

A Flexible GIS-based Computational Framework for the Early-Design of District Energy Networks

Félix Pouchain¹, Giuseppe Peronato¹ and Guillaume Meunier²

Eliothe (Egis Concept), ¹Lyon / ²Montreuil (France)

Abstract

The design of district heating and cooling (DHC) networks is crucial to increase the pooling of energy supply and demand between individual buildings, and thus to reduce the environmental and financial costs of the energy systems. DHC networks are also useful to increase the share of renewable energy sources in cities, their integration being more challenging in high-density environments. This paper presents a novel simulation framework for the optimization of building energy systems connected to a district heating and cooling network. The developed method is based on the open-source Python library PyPSA and is adapted for the early-design exploration of multiple scenarios and their optimization, based on GIS input data. We show the application of the proposed method into a fictitious district in France composed of mixed-use buildings. The results compare two scenarios of energy systems minimizing either greenhouse gas emissions or the energy cost.

Keywords: urban building energy modeling, district heating and cooling (DHC) networks;

1. Introduction

There is a growing interest towards urban energy simulations. Many of these tools fall into the category of urban building energy modeling (Cerezo Davila et al., 2016; Johari et al., 2020) and are generally aimed at the simulation of the building energy needs. Some tools tackle multiple scopes including also the simulation of the building energy systems and of the district energy networks (Fonseca et al., 2016).

We present here a flexible framework for the simulation and optimization of building energy systems and energy sources (including local photovoltaic modules) connected into a district heating and cooling (DHC) network. The simulation framework is constituted of different Python modules, each one performing a specific simulation task while being integrated into a single automated workflow based on the Luigi library (Bernhardsson and Freider, 2019). The optimization routine provides a sizing of the energy systems in the grid in order to optimize different environmental, energy or financial indicators, including both capital and operational expenditures.

Compared to existing urban energy simulation tools, the proposed framework is not bundled as a plugin to existing GIS or CAD modeling tools, but rather conceived as a software-agnostic workflow focusing on the data integration and optimization using open-source Python libraries. The exchange of input/output data between the different modules is performed using standardized formats. This eases the replacement of single modules with other external programs (e.g., a different building performance simulation tool), which could be possibly more adapted to solve specific design situations.

At the current development stage, the framework is particularly adapted for the early-design exploration of energy scenarios of new urban developments, where different design variants can be simulated and compared, with only minimal data input requirements. This is the case of the sample case-study application presented in this paper, where a small fictitious district is used to show the main inputs and outputs of the proposed simulation framework. In particular, two different scenarios minimizing either the environmental (CO₂ emissions) or the financial cost of the district operational energy are evaluated. For each scenario, the framework is used to pre-size the energy systems and the pipework, as well as to simulate the hourly energy demand. This information can be used, for example, to perform a preliminary check of the project compliance with environmental and/or financial objectives.

2. Methodology

The proposed framework is composed of six modules bundled as Python packages, which are exchanging data through file-based databases, and form all together a District Energy Modeling tool:

- Weather: translating weather files (EPW and French RT2012 database) into a standard tabular format;
- Geometry: creation of a 3D model from a 2.5D vector GIS model and calculation of the sky view maps for each building surface using the POV-Ray (Cason, 2013) simulation engine;
- Solar: calculation of the PV solar potential based on the PVLIB library (Holmgren et al., 2019, 2018);
- Climelieth: a dynamic building performance simulation tool based on the French building performance regulations (RT, 2012);
- GeoCAD: designing and sizing the district heating & cooling network based on the GeoPandas (Jordahl et al., 2019) and Networkx (Hagberg et al., 2019) libraries;
- Smartgrid: tool to size the energy systems by minimizing either the GHG emissions, the financial cost, the primary energy consumption or a combination of the three metrics, based on the PyPSA toolbox (Brown et al., 2018) and the CBC solver (Forrest et al., 2018).

Because of the use of separate Python modules performing tasks based on a command-line input and standardized files, the components can be easily replaced with other simulation engines, adapted to use different file formats or to solve specific design situations. Building energy needs can for example be computed using IES VE, EnergyPlus, or any equivalent BPS tool, before being used in SmartGrid. It is thus possible to integrate any design parameter available in these tools in the study and design of urban energy systems.

In the next sections, the GeoCAD and SmartGrid modules, composing the actual DHC simulation part of the proposed framework, as well as the workflow management system, will be presented in more detail.

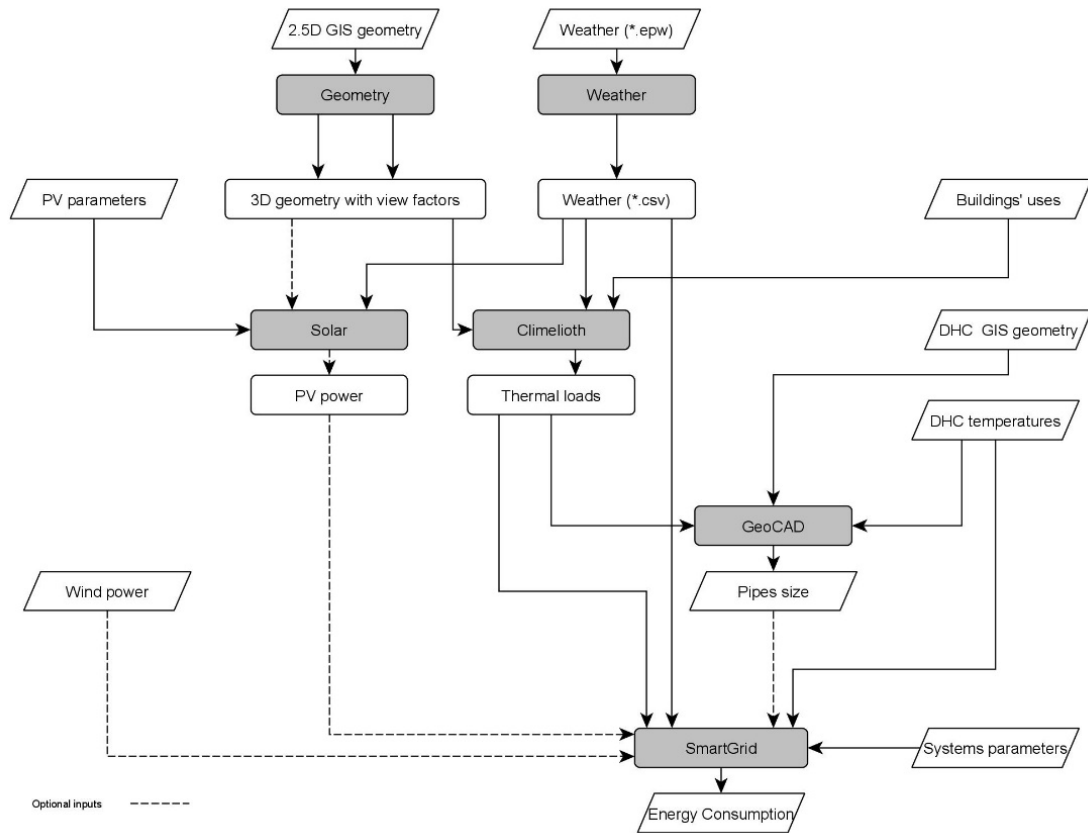


Fig. 1: Flowchart of the proposed simulation framework

3. Workflow Management System

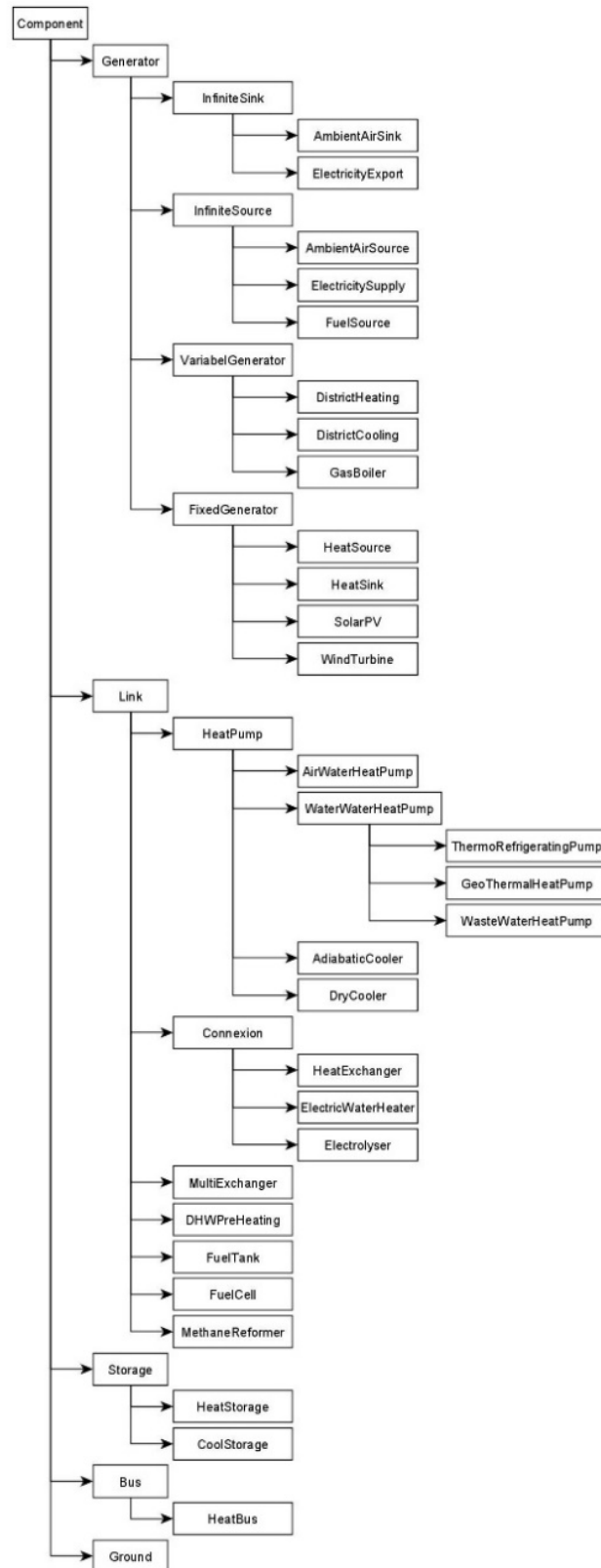


Fig. 1: Tree diagram representing the inheritance of the Python classes of the Smartgrid module.

A Python module named “Workflow” and based on the Luigi library (Bernhardsson and Freider 2012) is used as Workflow Management System (WMS) to organize the different simulation modules into a comprehensive and automated workflow.

When running a simulation, the Workflow module checks whether the input data already exist or have to be simulated and, in the latter case, run the modules that are required to produce the missing data inputs. This allows the user to run, for instance, the SmartGrid module, which is at the bottom of the workflow chain shown in Fig.1, with the Workflow module taking care of all missing dependencies in case they have not been already simulated.

4. The SmartGrid module

The SmartGrid module is a tool to size the energy systems by minimizing either the GHG emissions, the financial cost, the primary energy consumption or a combination of the three metrics. It is at the bottom of the dependency chain shown in Fig.1 and hence at the core of the developed workflow. It can also be run independently from the other tools, when its mandatory inputs (notably, the building energy needs and network characteristics) are known.

The tool’s application goal is twofold: a) modeling the energy systems and pipe network at an hourly timestep to cover the building energy needs, b) finding the combination and size of energy systems that minimize the capital and operational expenditures, in terms of primary energy, financial cost, or GHG emissions. The optimization is conceived as a single-objective problem, while multiple objectives can be weighted into a single one.

The module extends the Python package PyPSA (Brown et al., 2018), which provides basic energy systems components (generator, load, storage, link...) and the ability to optimize power flows given their fixed and marginal costs of use. SmartGrid builds upon these components to provide common buildings and district heating systems (boiler, heat pump, solar panels...). It also provides multiple helper functions, to ease the creation of large systems and the access to simulation results.

As in electricity grid simulation tools, the overall modeling concept revolves around the concepts of buses, links, sources and sinks, while the energy flux, in addition to electricity, can be also of thermal energy. The buildings are typically the sinks of the energy fluxes, while for cooling it is the opposite: the buildings are the sources of the heat waste that is either “sunk” in the environment (modeled here as an infinite sink), recovered by another building connected to the loop or injected into an external district cooling network. Sources and sinks are attached to a bus and the different buses are connected through simple connection links or energy transformation links, which can convert energy between its different forms at a given efficiency rate. Cooling and heating buses are provided with a temperature attribute, which is used to check that the connection is done at the same temperature level in case of simple connection links. Otherwise, the user is prompted to replace the simple connection with an energy transformation link, such as a heat pump, to raise or decrease the temperature to the same level.

The optimization solver implemented into the module solves the problem of the linear optimal power flow within the district energy network, finding the least-cost solution. The cost is expressed in terms of primary energy, financial cost, or GHG emissions and is composed of three components: investment, operational and marginal.

4.1 Programming paradigm

This module is coded using an Object-Oriented Programming (OOP) paradigm, where each physical component of the DHC network is a Python class.

The attributes of the generic components (bus, generator, load, link and storage) are inherited by the sub-objects and specified with other characteristics (Fig. 1). Each component is based on one or several PyPSA components (StorageUnit, Link, Generator, Bus).

When running a simulation, a SmartGrid object is initialized and the following networks are added:

- *RES*, the network including off-site Renewable Energy Sources, such as large wind power turbines or solar farms.
- *Environment*, used as a source/sink for the air- and ground- heat pumps and chillers. This network includes three thermal buses: *ambient_air_source*, *ambient_air_sink* and *ground*. The *ground* bus is connected to a storage, which allows the user to define a cyclic storage unit to balance the extractions and injections over the simulation period (for example over the year), while the *ambient air source* and *sink* correspond to an infinite source and sink.
- *Network*, the main network to which the shared energy systems are added.
- *Building*, an internal network for each building connected to the district network, to which the building loads are by default connected to, as well as possible building energy systems.

The connections between these networks are defined by the user in the input system parameters file.

4.2 Models

We will briefly describe the main models included in the SmartGrid and GeoCAD modules.

Heat pumps are modeled as PyPSA links connecting two input sources (electricity and a heat bus acting) to an output sink bus, where the efficiency is defined by a variable COP and the nominal electrical power. We use the COP models from (Staffell et al., 2012). The nominal electrical power necessary for the PyPSA model is obtained by dividing the nominal thermal power (defined by the used in the input systems parameters file) by a fixed COP. Refrigerating machines can be also modeled as a heat pump, with a source corresponding to the cooling network from which the calories are extracted. A geothermal heat pump can also be modeled and connected to the *ground bus*, whose temperature can be given in the input weather file or modeled.

The fuel cell is modeled as a cogeneration, which takes as input a dihydrogen flux, and gives as outputs two fluxes: electricity and thermal energy, both proportional to the ingoing dihydrogen flux. The considered efficiencies, corresponding to those of a commercial product, are the following ones:

$$\frac{\text{Heating+Electricity}_{out}}{\text{Hydrogen}} = 0.95$$

$$\frac{\text{Electricity}_{out}}{\text{Hydrogen}} = 0.45$$

The energy losses in the piping networks are calculated by a simple Heat Transfer Model, considering the thermal resistance of the pipe (steel and insulation layer), and the average temperature in the thermal loop.

4.3 Input and output

The systems parameters is the module's specific input file, where all energy systems and their parameters are allocated to the district and the building buses. It is structured as a *.csv file and provides the information about the connections of the different energy systems to the buses as well as some of their main design parameters, such as, for example, the nominal power and efficiency.

The input building energy network is in the form of a GeoJSON file, which can be created with the GeoCAD module, containing the geometry and information of the network segments (Origin, Destination, Nominal Diameter). Additional JSON properties are appended to the network input file, to define the depth and soil type, as well as the temperature ranges of all subnetworks (district-wide and building ones). This latter information is compulsory to run the simulation, while the geometry information is only needed to define piping-dependent thermal losses.

The simulation output including the origin, destination, transformation of all hourly energy fluxes is saved into a binary file that can be read through the SmartGrid module. A summary Excel file (including total energy and maximum power) and Sankey diagram are also automatically saved at each simulation.

5. The GeoCAD module

The GeoCAD module can be used for pre-sizing the pipework of heating and cooling networks. The size of the pipes is in fact required to calculate the network losses in the SmartGrid module.

The nominal diameter DN of the pipework components is sized based on the maximum thermal loads in each segment of the network. To this end, the maximum load of each building and each network connected to the network is chosen. This corresponds to a conservative approach, as the buildings are unlikely to have the maximum thermal load simultaneously. The system can be also forced to use the loads at a specific time step as the sizing factor.

The module uses input GIS files that can be created with any GIS software to define the network geometry and the directional flows of the pipes. The GIS files define the buildings (drawn as Points), as well as the networks (drawn as LineStrings) connecting the buildings. The points from where the line starts will be considered as the source of the energy flow, i.e. the position of the district thermal station.

The mandatory attributes for the network geometry are the direction of the flow (supply, return) and the type of network (heating, cooling, medium temperature). However, only the supply pipes are actually considered at this stage, the energy losses in the SmartGrid module being calculated on the average temperature of the supply/return pipes and on the double distance of the supply pipes.

The energy flows are balanced in each network segment using the network simplex algorithm as implemented in the Networkx library (Hagberg et al., 2019).

6. Case-study application

The tool has been already applied in several neighborhood-scale projects at the early-design phase. However, for the scope of this paper, we apply it in a fictitious mixed-use district in Lyon (France) composed of three buildings, which are connected through a local DHC low-temperature network. All systems are centralized in a district thermal station, while a Water-Water Heat Pump is installed in each building to raise the water temperature for domestic hot water. Photovoltaic panels are also installed in one building.

6.1 Description of the case study

The mixed-use district is composed of three buildings named A, B and C, which share the energy systems and are connected through a microgrid (Fig. 2). The energy systems are listed in Tab. 1.

All systems are installed in Building A, while the Water-Water Heat Pump is installed in each building to raise the water temperature and connects hence the heating network to the DHW network. The simulation considers the

climate of the H1c zone, including the city of Lyon, from the French building performance regulations (RT, 2012).

Tab. 1: Characteristics of the buildings of the case-study neighborhood: summary of the inputs of the SmartGrid simulation

	A	B	C	Total
Floor area (m ²)	3011	4017	2553	9581
Housing/Office/Retail (% _{floor area})	0/83/17%	0/88/12%	80/0/20%	21/63/15%
Heating needs (kWh/m ² _{floor area})	8.2	12.5	6.6	9.6
Cooling needs (kWh/m ² _{floor area})	39.7	42.0	7.2	32.0
DHW needs (kWh/m ² _{floor area})	1.8	1.9	14.5	5.3
Electricity needs (kWh/m ² _{floor area})	14.2	9.0	14.0	11.3
PV generation (kWh)	1 219	-	-	1 219
Heating network temperature (°C supply/return)	47/39	47/39	47/39	47/39
Cooling network temperature (°C supply/return)	8/15	8/15	8/15	8/15
DHW network temperature (°C supply/return)	55/50	55/50	55/50	55/50

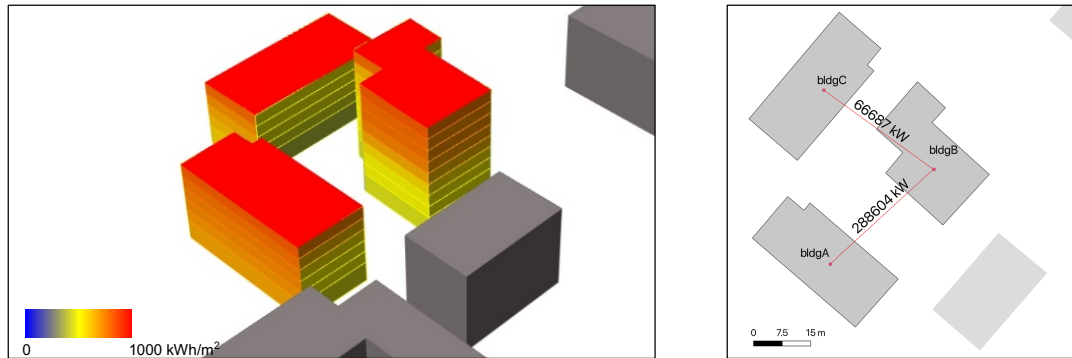


Fig. 2: False color visualization (left) of solar irradiation (based on Climelieth simulation) and map of the max heating power in each segment of the DHC network (based on the GeoCAD simulation)

6.2 Results

We present the results of a study of two simulation scenarios using the same energy sources and systems, while optimizing their use for either minimizing the Environmental Cost (expressed in gCO₂/kWh_{Final Energy}) or Financial Cost (expressed in Euros/kWh_{Final Energy}). In both cases, we will consider only the marginal cost per unit of energy consumption (kWh) over one year, without considering the investment or operational costs, as presented in the Appendix. However, the SmartGrid module includes the support for power-based capital costs (x/kW) in the simulations.

Tab. 2 shows the size (maximum power) and operating hours per each energy system. The maximum power in some cases exceeds the nominal power, because of the definition of PyPSA efficiency based on a fixed COP (see §4.2). In these cases, the remaining power need is provided by an alternative, yet more expensive, energy source, which is here modeled as “infinite”.

In Fig. 3, we can see the monthly energy consumption for the two scenarios. The energy mix is clearly changing between the two scenarios. In the second variant, we notice in particular the use of a natural gas boiler and a chiller as cheaper, yet with a higher CO₂-equivalent content, alternatives to using external district heating and, for most hours, cooling.

Tab. 2: Sizing results of the energy systems for the two optimization goals: Variant 1 – Minimization of CO₂-equivalent emissions (top) and Variant 2 – Minimization of Energy Financial Cost

Systems	Nominal power [kW]	Variant 1		Variant 2	
		Max Power [kW]	Annual operating hours [%]	Max Power [kW]	Annual operating hours [%]
GasBoiler	Infinite	-	-	365	19
DistrictHeating <i>DH</i>	Infinite	329	8	-	-
DistrictCooling <i>DC</i>	Infinite	739	2	-	-
Thermorefrigeratingpump <i>TRP</i>	50	23	36	23	39
GeoThermalHeatPump (heating)	25	24	73	24	71
GeoThermalHeatPump (cooling)	25	37	6	37	6
AirWaterHeatPump (heating) <i>HeatPump</i>	50	42	19	3	0.1
AirWaterHeatPump (cooling) <i>Chiller</i>	Infinite	644	18	738	19
WaterWaterHeatPump (heating) <i>HeatPump_DHW</i>	Infinite	28	62	28	62

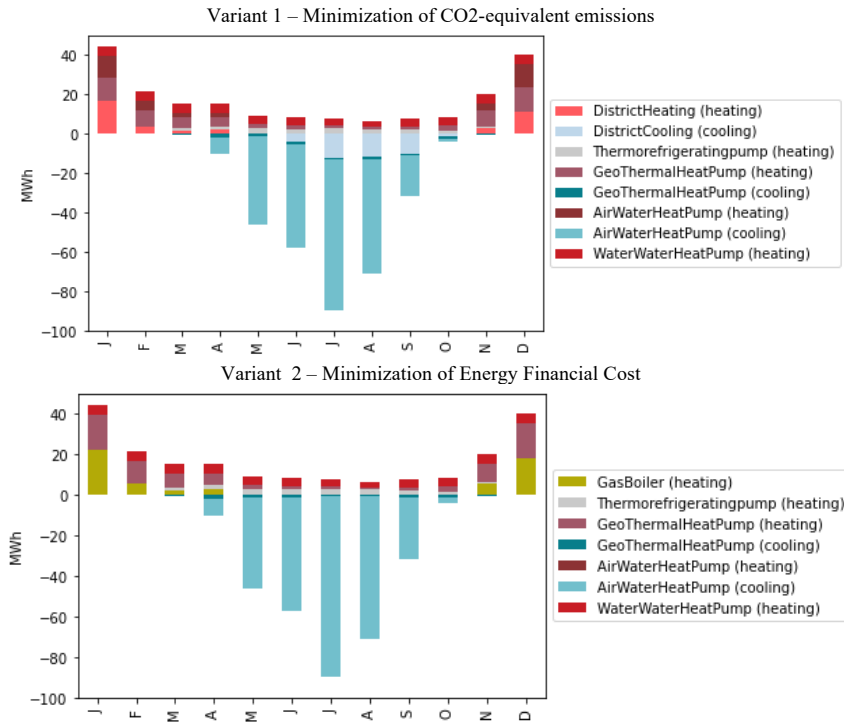


Fig. 3: Monthly thermal energy generation by systems (cooling is represented with negative values)

The Sankey plot representing the annual energy fluxes is also an output of the tool. In Fig. 4, we can see a simplified version, where the original networks and components used in the PyPSA simulation have been simplified to remove unnecessary nodes and the fluxes of building A, B, C have been represented as part of the same building. It can be seen how the heating needs are satisfied in both scenarios with a very diverse energy mix, whose main component is, in Variant 1, the geothermal heat pump.

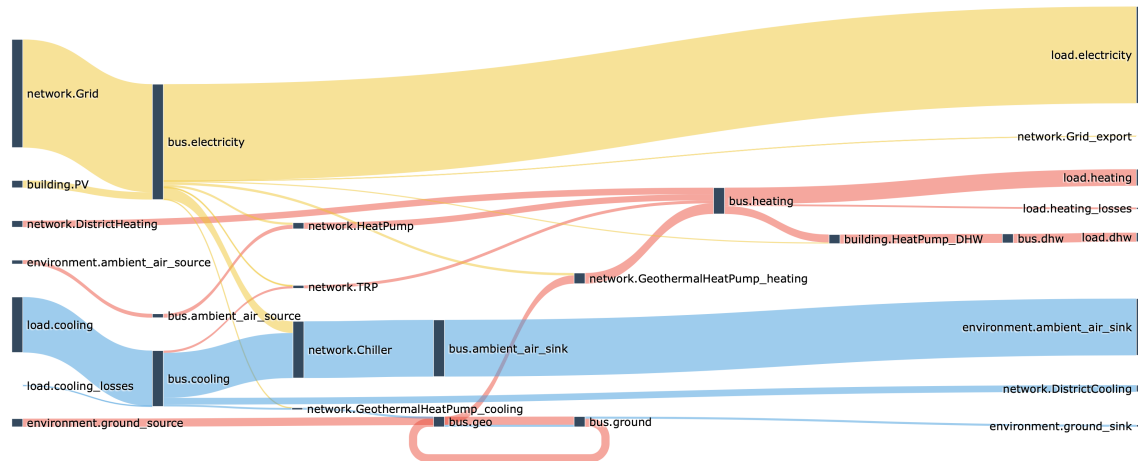


Fig. 4: Simplified Sankey representation of the annual energy fluxes for Variant 1 – Minimization of CO₂-equivalent emissions.
The height of each node is proportional to the energy flux.

7. Limitations and future work

The framework we have presented is still under development, and the available components are limited. The integration into the workflow of state-of-the-art energy simulation tools such as EnergyPlus is currently being tested.

Regarding the case-study evaluation presented in this paper, it should be noted that we only considered here environmental and marginal costs for the considered energy sources, while the results could change when analyzing the life-cycle costs.

8. Conclusions

This paper presented a simulation framework for the early simulation of DHC networks. The purpose of this tool is to study the impact of different energy mixes and energy systems on the performance of the network. It also allows a pre-sizing of the pipework and of the energy systems.

We have shown a sample application of this tool, simulating energy scenarios for a DHC network and optimizing its systems given environmental and financial objectives. Thanks to its flexible and platform-agnostic conception and its integration is a workflow automation tool, we argue that this framework is particularly adapted for the exploration of early design scenarios. The tool's source code is released under the GLPv3 license in a Git repository at this address: <https://gitlab.com/elieth/DESSim>

9. Acknowledgments

The software development presented in this work has been initiated by FP and further conducted by FP, GP and Arnaud Sanson, as part of different projects at Elioth. The authors would like to thank the Egis Group for providing funding to Elioth's "Le lab" R&D activities.

References

- Bernhardsson, E., Freider, E., 2019. Spotify/Luigi. Spotify, Stockholm, Sweden. <https://github.com/spotify/luigi>
- Brown, T., Hörsch, J., Schlachtberger, D., 2018. PyPSA: Python for Power System Analysis. J. Open Res. Softw. 6. <https://doi.org/10.5334/jors.188>

- Cason, C., 2013. POV-Ray. <http://povray.org>
- Cerezo Davila, C., Reinhart, C.F., Bemis, J.L., 2016. Modeling Boston: A workflow for the efficient generation and maintenance of urban building energy models from existing geospatial datasets. *Energy* 117, 237–250. <https://doi.org/10.1016/j.energy.2016.10.057>
- Fonseca, J.A., Nguyen, T.-A., Schlueter, A., Marechal, F., 2016. City Energy Analyst (CEA): Integrated framework for analysis and optimization of building energy systems in neighborhoods and city districts. *Energy Build.* 113, 202–226. <https://doi.org/10.1016/j.enbuild.2015.11.055>
- Forrest, J., Ralphs, T., Vigerske, S., LouHafer, Kristjansson, jpfasano, EdwinStraver, Lubin, M., Gambini Santos, H., rlougee, Saltzman, M., 2018. coin-or/Cbc. Zenodo. <https://doi.org/10.5281/zenodo.1317566>
- Hagberg, A., Schult, D., Swart, P., 2019. networkx/networkx. NetworkX: Python software for complex networks.
- Holmgren, W., Calama-Consulting, Lorenzo, T., Krien, bmu, Mikofski, M., Hansen, C., DaCoEx, konstant_t, Anton Driesse, mayudong, Leland Boeman, Ed Miller, Heliolytics, Marc A. Anoma, jforbess, pyElena21, Volker Beutner, Todd Hendricks, MLEEFs, Johannes Dollinger, Cedric Leroy, Cameron Stark, Kevin Anderson, Jonathan Gaffiot, Johannes Oos, Giuseppe Peronato, Birgit Schachler, Alan Mathew, Alaina Kafkes, 2019. pvlib/pvlib-python. Zenodo. <https://doi.org/10.5281/zenodo.2850192>
- Holmgren, W., Hansen, C., Mikofski, M., 2018. pvlib python: a python package for modeling solar energy systems. *J. Open Source Softw.* 3, 884. <https://doi.org/10.21105/joss.00884>
- Johari, F., Peronato, G., Sadeghian, P., Zhao, X., Widén, J., 2020. Urban Building Energy Modeling: State of the Art and Future Prospects. *Renew. Sustain. Energy Rev.* 128, 109902. <https://doi.org/10.1016/j.rser.2020.109902>
- Jordahl, K., Van den Bossche, J., Wasserman, J., McBride, J., Gerard, Tratner, J., Perry, M., Farmer, C., Sean Gillies, Micah Cochran, Matt Bartos, Martin Fleischmann, Lucas Culbertson, Nick Eubank, maxalbert, Aleksey Bilogur, Geir Arne Hjelle, Dani Arribas-Bel, Christopher Ren, Sergio Rey, Martin Journois, Levi John Wolf, Nick Grue, Joshua Wilson, Ömer Özak, YuichiNotoya, 2019. geopandas/geopandas. Zenodo. <https://doi.org/10.5281/zenodo.3333010>
- RT, 2012. Réglementation Thermique 2012. <http://rt-batiment.fr>
- Staffell, I., Brett, D., Brandon, N., Hawkes, A., 2012. A review of domestic heat pumps. *Energy Environ. Sci.* 5, 9291–9306. <https://doi.org/10.1039/C2EE22653G>

Appendix

Tab. 1: Factors for the energy sources

	Grid	DH	DC	PV	Natural gas
CO ₂ intensity (gCO ₂ eq/kWh _{Final Energy})	59 ^a	98 ^b	16 ^b	0 ^c	243 ^d
Cost of energy consumption (Euros/kWh _{Final Energy})	0.1324 ^e	0.0703 ^f	0.15 ^f	0	0.03608 ^g

^a Estimation of hourly CO₂ intensity for the French grid based on RTE Eco2mix (<https://www.rte-france.com/en/eco2mix>) for the 2013-2018 period. The CO₂ intensity of imported electricity has been also included. Hourly values used in the simulations are summarized here by a yearly average.

^b Average CO₂ intensity in 2018 of district heating and cooling providers in France based on “Base Carbone v17”
<https://www.data.gouv.fr/en/datasets/base-carbone-complete-de-lademe-en-francais-v17-0/>

^c Photovoltaic modules are considered as part of the building, and, as such, we assume that their footprint will be included in the building's life cycle assessment.

^d <https://www.bilans-ges.ademe.fr>, Boiler combustion, E+/C- label

^e Assumption based on the French regulated price for electricity, non-residential use, “Tarif bleu – option base”, 2020, excluding fixed and power-based costs. The gain for the surplus electricity injected into the grid is estimated at 0.06 €/kWh based for installations on buildings (June 2020), excluding other incentives.

^f Average energy price in 2017 for district heating and cooling providers in France (<https://amorce.asso.fr/publications/enquete-sur-le-prix-de-vente-de-la-chaleur-et-du-froid-en-2017-rce31/download>). The price includes both energy- and power-based components, with the latter usually representing around 50% of the final energy price.

^g Assumption based on the French regulated price for natural gas, zone 1, class B1 and B2i, 2020, excluding fixed and power-based costs.