B contains a complete numerical example of a simulated instance and its solution. That example refers to a typical decision making problem for a building manager with few options. In this case, two types of contracts are available and PV generation technologies are assessed. Real data from the FASAD test site in [8] are used to create all the parameter values. The solution obtained is consistent with FASAD's real costs and this solution includes operational and strategic decisions.

[1] See equations (1) - (4) in Appendix A and in D4.2 [10]))

4.2.1. Sets

The values for the sets to be included in the specific instance can be directly retrieved from the corresponding database tables available in the DSS Kernel through the routines implemented in the *sms* Python module of the SM Interface. These values come directly from the options and from information entered by end users through the GUI. Note that each set in the SMS is mapped to the corresponding database table in Appendix A.

Collected values for the three possible types of sets that can define the instance (sets, subsets, or multidimensional sets) are stored in database tables *SMS_instance_sets*, *SMS_instance_subsets*, *SMS_instance_mdsets*. In turn, these values are loaded in the actual problem instance, which is created when calling the SM Core Script using a generic data loading routine. Finally, the values defined for every instance are also internally stored in XML files by the SM Interface for backup purposes.

4.2.2. Deterministic Parameters

Deterministic parameters can be either fully specified directly by end users selecting a pre-defined series of values or determined by sets of previous states of these variables in the model. Therefore, these values are not affected by randomness or uncertainty. As a result, this information is also directly retrieved by the SM Interface from the DSS Kernel database, in a similar fashion as for sets. Note that each parameter in the SMS is mapped to the corresponding database table in Appendix A.

The information collected for these parameters is stored in the *SMS_instance_pars*, corresponding to entries with the *role* field equal to *"det"*. As in the case of sets, explained above, these series of values are also stored internally in the same XML file representing a single instance. The generic data-loading routine in the SM Core Script also loads this information before calling the Optimizer.

4.2.3. Scenario Tree (Stochastic) Parameters

Energy systems' performance is affected by a number of characteristics coming from their components, e.g., the building, technologies, demand, where not all information is perfectly known. The so-called random variables, or stochastic processes, that model such uncertain parameters, can be treated from different perspectives. The simplest one is replacing the random parameters by their expected value estimates. Within the EnRiMa project, uncertainty is tackled through stochastic optimization, which has been proven to be a more effective approach. This approach takes into account variability in addition to central values (see, for example, [4]). That variability is implemented through the use of scenario trees. A scenario tree is a discretized representation of the stochastic process of the system. See Appendix A for an explanation of the tree structure and its representation within the SMS. Kaut et al. [12] describe the scenario tree structure used for the EnRiMa DSS in more detail.

In order to retrieve the associated values for these parameters in a specific instance, the model relies on information stored in two different places of the Kernel database:

- ◇ Tables *StUncertType* and *StUncertValue* link individual parameters with the definition of the type of uncertainty affecting them, in those cases in which this variation only affects strategic nodes in the tree (i.e., not involving time profiles). In this case, uncertainty is specified as a typical probability distribution defined by its mean and standard deviation. The actual values for the parameters in each analysis are then obtained from a random sample of this reference distribution.

- ◇ Tables corresponding to the Scenario Generator module (*SG_**) include information to define the uncertainty for parameters in embedded scenario trees, comprising operational nodes (that is, involving time or seasonal profiles). It is worthwhile to remark that several parameters can share the same "uncertainty profiles" developed by the Scenario Generator. In other words, the same variability model can be applied to more than one parameter, thereby avoiding the need to repeat the same definition for different parameters.

Even though, in theory, there are a number of stochastic parameters that can be considered in each model, in practice it is expected that only a few of them will vary throughout the scenarios (the more volatile ones). The retrieved values are stored in table *SMS_instance_pars* and correspond to entries with the *role* field equal to *"stoc"*. As above, these values are stored by the SM Interface along with the rest of the data for every instance in a local XML file for backup purposes.

4.3. Optimizer Call

Once the interface has retrieved all the input data from the DB and performed all the computations to assign values to all the parameters, the data are saved in a file with the instance object, which can be treated by the SM Core Script. The SM Interface runs the script that eventually calls the solver. This script reads the SMS object and the instance object in order to generate the files in the appropriate format for the optimizer. For the prototype, a GAMS file is created with both the model definition and the values. If the files can be generated correctly, then a call to the optimizer is made. Otherwise, an error is returned to the SM Interface (see the manual accompanying the Solver Manager files for details).

4.3.1. Optimization Software

For the prototype, the GAMS software is used as optimizer. It is a well-known integrated system to model and solve optimization problems. As the SMS and the Solver Manager are flexible, data-driven structures, further options will be tested for the final version. The call to the optimizer is made from the script mentioned above using the `gdxrrw` library [2]. This script also manages the solver output. This output consists of two files: the GAMS listing file and the solution file (if the optimizer finds a solution). Moreover, the optimizer returns status codes for the model and the solver, which are used to log and eventually inform the interface.

4.3.2. Solvers

The stochastic strategic model is optimized using the OSICPLEX solver [3], which is a wrapper for the renowned CPLEX solver [11]. Further solvers and algorithms will be tested and benchmarked after the prototype delivery, while building real and more complex instances using data from the test sites.

4.4. Solution Delivery (Decision Variables)

If the optimizer finds a solution, then the SM Main Script imports that solution, consisting of the optimal values for the decision variables and the value of the objective funcion, into the instance object (which already had the sets and parameter values). This object is saved into a file for the SM Interface to deliver the solution to the DB. The SM Interface stores the solution in the *SMS_output_vars* database table to make them accessible for the GUI as explained in Section 3. Logging and messaging are performed throughout all the process. Thus, if the optimizer eventually does not find a solution error messages are passed to the GUI.

5. Conclusions

Along with the rest of the DSS components' prototypes, the Stochastic Optimization Prototype has been successfully finalized and is ready for the forthcoming activities of the project. The prototype is capable of solving instances using data provided by the other other modules as well as to return the decisions in an integrated manner. The Solver Manager's modular approach is adequate to face further development in the next period, when it will be tested on realistic cases with large amounts of data and increased complexity.

In this document, the Solver Manager has been described in detail in a sequential way. After putting it into context in the Introduction, an overview of the Solver Manager as a whole as well as the relationship with other modules of the DSS were provided. Then, the SM Interface that provides the necessary interoperability with the rest of the DSS components was described. Finally, the specific parts of the Optimization environment, i.e., the SMS, the Optimization Instance and the Optimizer were presented. This environment is handled by the SM Core Script, which interacts with the SM Interface.

The next steps will include the thorough testing of more complex and larger instances, the benchmarking of different algorithms and, eventually, the application of innovative ones. Another foreseen task is to implement the operational model within the Solver Manager as well as testing further optimizers. Both activities will be done in the framework of the remaining work in Task 4.5 (Stochastic optimization and solvers), which will lead to Deliverable D4.6 (Stochastic Optimization). Consequently, the components of the Solver Manager, including the SMS and the SM Interface, will be improved continuously during the development of the final DSS implementation. This corresponds to the remaining work in Task 4.6 (Implementation of the DSS Kernel), which will lead to Deliverable D4.7 (DSS Kernel implementation).

Finally, the upcoming development iteration will also focus on the integration and testing of the Stochastic Optimization Prototype in the consolidated server premises based at Stockholm University. This will involve a series of tests against different case studies in order to refine the whole process of the configuration and optimization of individual case instances. This corresponds to the remaining work in Task 4.7 (Testing). Nevertheless, the current Stochastic Optimization Prototype is already prepared for such an integration since the common data structures have been developed in agreement with the rest of partners involved in the development and deployment of the other modules included in the DSS.

Acknowledgments

The lead partner for this deliverable has been Universidad Rey Juan Carlos (URJC). The internal review has been done by SINTEF and the external review by UCL. Thanks to the EC reviewers and the Project Officer for their suggestions during the first review meeting regarding a better structure for the deliverables, which we have followed. We acknowledge the contributions of all other EnRima partners to the project: Stockholm University (SU), University College London (UCL), International Institute for Advanced Systems Analysis (IIASA), Center for Energy and Innovative Technologies (CET), SINTEF, Minerva Consulting and Communication (MC&C), HC Energía (HCE), and Tecnalia Research and Innovation (TECNALIA). Special thanks go to Afzal Siddiqui and Paula Rocha (UCL), Michal Kaut and Adrian Werner (SINTEF), and Markus Groissböck and Michael Stadler (CET) for their contribution to improve the SMS. Productive on-line discussions with Martin Henkel (SU) and Michal Kaut (SINTEF) led to the smooth integration of the Solver Manager Interface with the rest of the modules. Angel Luis Alvarez (HCE) provided help with real data from the FASAD test site to build the examples.

References

[1] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. *lpsolve : Open source (Mixed-Integer) Linear Programming system.*

[2] Original coding by Rishabh Jain. Adopted, packaged, and extended by Steve Dirkse. *gdxrrw: An interface between GAMS and R*, 2012. R package version 0.2.0.

[3] COIN-OR Foundation. COIN-OR open solver interface. Software. URL `https://projects.coin-or.org/Osi/`.

[4] A.J. Conejo, M. Carrión, and J.M. Morales. *Decision Making Under Uncertainty in Electricity Markets*. International Series in Operations Research and Management Science Series. Springer, 2010. ISBN 9781441974204. URL `http://books.google.es/books?id=zta0qWS_W98C`.

[5] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: a modeling language for mathematical programming*. Thomson/Brooks/Cole, 2003.

[6] GAMS Development Corporation. The general algebraic modeling system (GAMS). Software, 2013. URL `http://www.gams.com/`.

[7] W.E. Hart, C. Laird, and J.P. Watson. *Pyomo - Optimization Modeling in Python*. Springer Optimization and Its Applications. Springer, 2012.

[8] HCE, IIASA, SU, UCL, URJC, SINTEF, CET, and Tecnalia. Deliverable 1.1: Requirement assessment. EnRiMa Project Deliverable, 12 2011.

[9] HCE, IIASA, SU, UCL, URJC, SINTEF, CET, and Tecnalia. Deliverable 2.2: Mathematical formulation of energy balance and flow constraints. EnRima Project Deliverable, 03 2012.

[10] HCE, IIASA, SU, UCL, URJC, SINTEF, CET, and Tecnalia. Deliverable 4.2: Symbolic model specification. EnRiMa Project Deliverable, 03 2012.

[11] IBM. Cplex. Software. URL `http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/`.

[12] Michal Kaut, Kjetil T. Midthun, Adrian S. Werner, Asgeir Tomasgard, Lars Hellemo, and Marte Fodstad. Dual-level scenario trees – scenario generation and applications in energy planning. Optimization Online, 2012. URL `http://www.optimization-online.org/DB_HTML/2012/08/3551.html`. report 2012/08/3551.

[13] Mathworks. Matlab. Software. URL `www.mathworks.com/products/matlab/`.

[14] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. URL `http://www.R-project.org/`. ISBN 3-900051-07-0.

[15] L. Richardson and S. Ruby. *RESTful Web Services*. Oreilly Series. O'Reilly Media, 2008. ISBN 9780596529260.

[16] Guido van Rossum. The python language reference. Manual, 2013. URL `http://docs.python.org/py3k/reference/index.html`.

A. Symbolic Model Specification

This appendix contains the SMS implemented in the prototype. See subsection 4.1 for changes and justification with respect to the SMS presented in Deliverable D4.2 [10]. This appendix contains:

1. The mathematical representation of the entities in the model:
   - ◇ Sets, including standard sets, aliases, subsets, and conditional sets;
   - ◇ Constants (scalars, in GAMS language);
   - ◇ Decision variables (DV);
   - ◇ Parameters;
   - ◇ Equations, classified into operational and strategic constraints, and objective function.
2. The data representation of the SMS within the Solver Manager.

For all the sets and parameters whose values will form an instance, specific comments on how data are retrieved from the DB are included.

A.1. Scenario Tree Representation

This subsection contains an explanation of the symbols used for the scenario tree as well as the relationship between the tree nodes and other entities in the model. Scenario trees are widely used in stochastic programming to discretize the huge, usually infinite, number of possible outcomes of the random variables in a stochastic model. Thus, a scenario tree gathers the most probable scenarios resulting from a combination of all random variables (stochastic parameters using the SMS language). Several size-reduction techniques can be used in order to make the problems computationally tractable. See [4] for a thorough explanation of scenario trees applied to energy markets. Within the EnRiMa project, a new dual-level scenario tree approach has been used [12]. As a result, typical scenario trees are implicitly represented within the whole SMS. For the sake of clarity, we briefly explain here the scenario trees used in the models.

A scenario tree can be represented graphically as an acyclic graph consisting of nodes and arcs, where the root node has no parent (predecessor node), each node may have one or more children, and each node can only have one parent. The number of terminal nodes (leaves), which do not have children, determines the number of scenarios considered. Each scenario is a path from the root node to a leaf node. The nodes represent states of the system at a particular time, where decisions are made. The root node corresponds to the beginning of the planning horizon. Arcs represent the precedence relationship between nodes with an associated probability of occurrence. Therefore, in addition to the node identifier (v in the SMS), the following information is required:

- ◇ The parent node of each node. It is represented by the conditional set $\mathcal{V}^v_{Pa}$.
- ◇ The probability of each node. It is represented by the parameter PR^v.
- ◇ The time period of each node. It is represented by the parameter PT^v.

In addition, the conditional set $\mathcal{A}^{i,v}_{Ages}$ is defined as the possible technologies' ages at a given node for a given technology. For example, if period of node 3 is equal to 3, and at the beginning of the planning horizon a unit of CHP technology whose age was 5 years was already in the building, then the possible ages for that technology at node 3 are {0,1,2,3,8}.

Figure 5 shows a simple scenario tree with all the symbols and expressions used in the SMS. Circles represent nodes with the node index (v) displayed inside. Nodes in the same column correspond to the same time period (PT^v). Each node is linked to its parent through the conditional set $\mathcal{V}^v_{Pa}$ and has a probability (PR^v) associated to its parent's branching. The represented tree corresponds to a three-stage stochastic problem, where new information arrives at periods 1 and 4.

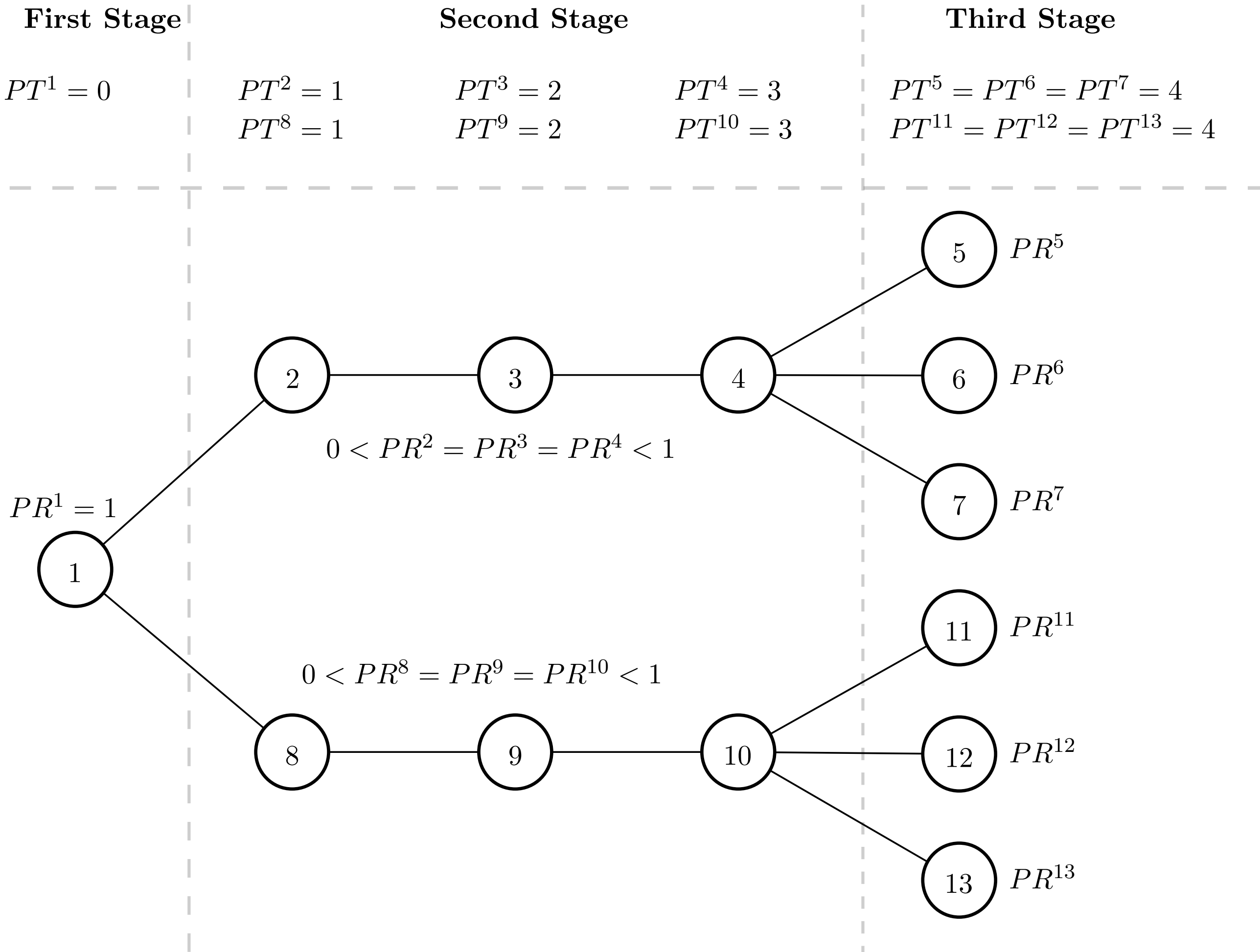


Figure 5: Scenario tree symbolic representation.

A.2. Sets

The information presented for each set is:

- ⋄ Symbol. Small latin letter for sets and aliases, see below for subsets and conditional sets.
- ⋄ Short description.
- ⋄ Domain, representing the set by calligraphic capital letters.
- ⋄ Tag, which is a label used for interfacing purposes, in square brackets.
- ⋄ Long description.
- ⋄ Data comment, for interfacing purposes.

a Technology age, $a \in \mathcal{A}$ `[age]`. This set is used to model the effect of aging on the capacity and the costs of the technologies.

Data. The content of this set is determined by the number of long term periods P. Thus, $a = 0, ..., P-1$. This can be obtained from the DB table SG_Output_ScenarioNodes, which contains a column PeriodId with the actual periods. The set must be extended in order to take into account initially installed technologies. Then $a = 0, ..., P-1+$ age-of-older-tech.

i Energy technology, $i \in \mathcal{I}$ `[enerTech]`. Equipment available in the building, or suitable to be installed. This equipment can be: (1) Energy generator, (2) Energy storage, or (3) Energy saver. Each element of the set is a specific model of a type of technology (e.g., CHP), with different features.

Data. Examples: 'PV panels model xxx', 'CHP Dachs 5.5', 'Wind turbine xxx', 'battery xxx', etc. Data input: Technologies are stored in the DB, and the user should select those to be included in the instance. DB table: StTechnology.

k Energy type, $k \in \mathcal{K}$ `[enerType]`. Type of energy that will be used in the building.

Data. Examples: Electricity, heat, natural gas, solar, etc. Data input: The energy types used in the building are stored in the DB and are retrieved by the solver manager. DB table: StEnergyType.

l Type of pollutant, $l \in \mathcal{L}$ `[pollutant]`. Energy generation and consumption generate emissions to the environment. The amount of a building's emissions of each pollutant depends on the emission ratios. The total emissions can be constrained by policy makers. Their minimization can also be an objective for certain decision makers.

Data. Examples: CO_2, NO_x, etc. Data input: Directly from the DB. The user might need to select which pollutants should be considered. DB table: StPollutant.

m Operational profile, $m \in \mathcal{M}$ `[profile]`. This set gathers the representative profiles considered in the model to link the short- and long-term performance of the energy systems in the building: this short-term performance is scaled to the long term through a weight factor given as a parameter value.

Data. Examples: 'average winter', 'average spring', 'extreme winter', 'extreme summer', etc. DB table: SG_Input_EmbProfiles.

n Energy tariff, $n \in \mathcal{N}$ `[tariff]`. This set contains the tariffs available throughout the decision time span. It is possible that not all the tariffs are available at each scenario tree node.

Data. Examples: 'Regulated electricity tariff 1', 'Regulated NG tariff 1', etc. Data input: From the DB tables. For each energy type, it must be specified which tariffs are available for purchasing and for selling energy, which tariffs are available for energy-creating technologies input, and which tariffs are available at each tree node. DB table: StTariff.

t Short-term period, $t \in \mathcal{T}$ `[time]`. These are the periods when operational decisions are made. Such decisions concern how much energy of each type must flow through the building energy systems, from markets to demand.

Data. Examples: 'hour1', ..., 'hour24'. Data input: From the operational profiles in the DB. The operational periods are supposed to be of the same length (e.g., one hour, 15 minutes, ...) and a parameter contains this information. The parameter is stored in the table SG_Input_EmbProfiles, so this set can be built as a sequence between 1 and 24/DT. For different short-term periods within profiles, a conditional set may be required.

v Tree node, $v \in \mathcal{V}$ `[node]`. This set contains the nodes in the scenario tree. For each node, its time period (cf., p index in the deterministic model), probability, and parent node must be specified

Data. Examples: 'node1', 'node2', etc. DB table: SG_Output_ScenarioNodes.

A.3. Aliases

k' Output energy type, $k' \in \mathcal{K}$ `[outEner]`. This index is used in order to distinguish the type of the energy input and output when using generators.

Data. An alias does not need data in the DB.

v' Parent nodes, $v' \in \mathcal{V}$ `[parentNode]`. This index is used to map the parent node of each node.

Data. An alias does not need data in the DB.

A.4. Subsets

Subsets are represented through the symbol of the set in which they are contained in calligraphic font and the symbol of the subset as subscript in *italics*.

$\mathcal{A}_{New}$ Age = 0, $\mathcal{A}_{New} \subset \mathcal{A}$ `[ageNew]`. This set contains only the element 0 from the age set.

Data. No data needed in the DB.

$\mathcal{A}_{Old}$ Age != 0, $\mathcal{A}_{Old} \subset \mathcal{A}$ `[ageOld]`. This set contains all the elements from the age set except 0.

Data. No data needed in the DB.

$\mathcal{I}_{Cn}$ Continuously-sized technologies, $\mathcal{I}_{Cn} \subset \mathcal{I}$ `[contTech]`. Technologies are continuously sized if they do not have a nominal capacity and the investment can be done by power units.

Data. Those technologies whose capacity is not equal to 1 (column Capacity in table StTechnology).

$\mathcal{I}_{Ds}$ Discretely-sized technologies, $\mathcal{I}_{Ds} \subset \mathcal{I}$ `[discTech]`. Technologies are discretely sized if they have a nominal capacity and the investment has to be done by devices.

Data. Technologies that are discretely sized can be deduced by the column Capacity in table StTechnology. There, 1 should be the default value.

$\mathcal{I}_{Gen}$ Energy-generation technologies, $\mathcal{I}_{Gen} \subset \mathcal{I}$ `[genTech]`. Technologies that receive energy as input and return other type(s) of energy as output.

Data. Table StTechnology, Type column = 'Production'.

$\mathcal{I}_{PU}$ Passive technologies (unitary), $\mathcal{I}_{PU} \subset \mathcal{I}$ `[passiveTech]`. Passive technologies which have a multiplicative effect on the demand, that is, the higher the demand, the higher the savings. They entail savings over the use of the energy regardless of the building dimensions.

Data. Table StTechnology, Type column = 'Passive'.

$\mathcal{I}_{Sto}$ Storage technologies, $\mathcal{I}_{Sto} \subset \mathcal{I}$ `[stoTech]`. Devices that store a type of energy from the market or the generation technologies and can release this energy to meet the demand. These technologies are subject to losses both at the input and at the output. The storage is also uncharged at a given ratio even if energy is not released to be consumed.

Data. Table StTechnology, Type column = 'Storage'.

$\mathcal{K}_{Cool}$ Type of energy for cooling, $\mathcal{K}_{Cool} \subset \mathcal{K}$ `[coolEner]`.

Data. This is one of the energy types in DB table StEnergyType.

$\mathcal{K}_{Dem}$ Types of energy on the demand side, $\mathcal{K}_{Dem} \subset \mathcal{K}$ `[demEner]`. This is the union of the subsets Elec, Heat and Cool.

Data. Subset automatically filled by the solver manager.

$\mathcal{K}_{Elec}$ Type of energy for electricity, $\mathcal{K}_{Elec} \subset \mathcal{K}$ `[elecEner]`.

Data. This is one of the energy types in DB table StEnergyType.

$\mathcal{K}_{EP}$ Energy to purchase, $\mathcal{K}_{EP} \subset \mathcal{K}$ `[enerPurchase]`. Types of energy which can be purchased.

Data. DB table StTariff, column Direction

$\mathcal{K}_{ES}$ Energy to be sold, $\mathcal{K}_{ES} \subset \mathcal{K}$ `[enerSale]`. Types of energy which can be sold.

Data. DB table StTariff, column Direction

$\mathcal{K}_{Heat}$ Type of energy for heat, $\mathcal{K}_{Heat} \subset \mathcal{K}$ `[heatEner]`.

Data. This is one of the energy types in DB table StEnergyType.

$\mathcal{K}_{Ren}$ Renewable energy, $\mathcal{K}_{Ren} \subset \mathcal{K}$ [enerRen]. Renewable energy does not need energy balance constraints.

Data. DB table EnergyType (to be implemented).

$\mathcal{N}_{TP}$ Purchasing tariffs, $\mathcal{N}_{TP} \subset \mathcal{N}$ [purchTariffs]. This subset contains the tariffs available to buy energy.

Data. DB table StTariff, column Direction.

$\mathcal{N}_{TS}$ Sales tariffs, $\mathcal{N}_{TS} \subset \mathcal{N}$ [salesTariffs]. This subset contains the tariffs available to sell energy.

Data. DB table StTariff, column Direction.

$\mathcal{V}_{Fut}$ Future nodes, $\mathcal{V}_{Fut} \subset \mathcal{V}$ [futureNodes]. All the nodes that are not the root node.

Data. Calculated by the Solver Manager.

$\mathcal{V}_{Root}$ Root node, $\mathcal{V}_{Root} \subset \mathcal{V}$ [rootNode]. This subset only contains the root node and is needed to identify states at time 0, for example, existing technologies.

Data. No data needed in the DB.

A.5. Conditional Sets

Conditional (or multidimensional) sets are represented as the symbol of the main set in calligraphic font, the symbol of the conditional set as subscript in *italics* font, and the index of the 'input' set(s) as superscript.

$\mathcal{A}^{i,v}_{Ages}$ Possible ages of a technology at a node, $i \in \mathcal{I}, v \in \mathcal{V}$ [agesNode]. This conditional set provides all the possible ages that technologies may have at a given node.

Data. The set is filled by the Solver Manager using other sets.

$\mathcal{K}^{i}_{In}$ Input energy types for a technology, $i \in \mathcal{I}_{Gen}$ [inputEner]. Generation technologies can utilize different types of energy to generate the output.

Data. DB table StTechProduction, column InEnergyTypeId. Note that even though in the technologies analyzed so far we only have one input energy type, in theory the SMS allows more than one.

$\mathcal{K}^{i}_{Out}$ Output energy types for a technology, $i \in \mathcal{I}_{Gen}$ [outEnergy]. Generation technologies provide one or more output energy types.

Data. DB table StTechEnergy, columns OutEnergyType1, OutEnergyType2. Note that even though in the technologies analyzed so far we have one or two output energy types, in theory the SMS allows more than two.

$\mathcal{K}^{i}_{Po}$ Principal energy of technologies, $i \in \mathcal{I}$ [principalEner]. Each generation technology has a principal output type of energy (when more than one). For storage technologies, the input and output types of energy are the same. For passive measures, it is the type of energy which is saved.

Data. DB table StTechStorage and table StTechPassive column IdEnergyTypeId, and DB table StTechProduction column OutEnergyType1Id.

$\mathcal{N}^{k}_{Pur}$ Purchase tariffs for each energy type, $k \in \mathcal{K}$ [enerBtariff]. Conditional set to make purchase tariffs for energy types available.

Data. The DB should specify which tariffs are available to buy each type of energy. DB table StTariff, columns Direction and EnergyTypeId.

$\mathcal{N}_S^k$ Sales tariffs for each energy type, $k \in \mathcal{K}$ [enerStariff]. Conditional set to make sales tariffs for energy types available.

Data. The DB should specify which tariffs are available to sell each type of energy. DB table StTariff, columns Direction and EnergyTypeId.

$\mathcal{N}_{Tr}^k$ Energy that can be traded in each market, $k \in \mathcal{K}$ [enerTrade]. This conditional set is the union of $\mathcal{N}_{Pur}^k$ and $\mathcal{N}_S^k$.

Data. The DB should specify which tariffs are available to purchase or sell each type of energy. DB table StTariff, column EnergyTypeId.

$\mathcal{T}_{Tm}^m$ Short-term periods by profile, $m \in \mathcal{M}$ [perProfile]. Each profile can contain several operational periods, whose duration is modeled through the DM parameter.

Data. In DB table SG_Output_EmbProfNodes, column EmbProfileId states profile, and EmbProfileTime column the short-term period.

$\mathcal{V}_{Pa}^v$ Parent for each node, $v \in \mathcal{V}$ [parentNodes]. This conditional set contains the relationship between each node and its parent. Note that the parent node is represented as $Pa(v)$ when it is used as an index in an expression.

Data. DB table SG_Output_ScenarioNodes.

A.6. Constants

DR Discount rate, per year [dRate].

A.7. Parameters

For each parameter (applies also to variables), the following information is shown:

- ◇ Expression, formed by the parameter/variable symbol and the sets' indices that apply;
- ◇ Short description;
- ◇ Measurement units;
- ◇ Nature (deterministic/stochastic);
- ◇ Tag and datatype, in square brackets and separated by ":", for interfacing purposes;
- ◇ Domain for the indices of the expression;
- ◇ Long description;
- ◇ Data comment, for interfacing purposes.

$AF_i^{v,m,t}$ Availability factor for a technology (kWh/kWh, stochastic, [genAvail:decimal]). $i \in \mathcal{I}$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. The capacity of a technology may change throughout the optimization horizon. For example, photovoltaic panels do not have the same performance during the day and they even do not work during the night. The factor can also be used to model the availability of future technologies.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechnology, column Availability.

AG_i^a Technology aging factor (kW/kWh, deterministic, [techAging:decimal]). $i \in \mathcal{I}$, $a \in \mathcal{A}$. This parameter adjusts the total capacity of a technology throughout its lifetime. The superindex is for the age. That is, at age 0, a given technology (e.g., CHP Dachs 5.5) has factor 1, which reduces at some rate each year.
Data. The number of entries must coincide with the number of strategic periods plus the age of technologies existing at start. DB table StTechMaintPerYear, column AgingFactor.

$B_{k,n}$ Primary energy needed to produce final-use energy (kWh/kWh, stochastic, `[marketPrimEff:decimal]`). $k \in \mathcal{K}$, $n \in \mathcal{N}^k_{Pur}$. Units of primary energy required to produce one unit of a type of energy available from a market where processed energy can be bought.
Data. This is a characteristic of the contracts. DB table StTariff, column PrimaryEnergyNeeded.

$CD^{v,a}_i$ Technology decommissioning cost (EUR/kW, stochastic, `[techDecomCost:decimal]`). $i \in \mathcal{I}$, $v \in \mathcal{V}$, $a \in \mathcal{A}$. Decommissioning a technology may lead to a removal cost or revenue from selling the equipment (in such a case, the value of the parameter is negative). It can be related to the installation cost.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechMaintPerYear, column Decommissioning cost.

CI^v_i Technology installation cost (EUR/kW, stochastic, `[techInstCost:decimal]`). $i \in \mathcal{I}$, $v \in \mathcal{V}$.
Data. Computed by the SM Interface using current investment cost and Scenario Generator output (SG_Output_* tables) for future cost development. Base values in DB table StTechnology, column Installcost.

$CM^{v,a}_i$ Technology maintenance cost (EUR/kW, EUR/kWh, stochastic, `[techMaintCost:decimal]`). $i \in \mathcal{I}$, $v \in \mathcal{V}$, $a \in \mathcal{A}$. This is a fixed cost per capacity installed. It may be linked to the installation cost.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechMaintPerYear, column MaintenanceCost.

$CO^v_{i,k}$ Technology operation cost (EUR/kWh, stochastic, `[genOperCost:decimal]`). $i \in \mathcal{I}_{Gen}$, $k \in \mathcal{K}^i_{Out}$, $v \in \mathcal{V}$. This parameter is used when the supplier/maintainer's tariff is quoted per operated 'kWh'.
Data. This is a characteristic of the technology for each type of energy. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechMaint, column OperatingCost.

$D^{v,m,t}_k$ Energy demand (kWh, stochastic, `[enerDemand:decimal]`). $k \in \mathcal{K}_{Dem}$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}^m_{Tm}$. Total energy load of the building for a type of energy, during each short-term (operational) period.
Data. The values for this parameter are retrieved from the scenario generator output. However, a sort of transformation is needed, as the scenario generator output returns values for the building occupancy, not for the demand. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StDemand, column Demand.

DM^m Weight (scaling factor) for the operational profile in the objective (days, deterministic, `[profileWeight:integer]`). $m \in \mathcal{M}$. This parameter is used to scale the operational system performance (energy, cost) to the strategic time resolution.
Data. Representative operational periods may have different durations. For example, for 'extreme summer day' it may be 10 days, whilst for 'average summer day' it may be 70 days. DB table SG_Input_EmbProfiles, column Weight.

DT^m Duration of the short-term period within a given profile. (hours, deterministic, `[durOper:decimal]`). $m \in \mathcal{M}$. The sum over the durations of all the operational periods must correspond to a whole day. This parameter is used to convert energy to power or vice versa.
Data. Example: with 24 operational periods, each period is one hour long, and the value of the parameter is equal to one. With 15-minutes-long operational periods, this parameter is equal to 0.25. DB table SG_Input_EmbProfiles, column DeltaT.

$EC^v_{i,k,k'}$ Output energy generated from one unit of input energy (kWh/kWh, stochastic, `[genConvFact:decimal]`). $i \in \mathcal{I}_{Gen}$, $k \in \mathcal{K}^i_{In}$, $k' \in \mathcal{K}^i_{Out}$, $v \in \mathcal{V}$. This is a conversion factor. It is applied to the input energy of a technology to compute the output energy of this technology. Both types of energy can be the same or different. We may also have several types of output

and input energy (e.g., gas, biogas)
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechEnergy, columns OutEnergyType2Eff, OutEnergyType1Eff.

EF^v Required building energy efficiency (unitless, stochastic, `[buildingEff:decimal]`). $v \in \mathcal{V}$.
Data. This is a characteristic of the building. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB StBuildReq, column RequiredBuildEff.

G_i Technology capacity (kW/Device, deterministic, `[techCapacity:decimal]`). $i \in \mathcal{I}$. Nominal capacity of each device of a given technology. For continuous technologies, its value is 1.
Data. DB table StTechnology, column Capacity.

IL^v Investment limit (EUR, stochastic, `[invLimit:integer]`). $v \in \mathcal{V}$. This is needed when the building has a budget limit for investing in technologies.
Data. This is a characteristic of the building for each year. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StBuildReq column invLimit.

$LC^v_{k,l,n}$ Pollution emissions by energy purchases (kg/kWh, stochastic, `[marketEmission:decimal]`). $k \in \mathcal{K}$, $l \in \mathcal{L}$, $n \in \mathcal{N}^k_{Pur}$, $v \in \mathcal{V}$. Mean rate of emission of a pollutant from processed energy purchased in the market.
Data. This is a characteristic of the type of energy for each market and pollutant. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StPollutionTariff.

$LH^v_{i,k,l}$ Pollution emissions by generating technologies (kg/kWh, stochastic, `[genEmission:decimal]`). $i \in \mathcal{I}_{Gen}$, $k \in \mathcal{K}^i_{In}$, $l \in \mathcal{L}$, $v \in \mathcal{V}$. Amount of pollutant that is emitted by a generation technology during its operation, for each type of input energy.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StPollutionTech.

LP^v_i Physical Limit (Devices/KW/kWh, stochastic, `[techPhysLim:decimal/integer]`). $i \in \mathcal{I}$, $v \in \mathcal{V}$. Number of units or capacity of a technology that can be installed at the site at a time.
Data. This parameter is a function of the building characteristics and the technology characteristics and is stored in the technologies table. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechnology, column PhysLimit)

$ME_{k,n}$ Maximum purchase/sale of a type of energy under a given contract (kW, deterministic, `[maxTrading:decimal]`). $k \in \mathcal{K}$, $n \in \mathcal{N}^k_{Tr}$.
Data. This parameter is a feature of the contracts. DB table StTariff, column MaximumTransaction.

$OA^v_{i,k}$ Fraction of storage lower limit (kWh/kWh, stochastic, `[stoLowerLim:decimal(0..1)]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. Minimum fraction of the capacity that must be charged in an energy-storage technology.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechStorage, column FractLowerLimit.

$OB^v_{i,k}$ Fraction of storage upper limit (kWh/kWh, stochastic, `[stoUpperLim:decimal(0..1)]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. Maximum fraction of the capacity that must be charged in an energy-storage technology.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechStorage, column FractUpperLimit.

$OD^v_{i,k}$ Energy demand reduction for a passive technology (kWh/kWh, stochastic, `[pasDemandRed:decimal]`). $i \in \mathcal{I}_{PU}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. For each unit of a passive technology, the total demand is reduced by some value.
Data. This is a characteristic of the technology for each type of energy. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechPassive, column Reduction.

$OI^v_{i,k}$ Charging ratio to storage (kWh/kWh, stochastic, `[stoChargRatio:decimal]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. Units of energy available for each unit sent to energy-storage technology.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechStorage, column ChargeRatioTo.

$OO^v_{i,k}$ Discharging ratio from storage (kWh/kWh, stochastic, `[stoDisChargRatio:decimal]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. Units of energy needed to be discharged from storage in order to deliver one unit of energy to the demand.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechStorage, column DischargRatio.

$OS_{i,k}$ Energy storage availability (kWh/kWh, deterministic, `[stoAvail:decimal(0..1)]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$. This parameter models the energy loss of a storage technology over the time. It represents the units of energy available for each unit of energy stored after each operational period.
Data. This parameter is a feature of the technology, and it can be stored in the technology table, assuming that a specific storing technology only stores one type of energy. DB table StTechStorage, column StorageAvail.

$OX^v_{i,k}$ Maximum discharge rate (kW/kWh, stochastic, `[stoDisRate:decimal]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. Maximum energy discharge rate per unit of storage capacity.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechStorage, column DischargingAvail.

$OY^v_{i,k}$ Maximum charge rate (kW/kWh, stochastic, `[stoChargeRate:decimal]`). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}^i_{Po}$, $v \in \mathcal{V}$. Maximum energy charge rate per unit of storage capacity.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechStorage, column DischargingAvail.

PL^v_l Pollution limit (kg, stochastic, `[emissLimit:integer]`). $l \in \mathcal{L}$, $v \in \mathcal{V}$. This is a constraint for the building for each year.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StPollutionLimit, column Pollution.

$PP^{v,m,t}_{k,n}$ Energy purchasing cost (EUR/kWh, stochastic, `[marketEnerPurCost:decimal]`). $k \in \mathcal{K}$, $n \in \mathcal{N}^k_{Pur}$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}^m_{Tm}$. This is the cost of the energy in markets where it can be bought. If there is no ToU tariff, the cost is equal for all operational periods within the same strategic period.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTariff.

PR^v Probability of the node (unitless, stochastic, `[nodeProb:decimal]`). $v \in \mathcal{V}$.
Data. This information is in the tree structure table SG_Output_ScenarioNodes, column AbsProb.

PT^v Time period of the node (unitless, stochastic, `[nodePeriod:decimal]`). $v \in \mathcal{V}$.
Data. This information is in the tree structure table SG_Output_ScenarioNodes, column PeriodID.

$SP_{k,n}^{v,m,t}$ Energy sales price (EUR/kWh, stochastic, [enerSellPrice:decimal]). $k \in \mathcal{K}$, $n \in \mathcal{N}_S^k$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. For the types of energy that can be sold, there is a price for each operational period.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTariff column Price.

SU_i^v Subsidies for a technology (EUR/kW, stochastic, [techSubsid:decimal]). $i \in \mathcal{I}$, $v \in \mathcal{V}$. Policy makers can subsidize the investment of some efficient technologies. Usually an amount per kW is paid.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechnology, column Subsidies.

XZ_i^a Existing devices (Devices/kW/kWh, deterministic, [techEdDevices:decimal/integer]). $i \in \mathcal{I}$, $a \in \mathcal{A}$. Number of existing devices of a given age of each technology at the start of the optimization horizon of.
Data. Computed by the SM Interface using scenario generator output (SG_Output_* tables). Base values in DB table StTechnology, columns InstDevices, InstAge).

A.8. Decision Variables

$e^{v,m,t}$ Primary energy consumed per operational period (kWh, Operational, [enerConsumed:decimal]). $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. This is a computed variable for the energy consumption of the building during each short-term period.
Data. This output can be also a good representation of the building's energy performance.

$h_{k,n}^v$ Tariff choice (unitless, Strategic, [marketChoice:binary]). $k \in \mathcal{K}$, $n \in \mathcal{N}_{Tr}^k$, $v \in \mathcal{V}$. This is the decision for selecting among different tariffs. The choice is done for the subsequent period.
Data. Equal to one if the tariff is chosen.

$r_{i,k}^{v,m,t}$ Energy stored (kWh, Operational, [stoEnerStored:decimal]). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}_{Po}^i$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. This is an inventory of the amount of energy that is stored in the energy-storage technologies during each short-term period. It is calculated using the operational decisions and the technology parameters.
Data. See comments for y.

$ri_{i,k}^{v,m,t}$ Energy input to storage (kWh, Operational, [stoInput:decimal]). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}_{Po}^i$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. Addition to energy storage for each type of energy during each short-term period.
Data. See comments for y. Storage usage over time can be represented by charts.

$ro_{i,k}^{v,m,t}$ Energy output from storage (kWh, Operational, [stoOutput:decimal]). $i \in \mathcal{I}_{Sto}$, $k \in \mathcal{K}_{Po}^i$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. Release from each energy-storage technology of each type of energy during each operational period.
Data. See comments for y.

$u_{k,n}^{v,m,t}$ Energy to purchase under a given tariff (kWh, Operational, [marketPurchEner:decimal]). $k \in \mathcal{K}$, $n \in \mathcal{N}_{Pur}^k$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. Energy purchased in the market for each type of energy, to be delivered during each operational period.
Data. See comment for y.

$w_{k,n}^{v,m,t}$ Energy to sell under a given tariff (kWh, Operational, [marketSellEner:decimal]). $k \in \mathcal{K}$, $n \in \mathcal{N}_S^k$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. Energy of each type of energy to be sold in the market during each operational period.
Data. See comment for y.

$x_i^{v,a}$ Installed units of a given age for each technology and node (Devices/kW/kWh, Strategic, [techInstalled:integer/decimal]). $i \in \mathcal{I}$, $v \in \mathcal{V}$, $a \in \mathcal{A}$. This is a computed variable.
Data. A table is returned with the installed units at each node.

xc_i^v Available capacity of a technology at each node (kW or kWh (storage), Strategic, [techAvailCap:decimal]). $i \in \mathcal{I}$, $v \in \mathcal{V}$. This capacity is computed through the decisions and the parameters.
Data. The table with these decisions can be used to represent the capacity of the building throughout the time, by technology, or aggregated. We may have a column for technologies for the 'type of technology', in order to aggregate, for example, the capacity of all the installed 'PV panels', regardless their model.

$xd_i^{v,a}$ Number of units of a technology to be decommissioned (Devices or kW, Strategic, [techDecom:integer/decimal]). integer for $i \in \mathcal{I}_{Ds}$, $v \in \mathcal{V}$, $a \in \mathcal{A}_{Old}$; continuous for $i \in \mathcal{I}_{Cn}$, $v \in \mathcal{V}$, $a \in \mathcal{A}_{Old}$. For continuously-sized technologies, this is the total capacity to be decommissioned. For discretely-sized technologies, it denotes number of devices to decommission.
Data. A table with the decision is returned for each year. It is used for the computed variable 'Installed units'. The cost of decommissioning can be computed for each year.

xi_i^v Number of units of a technology to be installed (Devices or kW, Strategic, [techInst:integer/decimal]). integer for $i \in \mathcal{I}_{Ds}$, $v \in \mathcal{V}$; continuous for $i \in \mathcal{I}_{Cn}$, $v \in \mathcal{V}$. For discretely-sized technologies, this is an integer variable, whilst for continuously-sized technologies, it is a continuous one.
Data. A table with the decisions for all technologies and long-term periods is returned by the solver manager. The decisions relevant for the decision maker are those for the first year. The rest can be stored and represented to see the 'expected' evolution of the building systems evolution. Examples: 'install 2 CHP Sachs 5.5 next year', 'install 20 PV panels model y in year 3', 'install 9 kW of technology xxx in year 5', ...

$y_{i,k}^{v,m,t}$ Energy generator input (kWh, Operational, [genInputEner:decimal]). $i \in \mathcal{I}_{Gen}$, $k \in \mathcal{K}_{In}^i$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. Amount of energy used as input to an energy-creating technology, for each type of energy, operational profile, and period.
Data. The values of this decision can be represented to see how the systems will perform in the short term when making decisions, but this is not a real decision for the decision maker, unless the prototype is run as the operational version (without long-term decisions).

$z_{i,k}^{v,m,t}$ Energy generator output (kWh, Operational, [genOutputEner:decimal]). $i \in \mathcal{I}_{Gen}$, $k \in \mathcal{K}_{Out}^i$, $v \in \mathcal{V}$, $m \in \mathcal{M}$, $t \in \mathcal{T}_{Tm}^m$. Amount of energy as output from an energy-creating technology, for each type of energy, operational profile, and period.
Data. This decision is calculated, using the input energy decisions and the technologies' parameters for each type of input/output energy. The interpretation is similar to the input energy decision.

A.9. Equations

(1) Strategic

Available new technologies (devices) at each node

The available new devices of a technology (age zero) are equal to the ones installed at each node.

$$x_i^{v,0} = xi_i^v \quad \forall\, i \in \mathcal{I},\ v \in \mathcal{V} \tag{1}$$

Available old technologies (devices) at future nodes

The number of available devices whose age is not zero is equal to the number available at the previous node minus the number of decommissioned ones.

$$x_i^{v,a} = x_i^{Pa(v),a-1} - xd_i^{v,a} \quad \forall\, i \in \mathcal{I},\ v \in \mathcal{V}_{Fut},\ a \in \mathcal{A}_{Ages}^{i,v} \cap \mathcal{A}_{Old} \tag{2}$$

Available old technologies (devices) at root node

For technologies existing before the start of the optimization horizon, the number of devices available at the root note is equal to the number of existing devices minus those decommissioned at the beginning of the first period.

$$x_i^{1,a} = XZ_i^{a-1} - xd_i^{1,a} \quad \forall\, i \in \mathcal{I},\ a \in \mathcal{A}_{Ages}^{i,v} \cap \mathcal{A}_{Old} \tag{3}$$

Technology capacity calculation

The total capacity of a technology is the sum of the capacities of the installed devices at any age, corrected by the aging factors and nominal capacity.

$$xc_i^v = G_i \cdot \sum_{a \in \mathcal{A}_{Ages}^{i,v}} AG_i^a \cdot x_i^{v,a} \quad \forall\, i \in \mathcal{I},\ v \in \mathcal{V} \tag{4}$$

Investment limit

An upper limit may be imposed on the total installation, decommissioning, and maintenance cost.

$$\sum_{i \in \mathcal{I}} \left((CI_i^v - SU_i^v) \cdot G_i \cdot xi_i^v + \sum_{a \in \mathcal{A}_{Ages}^{i,v}} CD_i^{v,a} \cdot G_i \cdot xd_i^{v,a} + \sum_{a \in \mathcal{A}_{Ages}^{i,v}} CM_i^{v,a} \cdot G_i \cdot x_i^{v,a} \right) \leq IL^v \quad \forall\, v \in \mathcal{V} \tag{5}$$

Purchase tariff choice

Only one purchase tariff is allowed.

$$\sum_{n \in \mathcal{N}_{Pur}^k} h_{k,n}^v = 1 \quad \forall\, v \in \mathcal{V},\ \ k \in \mathcal{K}_{EP} \tag{6}$$

Sales tariff choice

Only one sales tariff is allowed.

$$\sum_{n \in \mathcal{N}_S^k} h_{k,n}^v = 1 \quad \forall\, v \in \mathcal{V},\ \ k \in \mathcal{K}_{ES} \tag{7}$$

Physical limit

There is a limit for installing technologies, usually established by the space available in the site. Note that within the optimizer, it can be implemented as a variable upper limit rather than a constraint.

$$\sum_{a \in \mathcal{A}_{Ages}^{i,v}} x_i^{v,a} \leq LP_i^v \quad \forall\, i \in \mathcal{I},\ v \in \mathcal{V} \tag{8}$$

Required efficiency

The amount of energy consumed and sold must be larger than the amount of primary energy consumed corrected by the efficiency parameter.

$$\sum_{m \in \mathcal{M}} DM^m \cdot \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}_{Tm}^m} \left(D_k^{v,m,t} + \sum_{n \in \mathcal{N}_S^k} w_{k,n}^{v,m,t} \right) \geq EF^v \cdot \sum_{m \in \mathcal{M}} DM^m \cdot \sum_{t \in \mathcal{T}_{Tm}^m} e^{v,m,t} \quad \forall\, v \in \mathcal{V} \tag{9}$$

Emissions limit

The total emissions of each considered pollutant cannot exceed a specified limit.

$$\sum_{m \in \mathcal{M}} DM^m \cdot \sum_{t \in \mathcal{T}_{Tm}^m} \left(\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_{In}^i} LH_{i,k,l}^v \cdot y_{i,k}^{v,m,t} + \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{N}_{Pur}^k} LC_{k,l,n}^v \cdot u_{k,n}^{v,m,t} \right) \leq PL_l^v \quad \forall\, l \in \mathcal{L},\ v \in \mathcal{V} \tag{10}$$

(2) Operational

Output energy calculation

The amount of output energy is calculated from the input energy and the conversion factor.

$$z_{i,k'}^{v,m,t} = \sum_{k \in \mathcal{K}_{In}^{i}} EC_{i,k,k'}^{v} \cdot y_{i,k}^{v,m,t} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Gen},\ k' \in \mathcal{K}_{Out}^{i}, t \in \mathcal{T}_{Tm}^{m} \tag{11}$$

Technology output limit

The energy that can be supplied by a technology is constrained by the availability of the technology and its capacity.

$$z_{i,k}^{v,m,t} \le DT^{m} \cdot AF_{i}^{v,m,t} \cdot xc_{i}^{v} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Gen},\ t \in \mathcal{T}_{Tm}^{m}, k \in \mathcal{K}_{Po}^{i} \tag{12}$$

Storage available

The energy stored in each period is the energy stored in the previous period, plus the energy sent to storage, minus the energy released from storage. All flows are corrected by their respective loss ratio parameter.

$$\begin{gathered} r_{i,k}^{v,m,t} = OS_{i,k} \cdot r_{i,k}^{v,m,t-1} + OI_{i,k}^{v} \cdot ri_{i,k}^{v,m,t-1} - OO_{i,k}^{v} \cdot ro_{i,k}^{v,m,t-1} \\ \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Sto},\ t \in \mathcal{T}_{Tm}^{m}, k \in \mathcal{K}_{Po}^{i} \end{gathered} \tag{13}$$

Storage release limit

The amount of energy that can be discharged from any energy-storage technology is limited by the installed capacity and the maximum discharge rate.

$$ro_{i,k}^{v,m,t} \le OX_{i,k}^{v} \cdot DT^{m} \cdot xc_{i}^{v} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Sto},\ t \in \mathcal{T}_{Tm}^{m}, k \in \mathcal{K}_{Po}^{i} \tag{14}$$

Storage charge limit

The amount of energy that can be charged to any energy-storage technology is limited by the installed capacity and the maximum charge rate.

$$ri_{i,k}^{v,m,t} \le OY_{i,k}^{v} \cdot DT^{m} \cdot xc_{i}^{v} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Sto},\ t \in \mathcal{T}_{Tm}^{m}, k \in \mathcal{K}_{Po}^{i} \tag{15}$$

Storage level between periods

The storage level at the first short-term period must be equal to the level at the final period (in the same strategic period).

$$r_{i,k}^{v,m,1} = r_{i,k}^{v,m,Tmax} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Sto},\ k \in \mathcal{K}_{Po}^{i} \tag{16}$$

Lower storage limit

The amount of energy that can be stored in any energy-storage technology must be greater than the capacity installed, corrected by the minimum charge required.

$$r_{i,k}^{v,m,t} \ge OA_{i,k}^{v} \cdot xc_{i}^{v} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Sto},\ t \in \mathcal{T}_{Tm}^{m}, k \in \mathcal{K}_{Po}^{i} \tag{17}$$

Upper storage limit

The amount of energy that can be stored in any energy-storage technology must be lower than the capacity installed, corrected by the maximum charge allowed.

$$r_{i,k}^{v,m,t} \le OB_{i,k}^{v} \cdot xc_{i}^{v} \quad \forall\, v \in \mathcal{V},\ m \in \mathcal{M},\ \ i \in \mathcal{I}_{Sto},\ t \in \mathcal{T}_{Tm}^{m}, k \in \mathcal{K}_{Po}^{i} \tag{18}$$

Energy balance

The energy supplied must meet the energy demand minus the energy saved due to passive technologies. It is composed of the energy produced with energy-creating technologies plus the energy purchased in the market minus the energy sold, energy for storage and energy for production. On the demand side, the energy saved with passive technologies diminish the total demand.

$$\sum_{i\in\mathcal{I}_{Gen}} z_{i,k}^{v,m,t} - \sum_{i\in\mathcal{I}_{Gen}} y_{i,k}^{v,m,t} + \sum_{n\in\mathcal{N}_{Pur}^{k}} u_{k,n}^{v,m,t} - \sum_{n\in\mathcal{N}_{S}^{k}} w_{k,n}^{v,m,t} \tag{19}$$

$$+ \sum_{i\in\mathcal{I}_{Sto}} \left(ro_{i,k}^{v,m,t} - ri_{i,k}^{v,m,t}\right) = D_{k}^{v,m,t} \cdot \left(1 - \sum_{i\in\mathcal{I}_{PU}} OD_{i,k}^{v} \cdot xc_{i}^{v}\right)$$

$$\forall\, k\in\mathcal{K},\ v\in\mathcal{V},\ m\in\mathcal{M},\ t\in\mathcal{T}_{Tm}^{m}$$

Purchasing limit by contract

The amount of energy that can be purchased at a given node must not exceed the amount stipulated in the previously signed contract.

$$u_{k,n}^{v,m,t} \leq h_{k,n}^{v} \cdot ME_{k,n} \cdot DT^{m} \quad \forall\, k\in\mathcal{K},\ m\in\mathcal{M},\ n\in\mathcal{N}_{Pur}^{k}, t\in\mathcal{T}_{Tm}^{m} \tag{20}$$

Sales limit by contract

The amount of energy that can be purchased at a given node must not exceed the amount stipulated in the previously signed contract.

$$w_{k,n}^{v,m,t} \leq h_{k,n}^{v} \cdot ME_{k,n} \cdot DT^{m} \quad \forall\, k\in\mathcal{K},\ m\in\mathcal{M},\ n\in\mathcal{N}_{S}^{k}, t\in\mathcal{T}_{Tm}^{m} \tag{21}$$

Primary energy calculation

The primary energy (not from a fictitious market) consumed is the sum of the processed energy of each type and the energy used as an input fuel.

$$e^{v,m,t} = \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}_{Pur}^{k}} B_{k,n} \cdot u_{k,n}^{v,m,t} \quad \forall\, v\in\mathcal{V},\ m\in\mathcal{M},\ t\in\mathcal{T}_{Tm}^{m} \tag{22}$$

(3) Objective

Total discounted expected cost

The total cost consists of the installation cost (the term within the summation over i in (23a)), the decommissioning cost (first term in (23b)), the maintenance cost (second term in (23b)), the energy purchasing cost (first term in (23c)), and the operation cost (23d). Incomes from energy sales are subtracted in (23c) and subsidies directly diminish the technology installation costs in (23a).

$$\text{minimize} \sum_{v\in\mathcal{V}} (1+DR)^{-PT^{v}} \cdot PR^{v} \cdot \Bigg(\sum_{i\in\mathcal{I}} \Bigg((CI_{i}^{v} - SU_{i}^{v}) \cdot G_{i} \cdot xi_{i}^{v} \tag{23a}$$

$$+ \sum_{a\in\mathcal{A}_{Ages}^{i,v}} CD_{i}^{v,a} \cdot G_{i} \cdot xd_{i}^{v,a} + \sum_{a\in\mathcal{A}_{Ages}^{i,v}} CM_{i}^{v,a} \cdot G_{i} \cdot x_{i}^{v,a} \Bigg) \tag{23b}$$

$$+ \sum_{m\in\mathcal{M}} DM^{m} \cdot \sum_{t\in\mathcal{T}_{Tm}^{m,t}} \Bigg(\sum_{k\in\mathcal{K}_{EP}} \sum_{n\in\mathcal{N}_{Pur}^{k,n}} PP_{k,n}^{v,m,t} \cdot u_{k,n}^{v,m,t} - \sum_{k\in\mathcal{K}_{\mathcal{ES}}} \sum_{n\in\mathcal{N}_{S}^{k,n}} SP_{k,n}^{v,m,t} \cdot w_{k,n}^{v,m,t} \tag{23c}$$

$$+ \sum_{i\in\mathcal{I}_{Gen}, k\in\mathcal{K}_{Out}^{i,k}} CO_{i,k}^{v} \cdot z_{i,k}^{v,m,t} + \sum_{i\in\mathcal{I}_{Sto}, k\in\mathcal{K}_{Po}^{i,k}} CO_{i,k}^{v} \cdot r_{i,k}^{v,m,t} \Bigg)\Bigg) \tag{23d}$$

A.10. Data Structures

This section contains the data structures of the SMS within the Solver Manager. Descriptions are omitted for the sake of space.

Sets

```
SMSsets(enrimaSMS)[, -c(4, 5, 9)]

##    id symbol            tag loc inSet  setType ordered   setDom
## 1  11      a            age sup    NA      set    TRUE       NA
## 2   1      i       enerTech sub    NA      set   FALSE       NA
## 3   3      k       enerType sub    NA      set   FALSE       NA
## 4   4      l      pollutant sub    NA      set   FALSE       NA
## 5   5      n         tariff sub    NA      set   FALSE       NA
## 6   6      v           node sup    NA      set    TRUE       NA
## 7   7      m        profile sup    NA      set   FALSE       NA
## 8   8      t           time sup    NA      set    TRUE       NA
## 9  14   _p1_k        outEner sub     3    alias      NA       NA
## 10 32   _p1_v     parentNode sup     6    alias      NA       NA
## 11 30     Ds       discTech sub     1   subset      NA       NA
## 12 31     Cn       contTech sub     1   subset      NA       NA
## 13 26    Sto        stoTech sub     1   subset      NA       NA
## 14 27     PU    passiveTech sub     1   subset      NA       NA
## 15 12    New         ageNew sub    11   subset      NA       NA
## 16 13    Old         ageOld sub    11   subset      NA       NA
## 17 35    Gen        genTech sub     1   subset      NA       NA
## 18 36   Root       rootNode sub     6   subset      NA       NA
## 19 40     TS   salesTariffs sub     5   subset      NA       NA
## 20 41     TP   purchTariffs sub     5   subset      NA       NA
## 21 42     EP   enerPurchase sub     3   subset      NA       NA
## 22 43     ES       enerSale sub     3   subset      NA       NA
## 23 44    Ren        enerRen sub     3   subset      NA       NA
## 24 45    Fut    futureNodes sup     6   subset      NA       NA
## 25 46   Elec       elecEner sup     3   subset      NA       NA
## 26 47   Heat       heatEner sub     3   subset      NA       NA
## 27 48   Cool       coolEner sub     3   subset      NA       NA
## 28 49    Dem        demEner sub     3   subset      NA       NA
## 29 17    Pur    enerBtariff sub     5 multidim      NA     3, 5
## 30 18      S    enerStariff sub     5 multidim      NA     3, 5
## 31 19     In      inputEner sub     3 multidim      NA    35, 3
## 32 20    Out     outEnergy sub    14 multidim      NA   35, 14
## 33 33     Pa    parentNodes sub    32 multidim      NA    6, 32
## 34 34     Tr      enerTrade sub     5 multidim      NA     3, 5
## 35 37     Tm     perProfile sub     8 multidim      NA     7, 8
## 36 38     Po principalEner sub     3 multidim      NA     1, 3
## 37 39   Ages       agesNode sub    11 multidim      NA 1, 6, 11
```

Variables

```
SMSvars(enrimaSMS)[, -c(4, 5, 16)]

##     id symbol              tag      nature                units
## 1    1     xi         techInst   Strategic       Devices or kW
## 2    2     xd        techDecom   Strategic       Devices or kW
## 3    3     xc     techAvailCap   Strategic kW or kWh (storage)
## 4    4      x    techInstalled   Strategic      Devices/kW/kWh
## 5    5      h     marketChoice   Strategic            unitless
## 6    7      y     genInputEner Operational                 kWh
## 7    8      z    genOutputEner Operational                 kWh
## 8    9      u marketPurchEner Operational                 kWh
## 9   10      w   marketSellEner Operational                 kWh
## 10  11     ri         stoInput Operational                 kWh
## 11  12     ro        stoOutput Operational                 kWh
## 12  13      r     stoEnerStored Operational                 kWh
## 13  14      c        totalCost   Objective                 EUR
## 14  15      e     enerConsumed Operational                 kWh
##             dataType            setInd domCond             ind integer positive
## 1   integer/decimal               NULL      NA            1, 6    TRUE       NA
## 2   integer/decimal               NULL      NA        1, 6, 11    TRUE       NA
## 3           decimal               NULL      NA            1, 6      NA     TRUE
## 4   integer/decimal               NULL      NA        1, 6, 11      NA     TRUE
## 5            binary         3, 34, 6      NA         3, 5, 6      NA       NA
## 6           decimal 35, 19, 6, 7, 37      NA 1, 3, 6, 7, 8      NA     TRUE
## 7           decimal 35, 20, 6, 7, 37      NA 1, 3, 6, 7, 8      NA     TRUE
## 8           decimal  3, 17, 6, 7, 37      NA 3, 5, 6, 7, 8      NA     TRUE
## 9           decimal  3, 18, 6, 7, 37      NA 3, 5, 6, 7, 8      NA     TRUE
## 10          decimal 26, 38, 6, 7, 37      NA 1, 3, 6, 7, 8      NA     TRUE
## 11          decimal 26, 38, 6, 7, 37      NA 1, 3, 6, 7, 8      NA     TRUE
## 12          decimal 26, 38, 6, 7, 37      NA 1, 3, 6, 7, 8      NA     TRUE
## 13          decimal               NULL      NA            NULL      NA       NA
## 14          decimal          6, 7, 37      NA         6, 7, 8      NA     TRUE
##     binary                                   varType
## 1       NA          30, 6, integer, 31, 6, continuous
## 2       NA 30, 6, 13, integer, 31, 6, 13, continuous
## 3       NA                                      NULL
## 4       NA                                      NULL
## 5     TRUE                                      NULL
## 6       NA                                      NULL
## 7       NA                                      NULL
## 8       NA                                      NULL
## 9       NA                                      NULL
## 10      NA                                      NULL
## 11      NA                                      NULL
## 12      NA                                      NULL
## 13      NA                                      NULL
## 14      NA                                      NULL
```

Parameters

```
SMSpars(enrimaSMS)[, -c(4, 5, 13)]
```

```
##    id symbol              tag        nature          units
## 1   2      G      techCapacity deterministic       kW/Device
## 2   7     DT           durOper deterministic           hours
## 3   4     AG         techAging deterministic          kW/kWh
## 4  11     OS          stoAvail deterministic         kWh/kWh
## 5   3     ME        maxTrading deterministic              kW
## 6   8     DM     profileWeight deterministic            days
## 7  17     CI       techInstCost    stochastic          EUR/kW
## 8  18     CM      techMaintCost    stochastic EUR/kW, EUR/kWh
## 9  19     CD      techDecomCost    stochastic          EUR/kW
## 10  9     EC        genConvFact    stochastic         kWh/kWh
## 11 12     OI      stoChargRatio    stochastic         kWh/kWh
## 12 13     OO   stoDisChargRatio    stochastic         kWh/kWh
## 13 14     OX         stoDisRate    stochastic          kW/kWh
## 14 40     OY      stoChargeRate    stochastic          kW/kWh
## 15 15     OA        stoLowerLim    stochastic         kWh/kWh
## 16 16     OB        stoUpperLim    stochastic         kWh/kWh
## 17 31     IL           invLimit    stochastic             EUR
## 18 23     PP marketEnerPurCost    stochastic         EUR/kWh
## 19 24     SP      enerSellPrice    stochastic         EUR/kWh
## 20 25     SU         techSubsid    stochastic          EUR/kW
## 21  1      D         enerDemand    stochastic             kWh
## 22 32     OD       pasDemandRed    stochastic         kWh/kWh
## 23 21     CO        genOperCost    stochastic         EUR/kWh
## 24 29      B      marketPrimEff    stochastic         kWh/kWh
## 25 10     AF           genAvail    stochastic         kWh/kWh
## 26 26     LH        genEmission    stochastic          kg/kWh
## 27 27     LC     marketEmission    stochastic          kg/kWh
## 28 28     PL         emissLimit    stochastic              kg
## 29 30     EF        buildingEff    stochastic        unitless
## 30 36     PR           nodeProb    stochastic        unitless
## 31 37     PT         nodePeriod    stochastic        unitless
## 32 38     LP        techPhysLim    stochastic  Devices/KW/kWh
## 33 39     XZ      techEdDevices deterministic  Devices/kW/kWh
##            dataType          setInd domCond group            ind
## 1           decimal              NA      NA    NA              1
## 2           decimal              NA      NA    NA              7
## 3           decimal              NA      NA    NA          1, 11
## 4      decimal(0..1)         26, 38      NA    NA           1, 3
## 5           decimal           3, 34      NA    NA           3, 5
## 6           integer            NULL      NA    NA              7
## 7           decimal            NULL      NA    NA           1, 6
## 8           decimal            NULL      NA    NA       1, 6, 11
## 9           decimal            NULL      NA    NA       1, 6, 11
## 10          decimal   35, 19, 20, 6      NA    NA    1, 3, 14, 6
## 11     decimal(0..1)     26, 38, 6      NA    NA        1, 3, 6
## 12       decimal(>1)     26, 38, 6      NA    NA        1, 3, 6
## 13          decimal      26, 38, 6      NA    NA        1, 3, 6
## 14          decimal      26, 38, 6      NA    NA        1, 3, 6
```

```
## 15   decimal(0..1)       26, 38, 6       NA     NA       1, 3, 6
## 16   decimal(0..1)       26, 38, 6       NA     NA       1, 3, 6
## 17         integer            NULL       NA     NA             6
## 18         decimal 3, 17, 6, 7, 37       NA     NA 3, 5, 6, 7, 8
## 19         decimal 3, 18, 6, 7, 37       NA     NA 3, 5, 6, 7, 8
## 20         decimal            NULL       NA     NA          1, 6
## 21         decimal    49, 6, 7, 37       NA     NA    3, 6, 7, 8
## 22         decimal       27, 38, 6       NA     NA       1, 3, 6
## 23         decimal       35, 20, 6       NA     NA       1, 3, 6
## 24         decimal           3, 17       NA     NA          3, 5
## 25         decimal     1, 6, 7, 37       NA     NA    1, 6, 7, 8
## 26         decimal    35, 19, 4, 6       NA     NA    1, 3, 4, 6
## 27         decimal     3, 4, 17, 6       NA     NA    3, 4, 5, 6
## 28         integer            NULL       NA     NA          4, 6
## 29         decimal            NULL       NA     NA             6
## 30         decimal            NULL       NA     NA             6
## 31         decimal            NULL       NA     NA             6
## 32 decimal/integer            NULL       NA     NA          1, 6
## 33 decimal/integer           1, 11       NA     NA         1, 11
```

Constants

```
SMSconsts(enrimaSMS)[, -c(4, 5, 8)]

##   id symbol   tag value   aux
## 1  1     DR dRate  0.05 FALSE
## 2  2      1  <NA>     1  TRUE
## 3  3   Tmax  <NA>     T  TRUE
## 4  4      0  <NA>     0  TRUE
```

Equations

```
SMSeqs(enrimaSMS)[, -c(3, 4, 5, 9)]

##      id          symbol     nature relation           domain
## 1     1      eqAvailNew constraint       eq             1, 6
## 2     2      eqAvailOld constraint       eq    1, 13, 45, 39
## 3    19  eqExistingTech constraint       eq    1, 13, 36, 39
## 4     3      eqCapacity constraint       eq             1, 6
## 5     4        eqInvLim constraint      lte                6
## 6     5    eqTarChoiceP constraint       eq            6, 42
## 7     6    eqTarChoiceS constraint       eq            6, 43
## 8     7    eqOutputEner constraint       eq 35, 20, 6, 7, 37
## 9     8     eqtechLimit constraint      lte 35, 38, 6, 7, 37
## 10    9        eqStoInv constraint       eq 6, 7, 37, 26, 38
## 11   10  eqReleaseLimit constraint      lte 6, 7, 37, 26, 38
## 12   22   eqChargeLimit constraint      lte 6, 7, 37, 26, 38
## 13   23      eqStoLevel constraint       eq     6, 7, 26, 38
## 14   11      eqstolLimit constraint      gte 6, 7, 37, 26, 38
## 15   12      eqstouLimit constraint      lte 6, 7, 37, 26, 38
```

```
## 16  13     eqEnergyBal constraint       eq      3, 6, 7, 37
## 17  14        eqPurLim constraint      lte     3, 17, 7, 37
## 18  15      eqSalesLim constraint      lte     3, 18, 7, 37
## 19  16        eqEnergy constraint       eq         6, 7, 37
## 20  17     eqTotalCost  objective       eq             NULL
## 21  18 eqPhysicalLimit constraint      lte             1, 6
## 22  20           eqEff constraint      gte                6
## 23  21     eqEmissions constraint      lte             4, 6
## 24 101      auxParNode        aux       eq             NULL
## 25 102       auxPreAge        aux       eq             NULL
## 26 103      auxOutEner        aux       eq             NULL
## 27 110         auxPrev        aux       eq             NULL
## 28 111      period t 1        aux       eq             NULL
## 29 112   last period t        aux       eq             NULL
## 30 113              a0        aux       eq             NULL
```

B. Optimization Instance Example

This appendix contains a numerical example of the optimization problem defined in subsection 4.2 with a limited number of SMS entities. In the following, you will find:

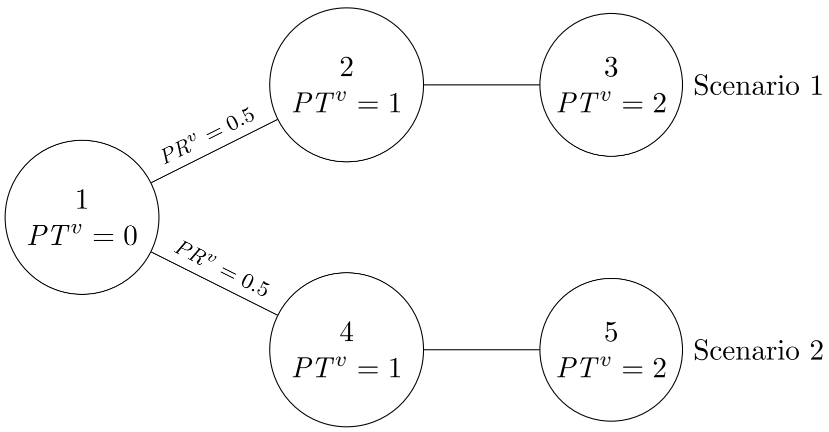
1. The set elements in each SMS set ($\emptyset$ when empty).
2. Parameter values used in the instance. If the table is very large, the first and last rows are shown. A graphical representation for some parameters is shown.
3. The result of the optimization returned by the Solver Manager. Similarly to the parameters, values and charts are presented.

B.1. Description

Highlights:

- Stochastic: two stages, two scenarios, equiprobable;
- Three strategic periods;
- Four operational profiles;
- Six operational periods per operational profile;
- PV technology, without operational costs;
- Only electricity demand;
- Two contracts;
- No efficiency, investment, pollution constraints.

Scenario tree:



B.2. Sets

$\mathcal{A} = \{0, 1, 2\}$
$\mathcal{I} = \{\text{PV}\}$
$\mathcal{K} = \{\text{electricity, radiation}\}$
$\mathcal{L} = \{\emptyset\}$
$\mathcal{N} = \{\text{normalRTEp, touRTEp}\}$
$\mathcal{V} = \{1, 2, 3, 4, 5\}$
$\mathcal{M} = \{\text{profile1, profile2, profile3, profile4}\}$
$\mathcal{T} = \{\text{time1, time2, time3, time4, time5, time6}\}$

Subsets

$\mathcal{I}_{Ds} = \{\text{PV}\}$
$\mathcal{I}_{Cn} = \{\emptyset\}$
$\mathcal{I}_{Sto} = \{\emptyset\}$
$\mathcal{I}_{PU} = \{\emptyset\}$
$\mathcal{A}_{New} = \{0\}$
$\mathcal{A}_{Old} = \{1, 2\}$
$\mathcal{I}_{Gen} = \{\text{PV}\}$
$\mathcal{V}_{Root} = \{1\}$
$\mathcal{N}_{TS} = \{\emptyset\}$
$\mathcal{N}_{TP} = \{\text{normalRTEp, touRTEp}\}$
$\mathcal{K}_{EP} = \{\text{electricity}\}$
$\mathcal{K}_{ES} = \{\emptyset\}$
$\mathcal{K}_{Ren} = \{\text{radiation}\}$
$\mathcal{V}_{Fut} = \{2, 3, 4, 5\}$
$\mathcal{K}_{Elec} = \{\text{electricity}\}$
$\mathcal{K}_{Heat} = \{\emptyset\}$
$\mathcal{K}_{Cool} = \{\emptyset\}$
$\mathcal{K}_{Dem} = \{\text{electricity}\}$

Conditional Sets

The only type of energy on the demand side (electricity) can be bought through any of the two contracts available:

$\mathcal{N}^{k}_{Pur}$

k	n
electricity	normalRTEp
electricity	touRTEp

The type of energy energy input of PV technology is radiation:

$\mathcal{K}^{i}_{In}$

i	k
PV	radiation

The type of energy input of PV technology is electricity:

$\mathcal{K}^{i}_{Out}$

i	kp
PV	electricity

This is the mapping of parent nodes (vp=v'=parent node), see scenario tree above:

$\mathcal{V}^{v}_{Pa}$

v	vp
2	1
3	2
4	1
5	4

The principal type of energy of PV technology is electricity:

$\mathcal{K}^{i}_{Po}$

i	k
PV	electricity

All the profiles have the same 6 operational periods:

$\mathcal{T}^{m}_{Tm}$

profile1	profile2	profile3	profile4
time1	time1	time1	time1
time2	time2	time2	time2
time3	time3	time3	time3
time4	time4	time4	time4
time5	time5	time5	time5
time6	time6	time6	time6

There are no existing technologies. Therefore, the possible ages of the PV panels at each node are the periods of that node and its predecessor nodes (e.g., at node 3, whose period is 2, there may be PV panels installed at period 0, 1, or 2 whose ages are 2, 1, and 0 respectively):

$\mathcal{A}^{i,v}_{Ages}$

v	a
1	0
2	0,1
3	0,1,2
4	0,1
5	0,1,2

B.3. Parameters

Period of nodes (PT)

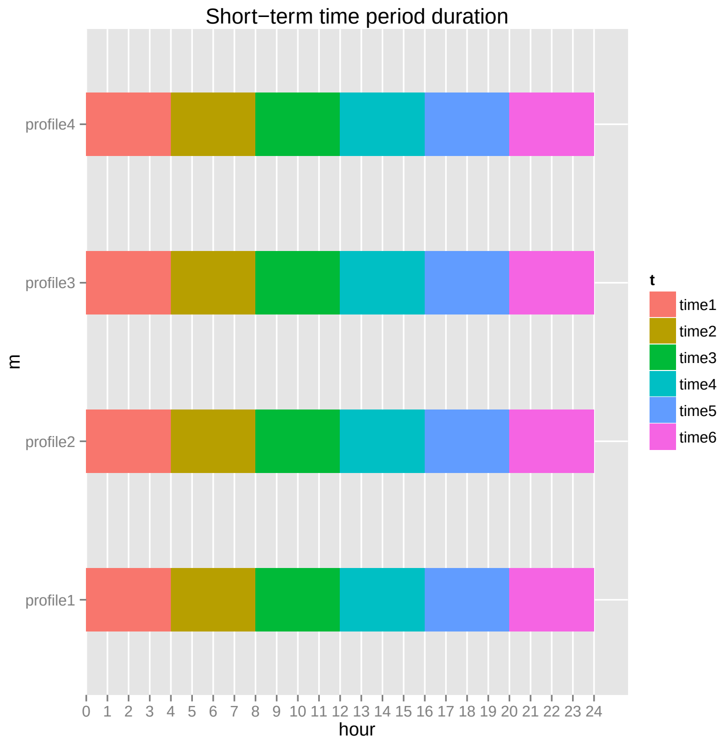
There are five nodes and three periods (see scenario tree above).

v	PT^v
1	0
2	1
3	2
4	1
5	2

Short-term periods duration time (DT)

The duration of each operational period within each profile is four hours as every profile (representing 24 hours) is divided into 6 operational periods.

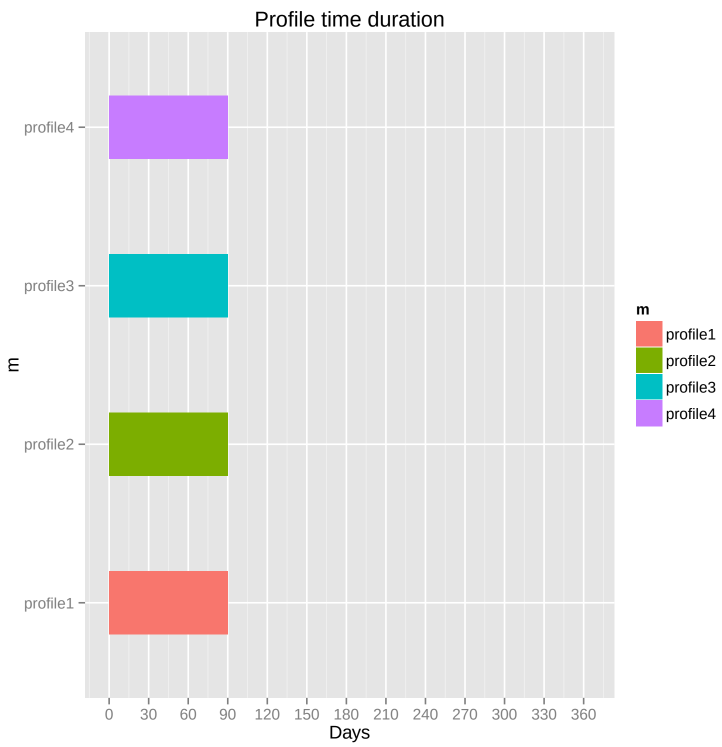
m	DT^m
profile1	4.00
profile2	4.00
profile3	4.00
profile4	4.00



Profile duration (DM)

For this example we consider four profiles, with 360 days in total. Thus, the weight of each profile is 90 days.

m	DM^m
profile1	90
profile2	90
profile3	90
profile4	90



Technologies Nominal Capacity (G)

We consider a PV sunmodule SW245, 245Wp by solarworld (`http://www.solarworld.com`).

i	G_i
PV	0.2450

Aging Factor (AG)

We do not consider aging effect for this example.

i	a	AG_i^a
PV	0	1
PV	1	1
PV	2	1

Conversion Rates (EC)

PV converts radiation to electricity directly (see AF parameter for availability throughout the time).

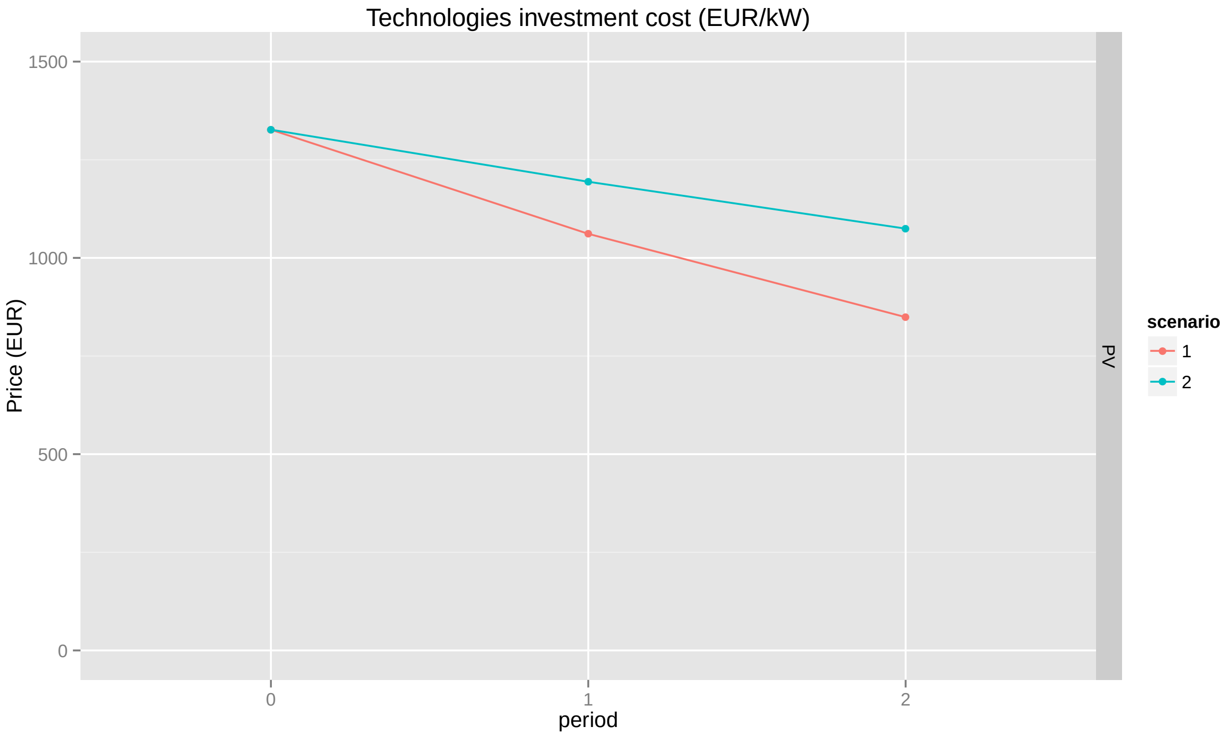
i	k	kp	v	$EC_{i,k,k'}^v$
PV	radiation	electricity	1	1.00
PV	radiation	electricity	2	1.00
PV	radiation	electricity	3	1.00
PV	radiation	electricity	4	1.00
PV	radiation	electricity	5	1.00

Technology Investment Cost (CI)

The base cost has been obtained from the preoc database (`http://www.preoc.es`). Price evolution has been simulated as follows:

- ◇ Scenario 1: -20% per year.
- ◇ Scenario 2: -10% per year.

i	v	CI_i^v
PV	1	1327
PV	2	1061
PV	3	849
PV	4	1194
PV	5	1075

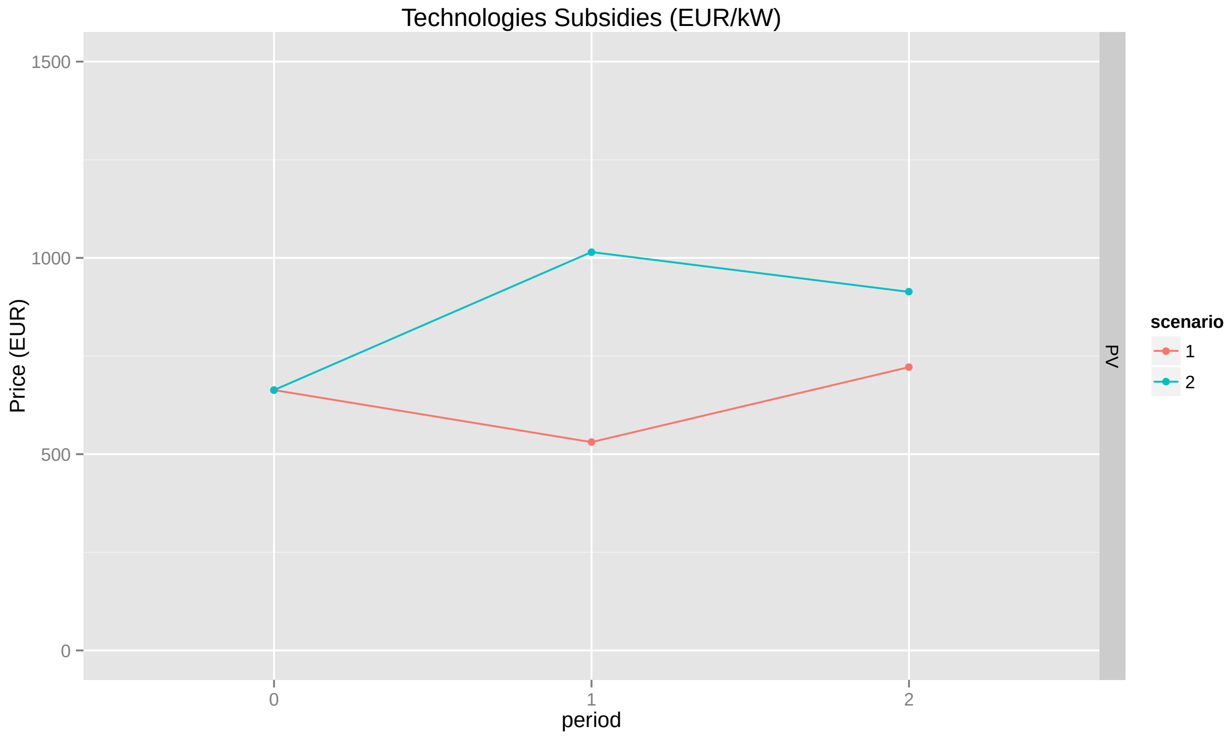


Technology Subsidies (SU)

As we have a very short decision horizon (three years), we assume artificial, very large subsidies (50%) in order to obtain strategic decisions for the technologies. Furthermore, subsidies evolution is simulated as follows:

- Scenario 1: -20% per year.
- Scenario 2: -10% per year.

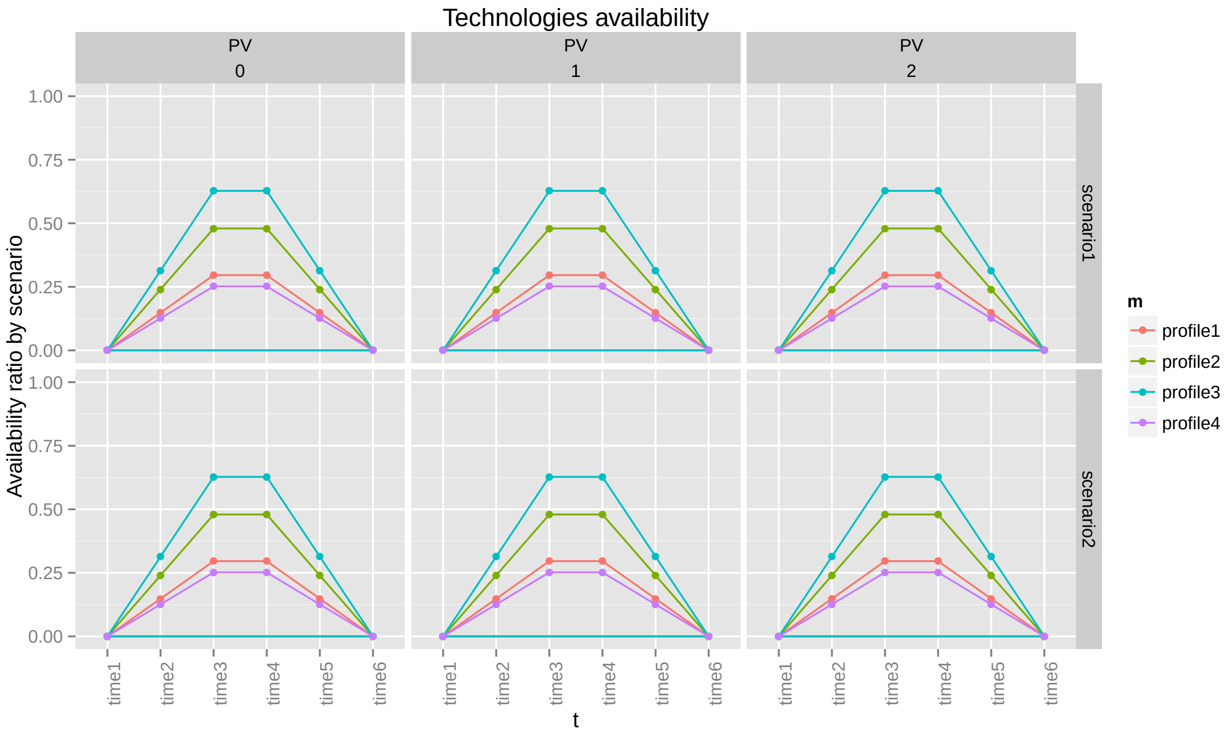
i	v	SU_i^v
PV	1	663.33
PV	2	530.66
PV	3	721.70
PV	4	1014.89
PV	5	913.40



Technology Availability Factor (AF)

Calculated using EC JRC PVGIS tool (`http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php`) for profiles. For the scope of the example, we assume weights {0,0.5,1,1,0.5,0} for operational periods. More precise calculations may be done using the same tool. Note that the factor may also have different values for each scenario / node.

i	v	m	t	$AF_i^{v,m,t}$
PV	1	profile1	time1	0
PV	1	profile1	time2	0.148
PV	1	profile1	time3	0.2959
PV	1	profile1	time4	0.2959
...	...	...	...	...
PV	5	profile4	time3	0.2517
PV	5	profile4	time4	0.2517
PV	5	profile4	time5	0.1259
PV	5	profile4	time6	0



Technologies Physical Limit (LP)

According to D1.1 [8] FASAD's roof area is 3237 m^2. If we assume 2 m^2 space requirement per PV module then we can allocate 1500 units.

i	v	PL_i^v
PV	1	1500
PV	2	1500
PV	3	1500
PV	4	1500
PV	5	1500

Trading limit by contract (ME)

FASAD's installation capacity is 276.80 kW.

k	n	$ME_{k,n}$
electricity	normalRTEp	276.80
electricity	touRTEp	276.80

Energy price (PP)

We use the energy price in [8] and simulate the price evolution as follows:

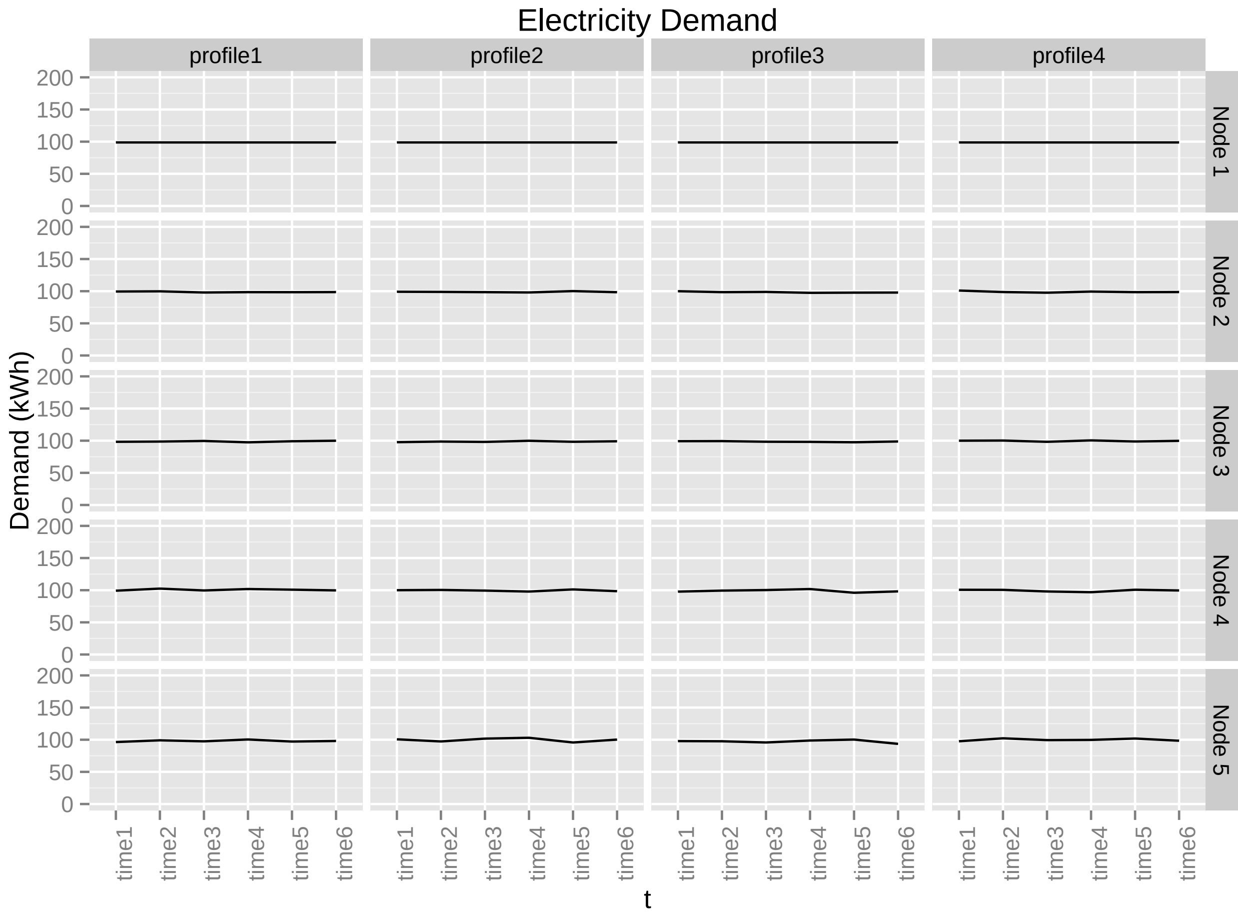
- Node 1: Base price for electricity: 0.1346 EUR/kWh.
- Scenario 1: 10% more expensive touRTEp compared to normalRTEp.
- Scenario 2: 5% cheaper touRTEp compared to normalRTEp.

k	n	v	m	t	$PP_{k,n}^{v,m,t}$
electricity	normalRTEp	1	profile1	time1	0.1346
electricity	touRTEp	1	profile1	time1	0.1346
electricity	normalRTEp	1	profile1	time2	0.1346
electricity	touRTEp	1	profile1	time2	0.1346
...	...	...	...	...	...
electricity	normalRTEp	5	profile4	time5	0.1346
electricity	touRTEp	5	profile4	time5	0.1278
electricity	normalRTEp	5	profile4	time6	0.1346
electricity	touRTEp	5	profile4	time6	0.1278

Demand (D)

From Deliverable D1.1 we know that FASAD's yearly electricity consumption was 213.50 MWh. We simulate the demand throughout the time for the example as follows: mean = annual demand distributed amongst 360 × 6 periods, standard deviation = 1 and 2 for scenarios 1 and 2, respectively.

k	v	m	t	$D_k^{v,m,t}$
electricity	1	profile1	time1	98.84
electricity	2	profile1	time1	99.47
electricity	3	profile1	time1	98.26
electricity	4	profile1	time1	99.25
...	...	...	...	...
electricity	2	profile4	time6	98.66
electricity	3	profile4	time6	99.77
electricity	4	profile4	time6	99.6
electricity	5	profile4	time6	98.43



Node Probability (PR)

Two equiprobable scenarios are considered (see tree above).

v	PR^v
1	1.00
2	0.50
3	0.50
4	0.50
5	0.50

B.4. Solution

Objective: 76,051 EUR (25,350 EUR per year)

Compare to FASAD's annual cost projection:

```
sum(1.05^(-1 * c(0:2)) * 28730.91)

## [1] 82153
```

The solution is consistent with FASAD's real data. The difference is due to the fact that parameter values have been simulated. Prior to this instance, simpler instances, i.e., with only one strategic period, were executed obtaining exact values.

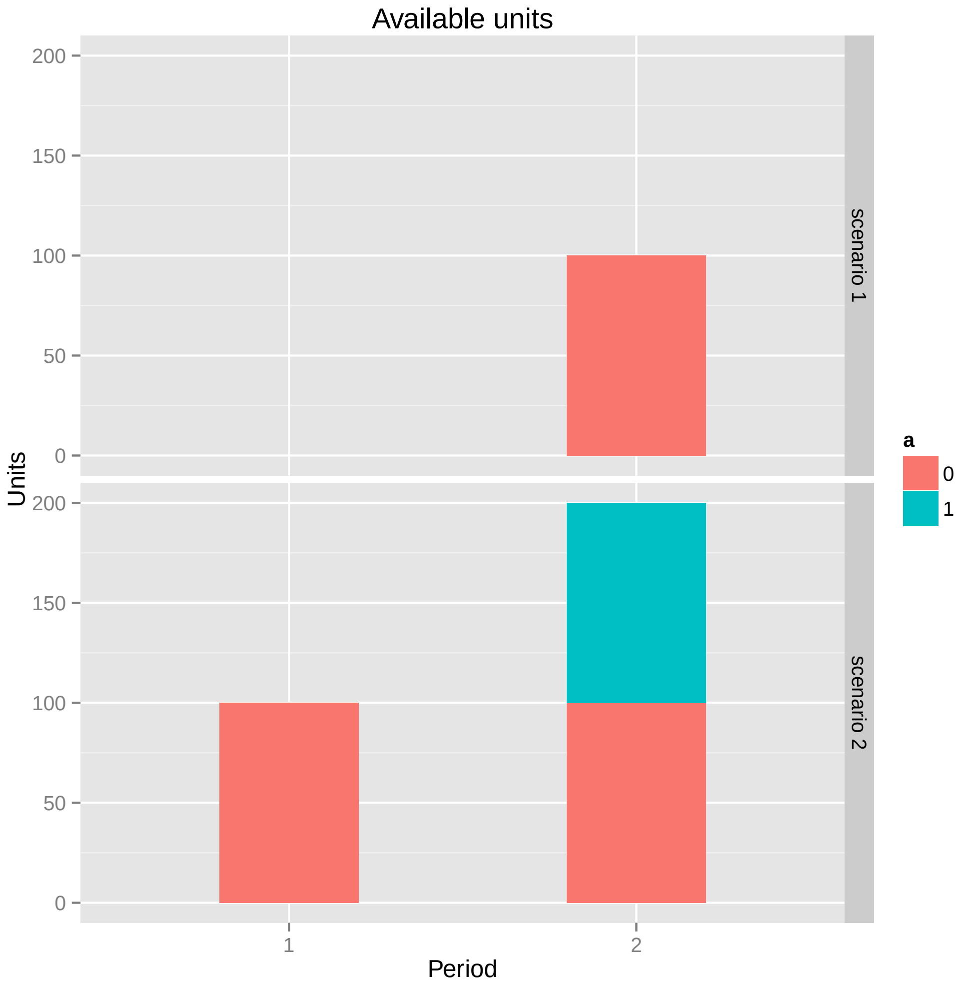
Technologies: units to be installed, xi

i	v	xi_i^v
PV	3	100
PV	4	100
PV	5	100

The solution obtained suggests no investment this year (time = 0). If scenario 1 occurrs, then investments will be likely done in period 2 but if scenario 2 occurs investments in PV will be likely done in periods 1 and 2.

Technologies: available units, x

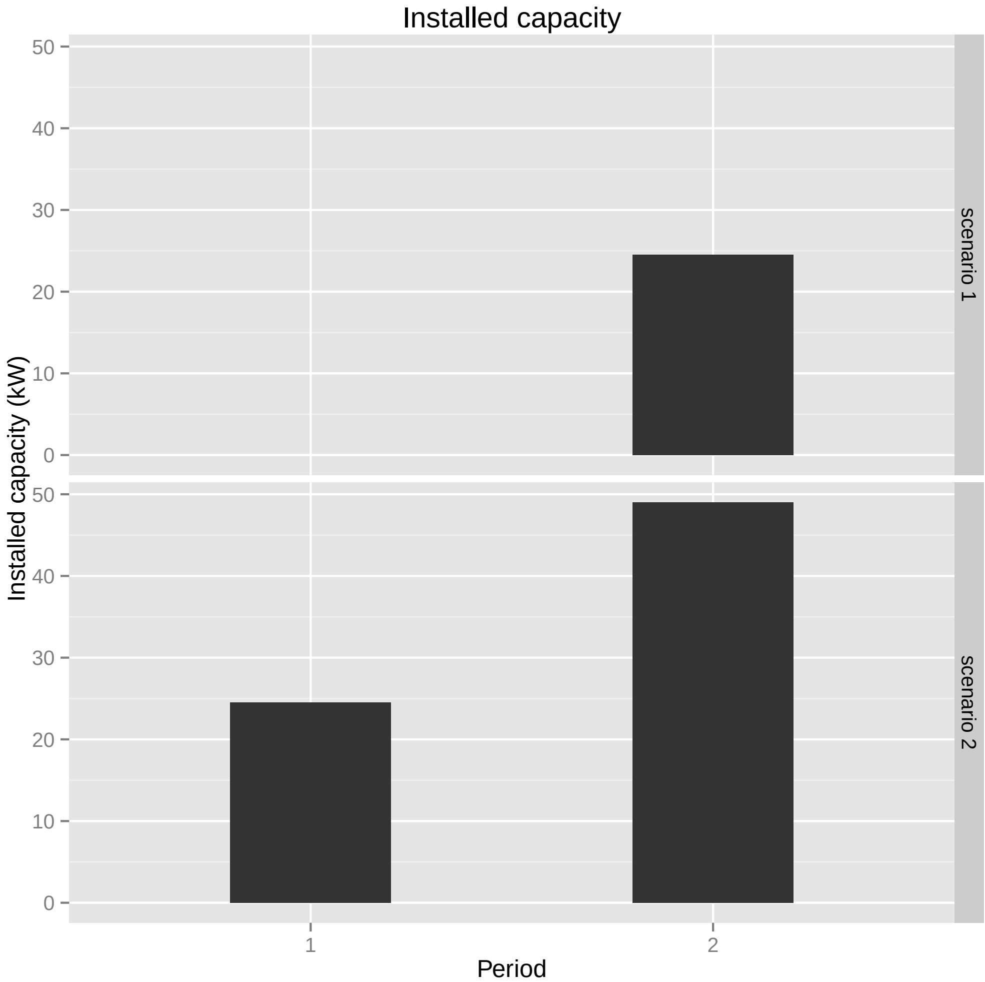
i	v	a	$x_i^{v,a}$
PV	3	0	100.00
PV	4	0	100.00
PV	5	0	100.00
PV	5	1	100.00



Technologies: installed capacity, xc

Given the previous investment decisions, the building has the following available capacity. Note that in scenario 2, year 2 we have technologies of different ages. If we had taken into account the aging of technologies, the model would have reflected that issue in xc_i^v.

i	v	xc_i^v
PV	3	24.50
PV	4	24.50
PV	5	49.00



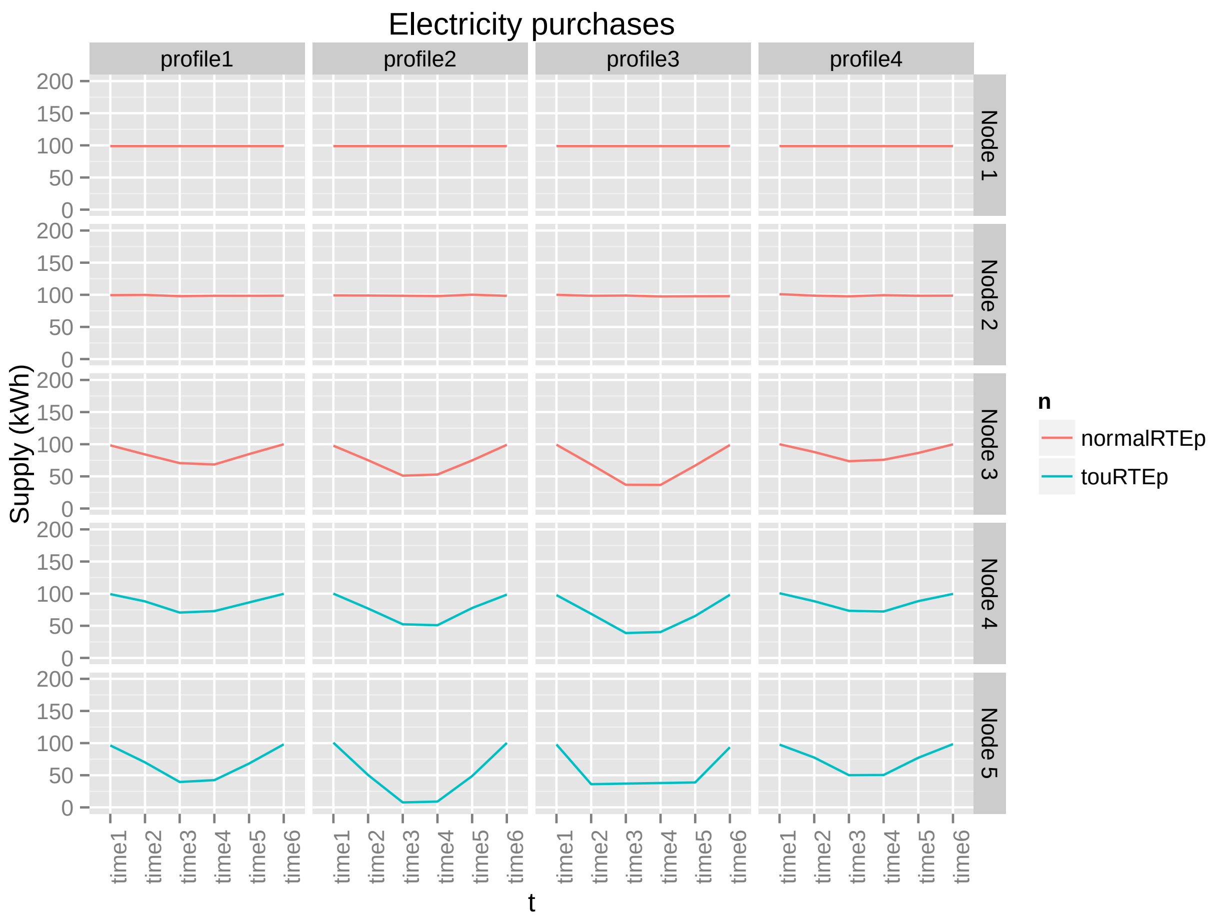
Contract selection

k	n	v	$h_{k,n}^v$
electricity	normalRTEp	1	1
electricity	normalRTEp	2	1
electricity	normalRTEp	3	1
electricity	touRTEp	4	1
electricity	touRTEp	5	1

The optimization suggests to select the normal tariff this year (time=0) and continue with it if scenario 1 occurs. However, if scenario 2 occurs, in period 1 we will have to change to the 'tou' tariff. The following decisions (energy purchases and energy generation) are the optimal operational decisions throughout the decision horizon in order to meet the demand of electricity, given the strategic decisions made.

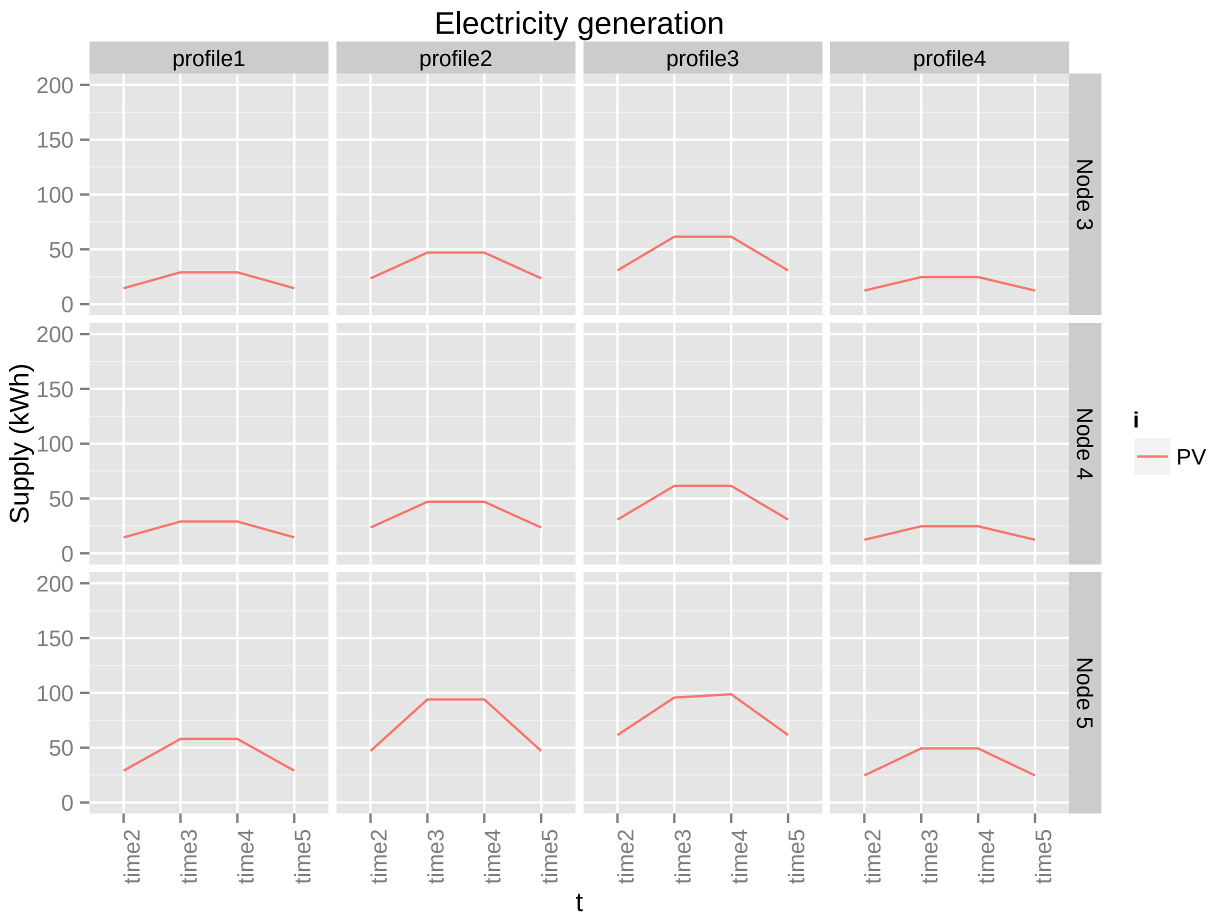
Energy Purchases (u)

k	n	v	m	t	$u_{k,n}^{v,m,t}$
electricity	normalRTEp	1	profile1	time1	98.84
electricity	normalRTEp	1	profile1	time2	98.84
electricity	normalRTEp	1	profile1	time3	98.84
electricity	normalRTEp	1	profile1	time4	98.84
...	...	...	...	...	...
electricity	touRTEp	5	profile4	time3	50.06
electricity	touRTEp	5	profile4	time4	50.38
electricity	touRTEp	5	profile4	time5	77.14
electricity	touRTEp	5	profile4	time6	98.43



Energy generation

i	k	v	m	t	$z_{k,n}^{v,m,t}$
PV	electricity	3	profile1	time2	14.5
PV	electricity	3	profile1	time3	29
PV	electricity	3	profile1	time4	29
PV	electricity	3	profile1	time5	14.5
...	...	...	...	...	...
PV	electricity	5	profile4	time2	24.67
PV	electricity	5	profile4	time3	49.33
PV	electricity	5	profile4	time4	49.33
PV	electricity	5	profile4	time5	24.67



Electricity generation
profile1
profile2
profile3
profile4
Node 3
Node 4
Node 5
Supply (kWh)
200
150
100
50
0
time2
time3
time4
time5
t
i
PV

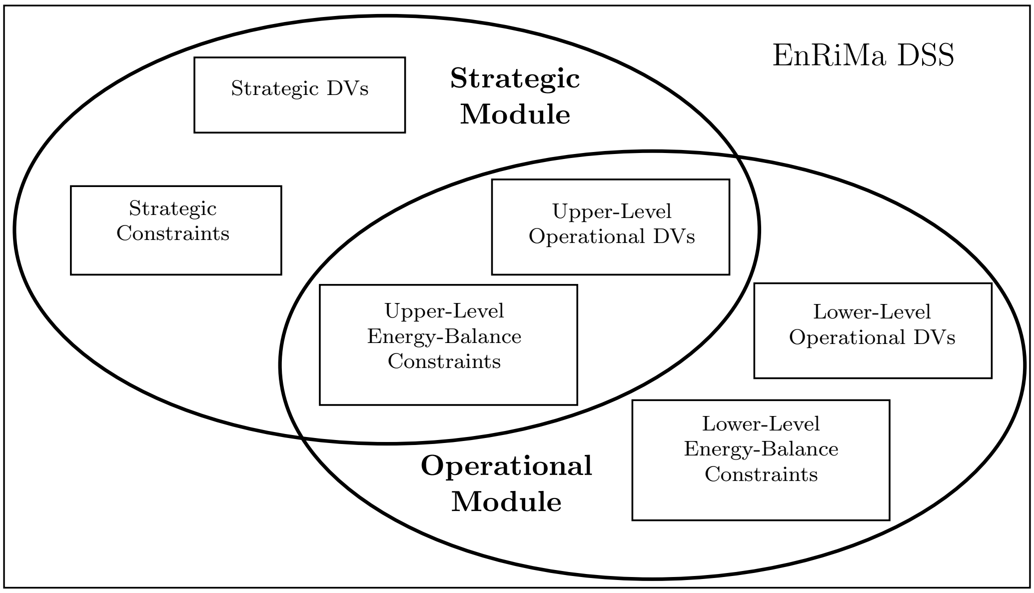


Figure 6: EnRiMa project DSS Schema.

C. Operational Upper-Level Constraints

C.1. Scope

This appendix is intended to pave the way for the integration of both strategic and operational models in the final Stochastic Optimization (Deliverable D4.6, due month 42). So far, the Solver Manager in the Stochastic Optimization prototype includes the strategic model, which will be demonstrated during the upcoming review meeting. The operational model described in D2.2 and D4.2 has been working independently using MatLab routines and can be tested through the GUI. Thus, with respect to the DSS schema outlined in both previous deliverables and reproduced in Figure 6, the strategic model covers the strategic and the upper-level operational elements, whilst the operational model covers the lower-level operational elements. Now both models are to be integrated by implementing the upper-level components in the operational model. The remaining subsections of this appendix update the operational model with the upper-level operational decision variables and constraints. Note that there may be further changes during the next stages of the project, specially after the validation of the equations. See subsection C.5 to see the notation.

C.2. Operational Constraints

Energy Balance

The energy supply must meet the energy demand. It is composed of the energy produced with energy-creating technologies plus the energy purchased in the energy market minus the energy for sale and the energy used for production.

$$\sum_{i \in \mathcal{I}_{\text{Gen}}} z^t_{i,k} - \sum_{i \in \mathcal{I}_{\text{Gen}}} y^t_{i,k} + \sum_{n \in \mathcal{N}^k_{\text{Pur}}} u^t_{k,n} - \sum_{n \in \mathcal{N}^k_{\text{S}}} w^t_{k,n} = D^t_k, \quad \forall\, k \in \mathcal{K},\, t \in \mathcal{T}_O \tag{24}$$

Output Energy

The amount of output energy produced by a given technology depends on the amount of input energy and the conversion factor.

$$z^t_{i,k'} = \sum_{k \in \mathcal{K}^i_{\text{In}}} EC_{i,k,k'} \cdot y^t_{i,k}, \quad \forall\, i \in \mathcal{I}_{\text{GenH}},\, k' \in \mathcal{K}^i_{\text{Out}},\, t \in \mathcal{T}_O \tag{25}$$

Technologies Output Limit

The energy that can be supplied by a technology is constrained by the availability of the technology and its capacity.

$$0 \le z_{i,k}^{t} \le \frac{\delta}{\eta} \cdot AF_i^t \cdot xc_i, \quad \forall\, i \in \mathcal{I}_{\text{GenX}},\, k \in \mathcal{K}_{\text{Po}}^{i},\, t \in \mathcal{T}_O \tag{26}$$

Purchasing Limit

The amount of energy that can be purchased cannot exceed the amount stipulated in the previously signed contract.

$$0 \le u_{k,n}^{t} \le \frac{\delta}{\eta} \cdot ME_{k,n} \cdot h_{k,n}, \quad \forall\, k \in \mathcal{K},\, n \in \mathcal{N}_{\text{Pur}}^{k},\, t \in \mathcal{T}_O \tag{27}$$

Sales Limit

The amount of energy that can be sold cannot exceed the amount agreed in the previously signed contract.

$$0 \le w_{k,n}^{t} \le \frac{\delta}{\eta} \cdot ME_{k,n} \cdot h_{k,n}, \quad \forall\, k \in \mathcal{K},\, n \in \mathcal{N}_{\text{S}}^{k},\, t \in \mathcal{T}_O \tag{28}$$

C.3. Conventional Heating System and HVAC Constraints

Zone Temperature Update

Eq. (29) updates the zone temperature in period t (Λ^t) based on the current zone temperature (Λ^{t-1}), the external temperature (χ^{t-1}), internal load (λ^{t-1}), and heat to be provided from both conventional (Ψ^t) and HVAC (Υ^t) systems while accounting for the building shell's characteristics.

$$\begin{aligned}\Lambda^t \;=\; & \left(\frac{1}{\frac{\gamma_{\text{air}}\cdot\rho_{\text{air}}\cdot\psi}{\delta} + \nu\cdot\alpha_{\text{wall}} + \Omega_{\text{vent}}^{t}\cdot\rho_{\text{air}}\cdot\gamma_{\text{air}}}\right)\cdot \\ & \left[\frac{\gamma_{\text{air}}\cdot\rho_{\text{air}}\cdot\psi}{\delta}\cdot\Lambda^{t-1} + \Psi^t\cdot\frac{\eta}{\delta} + \nu\cdot\alpha_{\text{wall}}\cdot\chi^{t-1}\right. \\ & +\sigma^{t-1}\cdot\epsilon\cdot\phi\cdot\alpha_{\text{glass}} + \lambda^{t-1}\cdot\alpha_{\text{floor}} \\ & \left.+\rho_{\text{air}}\cdot\gamma_{\text{air}}\cdot\Omega_{\text{vent}}^{t}\cdot\Upsilon^t\right], \forall\, t \in \mathcal{T}_O\end{aligned} \tag{29}$$

The terms inside the square brackets on the right-hand side reflect, in order, the natural zonal temperature change, the heat added by the radiator, the energy lost or gained due to the external temperature, the effect of solar gains through windows, any internal loads, and heating or cooling via the HVAC system.

Zone Temperature Bounds

Eq. (30) states that the zone temperature must lie between the lower and upper limits specified by the building manager during each short-term time period.

$$\underline{\kappa}^t \le \Lambda^t \le \overline{\kappa}^t, \forall\, t \in \mathcal{T}_O \tag{30}$$

Heat Transfer

Once heat is produced by the radiator, its effect on the zone temperature needs to be reflected. Eq. (31) determines how heat is transferred from the radiator to the air for a conventional heating system. This relationship is dependent on the desired zone temperature.

$$\Psi^t \;=\; \frac{\delta}{\eta}\cdot\xi\cdot\left(\frac{(\zeta - \Gamma^t)}{\ln\left(\frac{\zeta-\Lambda^t}{\Gamma^t-\Lambda^t}\right)}\cdot\frac{1}{\varrho}\right)^{\varphi}, \forall\, t \in \mathcal{T}_O \tag{31}$$

Here, φ and ϱ reflect the radiator's technical features: the former determines the temperature driving force for heat transfer in flow systems, while the latter describes the non-linear relation between the heat output and the mean transmission temperature (ζ) of the radiator.

Heat Exchange

The heat produced by the radiator is controlled by the rate at which heated water flows through it. Eq. (32) reflects how heat is exchanged inside the radiator and, thus, is a function of the water's flow rate, Ω^t_{water}.

$$\Psi^t \quad = \quad \frac{\delta}{\eta} \cdot \Omega^t_{\text{water}} \cdot \rho_{\text{water}} \cdot \gamma_{\text{water}} \cdot \left(\zeta - \Gamma^t\right), \forall\ t \in \mathcal{T}_O \tag{32}$$

Here, it is assumed that the supply-water temperature, ζ, is constant.

Heat Demand

After the radiator's operations are modeled, the demand for heat can be calculated. Eq. (33) determines the heat required inside the boiler ($D^t_{\text{space heat}}$) to change the temperature of water from the current return-water temperature (Γ^{t-1}) to the required supply-water temperature (ζ) for the period.

$$\begin{aligned} D^t_{\text{space heat}} \quad &= \quad \frac{\delta}{\eta} \cdot \Omega^t_{\text{water}} \cdot \rho_{\text{water}} \cdot \gamma_{\text{water}} \cdot \left(\zeta - \Gamma^{t-1}\right), \\ \forall\ t \in \mathcal{T}_O \end{aligned} \tag{33}$$

Heating System Capacity Limits

We set bounds on the capacity of the conventional heating system. Eq. (34) states the return-water temperature for each period cannot exceed the supply-water temperature and must be greater than the zone temperature for this period.

$$\Lambda^t \leq \Gamma^t \leq \zeta, \forall\ t \in \mathcal{T}_O \tag{34}$$

Heat Demand Bound

Eq. (35) restricts the maximum heat that can be demanded in any time period.

$$D^t_{\text{space heat}} \leq \iota, \forall\ t \in \mathcal{T}_O \tag{35}$$

Water Flow Rate Limits

Eq. (36) constrains the water flow rate in each period.

$$\underline{\mu}_{\text{water}} \leq \Omega^t_{\text{water}} \leq \overline{\mu}_{\text{water}}, \forall\ t \in \mathcal{T}_O \tag{36}$$

AHU Suppy-Air Temperature

Turning to the HVAC system, we reflect its temperature settings based on the external temperature. Eq. (37) describes the setting of the supply-air temperature for the HVAC's air-handling unit (AHU), which is a piece-wise linear function in case of ventilation with cooling.

$$\begin{aligned} \Upsilon^t \quad &= \quad \begin{cases} \Phi^t \cdot \chi^{t-1} + \left(1 - \Phi^t\right) \cdot \Lambda^{t-1} & \text{vent only} \\ \overline{\varsigma} & \text{cool \& } \chi^{t-1} < \underline{\chi} \\ \overline{\varsigma} + \left(\frac{\underline{\varsigma} - \overline{\varsigma}}{\overline{\chi} - \underline{\chi}}\right) \cdot \left(\chi^{t-1} - \underline{\chi}\right) & \text{cool \& } \underline{\chi} \leq \chi^{t-1} < \overline{\chi} \\ \underline{\varsigma} & \text{cool \& } \overline{\chi} \leq \chi^{t-1} \end{cases} \\ &\forall\ t \in \mathcal{T}_O \end{aligned} \tag{37}$$

Cooling Demand

Eq. (38) calculates the cooling demand for each period as the energy required to bring the temperature of the return air from the AHU mixed with the external air to the supply-air temperature.

$$\begin{aligned} D^t_{\text{cooling}} \quad &= \quad \Omega^t_{\text{vent}} \cdot \rho_{\text{air}} \cdot \gamma_{\text{air}} \cdot \frac{\delta}{\eta} \cdot \left(\Phi^t \cdot \chi^{t-1} \right. \\ &\left. + \left(1 - \Phi^t\right) \cdot \Lambda^{t-1} - \Upsilon^t\right), \forall\ t \in \mathcal{T}_O \end{aligned} \tag{38}$$

AHU Electricity Requirement

Depending on the operations of the HVAC system, a significant amount of electricity may be consumed. Eq. (39) determines the electric energy needed to meet either the ventilation or cooling requirements during each period depending on the kind of HVAC system installed.

$$y^t_{\text{HVAC,electricity}} = \begin{cases} \omega \cdot \Omega^t_{\text{vent}} & \text{vent only} \\ E_{\text{HVAC,electricity,cooling}} \cdot z^t_{\text{HVAC,cooling}} & \text{cooling} \end{cases} , \forall\ t \in \mathcal{T}_O \tag{39}$$

AHU External Air Limits

As with the conventional radiator, we impose technical limits on the AHU. First, Eq. (40) constrains the proportion of external air taken in by the AHU during each period.

$$\underline{\tau} \leq \Phi^t \leq \overline{\tau}, \forall\ t \in \mathcal{T}_O \tag{40}$$

AHU Air-Flow Rate Limits

Next, Eq. (41) constrains the AHU's air-flow rate during each period.

$$\underline{\mu}_{\text{vent}} \leq \Omega^t_{\text{vent}} \leq \overline{\mu}_{\text{vent}}, \forall\ t \in \mathcal{T}_O \tag{41}$$

C.4. Objective Function

Given Eqs. (29) through (41), an operational optimization problem may be formulated for meeting each site's temperature requirements given the existing building shell, energy equipment configuration and selected energy tariffs. The aim is to minimize the total energy trading costs

$$\text{Minimize} \sum_{t \in \mathcal{T}_O} \left\{ \sum_{k \in \mathcal{K}} \left(\sum_{n \in \mathcal{N}^k_{\text{Pur}}} PP^t_{k,n} \cdot u^t_{k,n} - \sum_{n \in \mathcal{N}^k_{\text{S}}} SP^t_{k,n} \cdot w^t_{k,n} \right) + \sum_{i \in \mathcal{I}_{\text{Gen}}} \sum_{k \in \mathcal{K}^i_{\text{Out}}} CO_{i,k} \cdot z^t_{i,k} \right\} \tag{42}$$

C.5. Notation

The following symbols are used in this appendix:

Sets

$\mathcal{I}$	technologies;
$\mathcal{I}_{\text{Gen}} \subset \mathcal{I}$	energy-generating technologies;
$\mathcal{I}_{\text{GenH}} \subset \mathcal{I}$	energy-generating technologies excluding HVAC systems;
$\mathcal{I}_{\text{GenX}} \subset \mathcal{I}$	energy-generating technologies excluding conventional heating and HVAC systems;
$\mathcal{K}$	energy types;
$\mathcal{K}^i_{\text{In}} \subset \mathcal{K}$	input energy types for technology $i \in \mathcal{I}_{\text{Gen}}$;
$\mathcal{K}^i_{\text{Out}} \subset \mathcal{K}$	output energy types for technology $i\mathcal{I}_{\text{Gen}}$;
$\mathcal{K}^i_{\text{Po}} \subset \mathcal{K}$	principal output energy type for technology $i\mathcal{I}$;
$\mathcal{N}$	energy tariffs;
$\mathcal{N}^k_{\text{Pur}} \subset \mathcal{N}$	purchase tariffs for energy type k;
$\mathcal{N}^k_{\text{S}} \subset \mathcal{N}$	sales tariffs for energy type k;
$\mathcal{T}_O$	short-term time periods.

Time

$\delta = 3600$	length of operational decision-making period (s);
$\eta = 3600$	number of seconds in an hour (s/h);

Physical Constants and Parameters

$\gamma_{\text{water}} = 4.1855$	specific heat capacity of water (kJ/(kg·K));
$\rho_{\text{water}} = 998.2071$	density of water (kg/m^3);
$\gamma_{\text{air}} = 1.0189$	specific heat capacity of air (kJ/(kg·K));
$\rho_{\text{air}} = 1.20$	density of air (kg/m^3).

Environmental Parameters

χ^t	external temperature during short-term period $t \in \mathcal{T}_O$ (°C);
σ^t	solar gains (weighted average over different wall directions) during short-term period $t \in \mathcal{T}_O$ (kW/m^2).

Building Parameters

ψ	volume of the zone (m^3);
ν	heat transition coefficient of the wall (kW/(m^2·K));
α_{wall}	heat transfer area of the wall (m^2);
α_{glass}	total area of windows (m^2);
ϵ	mean energy transmission coefficient of glass (unitless);
ϕ	mean sun protection factor of all components of the thermal envelope of building (unitless);
α_{floor}	area of the floor of the zone (m^2);
λ^t	internal load (from people, lighting, working machines, etc.) per area of the zone during short-term period $t \in \mathcal{T}_O$ (kW/m^2);
$\underline{\kappa}^t$	lower limit for the required zone temperature during short-term period $t \in \mathcal{T}_O$ (°C);
$\overline{\kappa}^t$	upper limit for the required zone temperature during short-term period $t \in \mathcal{T}_O$ (°C).

Heating System Parameters

ζ	supply-water temperature at the radiator inlet (°C);
ι	maximum amount of heat that can be provided by the conventional heating system in any given short-term time period (kWh);
$\overline{\mu}_{\text{water}}$	maximum water flow rate in radiator (m^3/s);
$\underline{\mu}_{\text{water}}$	minimum water flow rate in radiator (m^3/s);
ξ	mean nominal heat transfer capacity of all radiators installed (kW);
φ	radiator coefficient (unitless);
ϱ	mean logarithmic temperature difference (K, which is equivalent to °C since we are referring to a temperature difference).

HVAC System Parameters

$\overline{\mu}_{\text{vent}}$	maximum air flow rate of the heating, ventilation, and air conditioning (HVAC) system (m^3/s);
$\underline{\mu}_{\text{vent}}$	minimum air flow rate of the HVAC system (m^3/s);
$\underline{\tau}$	lower limit of the proportion of air that may be taken externally;
$\overline{\tau}$	upper limit of the proportion of air that may be taken externally;
$E_{\text{HVAC,electricity,cooling}}$	electricity required by the HVAC system to produce one unit of cooling (kWh_e/kWh);
ω	electricity required to pump the air at a given flow rate ($\text{kWh}_e/(\text{m}^3/\text{s})$);
$\overline{\chi}$	external temperature limit at which the air-handling unit (AHU) performs cooling (°C);
$\underline{\chi}$	external temperature limit at which the AHU performs heating (°C);
$\overline{\varsigma}$	AHU's supply-air temperature for heating (°C);
$\underline{\varsigma}$	AHU's supply-air temperature for cooling (°C).

Technology Parameters

AF_i^t	availability factor for technology $i \in \mathcal{I}$ during short-term time period $t \in \mathcal{T}_O$.
$CO_{i,k}^t$	technology operation cost for technology $i \in \mathcal{I}_{\text{Gen}}$ and energy type $k \in \mathcal{K}_{\text{Out}}^i$ (€/kWh).
$EC_{i,k,k'}$	amount of output energy $k' \in \mathcal{K}_{\text{Out}}^i$ generated by technology $i \in \mathcal{I}_{\text{Gen}}$ from one unit of input energy $k \in \mathcal{K}_{\text{In}}^i$ (kWh/kWh).
xc_i	available capacity of technology $i \in \mathcal{I}$ (kW).

Energy Market Parameters

$h_{k,n}$	existing contracts of type $n \in \mathcal{N}_{\text{Pur}}^k \cup \mathcal{N}_{\text{S}}^k$ for energy type $k \in \mathcal{K}$.
$ME_{k,n}$	maximum purchase/sale of energy $k \in \mathcal{K}$ allowed under a given contract $n \in \mathcal{N}_{\text{Pur}}^k \cup \mathcal{N}_{\text{S}}^k$ (kWh).
$PP_{k,n}^t$	price of energy type $k \in \mathcal{K}$ purchased under tariff $n \in \mathcal{N}_{\text{Pur}}^k$ during short-term time period $t \in \mathcal{T}_O$ (€/kWh).
$SP_{k,n}^t$	price of energy type $k \in \mathcal{K}$ sold under tariff $n \in \mathcal{N}_{\text{S}}^k$ during short-term time period $t \in \mathcal{T}_O$ (€/kWh).

State Variables

Γ^t	return-water temperature at the outlet of the radiator during short-term time period $t \in \mathcal{T}_O$ (°C).

Decision Variables

Λ^t	zone temperature during short-term time period $t \in \mathcal{T}_O$ (°C);
Ω^t_{water}	flow rate of water to conventional heating system during short-term time period $t \in \mathcal{T}_O$ (m^3/s);
Ω^t_{vent}	flow rate of air to HVAC system during short-term time period $t \in \mathcal{T}_O$ (m^3/s);
Υ^t	supply-air temperature from the HVAC system's AHU during short-term time period $t \in \mathcal{T}_O$ (°C);
Φ^t	fraction of external air used by the AHU during short-term time period $t \in \mathcal{T}_O$;
Ψ^t	heat from radiator during short-term time period $t \in \mathcal{T}_O$ (kWh);
D^t_k	demand for end-use energy type $k \in \mathcal{K}$ during short-term time period $t \in \mathcal{T}_O$ (kWh);
$u^t_{k,n}$	amount of energy of type $k \in \mathcal{K}$ purchased under tariff $n \in \mathcal{N}^k_{\text{Pur}}$ during short-term time period $t \in \mathcal{T}_O$ (kWh).
$w^t_{k,n}$	amount of energy of type $k \in \mathcal{K}$ sold under tariff $n \in \mathcal{N}^k_{\text{S}}$ during short-term time period $t \in \mathcal{T}_O$ (kWh).
$y^t_{i,k}$	requirement of energy type $k \in \mathcal{K}^i_{\text{In}}$ as input to energy-creating technology $i \in \mathcal{I}_{Gen}$ during short-term time period $t \in \mathcal{T}_O$ (kWh).
$z^t_{i,k}$	amount of energy of type $k \in \mathcal{K}^i_{\text{Out}}$ produced by energy-creating technology $i \in \mathcal{I}_{Gen}$ during short-term time period $t \in \mathcal{T}_O$ (kWh).