



Report on implementation of the planning tool

Deliverable Title	Report on implementation of the planning tool			
Deliverable number	5.2			
Nature of the deliverable⁽¹⁾	<input checked="" type="checkbox"/> R	<input type="checkbox"/> P	<input type="checkbox"/> D	<input type="checkbox"/> O
Dissemination level⁽²⁾	<input checked="" type="checkbox"/> PU	<input type="checkbox"/> PP	<input type="checkbox"/> RE	<input type="checkbox"/> CO
Due date	Month 30			
Status				
Date of Submission	31.8.2018			

-
- **(1)** Please indicate the nature of the deliverable using one of the following codes (according to the DOA):
Report (R), Prototype (P), Demonstrator (D), Other (O)
- **(2)** Please indicate the dissemination level using one of the following codes (according to the DOA):
PU = Public, 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)



This project has received funding
from European Union's Horizon 2020
research and innovation programme
under grant agreement n° 696098

Authors

Responsible Authors and Entities	
Author	Institution
Miika Rämä	VTT Technical Research Centre of Finland
Laura Sokka	VTT Technical Research Centre of Finland
Esa Pursiheimo	VTT Technical Research Centre of Finland
Krzysztof Klobut	VTT Technical Research Centre of Finland
Kati Koponen	VTT Technical Research Centre of Finland

Document Control

VERSION	DATE	INSTITUTION	CHANGE
0.1	15.6.2018	VTT	Drafting the table of contents
0.2	19.6.2018	VTT	First draft
0.8	29.8.2018	VTT	Preliminary version
0.9	31.8.2018	VTT	Version for internal quality control
1.0	2.9.2018	VTT	Revised for submission

Disclaimer

This work is licensed under a [Attribution-NonCommercial-NoDerivatives 4.0 International License](#)

You are free to:

- **Share** — copy and redistribute the material in any medium or format

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial** — You may not use the material for commercial purposes.

- **NoDerivatives** — If you remix, transform, or build upon the material, you may not distribute the modified material.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

The opinion stated in this report reflects the opinion of the authors and not the opinion of the European Commission.

All INDIGO consortium members are also committed to publish accurate and up to date information and take the greatest care to do so. However, the INDIGO consortium members cannot accept liability for any inaccuracies or omissions nor do they accept liability for any direct, indirect, special, consequential or other losses or damages of any kind arising out of the use of this information.

Executive Summary

Work package WP5 “Planning tool” provides means for evaluating the performance, benefits and potential of a District Cooling (DC) system. The aim is to create an easy-to-use open-source tool for analysing a cooling system in defined area with a group of buildings connected.

Program structure, data storage solutions and GUI design are all addressed in this report. The program consists of two main parts; the calculation algorithms including related data structures and the graphical user interface (GUI). In addition, there are several predefined input data files and files used for storing the systems and cases being studied by the planning tool.

The calculation algorithms and related data structures have been structured as a group of classes representing different components in one or more cooling systems. The most important class is called System. It acts as an interface between the calculation and the GUI. While the GUI is specific to the INDIGO project and the planning tool, the classes themselves could also be utilized independently by using the interface in System class.

System class includes all the needed functions for defining, optimizing and analyzing cooling systems by the planning tool as well as retrieving specific results, e.g. concerning a specific unit within a specific cooling system. The class acts as an interface between the calculation algorithms and the GUI. The class and its variables and functions are described in detail in Appendix A. The classes including their variables are described in Appendix B. The functions within the classes can be found in the program code.

The generated model is a simple linear programming (LP) model. Although oemof library would allow more complex mixed integer type of models (MILP) to be constructed, the increased calculation time with the free solvers available would make this less practical considering the original purpose of the planning tool.

The optimization can be run for a single representative year or a series of year-long optimization runs that take into account the projected development of commodity prices. The optimization finds the least-cost solution for utilising a given system to supply a defined cooling demand. Another element where optimization is relevant is the utilisation of energy storages.

The environmental analysis in the planning tool follows the framework of life cycle assessment (LCA) and focuses on climate impacts of the district cooling systems. The tool gathers together information on the components needed in the assessed systems, and combines them with emission data. The total GHG emissions over the whole life cycle of the system are expressed per MWh cooling produced [kg CO₂/MWh]. The tool allows comparison of the emissions produced by the district cooling system and the reference systems. In addition, it demonstrates how the emissions are formed, i.e. the contribution of different components and the energy consumption during the use to the total life cycle emissions of the assessed case.

Program implementation in Python takes advantage of object-oriented programming for data structures and storage. The predefined input data are stored in csv files. Graphic user interface (GUI) of the planning tool is used to manage model data, structure, and pass these values into optimization module of the tool, and import result data from optimization in order to form easy to read result summary. GUI consists of five functional tabs: Analysis, GIS Tool, Parameters, Results and Sensitivity Analysis. GIS tool of GUI is utilised in forming simple but functional network of cooling system based on map file of the area analysed.

The potential users for the planning tool include municipalities, master developers for a specific area and energy utilities. The results are useful as feasibility as for combining cooling and for energy planning with the city planning processes for both new and built areas. The results also provide input for more detailed, technical plans. The accuracy of the results can be greatly enhanced if coupled with a dedicated simulation model or measurement data for cooling demand.

Table of content

1	Introduction	5
1.1	Review of the related research	5
1.2	General approach in INDIGO	7
2	Overview of the new planning tool	8
3	Program structure	10
3.1	System class	10
3.2	Component classes	11
3.3	Input data files	11
3.4	GUI structure	11
3.5	Case and system specific data files	12
4	Input data	13
4.1	Cooling demand	13
4.2	Data files for calculation parameters	14
4.3	Data sources in emission calculations	15
5	Model generation	20
6	Optimisation	21
7	Analysis and results processing	21
7.1	Energy analysis	22
7.2	Economic analysis	23
7.3	Environmental analysis	24
7.3.1	Principles of life cycle assessment (LCA)	24
7.3.2	Climate impact assessment in the planning tool	26
7.3.3	Calculation of climate impacts	27
7.3.4	Evaluation of the quality of the input data	27
7.4	Comparative and combination indicators	28
8	Using the tool	28
8.1	Creating new case	29

8.1.1	Import map.....	29
8.1.2	Load case	30
8.1.3	Initialize model	31
8.1.4	Model construction	33
8.1.5	Modify and save case.....	36
8.2	Running the model and examining results.....	39
8.2.1	Start optimisation	39
8.2.2	Examine results.....	40
8.3	Sensitivity analysis	43
9	Further development plans.....	45
10	Summary	45
11	References	48
12	Appendix A: class System documentation	50
13	Appendix B: Component classes documentation	56

Nomenclature

DC	District Cooling
WP	Work Package
DoW	Description of Work
KPI	Key Performance Indicator
PV	Photovoltaic
GUI	Graphical user interface
LCA	Life cycle assessment
GHG	Greenhouse gas
NPV	Net present value

1 Introduction

Heating and cooling in industry and buildings together account for ca. half of the European energy consumption¹. The share of cooling of the energy consumption is still fairly small, but it is growing in importance in both households and industry. Warming induced by climate change further contributes to this trend. On the other hand, ambitious climate change mitigation requires drastic reductions in energy use. The new energy efficiency directive includes an energy efficiency target of 32.5% for the EU for 2030². District cooling (DC) can increase the cooling system efficiency by 5 to 10 times compared to on-site cooling systems. It therefore offers a very potential way of improving the energy efficiency, and reducing greenhouse gas emissions of cooling systems,

The goal of project INDIGO is to provide tools supporting design, planning and operation of DC systems. The project covers all the parts of the system; generation, distribution, storage, and demand. Within INDIGO, an open-source planning tool focused on the design stage of a DC system, has been developed. The purpose of the tool is to support the optimal design of new DC systems, and the assessment of existing DC systems' potential for performance/efficiency improvement.

In this report, this dedicated new planning tool of a DC is documented. The section below presents a limited background review of other ongoing projects with a similar topic. The rest of the document describes in more detail the INDIGO planning tool, and its contribution to the state of the art.

1.1 Review of the related research

The OPTi project will deliver methodologies and tools that will enable accurate modelling, analysis and control of current and envisioned district heating and cooling (DHC) systems. The methodology will be deployed both on a complete system level, and on the level of a building(s). The DHC system will be a subject to dynamic control, and will treat thermal energy as a resource to be controlled for saving energy and reducing peak load. OPTi-Sim facilitates a virtual representation of the real DHC system and is a central component of the OPTi framework. It integrates different models in a co-simulation approach. (OPTi)

FLEXYNETS project will develop, demonstrate and deploy a new generation of intelligent DHC network that reduces energy transportation losses by using low distribution temperature levels (15-20°C). The network is used as a source of both heating and cooling by utilising heat pump technology. GIS Software Tool for Designing Networks is being developed. The analysis is based on steady state conditions for fixed temperature levels. (FLEXYNETS)

In STORM project, a generic DHC network controller will be developed with the ambition to increase the use of excess heat and renewable energy sources in the DHC system. The self-learning control is used instead of model-based control approaches. It will make the controller easy to implement in different configuration and generations of DHC networks. The STORM controller includes modules for forecasting, planning and dispatching heating and cooling

¹ <https://ec.europa.eu/energy/en/topics/energy-efficiency/heating-and-cooling>

² http://europa.eu/rapid/press-release_STATEMENT-18-3997_en.htm

production. The control strategies benefit the whole energy chain, from generation and distribution to consumption. (STORM)

InDeal project will offer an innovative platform that will impose an adequate distribution of heating and cooling among the buildings in the network. A number of modules have been integrated in an advanced automated Decision Support System (DSS) including artificial intelligence metering devices, energy demand prediction of the network, data-driven short-term weather forecasting, energy storage monitoring and modelling, energy harvesting, web-based control platform along with solutions for minimizing heat losses. (InDeal)

The innovative thermo-chemical network technology of H-DisNet project will contribute to the next-generation district energy networks. The technology will exploit high chemical potential of absorption processes for loss-free transport and storage of energy potential. The concept forms an intelligent district network with thermal, electric and gas networks. The modelling of H-DisNet will provide the input for control strategies, large network examination and potential assessment. (H-DisNet)

E2District project aims to develop and demonstrate a novel cloud enabled management framework for DHC systems, which will deliver energy cost savings through intelligent adaptive DHC control and optimisation methods. A scalable District Operation System (DOS) will integrate all control, optimisation, diagnostics and prosumer engagement into a DHC management platform. (E2District)

The objective of CoolHeating is to support the implementation of "small modular renewable heating and cooling grids" for communities in South-Eastern Europe. Economic calculation tool has been developed for small modular district heating and cooling projects, mainly meant for non-experts of DH technology, e.g. decision-makers in municipalities. Tool focuses on economic aspects of the system and the main input consists of investment and financing, operational costs, revenues and economic parameters. The results include assets, liabilities and equities, incomes, a balance sheet, cash flow and profitability. (CoolHeating)

PLANHEAT project aims to develop a tool for mapping, planning and simulating a local energy demand and supply to investigate future scenarios, potential identification of an existing network and describe benefits from energetic, economic and environmental points of view. Tool will be easy-to-use, open-source and free. Potential mapping includes demand estimation (degree hours for cooling and heating), database for retrofitting and energy efficiency measures, models for current and future energy demand estimation, available energy sources and industrial excess heating and cooling. GIS based platform will be developed to visualize potential of both demand and supply. (PlanHeat)

SDHp2m project mission is to develop, improve and implement advanced policies and support measures for Solar District Heating (SDH) in 9 participating European regions. On their website, evaluations and a number of calculation tools dedicated to solar district heating can be found enabling feasibility studies. For example, the SDH Online Calculation Tool is based on TRNSYS simulation results and provides a first idea of dimensioning and economics of a solar district heating plant. The tool is suitable for centralised distributed solar district heating plants. (SDHp2m)

The aim of THERMOSS is to define a set of retrofitting heating and cooling packages based on cutting-edge, high-potential, market-ready technologies that are connected together in an

open ICT platform for smart energy management at building and district level. It employs DIMOSIM, an integrated simulation tool for the analysis of feasibility, conception and operation of district energy systems. The tool is meant for the development, test and analysis of the optimisation of concepts and operation of district energy systems. (THERMOSS)

The THERMOS project aims to provide the methods, data, and tools (free and open source) to enable more sophisticated thermal energy system planning rapidly and cheaply. The THERMOS Tool is a map-driven open-source web-based application tailored to the real world requirements of energy planners to make heat network planning faster and more efficient. (THERMOS)

HotMaps project will produce a web-based GIS toolbox for analyzing heating and cooling demand as well as supply options on a local, regional and national levels. All the source code produced during the development of Hotmaps will be accessible on the open source platform GitHub. Main features will be to identify location of heating and cooling demand and potential supply on a map (EU28) with renewable and industrial excess potential highlighted, to estimate potential district heating options for a selected area, to estimate and compare costs of individual heating vs. district heating and to compare the results with national plans for decarbonisation. (HotMaps)

District Heating Assessment Tool (DHAT) is a tool for economic feasibility studies for establishing district heating with a comparison with individual heating. It can be adjusted to local conditions globally. The model can be used to show the economic and environmental benefits of DH. The tool is based on Danish technical data and price projections which can be adjusted, making it applicable to other countries as well. The model is based on MS Excel with no hidden data or VBA code – the user is free to modify all data inputs and variables.

As the limited review of the 13 projects shows, there is a need for tools for assessment of DHC systems. The tool being developed within INDIGO is comparable to the products of related projects. However, while focusing only on DC and cooling systems in general, it outstands by the approach and comprehensive capabilities, analyzing the system from all energy, economic and environmental perspectives. The deliverable D5.1 (Rämä et al. 2017) provided the main framework of our dedicated tool. The following of this deliverable provides details of the solution.

1.2 General approach in INDIGO

INDIGO project focuses on supporting the development of more efficient, intelligent and economic district cooling (DC) systems by improving system planning, control and management. To support the planning phase, an open-source stand-alone planning tool for evaluating the performance and benefits of a DC system has been developed. The tool is an easy-to-use software enabling analysis of cooling systems for a defined area consisting of a group of buildings. The scope, the calculation methods and the program implementation were sketched within the deliverable *D5.1 District cooling planning tool framework and specification* (VIITE). In this deliverable, the developed planning tool is described in detail.

As stated in the Description of Work (DoW), WP5 is carried out in parallel with WP3 and WP4 focusing on component predictive controllers and overall management strategy, respectively. The interaction between the WPs consists of information exchange and result comparison.

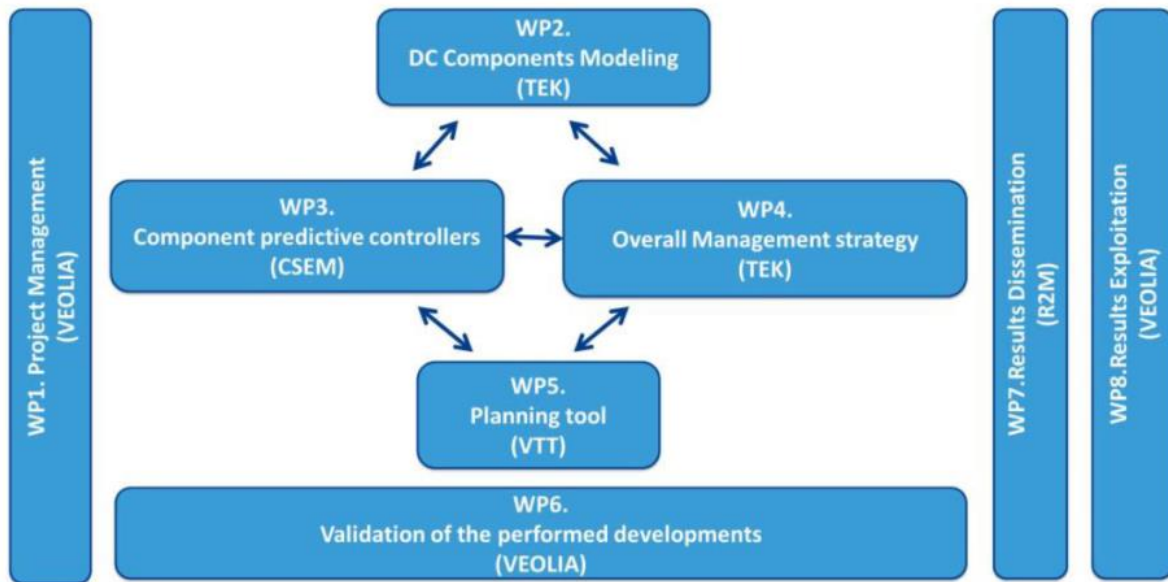


Figure 1. Work package structure of the INDIGO project.

The calculation tool covers energy, economic and environmental aspects of a cooling system. The tool provides information on primary energy consumption, greenhouse gas emissions (GHG, expressed in kg CO_{2eq.}) and costs of the system. In the tool, the potential defined district cooling system can be compared against building and space specific cooling systems providing the same cooling service.

The results of the analysis will be reported also as a diverse set of key performance indicators (KPIs). The user can easily try out and compare impacts of different assumptions on e.g. component specifications, costs and other input data.

The structure of the present report is the following: Section 2 gives an overview of the tool. Its structure, and the input data are described in more detail in Sections 3 and 4. Model generation and optimisation are introduced in Sections 5 and 6, while analysis and results processing are dealt with in Section 7. Model usage is discussed in Section 8, and Section 9 provides a summary of the report content.

2 Overview of the new planning tool

The aim of the planning tool is to be simple to use, and easily configurable. The tool enables the user to try out and compare different cooling systems and particularly to see the effects of different assumptions on component specifications, costs and other input data. With the tool, one can study the feasibility of a DC system, and compare it to the building and/or space specific systems. It thereby enables the demonstration of the benefits of DC. Also, different configurations of a DC system can be compared.

The potential users for the planning tool include municipalities, master developers for a specific area and energy utilities. The results are useful as feasibility and pre-feasibility studies as well as for combining cooling and energy planning with the city planning processes for both new and built areas. The results also provide input for a more detailed, technical plans. The accuracy of the results can be greatly enhanced if coupled with a dedicated simulation model or measurement data for cooling demand. The model generation and optimisation process can also be useful for research purposes.

The planning tool includes all the main components of a cooling system. However, details on external processes are excluded from it. These external processes include industrial activities providing excess heat for absorption based cooling production or electricity production other than solar PVs and CHP units located on-site.

DC system is the most complete of the systems considered in terms of system definition. Both the building and space specific systems can be seen as special cases of the district cooling system. The two main differences between DC and cooling systems are the higher total cooling capacity in building and space specific cooling systems due to simultaneity factor (i.e. peak demand not taking place exactly at the same time) and the distribution network.

The resources considered in a district cooling system are fuels (both fossil and biomass), solar irradiance, electric grid and other external sources of energy such as industrial waste heat. These resources are in turn utilised in energy supply by CHP units, boilers, solar collectors or photovoltaics (PVs) and directly as electrical energy, excess heat or natural source of cooling. The output of the energy supply in the district cooling system can be electricity, heating or cooling. They are utilised in production of cooling by a electric chiller, an absorption chiller or free cooling. The produced district cooling is distributed to the consumers to satisfy a given consumption of all consumers through a network of pipes. The distribution component also includes optional centralised storage and pumps. This process is illustrated in Figure 2.

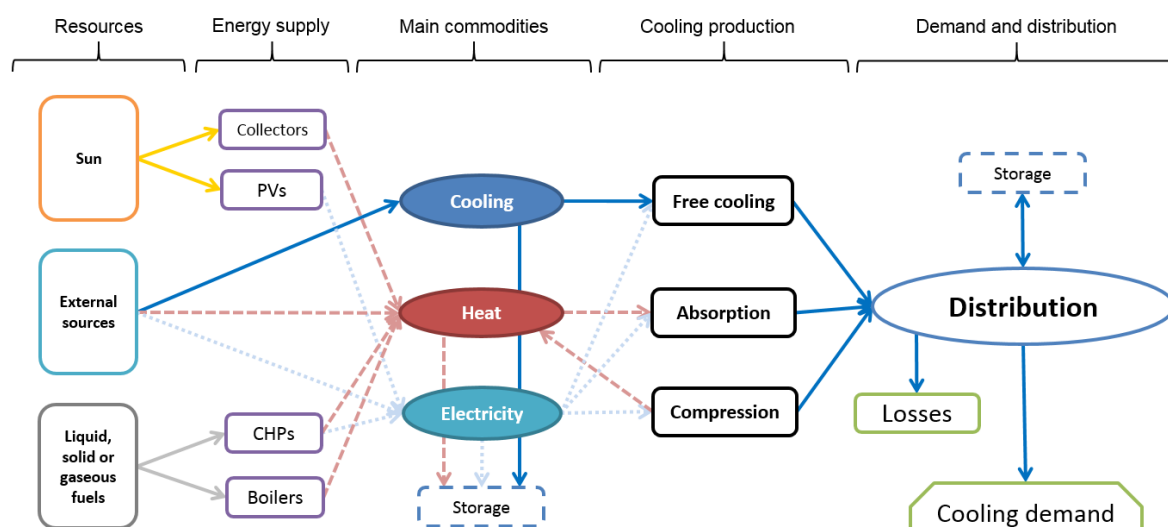


Figure 2. Structure and elements of the energy system model representing a cooling system.

The building specific cooling system is identical to the district cooling system except that it does not include a distribution network and that the whole system consists of several sub-systems (i.e. consumers, individual buildings). In systems consisting of units with building-

specific cooling, the total cooling demand in terms of capacity is usually higher than in a district cooling system. In the space specific cooling system, each cooled space has its own cooling supply. The only technology considered as a source of cooling is heat pumps. Solar PVs are taken into account on a building level, not for single cooling units.

The analysis process begins with system definition by the user. While default assumptions on technical parameters and costs are given, the user can freely influence the specification. After the system has been defined, the planning tool automatically generates an optimisation model based on the input and solves this model to find out how the system is operated in a cost-optimal way. The resulting energy flows are then analysed to summarise the costs and energy consumption as well as to calculate the emission impact of the cooling system operation. Results corresponding to one or many systems can then be presented and compared.

The scope and level of detail during this process is one of the main topics within this deliverable.

3 Program structure

The program consists of two main parts; the calculation algorithms including related data structures and the graphical user interface (GUI). In addition, there are several predefined input data files and files used for storing the systems and cases being studied by the planning tool.

The calculation algorithms and related data structures have been structured as a group of classes representing different components in one or more cooling systems. The most important class is called System. It acts as an interface between the calculation and the GUI. While the GUI is specific to the INDIGO project and the planning tool, the classes themselves could also be utilized independently by using the interface in System class.

Two external Python libraries are utilized by the planning tool; Qt for the GUI implementation and oemof for the building and solving an optimization model for the cooling systems studied. The optimization model also needs an external solver for the optimization process. In this case, a free solver recommended by the oemof community, the CBC (Coin-or branch and cut) solver.

This chapter provides a detailed description of the classes (especially the System class and the interface), GUI structure in a programming sense, input data files including their format, and finally, the case and system specific data file generated by the GUI.

3.1 System class

System class includes all the needed functions for defining, optimizing and analyzing cooling systems by the planning tool as well as retrieving specific results, e.g. concerning a specific unit within a specific cooling system. The class acts as an interface between the calculation algorithms and the GUI.

The main variables contained in the class definition are called *systems* and *resources*. Variable *systems* is a list consisting of individual cooling systems, DC or building/space specific. An element in the list is an instance of the class CoolingSystem. Variable *resources* is a list as well, consisting of resources available for all cooling systems. Its elements are instances of the class Resources. Contents of both classes CoolingSystem and Resources are described in more detail in next chapter.

Other variables are reserved for the needs of the economic analysis; *interest_rate* is used in calculation of the discount factor while *costs* and *profitability* are lists with float values representing the results of the analysis.

The class and its variables and functions are described in detail in Appendix A.

3.2 Component classes

The component classes are the six programming structures that are used to represent one or more cooling systems and the components within these systems. They are created and manipulated by calling the interface functions in class System presented in previous chapter.

A single cooling system is represented by class CoolingSystem. It can in turn include a number of energy supply units, cooling production units, consumers, energy storages and in case of a DC system, a distribution network. These are represented by classes EnergySupply, Producer, Consumer, EnergyStorage and Network, respectively.

The classes including their variables are described in Appendix B. The functions within the classes can be found in the program code.

3.3 Input data files

Input data files used by the planning tool are all located under the folder “Data”. The data files are in tab-separated values format with a .txt extension. Some input data files are shared by all cases studied by the planning tool while others are case specific. These case specific files are named after their location, e.g. a country or a city. All types of files can be freely modified if more accurate information is available.

The content and format of the input data files is described in Chapter 4.2.

3.4 GUI structure

GUI of planning tool is represented in detail in Chapter 8. In Figure 3 structure of planning tool in terms of data flow is illustrated. In addition to GIS based construction of the DC system model, the main function of GUI is to manage following data flows:

1. Saving model structure and data (.pickle files)
2. Loading model structure and data (.pickle files)
3. Creation of model files (.txt files)
4. Feeding model generator and optimization solver with model files
5. Load results from optimization into result files (.pickle)
6. Illustrate results from result files

See Chapter 3.5 for details of case specific model files.

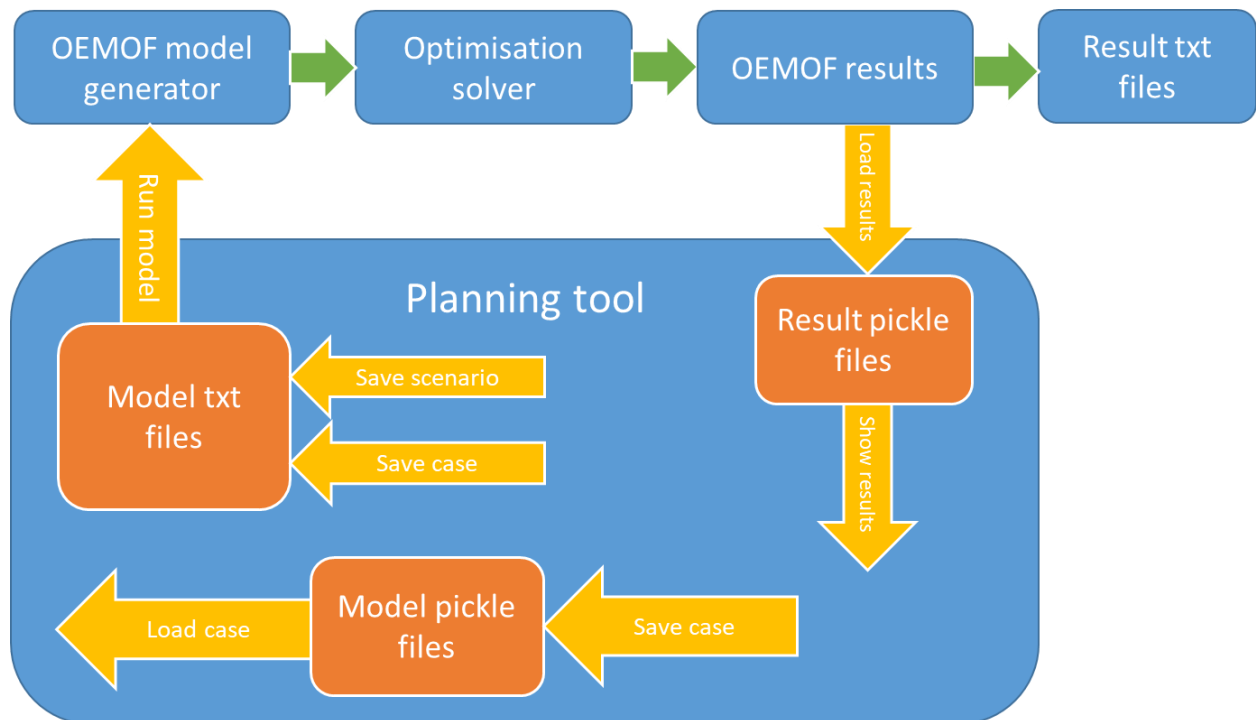


Figure 3. Flow of data in planning tool.

3.5 Case and system specific data files

In the folder application is installed, there is subfolder named .\GIS\ in which case folders are created. There are several files in a case folder (see Table 1). During case creation map file and coordinate files are inserted in the case folder. When user saves case during model construction, application saves data structures into pickle files (.pickle), which store data objects and can be utilized easily when case is loaded into model construction tool. Furthermore, when user saves case, application creates model text files with scenario name BASE. If user saves scenarios, similar model files named after scenario are created in the case folder. Optimization procedure uses these model text files to construct optimization model. User can delete scenario files, except BASE scenario, in order to delete a scenario (all the model .txt files for deleted scenario should be removed).

Table 1. Files located in the case folder.

File type	Description
GIS case files	User should not delete or edit
Map.png	Map file used in GIS tool
Coordinates.txt	Longitude and latitude of left top corner and right bottom corner
Model .txt files	User can edit numeric values and delete files (except BASE scenario!)
dh_network_<scenario>.txt	Topology of DC network

dh_nodes_<scenario>.txt	Location of network nodes
prod_con_<scenario>.txt	Location of producers and consumers
reference_data_<scenario>.txt	Building specific reference case data
System_cost_<scenario>.txt	System data - tariffs and location
system_data_<scenario>.txt	System data - consumers, production and supply units, storages and pipeline
Pickle files	User should not delete or edit
arc.pickle	System data object structure
con_ts.pickle	Consumer time series data structure
cons.pickle	Consumer location data structure
lines.pickle	Pipeline location data structure
prod.pickle	Producer location data structure
ref.pickle	Building specific reference case data structure
res_ts.pickle	Resource time series data structure
scen_af_<scenario>.pickle	Unit utilization factors - scenario result file
scen_emission_<scenario>.pickle	Emissions - scenario result file
scen_months_<scenario>.pickle	Monthly data - scenario result file
scen_profit_<scenario>.pickle	Profit data - scenario result file
scen_results_<scenario>.pickle	Operational time series - scenario result file
scen_types_<scenario>.pickle	Unit types - scenario result file
scen_values_<scenario>.pickle	Key performance indicators - scenario result file

4 Input data

The cooling demand time series are the most relevant piece of input data, but there are several data files that include technology specific key parameters for all the included technologies. These values can also be defined and modified within the planning tool GUI. This chapter describes the input data needed by the model. Special attention is paid on the input data needed by the emission calculations which is and will remain the most difficult to evaluate.

4.1 Cooling demand

While being the most important piece of input and a requirement for using the tool, the data format for cooling demand specification is very simple; it is a .txt file with consumer name or

identification on the first header row followed by an hourly (preferably) year-long (8760 h) time series in kWh/h of cooling.

The definition of cooling demand for the planning tool is described in Chapter 8.1.3 on initialising the model.

4.2 Data files for calculation parameters

The data files include the techno-economic and emission calculation parameters used by the planning tool.

Following Table 2 lists and describes the data files. For location specific data files in the table, “<location>” is the used specified file name, e.g. “emissions_Spain.txt”.

Table 2. Shared and case specific input data files with descriptions.

Filename	Description
Shared by all cases	
cooling_techs.txt	Techno-economic and specific emission parameters for the cooling production technologies; compression, absorption and free cooling. Values for small (< 1000 kW) and large units.
supply_techs.txt	Techno-economic and specific emission parameters for the energy supply technologies; CHP units and boilers with different fuels, PVs and solar collectors. Values for small (< 1000 kW or m ²) and large units.
pipedata.txt	Pipe size (DN) specific pipe properties for steel pipes with insulation, including specific emissions per pipe meter for production (materials) and construction.
emissions_storage.txt	Embedded (production, construction) emissions for different storage types. Values for small (<1 MWh) and large units.
techs_and_resources.txt	Allocation of fuels into supply technologies e.g. as follows: CHP:HFO,LFO,NG,Chips,Pellets Boiler:HFO,LFO,NG,Chips,Pellets Collectors:Sun PVs:Sun
Case specific	
emissions_<location>.txt	Location specific emissions from consumption/combustion of specified resources, production of these resources and possible other local, site specific emissions (gCO ₂ /kWh).
resources_<location>.txt	Location specific resources and their prices as year-long time series (€/MWh).
resources_projection_<location>.txt	Location specific resource price development for next 32 years as index values (year 1 has a value of 1.000)
solar_<location>.txt	Solar irradiation (diffuse and direct radiation) and outdoor temperature for a specific location (W/m ² and °C)

The content of the input files for cooling production, energy supply and storage technologies are presented in Table 3, Table 4 and Table 5.

Table 3. Parameters for cooling production technologies.

Technology	Compression	Absorption	Free cooling
<i>Capacity (MW)</i>	5	5	20
<i>Natural efficiency (-)</i>	0	0	1
<i>Electrical efficiency (-)</i>	0.2	0.02	0.02
<i>Heat efficiency (-)</i>	0	1.25	0
<i>Investment cost (€/kW)</i>	700	1000	1200
<i>O&M</i>	0.05	0.03	0.02
<i>Lifetime (a)</i>	10	20	20
<i>Emissions (small, kgCO₂/kW)</i>	96	127	12
<i>Emissions (large, kgCO₂/kW)</i>	12	30	46

Table 4. Parameters for energy supply technologies.

Technology	CHP	Boiler	Collectors	PVs
<i>Electric efficiency (-)</i>	0.4	0	0	0.13
<i>Heat efficiency (-)</i>	0.5	0.8	0.5	0
<i>Capacity (kW, kW/m²)</i>	5000	2000	0.6	0.1
<i>Investment (€/kW)</i>	2500	600	1200	1000
<i>O&M (%-inv)</i>	0.04	0.02	0.01	0.01
<i>Lifetime (a)</i>	20	25	20	20
<i>Emissions (small, kgCO₂/kW)</i>	40	40	478	2353
<i>Emissions (large, kgCO₂/kW)</i>	114	114	158	1584

Table 5. Parameters for storage technologies.

Storage	DC	Heat	Cold	Electricity
<i>Investment (€/kWh,m³)</i>	150	150	150	1000
<i>Low (kgCO₂/m³, kWh)</i>	678	678	678	0.08
<i>High (kgCO₂/m³, kWh)</i>	212	212	212	0.08

Input data file concerning pipes used in the distribution network has following column headers; DN (DN size), Dout (outer diameter, mm), d (thickness of the pipe, mm), Din (inner diameter, mm), Dcase (insulation casing diameter, mm), Cd (insulation casing thickness, mm), Distance (distance between two pipe centers, mm), Costs (specific costs, €/m), Production (specific emissions, kgCO₂/m) and Construction (specific emissions, kgCO₂/m).

4.3 Data sources in emission calculations

As described previously in this report, the systems included in the model consist of district cooling systems, and space or building specific systems. For all of these systems, their typical emission profiles are included in the model. All the major energy and material flows related to

the components needed for the systems are included in the planning tool in order to assess the GHG emission flows and the total emissions (Figure 4).

The systems consist of elements that can be categorised into following categories:

- resources,
- energy supply,
- cooling production,
- distribution and
- consumption

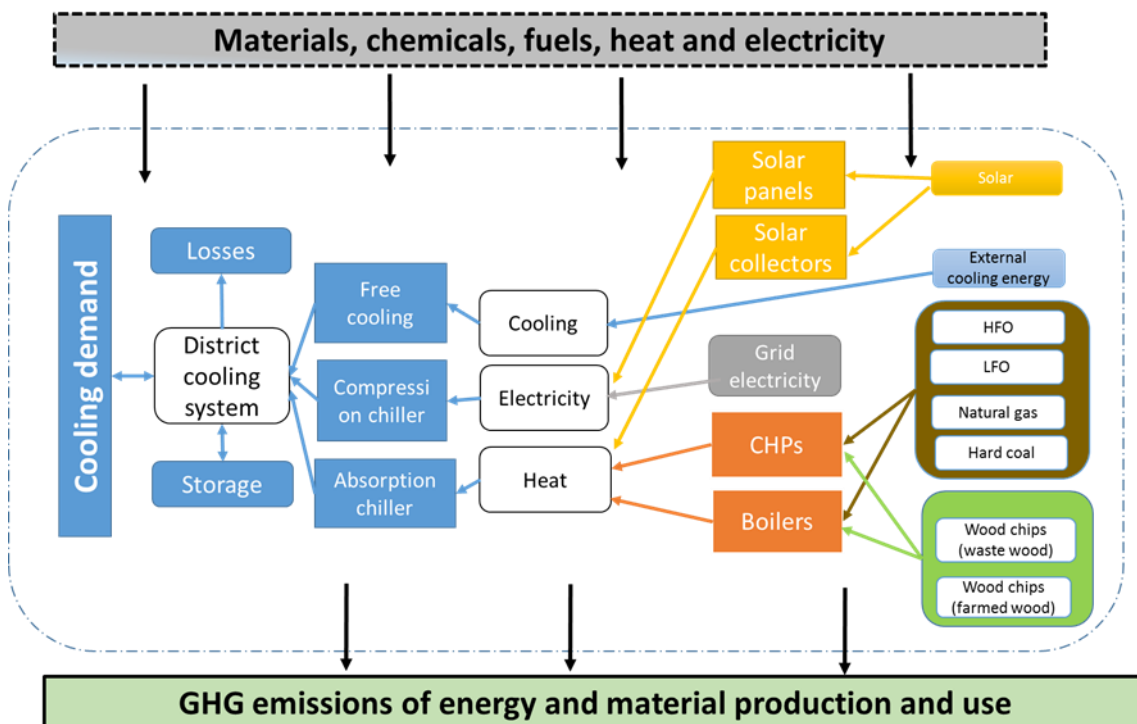


Figure 4. Framework of the assessment of the greenhouse gas emissions in the study for an example district cooling system. Energy and material flows related to the system components are assessed enabling the evaluation of the GHG emissions of the system.

Table 6 presents the processes included in the different phases. The emissions of the utilisation phase refer to the emissions of energy consumption in the studied systems. These include for example the emissions of fuels or electricity used in the cooling production. The district cooling systems and the components utilised are supposed to be long-lived. It is challenging to know how the components would be treated at the end of their life cycle, and what the recycling possibilities and emissions/emission benefits at that point are. Thus, for simplicity, the end of life emissions are left outside the system boundary of the planning tool, and only the establishment and use phase of the district cooling and the reference system are included in the assessment.

Table 6. Processes and data sources included in the different life cycle phases.

Life cycle phase/process	Reference	Value (CO ₂ eq)	Other notes
Energy production			
Grid electricity	ELCD database	4,67E-01 (EU27); 4,49E-01 (Spain); 5,57E-01 (Italy)	Average grid electricity for the countries / country groups in question. Unit of all figures g CO _{2eq} /kWh
Heavy fuel oil (HFO)	ELCD database	10.3 g/MJ	
Light fuel oil	JEC - Joint Research Centre- EUCAR- CONCAWE collaboration 2014	15,4 g/MJ	
Natural gas	JEC - Joint Research Centre- EUCAR- CONCAWE collaboration 2014	13,0 g/MJ	
Wood fuels	JEC - Joint Research Centre- EUCAR- CONCAWE collaboration 2014	4.1-6.5 g/MJ	Covers two types of wood fuels: “farmed Wood, industrial” and “waste wood, industrial”.
Photovoltaic panels	Frischknecht et al. 2014 (panels <20 kW); Corona et al. 2017	1584-2353 g/kW	Smaller figure represents the 1 MW-size panel, and the larger

	(panel size 1000 kW).	figure the kW-size panel.
Photovoltaic collectors	Ardente et al. 158-478 kg/kW 2005 (collector size 1,4 kW); Heath et al. 2009 (collector size 50 MW)	Smaller figure represents the large plant, larger the kW-size collector.
Oil / Wood boiler	Expert 50-90 kg/kW estimation on the basis of different references	Smaller figure for 10 kW boiler, larger for a 30 kW boiler.
Gas boiler	IEA solar 38 kg/kW heating LCA tool	Size of the boiler 10 kW
Hard coal power plant	Bauer 2008 415 kg/kW	Supercritical plant, net capacity 800 MW
Natural gas power plant	Bauer 2008 145 kg/kW	Combined cycle power plant, net capacity 400 MW
Cooling production		
Compression chillers	Shah et al. 12-96 kg/kW 2008 (kW- size chiller); EPD of the Trane Centrifugal Chiller (MW- size chiller)	Smaller figure represents the larger chiller, larger the smaller chiller.
Absorption chillers	Beccali et al. 127 kg/kW 2010 & 2012 (kW-size chiller)	No data found for MW-size absorption chillers
Free cooling (Heat exchangers)	Hang et al. 12-48 kg/kW 2014 (size <100 kW); Adolfsson & Rashid 2016	Smaller figure the smaller heat exchanger, and the larger figure the

	(size 490 kW)		larger heat exchanger.
Distribution network and storage			
Insulated pipes - pipe production	Fröhling & Svanström 2005a	7 - 270 kg/m	Pipe sizes covered: DN25, DN100 and DN500
Pipes - network construction	Fröhling & Svanström 2005b	27 - 110 kg/m	Pipe sizes covered: DN25, DN100 and DN500. Network construction represents construction in urban environments.
Insulated water storage	Oró et al. 2012 (size of the tank 10.3 m ³); Raluy et al. 2014 (size of the storage 15 000 m ³)	212-678 kg CO ₂ /m ³	Smaller figure represents the smaller water storage unit, and the larger figure the larger storage unit.
Lithium-ion batteries	Guillaume Majeau-Bettez et al. 2011	0,08 kg CO ₂ /kWh	Data represents nickel cobalt manganese lithium-ion (NCM) batteries
<i>Fan coils and cooling coils</i>	Shah et al. 2008	289 kg CO ₂ /unit	Data represents fan sizes with cooling capacity of ca. 5-18 kW.

However, some minor components are excluded, as their contribution to the final results was expected to be insignificant. For this, the 1% cut-off rule is applied in the tool, i.e. those processes whose contribution to the total emissions of the cooling system is estimated to be less than 1% will be excluded. In addition, installation of components (e.g. energy used by manpower) and end of life emissions (emissions in landfill/recycling/recovery) have been excluded.

5 Model generation

After defining a cooling system in the planning tool, an optimization model can be built and solved. This dynamic generation of the cooling system based on user input is at the core of the planning tool.

The process makes heavy use of oemof library as described in the Chapter 3. In the oemof framework, an energy system consists of sources, sinks, transformers and buses between the first three. Energy storage is also available as a dynamic element. Sources are where the primary energy to run the system is drawn (e.g. fuels, electric grid) while the sinks represent cooling demand itself, and electric grid or heating system for possible surplus production. Transformers are components representing energy conversion units, such as CHPs or electric chillers, that in the end supply the cooling needed. Energy storages can be connected to the busses to store the corresponding commodity such as electricity or district cooling.

The first step in the model generation is to define the cooling demand. In case of a DC system, it is the sum of cooling demands of every connected consumer and the heat (or cooling) losses in the distribution network. For a building or space specific cooling system, it is the defined cooling demand.

Next all the buses between sinks, sources, transformers and possible energy storages are defined. Some of these buses might be left unused if no components are connected.

Sources (i.e. resources) are generated next and connected to the respective busses. In case of fuels or external sources of energy such as electric grid, a variable cost is defined. This definition takes into account the forecasted development of the costs if the optimization is run for a year in the future. The sources/resources can be enabled and disabled. In practice this means that they have a very high costs (fuels, commodities) or very low potential (solar).

Next a set of (possibly) defined energy supply units are generated as transformers. Each of the transformers have a connected input (resource) and one or more outputs (heat, electricity) with capacities and conversion efficiencies defined.

The output of the energy supply units or external sources of heating or cooling are then connected to a set of defined cooling production units. Similar to energy supply units, cooling production units have capacities and efficiencies defined. However, they all produce cooling.

Possible storage units for electricity, heating and cooling (produced or external) may be defined according to the specifications given by the user.

Lastly, the sinks are generated. These include cooling demand, distribution network losses (if a network exists, building or space specific cooling does not include a network), an electric grid for surplus electricity sold and a heating system connection for surplus heat sold. The cooling demand and losses are fixed, meaning that they need to be fulfilled. The connections for surplus production are open with a negative cost (revenue) defined.

One special sink is defined for the pump electricity consumption if a distribution network exists. While the planning tool does not include a network simulation model, a simplified calculation algorithm has been implemented that returns the hourly electricity consumption based on design pressure loss (i.e. pressure loss at peak cooling demand) and cooling demand. The details of the method are described in class Network description in Appendix B.

All the components derive their input from user specifications defined within the component classes presented in Chapter 3. The structure of the generated system follows closely the system definition illustrated in Figure 2.

6 Optimisation

The generated model is a simple linear programming (LP) model. Although oemof library would allow more complex mixed integer type of models (MILP) to be constructed, the increased calculation time with the free solvers available would make this less practical considering the original purpose of the planning tool.

The current recommendation of the oemof community is to use CBC (Coin-or branch and cut) solver with the generated models. Other free alternative is GLPK³ and commercial solvers such as Gurobi or Cplex can be utilized as well.

The optimization can be run for a single representative year or a series of year-long optimization runs that take into account the projected development of commodity prices. Several years' optimization is obviously a lot slower process than one representative year. The optimization finds the least-cost solution for utilising a given system to supply a defined cooling demand. The costs consist of variable costs from the use of resources. As the resource prices are defined as time series and can change (e.g. electricity), this can potentially change how the system is operated. Another element where optimization is relevant is the utilisation of energy storages.

See Chapter 3.3 for details concerning the costs data defined in the input files.

At the current version, there is no extensive feasibility testing or error detection. The optimization functions simply return 1 if successfully finding an optimal solution and 0 if not. If successful, the results are saved within the component classes and can be retrieved for visualization by the GUI. The optimization function also writes a .txt file "Optimisation results.txt" for further processing by the user.

7 Analysis and results processing

The overall analysis process follows closely the plans set in D5.1 "District cooling planning tool framework and specification" as presented in Figure 5. In overall, there are very little significant deviations from principles described in the previous deliverable. However, more details are presented.

³ GNU Linear Programming Kit, <https://www.gnu.org/software/glpk/>

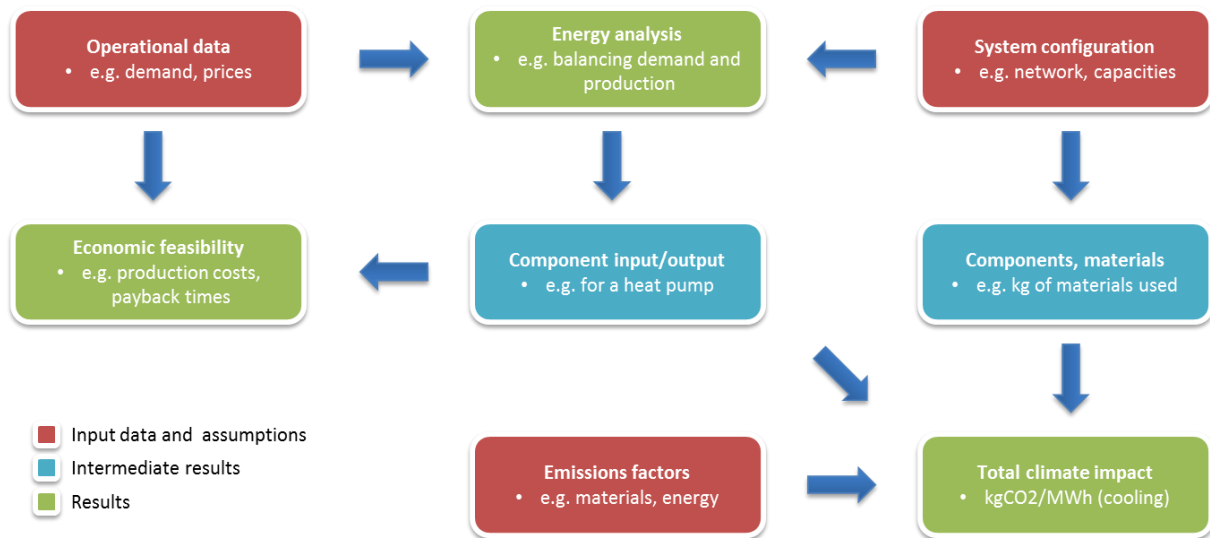


Figure 5. Overall description of the analysis to be carried out.

Starting from the top left corner of Figure 5, the idea is first to define the system being investigated and prepare the required input data. Then, to find out how the system is operated and what are the energy flows within the system. Using both information on the system components (investment) and their energy consumption (operational costs), the economic feasibility can be studied. On the other hand, energy consumption and component specification is used to calculate the climate impact in terms of GHG emissions. These emissions are calculated using life cycle assessment both for operational phase and for production of components, i.e. materials used.

Although not yet added to the planning tool, there are preliminary plans to include a weighting functionality into the tool. According to the plan, all energy efficiency, economic performance and emission impacts would be ranked with a scale of 1 to 5. Weighing these aspects of the system analysis, a overall score could be presented. This would allow:

- a better consideration of client's preferences if e.g. emission impact is by far the most important aspect
- studying greater amount of parameter combinations, including variation of weights of e.g. costs and energy consumption

7.1 Energy analysis

The objective of the energy analysis is to provide information on the energy flows within a given system with a specified cooling demand. These energy flows include both the input of energy into the system as consumption of available resources (primary energy consumption) and the consumption and production of energy by individual components within the system such as e.g. CHP units or electric chillers.

The energy flows are based on results of the optimisation and as it is based on minimising the operational costs of the system, energy analysis can in fact be considered as a techno-economic analysis. The value of the objective function of the optimisation is the total operational cost of running the system. However, in practice the operational costs of the

system are recalculated using the result time series of the resource consumption. Although not yet fully utilised this way, it is possible to differentiate the origin of the costs.

A single optimisation run represents a single year of operation, a so-called representative year. In both economic and environmental analysis, a longer time period is investigated based on the results of the energy analysis for this representative year. However, it is also possible to run the optimisation for multiple years. This takes into account the projected development of the commodity prices. In this case, the yearly optimisation results are recorded and economic and environmental impact analysis is based on these yearly results.

The energy analysis for district cooling systems and building/space specific systems differs in a way that for district cooling simply one system is investigated. For building specific cooling systems, the number of systems is same as number of buildings included in the analysis. For space specific cooling system analysis, the number of individual systems is even higher but many can be considered to be identical in terms of cooling demand and technical specifications. The results for building/space specific analyses are finally summed up to represent the operation of a whole system comparable to a district cooling system. This approach is visualised in Figure 6.

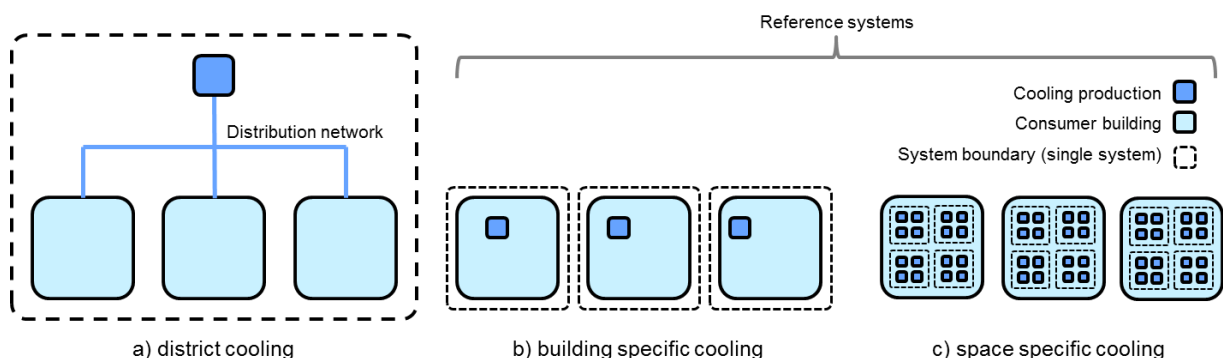


Figure 6. Simplified illustrations of the systems investigated.

Naturally, results of the energy analysis from two different district cooling systems can be compared as well.

While currently both the reference systems can be analysed by the calculation algorithms implemented, the GUI supports only building specific cooling system as a reference case. This is due to the inconvenient amount of required input data which would make the use of the tool very laborious and not practical at all. However, there are plans on addressing this issue by treating space specific cooling systems similar to building specific cooling systems, but by dividing the installed cooling capacity by a simultaneity factor based on the number of cooled spaces within the building. Also, the electric chiller efficiency can be set to a different level than (presumably) more efficient building specific unit. This way, both the operational costs and the investment required would be more realistic and representative of the case of space specific cooling.

7.2 Economic analysis

In economic analysis the results of the energy analysis (resource consumption in kWh) are used to calculate operational costs (€/kWh) and system definition, i.e. capacities for

components (kW) within the system, is used for calculating investment costs (€) by using defined specific investments for each technology (€/kW). Other yearly operation and maintenance related costs are added as percentages of initial investment, also defined within the components of the system.

The economic analysis can be run by utilising results from “a representative year” or it can make use of optimisation results of multiple years. In the first case, the time span of the study can be defined and in the second it is the same number of years used for the multiple year optimisation.

The analysis takes into account the discount factor (set to 5 % as default) and calculates the net present value (NPV) of future costs and revenues. The analysis provides two main results; cumulative costs and profitability. The profitability is cumulative costs taking into account the yearly revenues. This allows also payback time to be calculated.

Figure 7 presents an example of this type of visualisation. The example graph on the right hand side has one line that intersects zero. The corresponding year (10.5 a) is the payback time of the investment.

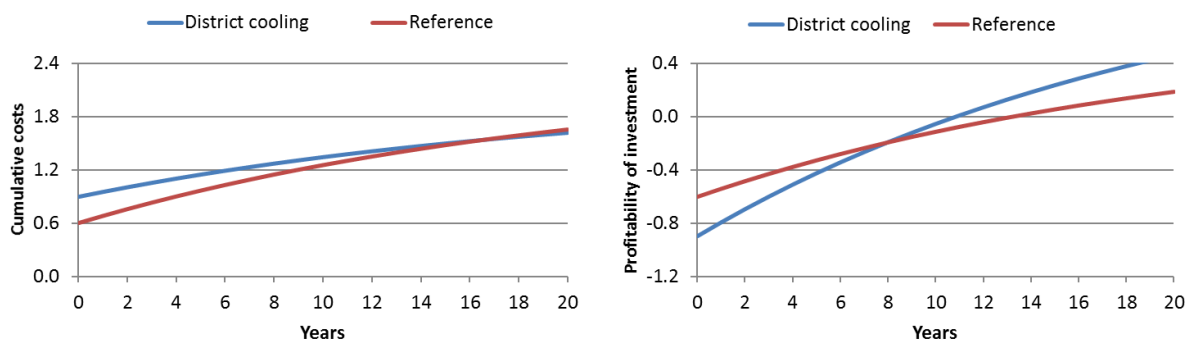


Figure 7. Examples of curves representing cumulative costs (left) and profitability of investment.

7.3 Environmental analysis

The environmental analysis in the planning tool follows the framework of life cycle assessment (LCA) and focuses on climate impacts of the district cooling systems. The planning tool enables assessment of the GHG emissions of the studied district cooling system. It also allows for comparison of the emission impacts between planned district cooling systems and reference cooling systems (i.e. space specific and building specific cooling systems). Thus, the tool will provide information on potential emission savings achieved with the district cooling systems.

7.3.1 Principles of life cycle assessment (LCA)

Life cycle assessment (LCA) is a widely utilised tool for systematically evaluating the potential environmental impacts of a product or a system throughout its whole life cycle. The principles of LCA have been standardised in two ISO standards; ISO 14 040 and ISO 14 044 [1,2]⁴. In

⁴ ISO 14040 provides a clear overview of the practice, applications and limitations of LCA to a broad range of potential users and stakeholders, also those with a limited knowledge of life cycle assessment,

LCA, all the inflows and outflows of substances are considered, and the impacts of these are analysed. Typically, an LCA covers the life cycle of a product from ‘cradle to grave’ but it can also be limited to a certain part of the life cycle, for instance the use phase. LCA can help to recognise the most emission intensive components and functions of a system and thus enables efficient emission reduction measures. As LCA concentrates on all the inflows and outflows of substances it also reduces the risk of shifting problems from one part of the life cycle to another.

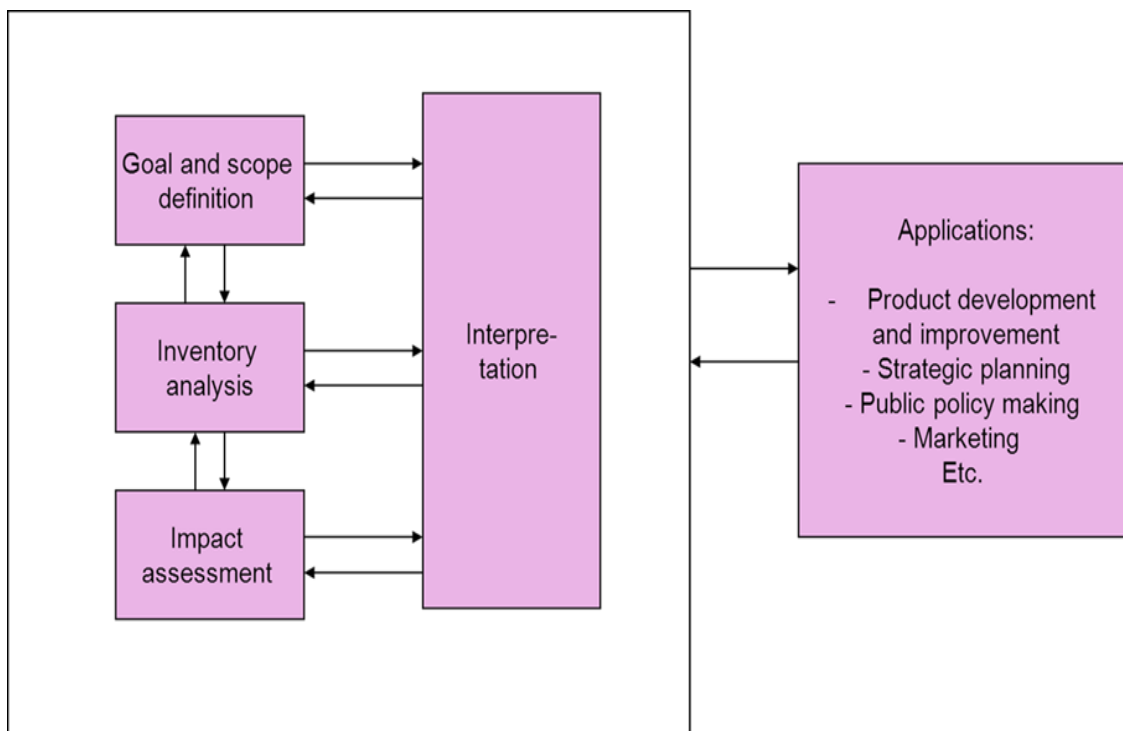


Figure 8. The four phases of life cycle assessment (LCA) based on ISO 14040 and 14044.

In an LCA study, the emissions of a system are studied per functional unit. Functional unit means a reference to which the inputs and outputs of the system are related, and by which the different systems can be compared to each other.

LCA studies can be roughly divided to two different types of analyses based on their approach to data collection and system modelling. In the so-called attributional LCA, the average emissions of a system are studied, whereas in so called consequential LCA the marginal emissions due to a change in the system are studied. In the planning tool, we follow the principles of attributional LCA and define the average emissions of the district cooling systems

while ISO 14044:2006 has been designed for the preparation, execution and critical review of life cycle inventory analysis. It also gives guidance on the impact assessment phase of LCA and on the interpretation of LCA results, as well as on data collection. The standards complement one another as 14040 sets the basics and 14044 goes into more detail.

and the reference systems⁵. This will enable the comparison of the different choices to produce a certain amount of cooling.

7.3.2 Climate impact assessment in the planning tool

In LCA, a system boundary defines the limits of which parts of the system are included in and which parts excluded from the study. Normally, an LCA study covers the whole life cycle until the disposal of the product studied. In the modelling tool, the system boundary covers the district cooling system, including the local resources and energy supply, cooling production, distribution network and consumers. In principle, all the inputs to and outputs from the system are studied. Emissions related to both the establishment and use of the system will be analysed. As mentioned in Section 4.3, in the tool, some minor components are excluded by applying the 1% cut-off rule, i.e. those processes whose contribution to the total emissions of the cooling system is estimated to be less than 1% will be excluded. In addition, installation of components (e.g. energy used by manpower) and end of life emissions (emissions in landfill/recycling/recovery) have been excluded. As the district cooling systems and the components utilised are supposed to be long-lived, for the technical components, the emissions of production are allocated over 20 years, which is expected to be their average life time. For the distribution network (pipes), the emissions are allocated over 50 years life time.

In the planning tool, the user will select the parameters to be used in his system. The tool will provide LCA data for base cases, but the user can also include also own emission data. Due to limited resources, it was not possible to conduct LCAs within the project. Thus, the LCA data in the model stems from literature and open LCA databases (see Section 4.3). If appropriate data was not found, it was calculated on the basis of the raw materials used in producing the product in question. Moreover, data is offered in aggregated format, and not all technologies or sizes on the market are covered. The user also has the choice of replacing the model data with his/her own data.

The planning tool will gather together the information of the components needed in the district cooling system and the reference system. This information will be combined with the emission data. As a result, comparison of the emissions produced by the district cooling system and the reference systems is given. The tool will also demonstrate the contribution of different components and the energy consumption during the use to the total emissions. In addition, the user can calculate additional indicators, including the average annual emissions (allocated emission to components) / average annual cooling produced, and the distribution of the emissions over different life cycle phases.

In the tool, functional unit (i.e. unit per which the total GHG emission over the whole life cycle of the system are expressed) is MWh cooling produced [kg CO_{2eq}/MWh] by the system. Different greenhouse gases have different effects on global warming⁶. In order to

⁵ In a planning tool like the one developed in this project, the assessment of overall consequences of a decision to build a district cooling system would be very challenging, and e.g. energy system modelling would be needed to track the market impacts.

⁶ Two main ways how gases differ from one another regarding their contribution to global warming are their ability to absorb energy (their "radiative efficiency"), and the time they stay in the atmosphere (also known as their "lifetime"). Carbon dioxide by definition has a global warming potential of 1 as it acts as a reference, and the warming potential of the other gases is calculated in relation to it. More specifically, global warming potential measures how much energy the emission of 1 ton of a particular gas will absorb over a given period of time, relative to the emission of 1 ton of carbon dioxide (EPA 2017).

commensurate the warming impact of the different gases, so-called global warming potentials have been used in order to transfer the emissions into carbon dioxide equivalents (CO_{2eq}).

7.3.3 Calculation of climate impacts

As explained in the previous section, the total GHG emissions over the whole life cycle of the system are expressed per MWh cooling produced [kg CO₂/MWh]. The planning tool will gather together the information of the components needed in the district cooling system and the reference system. This information will be combined with the emission data. Thus, the calculation of the GHG emissions in the tool is straightforward. Using the emission per component / resource unit E_C , the total emission of these components E_{tot} can be calculated as:

$$E_{tot} = i * E_C,$$

where

i is the number of components in the system.

The emissions of energy consumption (electricity/heat/fuel) are calculated as:

$$E_{tot} = E_E * Q,$$

where

E_E is the emission of Energy [gCO₂/kWh or gCO₂/MJ]

Q is e.g. the energy consumption [kWh or MJ].

The total emissions of the system E_{system} will then be calculated as a sum of these emission components E_{tot} :

$$E_{system} = \sum_{i=1}^n E_{tot}.$$

The tool then enables comparison of the emissions produced by the district cooling system and the reference systems. It also shows how the emissions are formed, e.g. the contribution of different components and the energy consumption during the use. In addition, the user can calculate additional indicators, including the average annual emissions (allocated emission to components) / average annual cooling produced, and the distribution of the emissions over different life cycle phases

7.3.4 Evaluation of the quality of the input data

The LCA data in the model is generic by nature. It has been selected for the model through a thorough literature review, and it has been evaluated to be the most suitable and representative data available for the processes in question. The data is appropriate for screening studies and simplified LCAs but for more specific assessment the user is encouraged to replace the model data with more specific data representing the exact system analysed, particularly data on processes that most affect the results.

7.4 Comparative and combination indicators

The main results of the planning tool consist of optimisation results presenting the energy flows within a cooling system, economic analysis of costs, revenues and required investments, and operational and embedded emissions. They are all visualised in the planning tool.

The optimisation results are hourly year-long time series (kWh/h). The value are also saved into a dynamically created .txt file for further processing. The time series include all the energy flows between components from resources to cooling demand. They represent the energy analysis in the planning tool.

The economic analysis is carried out for a specified number of year. It calculates the costs of resource consumption based on results of the energy analysis, adds other operation and maintenance related costs based on the needed investment and the most importantly, considers the investment itself. Future costs (and revenues) are discounted.

Operational and embedded emission calculation is based on the results of the energy analysis (resource consumption) and the system specification, respectively. The embedded emissions are technology specific with different values used for small and large units. The operational emissions are calculated using location specific data, e.g. average specific emission factor of electricity for a specific country.

The results are also summarised as a series of key performance indicators. The current set of these indicators are:

- Total yearly cooling demand (MWh)
- Total costs (k€)
- Total yearly emissions (tCO₂)
- Yearly operational emissions (tCO₂)
- Embedded emissions per year (tCO₂)
- Share of embedded emissions (%)
- Cooling production costs (€/MWh)
- Cooling emissions (kgCO₂/MWh)

Analysing and comparing cases can also lead to additional so-called comparative indicators, such as price of achieved emission reduction. While the comparisons are already possible, the related key indicators are not yet fully developed in current version of the planning tool.

8 Using the tool

Planning tool contains three major tabs:

1. **Model analysis:** In this tab, existing DC model is optimized and results can be examined.
2. **Model construction:** In this tab, DC model is constructed and parameters defined.
3. **Sensitivity:** In this tab, sensitivity analysis in terms of model parameters can be performed.

In this brief manual, user of planning tool is guided through construction of new DC model by using GIS tool and performing optimization and result analysis of existing models.

8.1 Creating new case

8.1.1 Import map

Creation of new model begins by choosing **Model construction** tab and clicking **Create case** button. See Figure 9.

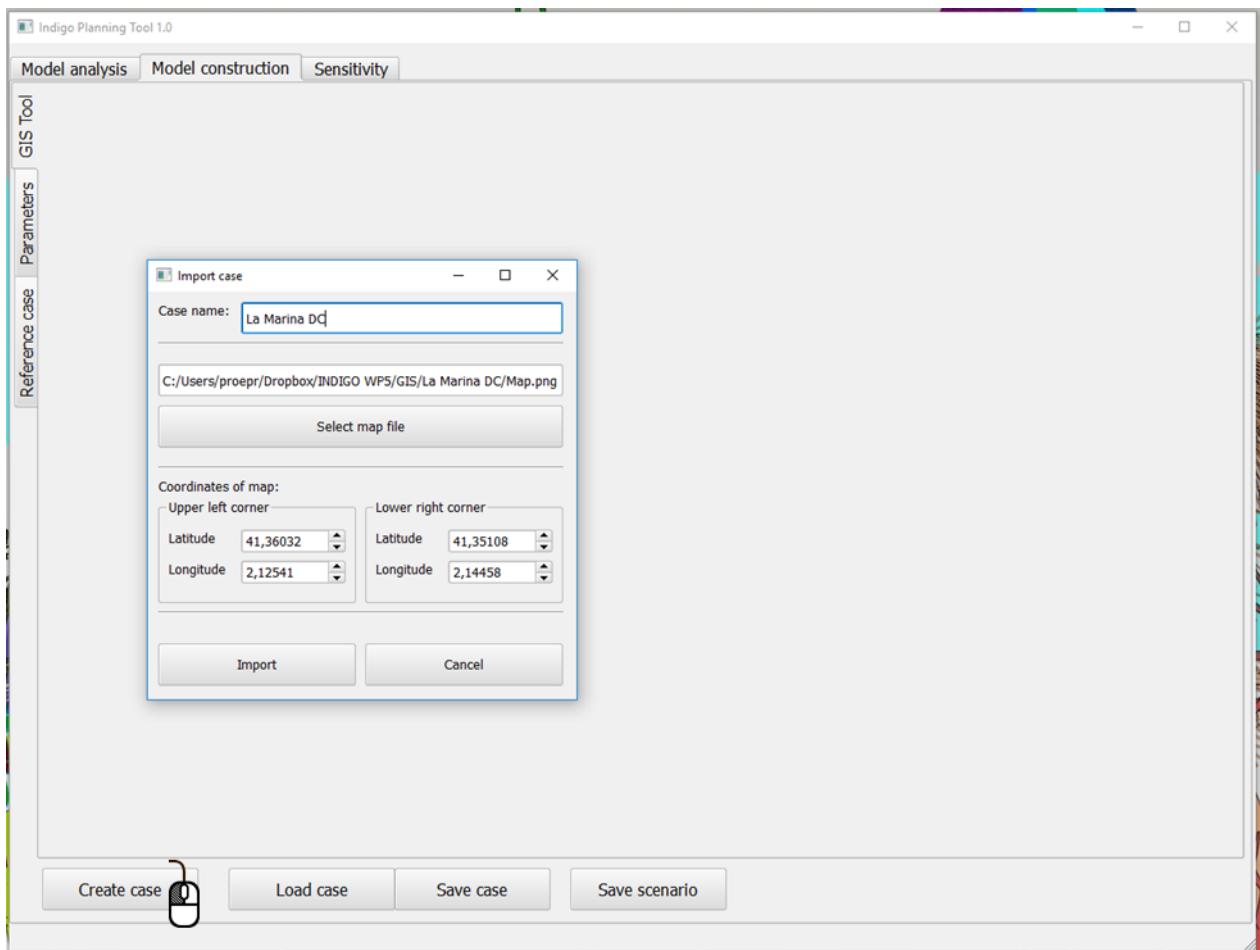


Figure 9. Creating new case by clicking Create case button.

Create case button ejects **Import case** window illustrated in Figure 10. Importing model includes following features:

1. **Case name:** case is not created if case name is missing or case folder already exists.
2. **Select map file** button: user browses for map file (.png format)
3. **Coordinates of map:** user inserts latitude and longitude of upper left and lower right corner of map file. Import window warns if coordinates are inconsistent.
4. **Import** button: model is created (model folder is created) if case requirements are met.

Corner coordinates of map file can be obtained from e.g. google maps.

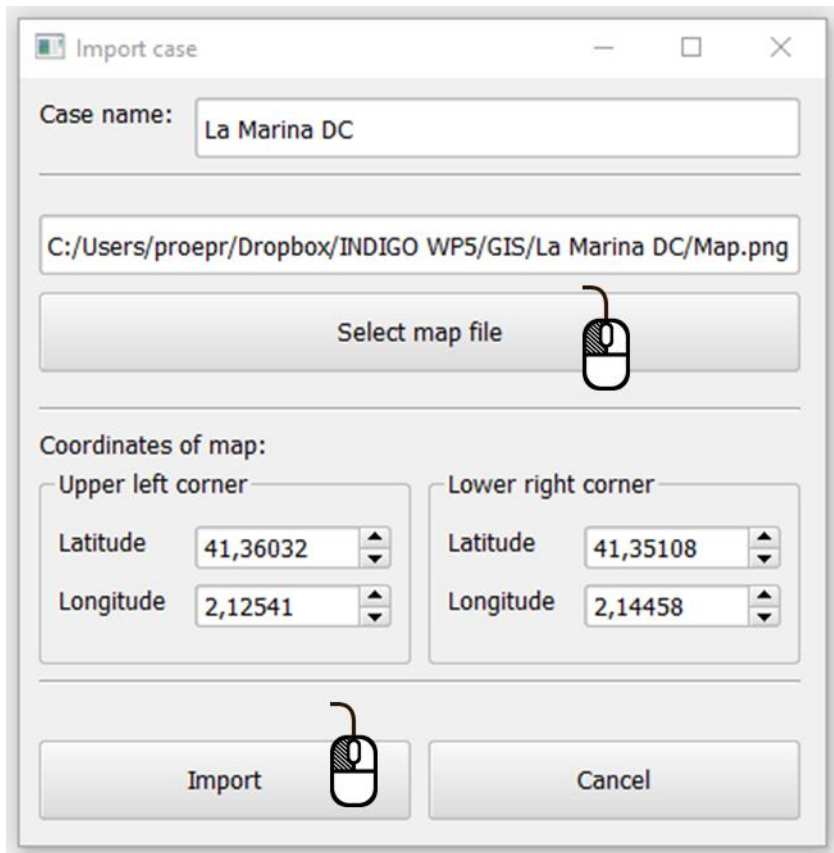


Figure 10. Case import window.

8.1.2 Load case

Once case is imported, and case folder with necessary files is created with import tool, cases can be loaded for model construction by pressing **Load case** button in **Model construction** tab. Pressing **Load case** button brings out selection of cases, illustrated in Figure 11, (all the case folders in GIS folder) and user can select case to be modified.

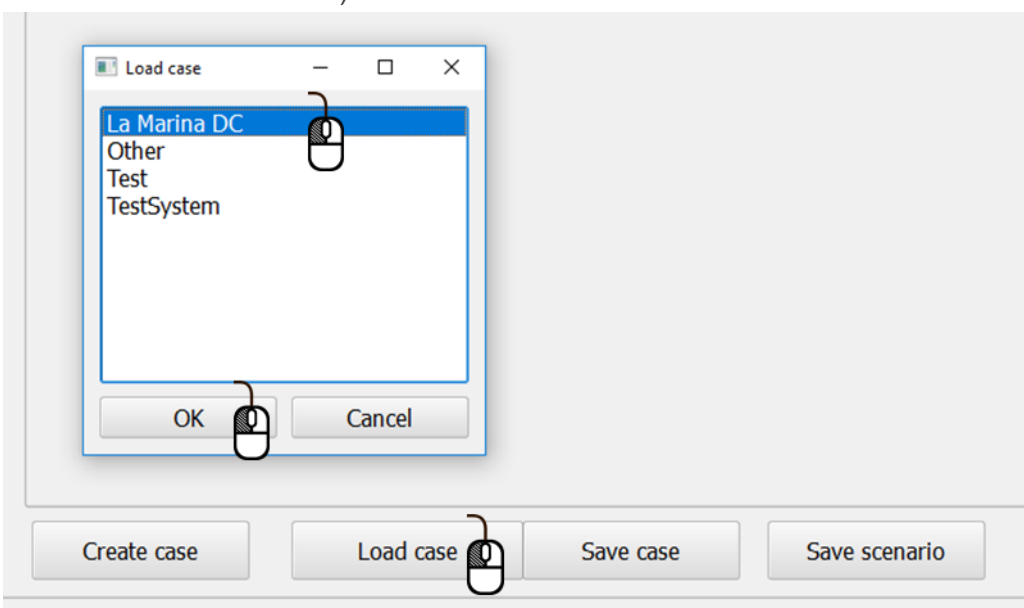
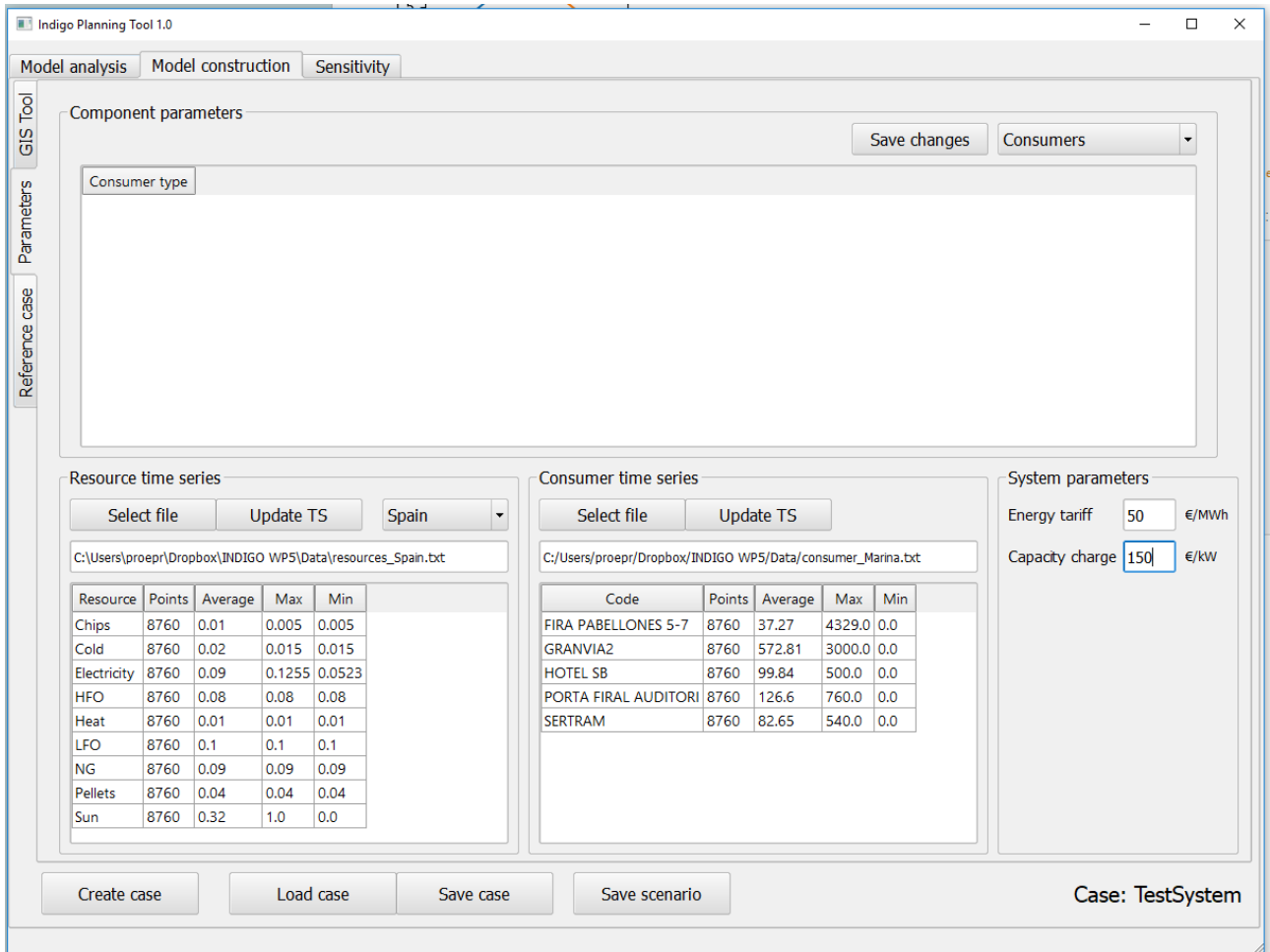


Figure 11. Loading case to be modified.

8.1.3 Initialize model

Once case is loaded, DC model can be constructed or modified in three different tabs:

1. **GIS tool:** In this tab, user can construct DC network in GIS interface by inserting consumers, producers, pipeline network and energy suppliers.
2. **Parameters:** In this tab, user can define and modify technical and economical parameters of DC system.
3. **Reference case:** In this tab, user can modify details of building specific reference case.



Indigo Planning Tool 1.0

Model analysis | Model construction | Sensitivity

GIS Tool | Parameters | Reference case

Component parameters

Save changes | Consumers

Consumer type

Resource time series

Select file | Update TS | Spain

C:\Users\proepr\Dropbox\INDIGO WP5\Data\resources_Spain.txt

Resource	Points	Average	Max	Min
Chips	8760	0.01	0.005	0.005
Cold	8760	0.02	0.015	0.015
Electricity	8760	0.09	0.1255	0.0523
HFO	8760	0.08	0.08	0.08
Heat	8760	0.01	0.01	0.01
LFO	8760	0.1	0.1	0.1
NG	8760	0.09	0.09	0.09
Pellets	8760	0.04	0.04	0.04
Sun	8760	0.32	1.0	0.0

Consumer time series

Select file | Update TS

C:\Users\proepr\Dropbox\INDIGO WP5\Data\consumer_Marina.txt

Code	Points	Average	Max	Min
FIRA PABELLONES 5-7	8760	37.27	4329.0	0.0
GRANVIA2	8760	572.81	3000.0	0.0
HOTEL SB	8760	99.84	500.0	0.0
PORTA FIRAL AUDITORI	8760	126.6	760.0	0.0
SERTRAM	8760	82.65	540.0	0.0

System parameters

Energy tariff 50 €/MWh

Capacity charge 150 €/kW

Create case | Load case | Save case | Save scenario

Case: TestSystem

Figure 12. Model construction tool.

Model construction starts with defining consumer time series in Parameters tab (see Figure 12). User selects from Data folder (or any other folder) a .txt file, in which hourly consumption data for different buildings are listed, by clicking **Select file** button and proceeds to setting these time series into model by clicking **Update TS** button (see Figure 13). User can examine properties, such as number of data points, average value etc., of time series in a table. Also, user can similarly examine (see Figure 12) time series for resources and setting country in which DC network locates. Furthermore, in Parameters tab user defines local energy tariff and capacity charge values for DC.

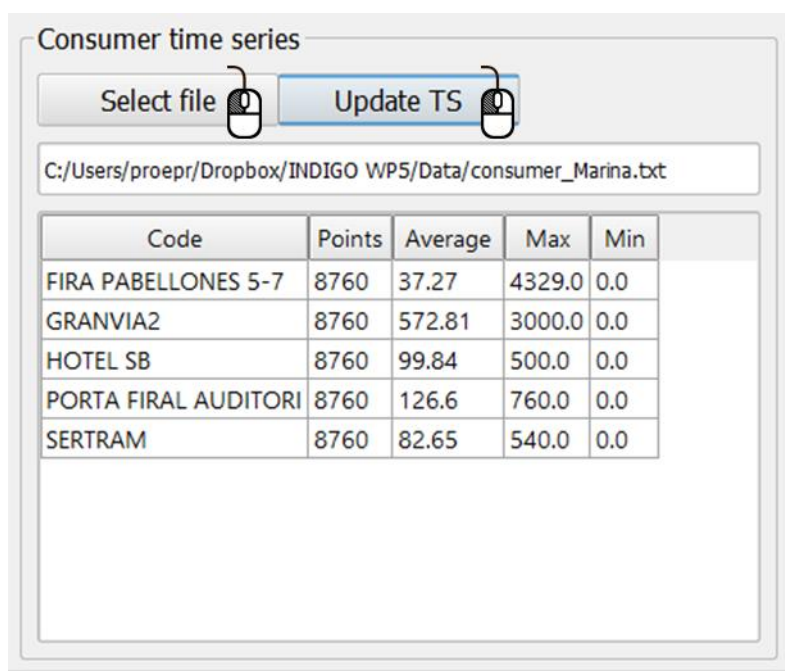


Figure 13. Import of consumer data.

8.1.4 Model construction

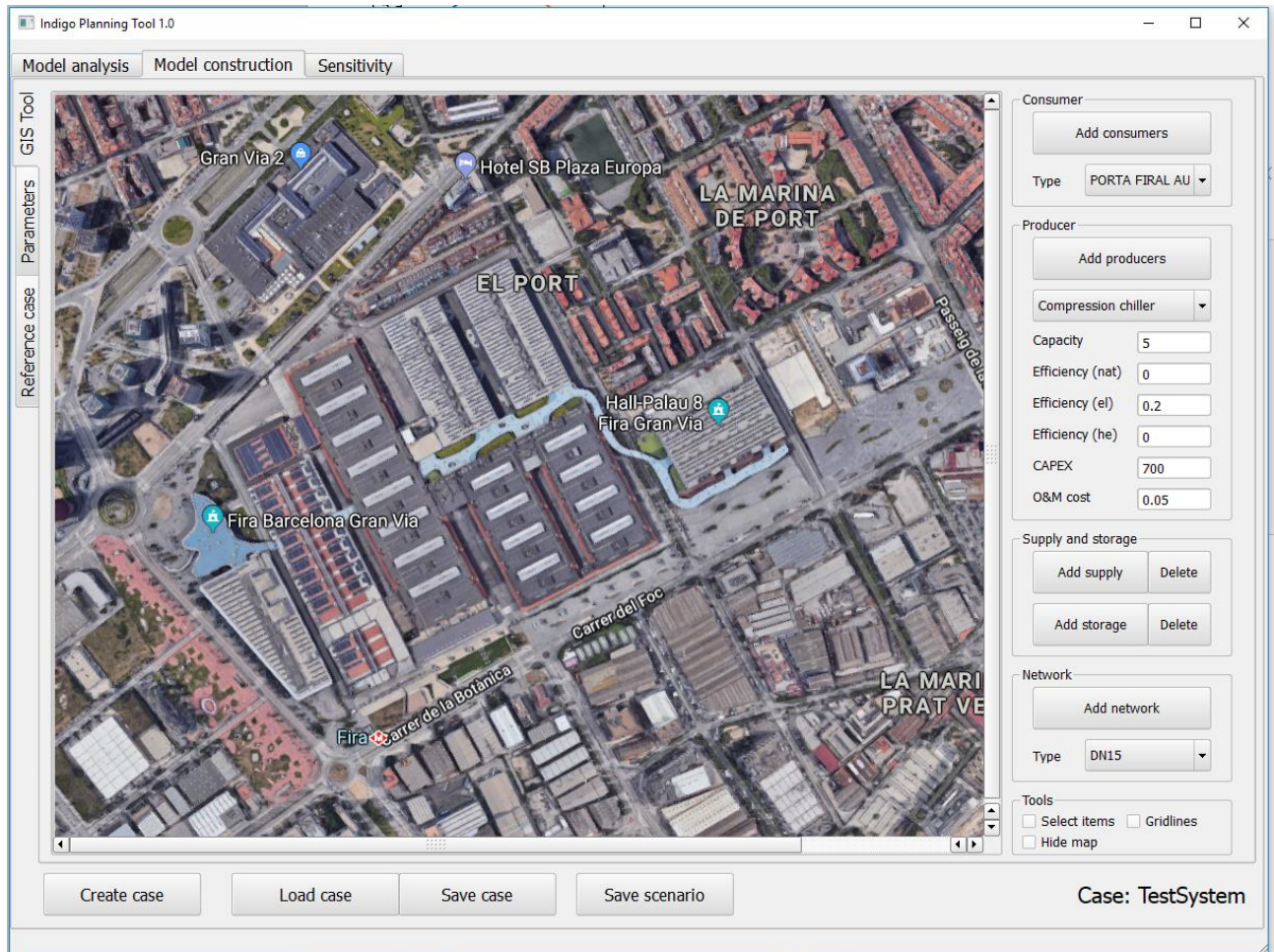


Figure 14. GIS construction tool.

After consumer time series have been defined, GIS based model construction can start in **GIS tool** tab (see Figure 14). User can insert model components by selecting the unit to add and clicking on the map screen. Map is based on a grid (size of 10 pixels) and added units snap to this grid in order to ease construction of continuous network.

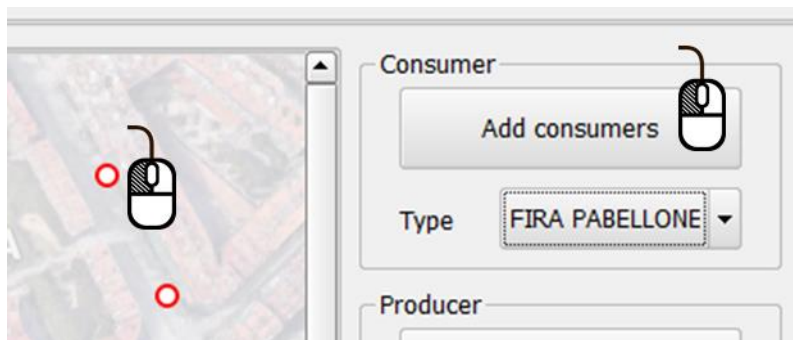


Figure 15. Adding consumers into the model.

Consumers can be added after pressing **Add consumers** button (see Figure 15). User can select type of building from combo box and insert consumer into system by clicking on the map. Consumers are illustrated by red circles.

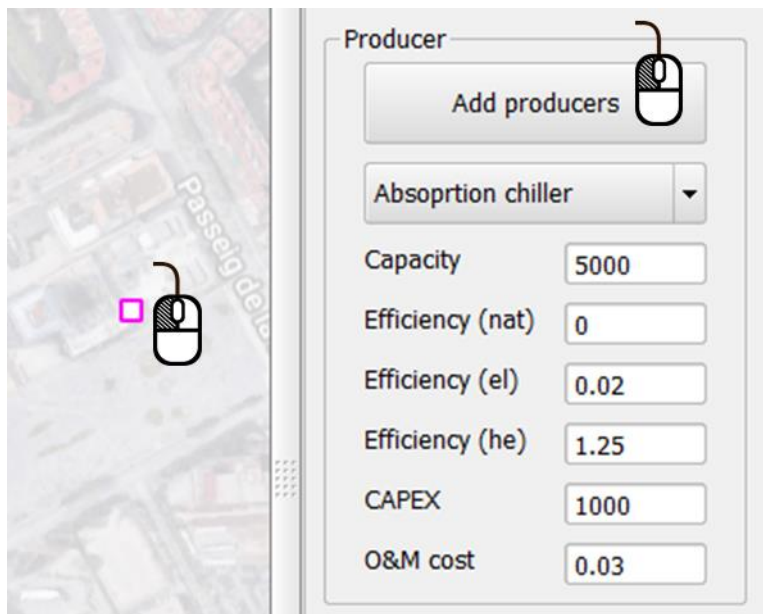


Figure 16. Adding producers into the model.

Producers of DC can be added after pressing **Add producers** button (see Figure 16). User can select type of producer technology (compression, absorption or free cooling) from combo box and edit technical and economical parameters of the unit (default values are based on a data file in Data folder). User inserts producer into system by clicking on the map. Producers are illustrated by violet squares.

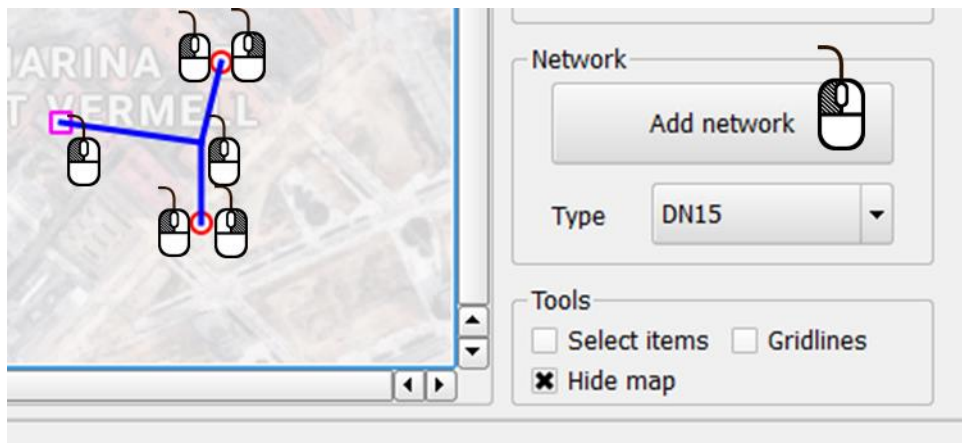


Figure 17. Drawing DC pipeline network.

DC network must connect producers and consumers by continuous pipeline. User can draw pipeline after pressing **Add network** button (see Figure 17). User can select type of each pipeline component from combo box before drawing. Drawing starts by left clicking on location of a producer or a consumer and proceeds by left clicking on the trail of pipeline (blue lines in Figure 17). Once a continuous pipeline section is finished, user right clicks and stops this section. User can then proceed from another location and continue left clicking as before. It must be noted, that network must be continuous, that is, all the producers and consumers must be linked to each other and pipeline cannot have a loose ends.

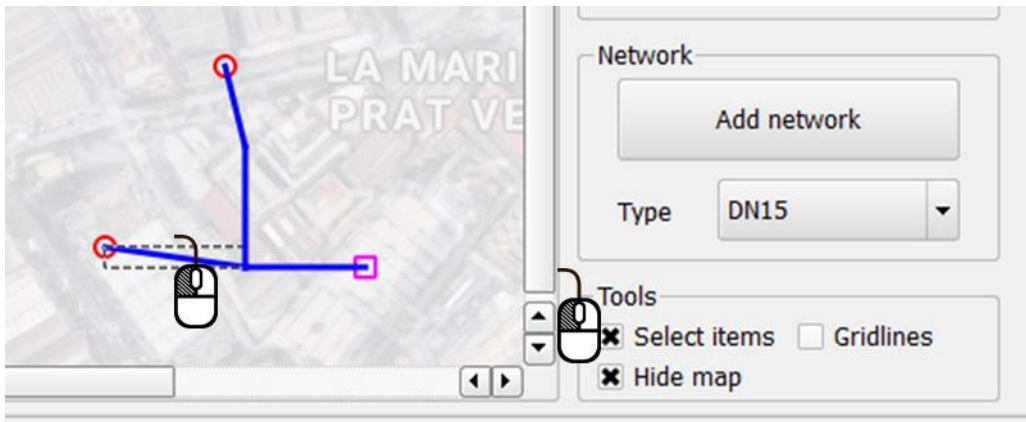


Figure 18. Editing DC system.

When user constructs DC system, components can be spatially edited by checking **Select items** check box (see Figure 18). When this option is checked, any component on map can be selected and either moved by directional keys or removed by **Delete** key. Furthermore, in **Tools** section user can see gridlines or hide map as partially transparent.

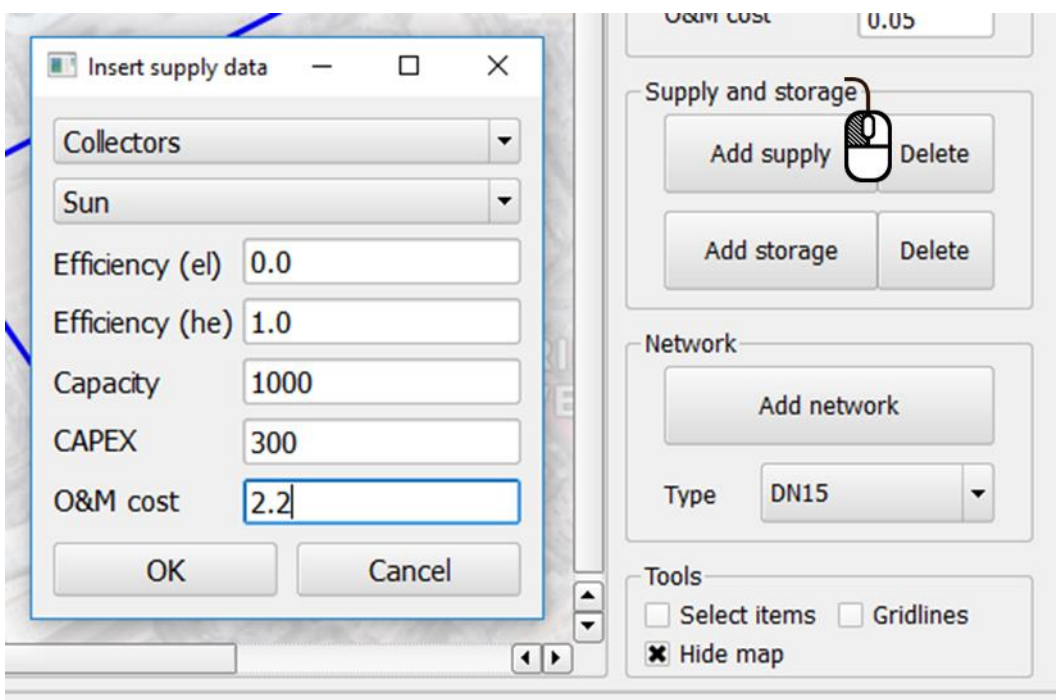


Figure 19. Inserting supply unit.

In DC system, energy (e.g. heat for absorption chiller) can be supplied by production units, which are inserted into model by clicking **Add supply** button (see Figure 19). User can select type of supply unit (e.g. CHP or solar PV) and fuel used (e.g. wood chips or fuel oil), and define technical and economical parameters. Supply units are not spatially modeled and therefore not inserted into map. Supply units can be deleted by clicking **Delete** button and selecting unit to be removed.

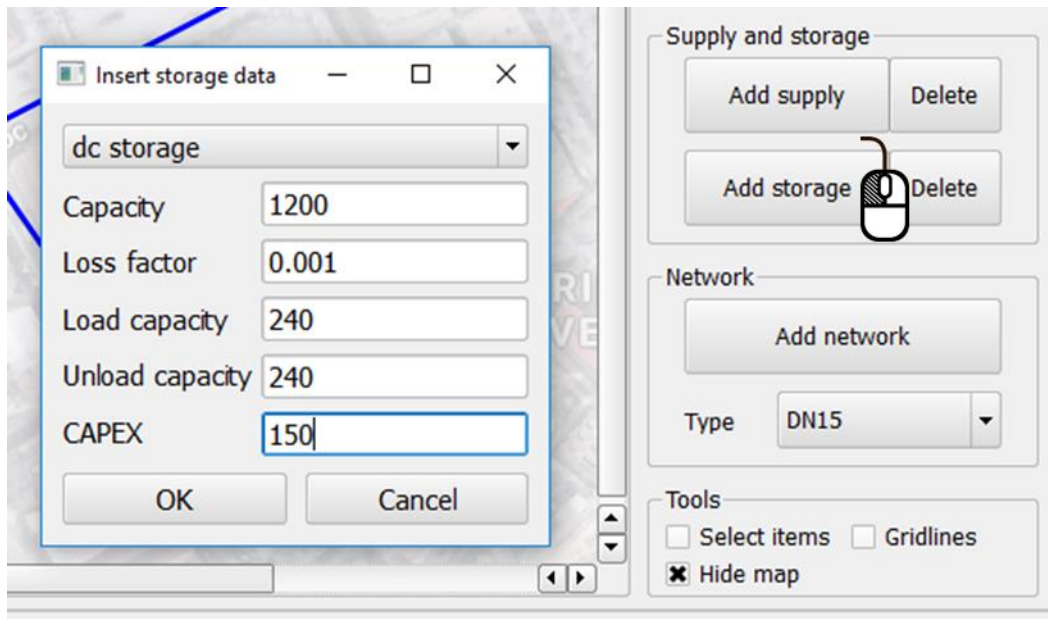


Figure 20. Inserting storage unit.

Storage unit concerning cooling, heat or electricity can be added into the system by clicking **Add storage** button (see Figure 20). User can select type of storage unit (dc storage, heat storage, electricity storage) and define technical and economical parameters. Storage units are not spatially modeled and therefore not inserted into map. Storage units can be deleted by clicking **Delete** button and selecting unit to be removed.

8.1.5 Modify and save case

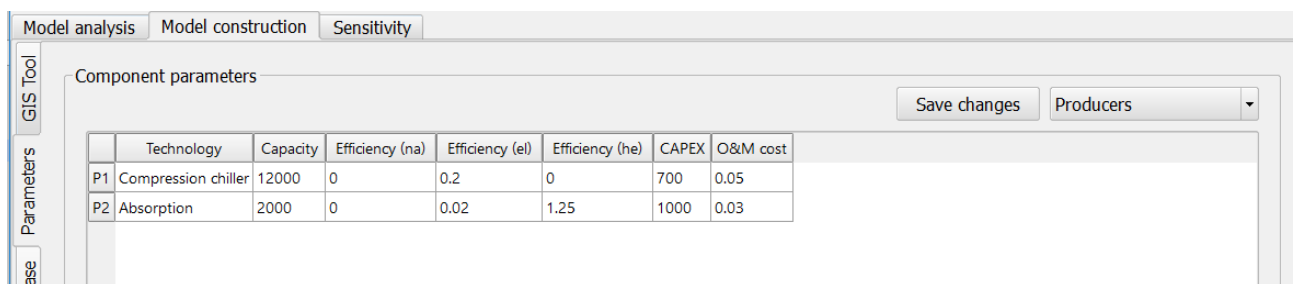


Figure 21. Component parameters of the system.

Elements and units of DC model can be examined in terms of parameters from **Component parameters** section in **Parameters** tab. User can select type of unit to be examined from the combo box. Also, user can edit parameter values and save modifications by pressing **Save changes** button. Note - this saving affects only current data structure and this change does not make any changes in case folder data files.

Model analysis
Model construction
Sensitivity

GIS Tool
Parameters
Reference case

House specific cooling

Save changes Update table

	Consumer type	Cooling method	Cooling capacity	Heating method	Heating fuel	Heating capacity	Solar option	Solar capacity
C1	GRANVIA2	Compression	3000.0	Boiler	HFO	3750.0	No	0.0
C2	HOTEL SB	Absorption	500.0	Boiler	Chips	625.0	No	0.0
C3	PORTA FIRAL AUDITORI	Compression	760.0	Boiler	HFO	950.0	No	0.0
C4	PORTA FIRAL AUDITORI	Compression	760.0	Boiler	HFO	950.0	No	0.0
C5	FIRA PABELLONES 5-7	Compression	4329.0	Boiler	HFO	5411.25	No	0.0
C6	SERTRAM	Absorption	540.0	CHP	HFO	675.0	Yes	200.0

Figure 22. Building specific reference case.

In **Reference case** tab, user can examine and modify building specific reference case details by pressing **Update table** button (see Figure 22). In this table each consumer unit is listed with fitted cooling method capacity, that is, cooling capacity is calculated from peak load power divided by cooling efficiency. Furthermore, in the case of absorption cooling heat supply capacity is calculated by dividing cooling capacity by heating efficiency. Finally, there is option for user to include solar collectors in the building. User can choose between options by using combo boxes and then **Save changes**. Note - this saving affects only current data structure and this change does not make any changes in case folder data files.

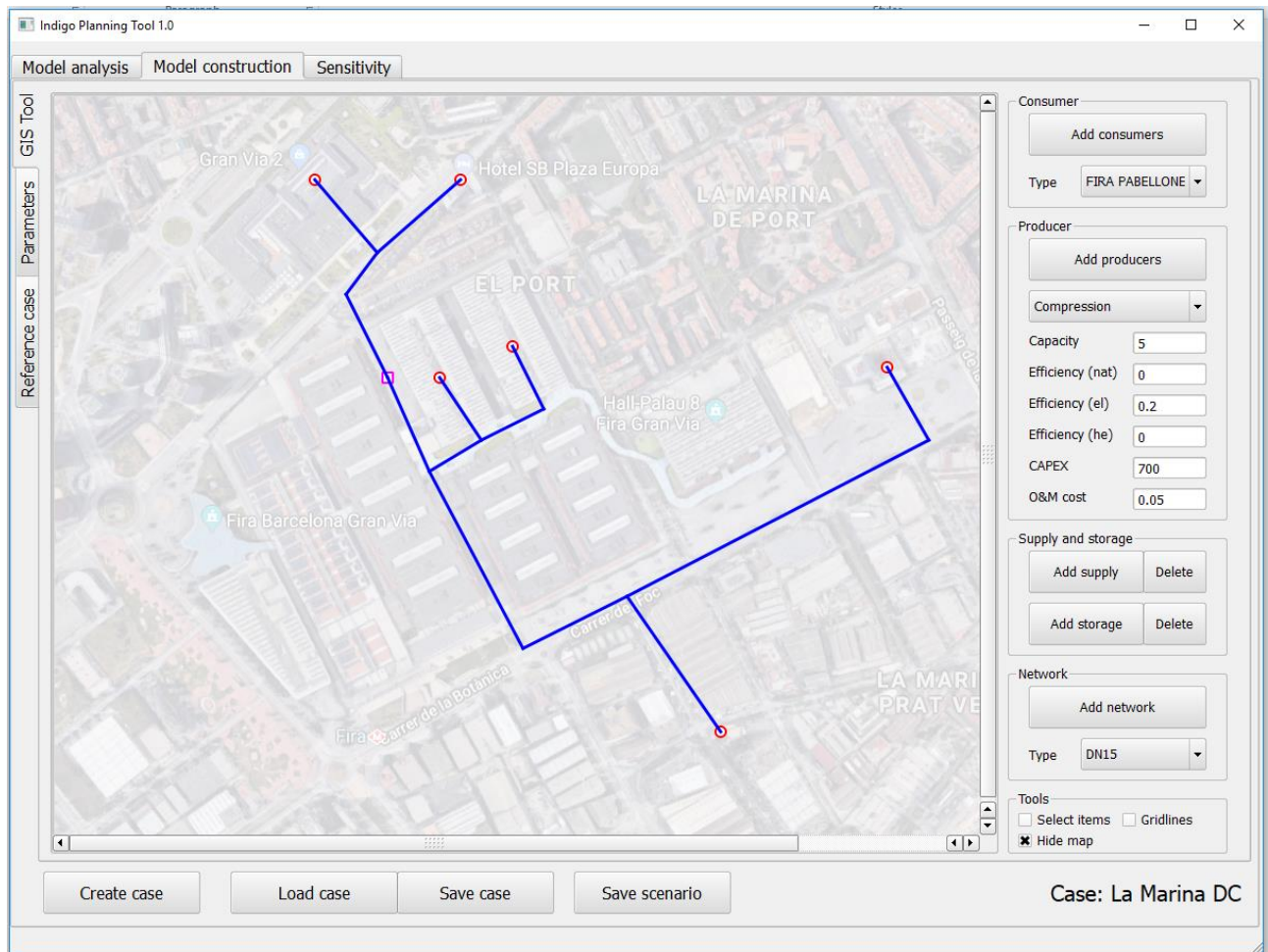
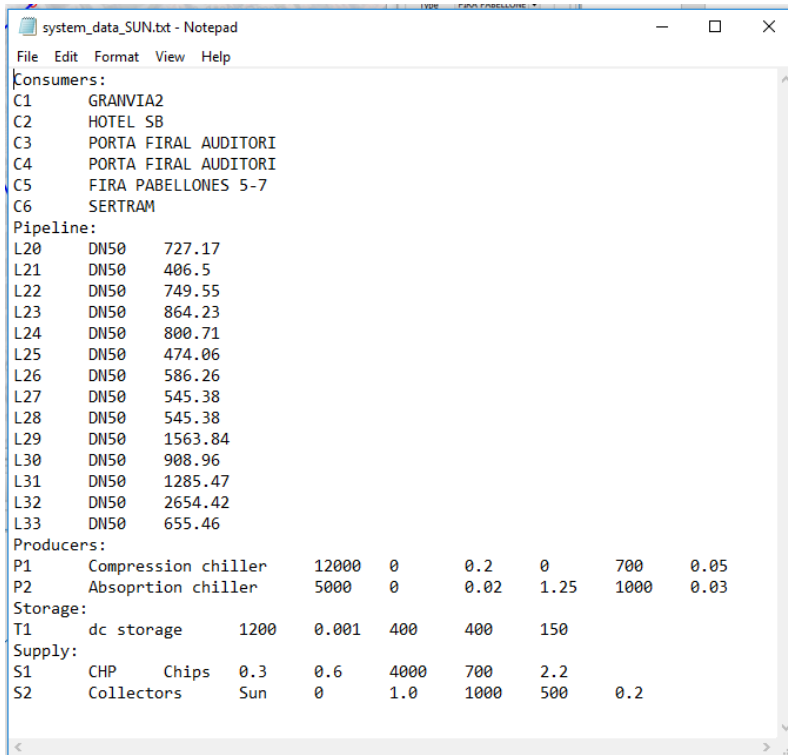


Figure 23. Finished DC model.

User can save model any time by pressing **Save case** button (see Figure 23). This saving procedure saves all the data structures, and these data structures are recovered from files when user loads case, that is, after loading user continues from the state in which model when saved. Also, **Save case** produces BASE case model files for optimization. If user makes modifications into model and presses **Save scenario** button, planning tool creates a snapshot of the current model in terms of model files (see Figure 24 for scenario model file example). Note - **Save scenario** button does not save data structures.



```

system_data_SUN.txt - Notepad
File Edit Format View Help
Consumers:
C1 GRANVIA2
C2 HOTEL SB
C3 PORTA FIRAL AUDITORI
C4 PORTA FIRAL AUDITORI
C5 FIRA PABELLONES 5-7
C6 SERTRAM
Pipeline:
L20 DN50 727.17
L21 DN50 406.5
L22 DN50 749.55
L23 DN50 864.23
L24 DN50 800.71
L25 DN50 474.06
L26 DN50 586.26
L27 DN50 545.38
L28 DN50 545.38
L29 DN50 1563.84
L30 DN50 908.96
L31 DN50 1285.47
L32 DN50 2654.42
L33 DN50 655.46
Producers:
P1 Compression chiller 12000 0 0.2 0 700 0.05
P2 Absorption chiller 5000 0 0.02 1.25 1000 0.03
Storage:
T1 dc storage 1200 0.001 400 400 150
Supply:
S1 CHP Chips 0.3 0.6 4000 700 2.2
S2 Collectors Sun 0 1.0 1000 500 0.2

```

Figure 24. Example of scenario model file.

8.2 Running the model and examining results

8.2.1 Start optimisation

When user has created models in Model construction tab, user proceeds to run models in Model analysis tab, which contains following sections:

1. **Run model:** In this tab, user launches optimization of selected scenario in selected case.
2. **Model results:** In this tab, user can analyze results from optimized model or load previous model results

Model optimization takes place in **Run model** tab (see Figure 25). User selects case combo box and then selects a scenario of this case from the scenario list. User can modify model period length in hours, but user is recommended to use hours of one year (8760 hours). If user wants to run in addition building specific reference case, user can fill **Run reference case** check box.

When optimization proceeds details of model and simulation progress is listed in **Simulation log**. Optimization is completed when simulation log states “*END - Optimization ready*”. Also, there can be two different notifications, when ready:

1. If optimization produces feasible result, notification is: *Optimal solution found - see results*
2. If no feasible solution is found, notification is: *Infeasible model - check model parameters*

Infeasible result indicates that parameters of DC system are set in a way that cooling demand cannot be covered with set of units in DC system.

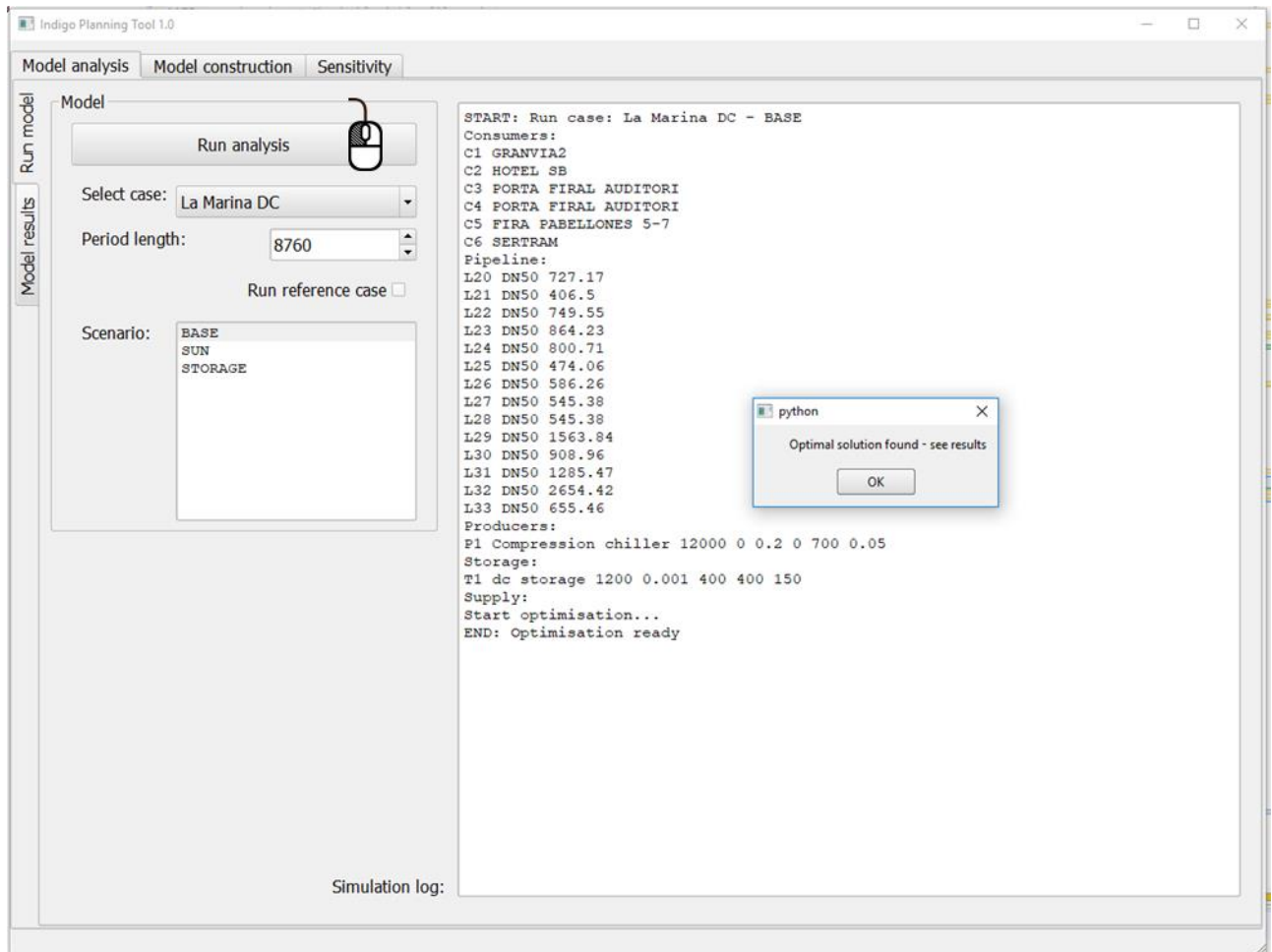


Figure 25. Running optimization model.

8.2.2 Examine results

After running the model and having received a feasible solution, model results can be examined in **Model results** tab (see Figure 26). In this tab there are two result sections:

1. **Result summary:** this tab contains operational result data and key performance indicators.
2. **Economy and emissions:** this tab contains economic data from case (and reference case) and emissions from DC system.

Result summary (see Figure 26) contains four sections:

1. Top row graph: this graph contains operational hourly data displayed for selected week. There are four different views, to be selected by user (see Figure 27) - DC production, heat supply, storage state and resource use.
2. Low left graph: this graph contains monthly sums of cooling production and heat supply.
3. Low middle table: this table contains utilization rates of production and supply units, and number of cycles in storage units.

4. Low right table: this table contains key performance indicators of DC system concerning costs and emissions.

In **Model results** tab previous results can be loaded by selecting case from combo box and pressing **Load result** button.

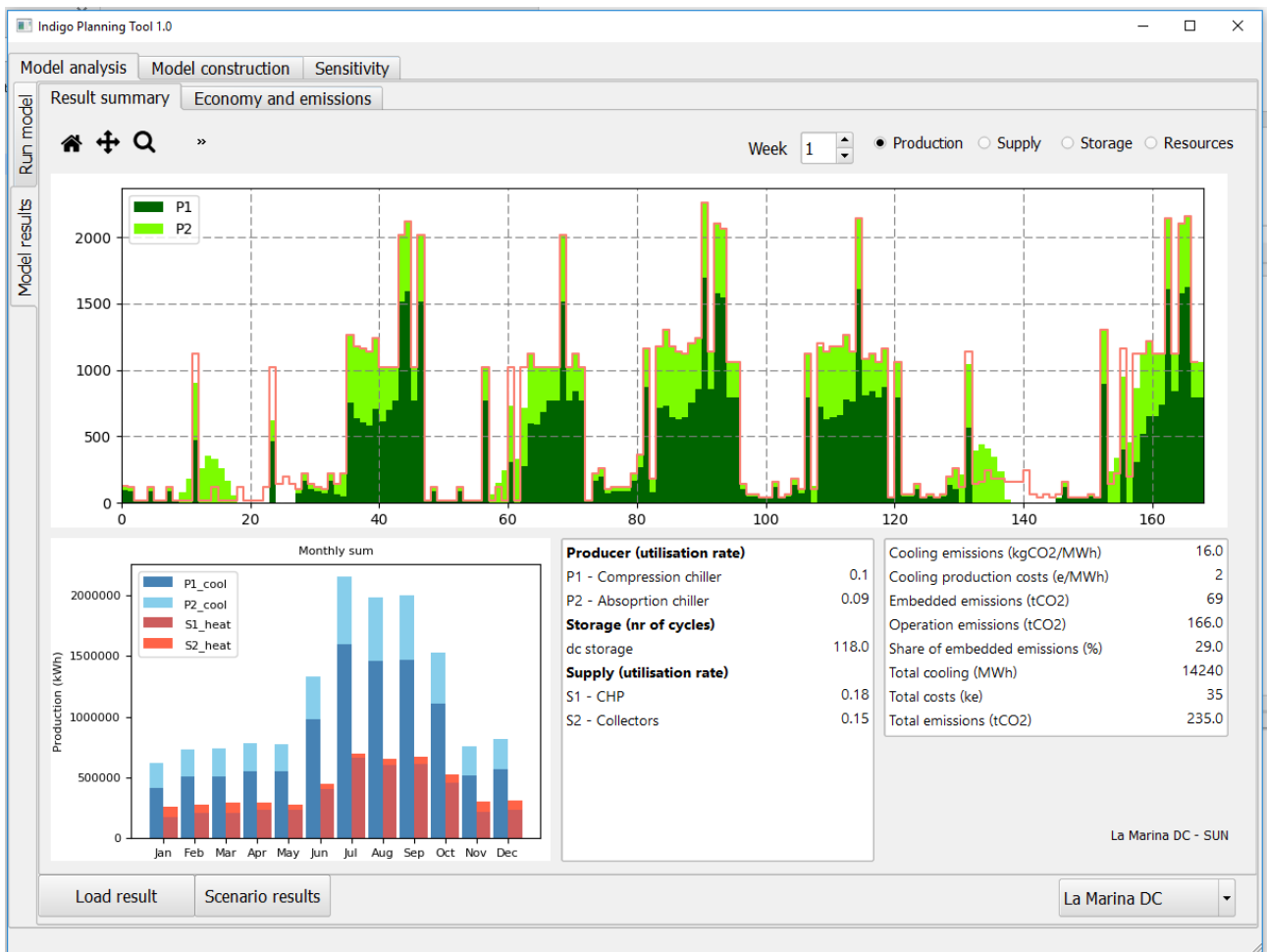


Figure 26. Result summary.

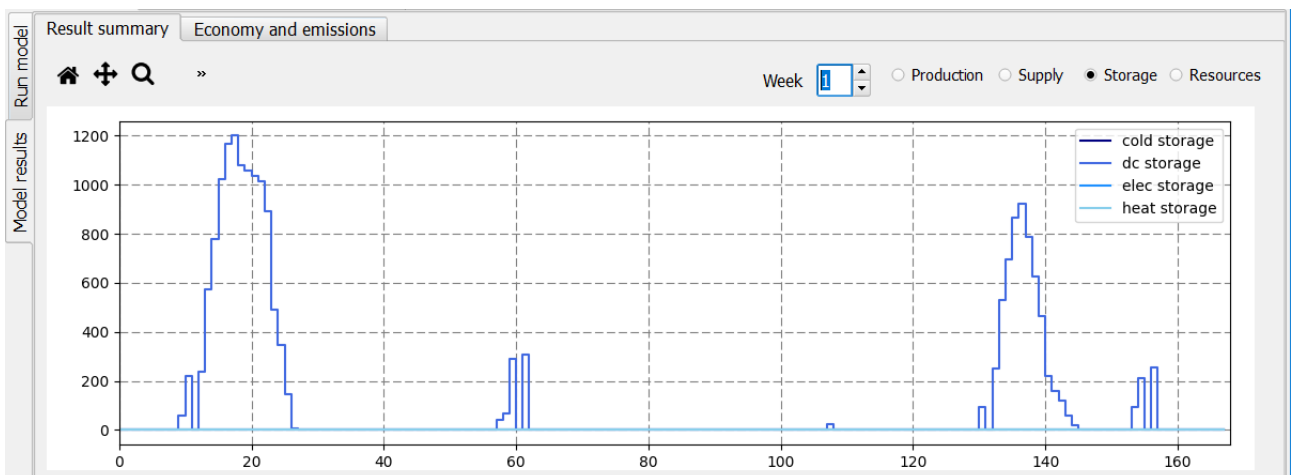


Figure 27. Operational data graph.

Economy and emissions (see Figure 28) contains two sections:

1. Economy of investment: In this section, cost indicators from case and reference case are compared. Profitability curve illustrates how investment cost is amortized by profits from DC system.
2. System emissions: In this section, emission based indicators are listed and system emissions divided into embedded and operational emissions are illustrated.

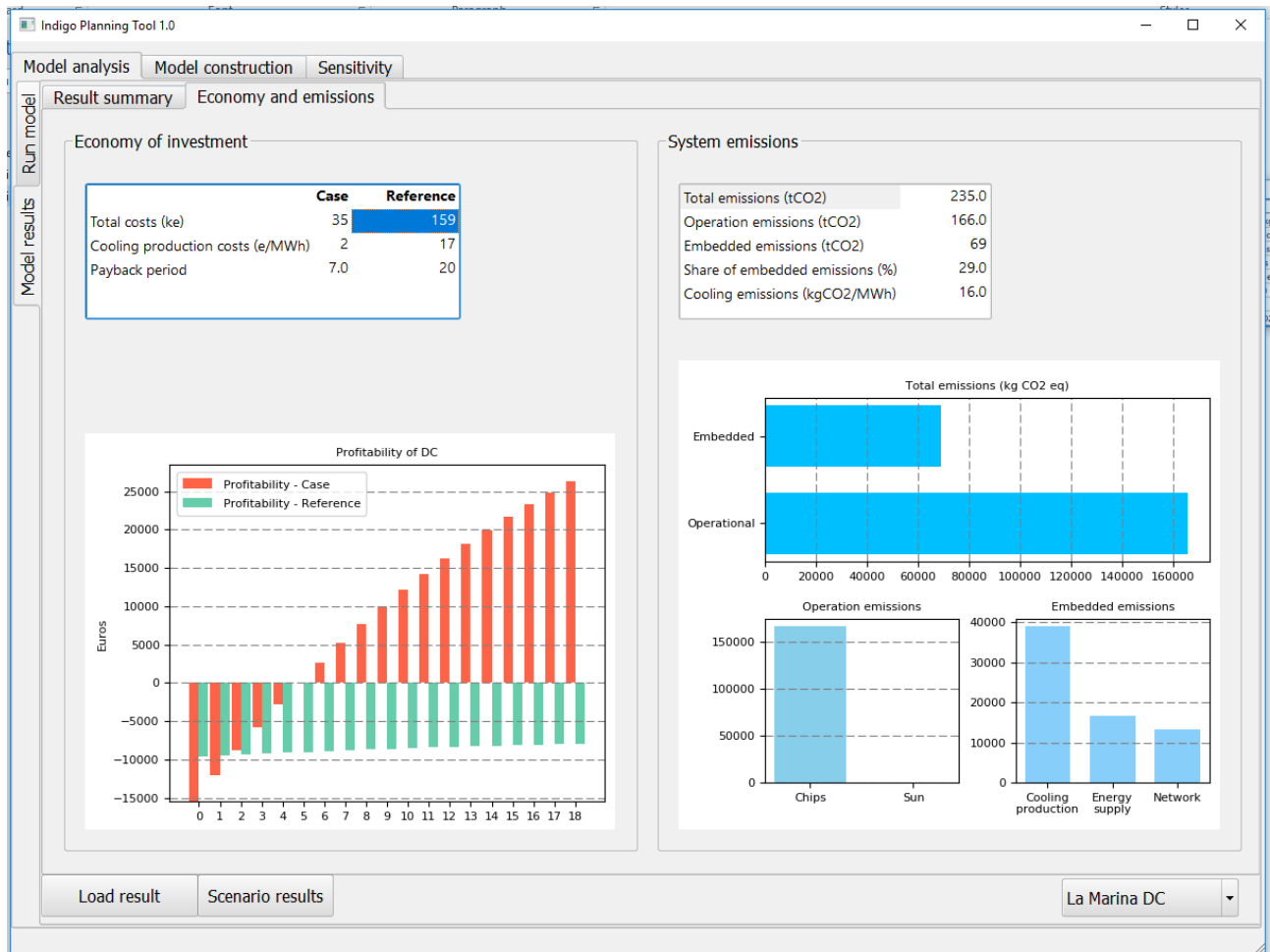


Figure 28. Economy and emission results.

In Model results tab user can view summary results from several scenarios of selected case by pressing **Scenario results** button. This button launches separate windows (see Figure 29) containing:

1. Graphs of monthly sums of cooling production and heat supply from all scenarios.
2. Table of key performance indicators from all scenarios.

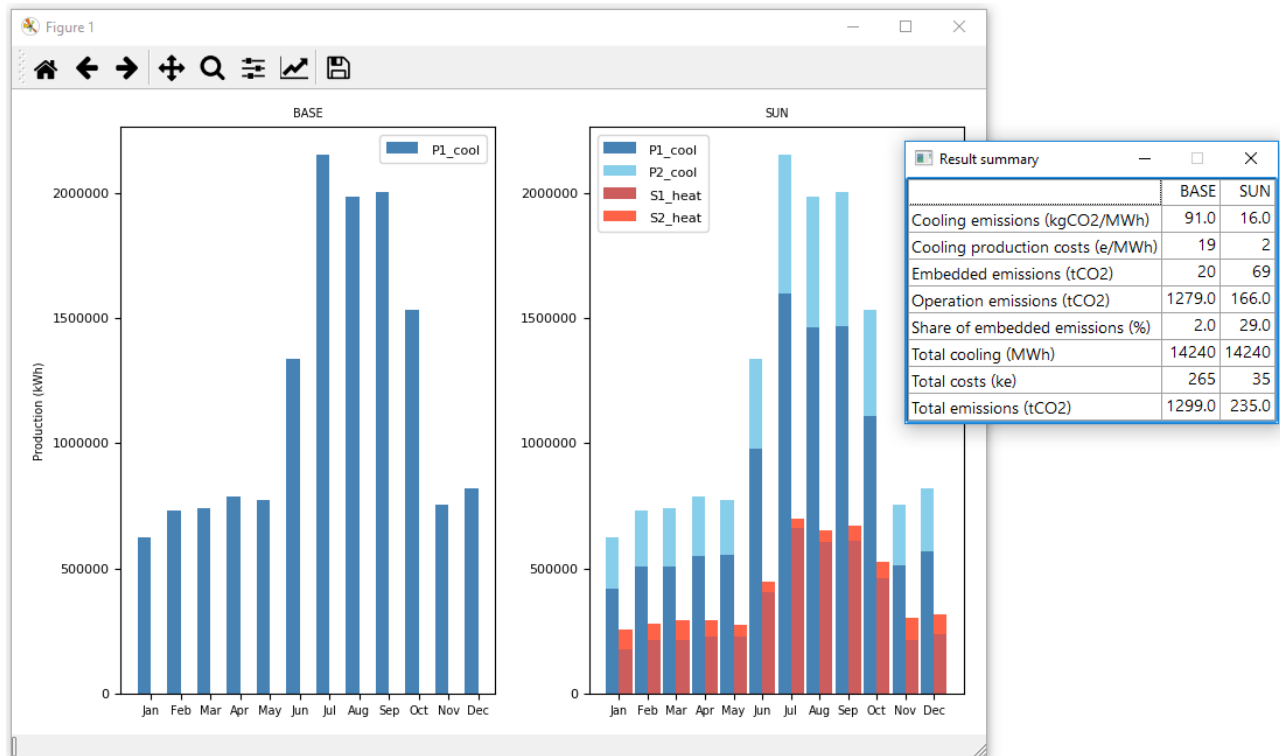


Figure 29. Scenario results.

8.3 Sensitivity analysis

In **Sensitivity** tab (see Figure 30) user can perform sensitivity analysis on selected case in terms of unit capacity parameter. User selects a case from combo box and then selects one scenario from this case. After selection, application presents list of units (producer, supply or storage) in this scenario. User can double-click one unit and activate that unit to be submitted to sensitivity analysis. Activated unit can be examined in Sensitivity case section (see Figure 31). User sees current capacity value of analyzed unit and can set minimum and maximum value and number of values (min + max + points between) to be simulated. By pressing **Run sensitivity analysis**, user initiates as many optimization runs as there are number of values defined. Progression of sensitivity analysis can be examined in sensitivity log.

Results from sensitivity analysis are illustrated in graph and table:

1. In top graph, total costs and total emissions are illustrated as function of capacity value.
2. In bottom table, key performance indicators are listed as function of capacity value.

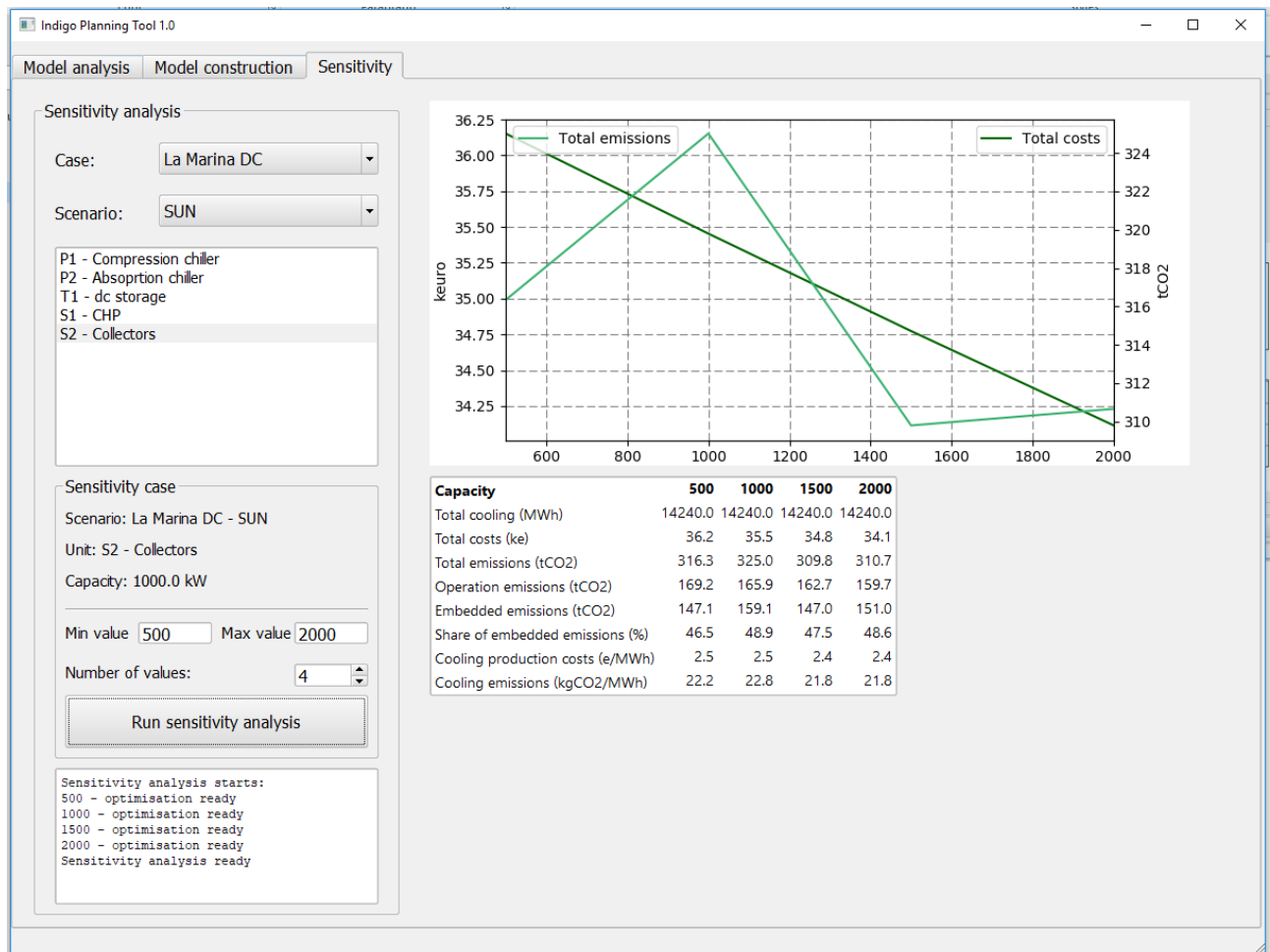


Figure 30. Sensitivity analysis.

Sensitivity case

Scenario: La Marina DC - SUN

Unit: S2 - Collectors

Capacity: 1000.0 kW

Min value: **500** Max value: **2000**

Number of values: **4**

Run sensitivity analysis

Sensitivity analysis starts:
500 - optimisation ready
1000 - optimisation ready
1500 - optimisation ready
2000 - optimisation ready
Sensitivity analysis ready

Figure 31. Unit activated for sensitivity analysis.

9 Further development plans

The planning tool development will continue during the last year of the project. Both testing and applying the tool for the case system studies within Task 5.3 are expected to provide guidance on how to make the tool more user-friendly and, in general, more useful.

However, some plans have already been made concerning further development. Implementation of some have already begun while others are still in planning process and their usefulness and required work effort will be evaluated during the testing and application phase. Few are minor additions, but some can potentially change the outlook of the planning tool. In this case, the changes will be reported within Appendixes of deliverables due during Task 5.3.

The current plans include:

- Refining the overall outlook of the tool based on user feedback during the Task 5.3
- Multi-criteria analysis (energy, economic, emissions) by ranking system performance on each category (1-5 points), allowing the user to define weighting for the categories
- More options for comparing different cases and more key performance indicators
- More indepth analysis of cases with multiple years; while calculation is possible, the results presentation is not complete
- Possibility and specification for reinvestments
- Fixed and (possible) connection costs for linking external sources such as electric grid
- Functionality making analysis space specific cooling systems easier to define, currently possible but practically very laborous

10 Summary

Within INDIGO, an open-source planning tool focused on the design stage of a DC system, has been developed. The purpose of the tool is to support the optimal design of new DC systems, and the assessment of existing DC systems' potential for performance/efficiency improvement. This report provides documentation of this new planning tool of a DC or a district heating system.

A limited background review of other ongoing projects with a similar topic showed that there is a need for tools for assessment of DHC systems. The tool being developed within INDIGO outstands from others by the approach and comprehensive capabilities, analysing the system from all energy, economic and environmental perspectives.

The aim of the planning tool is to be simple to use, and easily configurable. The tool enables the user to try out and compare different cooling systems and particularly to see the effects of different assumptions on component specifications, costs and other input data. With the tool, one can study the feasibility of a DC system, and compare it to the building and/or space specific systems. It thereby enables the demonstration of the benefits of DC. In addition, different configurations of a DC system can be compared.

The planning tool includes all the main components of a cooling system. However, details on external processes are excluded from it. These external processes include industrial activities providing excess heat for absorption based cooling production or electricity production other than solar PVs and CHP units located on-site. Also, the utilisation of excess heat within the specified system is excluded. (delete?)

The programs consists of two main parts; the calculation algorithms including related data structures and the graphical user interface (GUI). In addition, there are several predefined input data files and files used for storing the systems and cases being studied by the planning tool.

The calculation algorithms and related data structures have been structured as a group of classes representing different components in one or more cooling systems. The most important class is called System. It acts as an interface between the calculation and the GUI. While the GUI is specific to the INDIGO project and the planning tool, the classes themselves could also be utilized independently by using the interface in System class.

Two external Python libraries are utilized by the planning tool; Qt for the GUI implementation and oemof for the building and solving an optimization model for the cooling systems studied. The optimization model needs an external solver for the optimization process. In this case, a free solver recommended by the oemof community, the CBC (Coin-or branch and cut) solver is used.

System class includes all the needed functions for defining, optimizing and analyzing cooling systems by the planning tool as well as retrieving specific results, e.g. concerning a specific unit within a specific cooling system. The class acts as an interface between the calculation algorithms and the GUI. The class and its variables and functions are described in detail in Appendix A. The classes including their variables are described in Appendix B. The functions within the classes can be found in the program code.

Input data files used by the planning tool are all located under the folder “Data”. The data files are in tab-separated values format with an *.txt extension. Some input data files are shared by all cases studied by the planning tool while others are case specific. These case specific files are named according to their location, e.g. a country or a city.

The systems included in the model consist of district cooling systems, and space or building specific systems. For all of these systems, their typical emission profiles are included in the model. All the major energy and material flows related to the components needed for the systems are included in the planning tool in order to assess the GHG emission flows and the total emissions

After defining a cooling system in the planning tool, an optimization model can be built and solved. This dynamic generation based on user input is at the core of the planning tool.

The process involves heavy use of oemof library. In the oemof framework, an energy system consists of sources, sinks, transformers and buses between the first three. Energy storage is also available as a dynamic element. Sources are where the primary energy to run the system is drawn (e.g. fuels, electric grid) while the sinks represent cooling demand itself, and electric grid or heating system for possible surplus production. Transformers are components representing energy conversion units, such as CHPs or electric chillers, that in the end supply the cooling needed. Energy storages can be connected to the busses to store the corresponding commodity such as electricity or district cooling.

The output of the energy supply units or external sources of heating or cooling are then connected to a set of defined cooling production units. Similar to energy supply units, cooling production units have capacities and efficiencies defined. However, they all produce cooling. Possible storage units for electricity, heating and cooling (produced or external) may be defined according to the specifications given by the user.

The generated model is a simple linear programming (LP) model. Although oemof library would allow more complex mixed integer type of models (MILP) to be constructed, the increased calculation time with the free solvers available would make this less practical considering the original purpose of the planning tool.

The optimization can be run for a single representative year or a series of year-long optimization runs that take into account the projected development of commodity prices.

Several years' optimization is obviously a considerably slower process than one representative year. The optimization finds the least-cost solution for utilising a given system to supply a defined cooling demand. The costs consist of variable costs from the use of resources. Because the resource prices are defined as time series and can change (e.g. electricity), this can potentially affect how the system is operated. Another element where optimization is relevant is the utilisation of energy storages.

In economic analysis the results of the energy analysis (resource consumption in kWh) are used to calculate operational costs (€/kWh) and system definition, i.e. capacities for components (kW) within the system, is used for calculating investment costs (€) by using defined specific investments for each technology (€/kW). Other yearly operation and maintenance related costs are added as percentages of initial investment, also defined within the components of the system.

The environmental analysis in the planning tool follows the framework of life cycle assessment (LCA) and focuses on climate impacts of the district cooling systems. The planning tool enables assessment of the GHG emissions of the studied district cooling system. It also allows for comparison of the emission impacts between planned district cooling systems and reference cooling systems (i.e. space specific and building specific cooling systems). Thus, the tool will provide information on potential emission savings achieved with the district cooling systems and enables comparison of the emissions produced by the district cooling system and the reference systems. It also shows how the emissions are formed. For example, it provides the user with information on the contribution of different components and the energy consumption during the use. In addition, the user can calculate additional indicators, including the average annual emissions (allocated emission to components) / average annual cooling produced, and the distribution of the emissions over different life cycle phases.

Combining the results of the different analysis phases, combination indicators such as price of cooling or comparative indicators such as costs of emission reduction can be calculated. The planning tool will also include functionalities for performing a sensitivity analysis.

Program implementation in Python takes advantage of object-oriented programming for data structures and storage. The predefined input data are stored in csv files. Graphic user interface (GUI) of the planning tool is used to manage model data, structure, and pass these values into optimization module of the tool, and import result data from optimization in order to form easy to read result summary. GUI consists of five functional tabs: Analysis, GIS Tool, Parameters, Results and Sensitivity Analysis. GIS tool of GUI is utilised in forming simple but functional network of cooling system based on map file of the area analysed.

The potential users for the planning tool include municipalities, master developers for a specific area and energy utilities. The results are useful as feasibility and pre-feasibility studies as well as for combining cooling and energy planning with the city planning processes for both new and built areas. The results also provide input for a more detailed, technical plans. The accuracy of the results can be greatly enhanced if coupled with a dedicated simulation model or measurement data for cooling demand. The model generation and optimisation process can also be useful for research purposes.

11 References

- Ardente, F., Beccali, M., Cellura, M., & Mistretta, M. 2011. Energy and environmental benefits in public buildings as a result of retrofit actions. *Renewable and Sustainable Energy Reviews*, 15(1), 460-470.
- Bauer 2008. Life Cycle Assessment of Fossil and Biomass Power Generation Chains. An analysis carried out for the ALSTOM Power Services. PSI Bericht Nr. 08-05. Paul Scherrer Institut. Technology Assessment / GaBE.
- Beccali, M., Cellura, M., Longo, S., Nocke, B., & Finocchiaro, P. (2012). LCA of a solar heating and cooling system equipped with a small water-ammonia absorption chiller. *Solar Energy*, 86(5), 1491-1503.
- Beccali, M., Cellura, M., Ardente, F., Longo, S., Nocke, B., Finocchiaro, P., Kleijer, A., Hildbrand, C., Bony, J., 2010. Life Cycle Assessment of Solar Cooling Systems – A technical report of subtask D Subtask Activity D3, Task 38 Solar Air-Conditioning and Refrigeration, IEA. Solar Heating & Cooling Programme.
- Bøhm B, Kristjansson H. Single, twin and triple buried heating pipes: on potential savings in heat losses and costs. *Int J Energy Res* 2005;29:1301–12. doi:10.1002/er.1118.
- Corona, B., Escudero, L., Quéméré, G., Luque-Heredia, I. and San Miguel, G., 2017. Energy and environmental life cycle assessment of a high concentration photovoltaic power plant in Morocco. *The International Journal of Life Cycle Assessment*, 22(3): 364-373.
- Frischknecht, R., Itten, R., Wyss, F., Blanc, I., Heath, G., Raugei, M., Sinha, P. and Wade, A., 2015. Life cycle assessment of future photovoltaic electricity production from residential-scale systems operated in Europe. Subtask 2.0. "LCA", IEA-PVPS, Task 12. International Energy Agency, Geneva, Switzerland.
- Fröling, M., Holmgren, C., & Svanström, M. (2004). Life cycle assessment of the district heat distribution system. *The International Journal of Life Cycle Assessment*, 9(2), 130-136.
- Fröling, M., & Svanström, M. (2005). Life cycle assessment of the district heat distribution system-Part 2: network construction (11 pp). *The International Journal of Life Cycle Assessment*, 10(6), 425-435.
- Hang, Y., Qu, M., Winston, R., Jiang, L., Widyolar, B., & Poiry, H. (2014). Experimental based energy performance analysis and life cycle assessment for solar absorption cooling system at University of Californian, Merced. *Energy and Buildings*, 82, 746-757.
- Heath, G., Turchi, C., Decker, T., Burkhardt, J., & Kutscher, C. 2009. Life cycle assessment of thermal energy storage: two-tank indirect and thermocline. In *ASME 2009 3rd International Conference on Energy Sustainability collocated with the Heat Transfer and InterPACK09 Conferences* (pp. 689-690), July 19-23, 2009. American Society of Mechanical Engineers.
- JEC - Joint Research Centre-EUCAR-CONCAWE collaboration. (2014). Well-to-Wheels analysis of future automotive fuels and powertrains in the European context. Well-to-Tank report, Version 4a. European Commission, Joint Research Centre, Institute for Energy and Transport. Available in: https://iet.jrc.ec.europa.eu/about-jec/sites/iet.jrc.ec.europa.eu/about-jec/files/documents/report_2014/wtt_appendix_4_v4a.pdf

- Majeau-Bettez, G., Hawkins, T.R. and Strømman, A.H., 2011. Life cycle environmental assessment of lithium-ion and nickel metal hydride batteries for plug-in hybrid and battery electric vehicles. *Environmental science & technology*, 45(10), pp.4548-4554.
- Nouidui, T.S.; Wetter, M.; Zuo, W. 2014. Functional mock-up unit for co-simulation import in EnergyPlus, *Journal of Building Performance Simulation*, 7(3): 192–202. DOI: <http://dx.doi.org/10.1080/19401493.2013.808265>
- Oró, E., Gil, A., De Gracia, A., Boer, D., & Cabeza, L. F. (2012). Comparative life cycle assessment of thermal energy storage systems for solar power plants. *Renewable Energy*, 44, 166-173.
- Raluy, R. G., Serra, L. M., Guadalfajara, M., & Lozano, M. A. (2014). Life cycle assessment of central solar heating plants with seasonal storage. *Energy Procedia*, 48, 966-976.
- Rämä, M.; Klobut, K.; Pursiheimo, E.; Koponen, K.; Sokka, L. 2017. District cooling planning tool framework and specification. INDIGO project deliverable D5.1.
- Shah, V.P. David, Debella, C., Ries, R.J. 2008. Life cycle assessment of residential heating and cooling systems in four regions in the United States, *Energy and Buildings* 40 (4): 503-513.

Reviewed projects

- OPTi project. Available online: <http://www.opti2020.eu> (accessed on 5 July 2018)
- FLEXYNETS project. Available online: <http://www.flexynets.eu> (accessed on 5 July 2018)
- STORM project. Available online: <http://www.storm-dhc.eu> (accessed on 5 July 2018)
- InDeal project. Available online: <http://www.indeal-project.eu> (accessed on 5 July 2018)
- H-DisNet project. Available online: <http://www.h-disnet.eu> (accessed on 5 July 2018)
- E2District project. Available online: <http://e2district.eu> (accessed on 5 July 2018)
- CoolHeating project. Available online: <http://www.coolheating.eu> (accessed on 5 July 2018)
- PlanHeat project. Available online: <http://planheat.eu> (accessed on 5 July 2018)
- SDHp2m project. Available online: <http://solar-district-heating.eu> (accessed on 5 July 2018)
- THERMOSS project. Available online: <http://thermoss.eu> (accessed on 5 July 2018)
- THERMOS project. Available online: <http://www.thermos-project.eu> (accessed on 5 July 2018)
- HotMaps project. Available online: <http://www.hotmaps-project.eu> (accessed on 5 July 2018)
- DHAT tool. Available online (accessed on 5 July 2018): <https://ens.dk/en/our-responsibilities/global-cooperation/district-heating-assessment-tool-dhat>

12 Appendix A: class System documentation

Table 7. System class definition; variables and functions.

class System	
Variables	
<i>systems</i> (list, CoolingSystem)	List of cooling systems included in the study, normally one in the case of district cooling or many in the case building or space specific cooling systems.
<i>resources</i> (list, Resource)	List of resources (and their properties) shared by all the cooling systems, e.g. fuels, electricity grid.
<i>interest_rate</i> (float)	Value for the interest rate used in economic analysis.
<i>costs</i> (list, float)	Cumulative yearly costs as a result of the economic analysis.
<i>profitability</i> (list, float)	Cumulative yearly costs including revenues from cooling sold.
Functions	
def __init__ ()	
Class constructor, sets up the default values or initializes the variables. Default interest rate is set to 5 %, no input parameters are needed	
def initialise (location)	
<i>location</i> (string)	Name of the country which the system is located, corresponds to input data files <i>resource_<location>.txt</i> and <i>resource_projection_<location>.txt</i> located in <i>Data</i> subfolder.
Initialises resource data (yearly times series, future projections) . Needs to be done before any analysis.	
def setCoolingPrices (allSystems, systemName, energyTariff, capacityCharge)	
<i>systemName</i> (string)	name of the system for which the prices are set, an empty string means the change will be done for all systems in systems (list, CoolingSystem).
<i>energyTariff</i> (float)	a new price for energy in €/MWh
<i>capacityCharge</i> (float)	a new yearly capacity charge for the cooling connection (€/kW)
Defines prices for consumers for a defined system or for all systems.	
def initNetwork (systemName, scenName)	
<i>systemName</i> (string)	Name of the system for which the network is initialized.
<i>scenName</i> (string)	Name of the corresponding scenario.
Initialises the district cooling network for previously specified case.	
def setResourceStatus (resource, status)	
<i>resource</i> (string)	Name of the resource for which the change is made.
<i>status</i> (boolean)	Whether to enable (True) or disable (False) a resource.
Enables or disables the status of a specific resource. This is done by specifying a very high price for commodities (fuels, electricity, external sources) or very low value in case of solar irradiation.	
def setYear (year)	

year (int) Years from start of the analysis, from 1 to e.g. 20 years.

Sets a year used in defining commodity prices in optimization. Based on projected prices, see **initResources** for details.

def **addCoolingSystem**(name, hasNetwork, tariff, charge)

name (string) Name for the cooling system to be created.
hasNetwork (boolean) Whether the new system has a distribution network (DC)
tariff (float) Energy tariff for the new cooling system (€/MWh)
charge (float) Capacity charge for the new cooling system (€/kW)

Adds a cooling system to be studied. A *name*, energy *tariff* and capacity *charge* needs to be defined and whether the system is a DC system (*hasNetwork*, *True* or *False*)

def **addConsumer**(systemName, id_number, lat, lon, demand)

systemName (string) Name of the system where the consumer is added.
id_number (int) Identification number for further modifications.
lat, lon (float) Location of the consumer added.
demand (list, float) Cooling demand of the consumer, year-long time series.

Adds a consumer into a specified system.

def **addProducer**(systemName, id_number, lat, lon, cap, ne, ee, he, inv, om)

systemName (string) Name of the system where the producer is added.
id_number (int) Identification number for further modifications.
lat, lon (float) Location of the unit added.
cap (float) Cooling capacity (in kW)
ne (float) Natural efficiency (unit of resource consumed / unit of cooling produced)
ee (float) Electric efficiency (unit of electricity consumed / unit of cooling produced)
he (float) Heat efficiency (unit of heat consumed / unit of cooling produced)
inv (float) Specific investment corresponding to the technology (€/kW)
om (float) Operational and maintenance costs (% of investment, 0-1)

Adds a producer (of cooling) into a specified system.

def **addSimpleProducer**(systemName, tech, id_number, cap)

systemName (string) Name of the system where the producer is added.
tech (string) Technology of the production unit added.
id_number (int) Identification number for further modifications.
cap (float) Cooling capacity (in kW)

Adds a producer into a specified system, but by providing a name for the technology ("Compression chiller", "Absorption chiller" or "Free cooling") and retrieving other parameters from input data file *cooling_techs.txt* in *Data* subfolder.

def **addSupply**(systemName, id_number, tech, resource, ee, he, cap, inv, om)

systemName (string) Name of the system where the energy supply unit is added.
id_number (int) Identification number for further modifications.
tech (string) Technology of the energy supply unit added.
resource (string) The resource the energy supply unit is using
ee (float) Electric efficiency (0-1, can be 0 if no electricity production)
he (float) Heat efficiency (0-1, can be 0 if no heat production)
cap (float) Total capacity and in case of CHP unit; fuel capacity (in kW, m² for solar)
inv (float) Specific investment (€/kW) of capacity
om (float) Operation and maintenance costs (% of investment, 0-1)

Adds an energy supply unit into a specified system. Solar PVs and collectors do not utilize efficiencies, and consider capacity as m², not kW.

```
def addSimpleSupply(systemName, tech, fuel, id_number, cap)
```

<i>systemName (string)</i>	Name of the system where the energy supply unit is added.
<i>id_number (int)</i>	Identification number for further modifications.
<i>tech (string)</i>	Technology of the energy supply unit added.
<i>fuel (string)</i>	The fuel the energy supply unit is burning. If no fuel, can be left empty.
<i>cap (float)</i>	Total capacity and in case of CHP unit; fuel capacity (in kW, m ² for solar)

Adds an energy supply unit into a specified system, but with less input. The parameters are read from *supply_techs.txt* in *Data* subfolder.

```
def defineStorage(systemName, storageType, props)
```

<i>systemName (string)</i>	Name of the system where the energy supply unit is added.
<i>storageType (string)</i>	Identification number for further modifications.
<i>props (list, float)</i>	Parameters for defining the storage.

Adds a energy storage into a specified system. The valid *storageType* values are "dc storage", "elec storage", "heat storage", "cold storage" indicating what is stored. The parameter *props* is a 9 element long list (float) including capacity (kWh), losses, initial charge (kWh), loading cost (€/kWh), unloading cost, loading ratio (% of capacity per hour), unloading ratio, loading efficiency, unloading efficiency and investment (€/kWh).

```
def defineSimpleStorage(systemName, storageType, cap, loss, load, unload, inv)
```

<i>systemName (string)</i>	Name of the system where the energy supply unit is added.
<i>storageType (string)</i>	Identification number for further modifications.
<i>cap (float)</i>	Capacity of the storage (kWh)
<i>loss (float)</i>	Loss factor (0-1)
<i>load (float)</i>	Loading capacity (kWh/h)
<i>unload (float)</i>	Unloading capacity (kWh/h)
<i>inv (float)</i>	Specific investment for the storage (€/kWh)

Adds a energy storage into a specified system, similarly than *defineStorage()* function. The resulting storage is more simple (no costs, 100 % efficiency, initial charge zero).

```
def runOptimisation(systemName, period)
```

<i>systemName (string)</i>	Name of the system optimized, if left blank runs all the systems specified.
<i>period (int)</i>	Length of the optimization in hours

Runs the optimization model returning 1 if succesful (optimal) and 0 if not. The results of the optimization can be retrieved by functions described below and a result text file *Optimisation results (<systemName>).txt* is also generated.

```
def runOptimisationForMultipleYears(systemName, years)
```

<i>systemName (string)</i>	Name of the system optimized, if left blank runs all the systems specified.
<i>years (int)</i>	Number of years optimized.

Runs the optimization model similarly to the previous function, but for several years taking into account the projected costs of resources (see *initResources()* function). Warning! The optimization can take a lot of time.

```
def getUsedResources(systemName)
```

<i>systemName (string)</i>	Name of the system for which the used resources are retrieved.
----------------------------	--

Returning results (time series) of the optimization from a specified system; a dictionary with the name of the used resource as a key, and resource consumption (in kWh) as a value.

```
def getSupplyInput(systemName, id_number)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>id_number (int)</i>	Id number of the energy supply unit.

Returning results (time series) of the optimization from a specified system; resource consumption of a specified energy supply unit (kWh/h).

```
def getSupplyOutput(systemName, id_number, commodity)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>id_number (int)</i>	Id number of the energy supply unit.
<i>commodity</i>	Commodity produced by the unit ("Electricity" or "Heat")

Returning results (time series) of the optimization from a specified system; energy production of a specified commodity by a specified energy supply unit (kWh/h).

```
def getProducerInput(systemName, id_number, commodity)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>id_number (int)</i>	Id number of the cooling production unit.
<i>commodity</i>	Commodity consumed by the unit ("Electricity", "Heat" or "Natural")

Returning results (time series) of the optimization from a specified system; energy consumption of a specified commodity by a specified cooling production unit (kWh/h).

```
def getProducerOutput(systemName, id_number)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>id_number (int)</i>	Id number of the cooling production unit.

Returning results (time series) of the optimization from a specified system; cooling production of a specified unit (kWh/h).

```
def getStorageInput(systemName, storageType)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>storageType (string)</i>	Type of the storage (see defineStorage() for valid format)

Returning results (time series) of the optimization from a specified system and specific storage type; what is loaded into the storage.

```
def getStorageOutput(systemName, storageType)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>storageType (string)</i>	Type of the storage (see defineStorage() for valid format)

Returning results (time series) of the optimization from a specified system and specific storage type; what is unloaded from the storage.

```
def getStorageState(systemName, storageType)
```

<i>systemName (string)</i>	Name of the system for which the results are retrieved.
<i>storageType (string)</i>	Type of the storage (see defineStorage() for valid format)

Returning results (time series) of the optimization from a specified system and specific storage type; what is the state (charge in kWh) of the storage.

```
getTotalDemand(systemName)
```

systemName (string) Name of the system for which the results are retrieved.

Returning results (time series) of the optimization from a specified system; the total cooling demand (in kWh/h, including possible distribution losses) of a specified system.

getCoolingDemand(systemName)

systemName (string) Name of the system for which the results are retrieved.

Returning results (time series) of the optimization from a specified system; the cooling demand (kWh/h) of all consumers for a specified system.

getUtilisationRate(systemName, isSupply, id_number)

systemName (string) Name of the system for which the results are retrieved.
isSupply (boolean) Indication if the unit in question is an energy supply unit.
id_number (int) Id number of the unit

Returning results of the optimization from a specified system; the utilisation rate of an energy supply or a cooling production unit with a specified id number.

getNumberOfCycles(systemName, storageType)

systemName (string) Name of the system for which the results are retrieved.
storageType (string) Type of the storage (see defineStorage() for valid format)

Returning results of the optimization from a specified system; the number of cycles (i.e. loading capacity / total storage capacity) for a specified *storageType* (see defineStorage() for details).

runEconomicAnalysis(years)

years (int) Number of years the analysis is run.

Runs the economic analysis for a specified number of *years* based on optimization results. Calculates and saves cumulative costs and cumulative costs and revenues into variables of the System class. The interest rate (variable of the System class) is used to calculate discount factor for each year. All investments are taking place at year 0.

getCumulativeCosts()

Returns the cumulative costs list (float) from the economic analysis.

getCumulativeCostsAndRevenues()

Returns the cumulative costs and revenues list (float) from the economic analysis.

getPaybackPeriod()

Calculates and returns the payback period based on cumulative costs and revenues calculated in the economic analysis.

runEconomicAnalysisForMultipleYears(optimisationRuns)

optimisationRuns (list, Cooling system) List of individual optimized cooling systems.

Runs the economic analysis for optimization runs that are based on changing commodity prices (see runOptimisationForMultipleYears() for details). The length of the analysis is the same as number of years optimized. As with the runEconomicAnalysis() function, investments are all taking place at year 0.

getOperationalEmissions(systemName, location)

systemName (string) Name of the system for which the results are retrieved, if left blank gets the results from all specified systems.

location (string)

Name of the country which the system is located, corresponds to input data files *emissions_<location>.txt* located in *Data* subfolder.

Calculates and returns the yearly operational emissions (kgCO₂) of a specified system. As specific emissions can be country specific, *location* is given as a parameter and used in finding right values in *emissions_<location>.txt* data file in *Data* subfolder.

getEmbeddedEmissions(systemName)

systemName (string)

Name of the system for which the results are retrieved, if left blank gets the results from all specified systems.

Calculates and returns the embedded emissions of energy supply, cooling production and the (possible) distribution network as a list (float, kgCO₂). Data files *emissions_supply.txt*, *emissions_cooling.txt* and *emissions_storage.txt* in *Data* subfolder are used. Life time of the units is taken into account making the results comparable with yearly operational emissions.

getResultsSummary(systemName, location)

systemName (string)

Name of the system for which the results are retrieved, if left blank gets the results from all specified systems.

location (string)

Name of the country which the system is located, corresponds to input data files *emissions_<location>.txt* located in *Data* subfolder.

Calculates and returns a dictionary representing a summary of results; total demand, costs and emissions, operational and embedded emissions, share of embedded emissions in total emissions, cooling production costs and cooling emissions.

13 Appendix B: Component classes documentation

Table 8. System class definition; variables and functions.

class CoolingSystem
Description <p>Class CoolingSystem contains all the specifications for a single cooling system and its components. This system can be a DC, a building or a space specific system. The model generation launched from the class System takes place within class CoolingSystem.</p>
Variables <p><i>name (string)</i> Name of the cooling system, used for identification purposes. Needs to be unique among cooling systems.</p> <p><i>supply (list, EnergySupply)</i> A list of energy supply units within the cooling system.</p> <p><i>consumers (list, Consumer)</i> A list of consumers within the cooling system.</p> <p><i>producers (list, Producer)</i> A list of cooling production units within the cooling system.</p> <p><i>network (Network)</i> In case of district cooling, the network element of the cooling system.</p> <p><i>mastersystem (System)</i> A reference to the master system where the cooling system specified is part of.</p> <p><i>energyTariff (float)</i> Price for cooling energy (€/kWh) used in economic analysis.</p> <p><i>capacityCharge (float)</i> Connection charge for a single consumer (€/kW) used in economic analysis.</p> <p><i>network (Network)</i> A definition of the distribution network in case of DC system.</p> <p><i>usedResources (dict)</i> Results dictionary containing the amount of used resource. Specified after optimizing the system, key is the name of the used resource and value a time series resource use (kWh/h).</p> <p><i>totalCosts (float)</i> The total operational costs as a result of the optimisation.</p> <p><i>storages (dict)</i> A dictionary containing the possible storage units specified for the cooling system. The key is the storage type (acceptable values being "dc storage", "elec storage", "heat storage" or "cold storage") and the value an instance of the EnergyStorage class.</p> <p><i>period (int)</i> The length of the optimized period in hours.</p> <p><i>year (int)</i> The year for which the optimization is carried out, counting onwards from the year of commission (1).</p>

Table 9. EnergySupply class definition; variables.

class EnergySupply
Description <p>Class EnergySupply contains the specification for a energy supply unit. It is mainly used as data structure with functions within class CoolingSystem accessing its variables.</p>
Variables <p><i>node_id (int)</i> Identification number for identifying the energy supply unit. Must be unique among energy supply units.</p> <p><i>name (string)</i> Name for the energy supply unit, generated automatically by using the name of the technology and id.</p> <p><i>tech (string)</i> Name of the technology the energy supply is representing.</p> <p><i>resource (string)</i> The resource the energy supply unit is consuming/utilising when producing energy.</p> <p><i>elec_eff (float)</i> Electricity production efficiency (0-1) of the energy production unit.</p> <p><i>heat_eff (float)</i> Heat production efficiency (0-1) of the energy production unit.</p> <p><i>cap (float)</i> Capacity of the unit. In case of CHP unit, the fuel capacity.</p> <p><i>spec_inv (float)</i> Specific investment (€/kW)</p> <p><i>om (float)</i> Operation and maintenance costs (0-1) as % of the total investment.</p> <p><i>lifetime (int)</i> Life time of the unit in years.</p> <p><i>elec_production (list, float)</i> Results of the optimization; electricity production time series of the unit (kWh/h).</p> <p><i>heat_production (list, float)</i> Results of the optimization; heat production time series of the unit (kWh/h).</p> <p><i>consumption (list, float)</i> Results of the optimization; resource consumption of the unit (kWh/h).</p> <p><i>utilisation (float)</i> Results of the optimization; utilisation rate (0-1).</p>

Table 10. Producer class definition; variables.

class Producer
Description <p>Class Producer contains the specification for a cooling production unit. Same as EnergySupply, it is mainly used as data structure.</p>
Variables <p><i>node_id (int)</i> Identification number for identifying the energy supply unit. Must be unique among energy supply units.</p> <p><i>name (string)</i> Name for the cooling production unit, generated automatically by combining string “Chiller” and id.</p> <p><i>capacity (float)</i> Cooling production capacity (kW).</p> <p><i>natr_eff (float)</i> Conversion efficiency from a natural source to cooling production (natural source / cooling produced)</p> <p><i>elec_eff (float)</i> Conversion efficiency from electricity to cooling production (electricity / cooling produced)</p> <p><i>heat_eff (float)</i> Conversion efficiency from heat to cooling production (heat / cooling produced)</p> <p><i>natr_eff (float)</i> Conversion efficiency from a natural source to cooling production (natural source / cooling produced)</p> <p><i>spec_inv (float)</i> Specific investment (€/kW)</p> <p><i>natr_cons (list, float)</i> Result of the optimization; time series of consumption of a natural source (kWh/h).</p> <p><i>elec_cons (list, float)</i> Result of the optimization; time series of consumption of electricity (kWh/h).</p> <p><i>heat_cons (list, float)</i> Result of the optimization; time series of consumption of heat (kWh/h).</p> <p><i>production (list, float)</i> Result of the optimization; time series of cooling production (kWh/h).</p> <p><i>utilisation (float)</i> Result of the optimization; utilisation rate of the cooling production unit.</p>

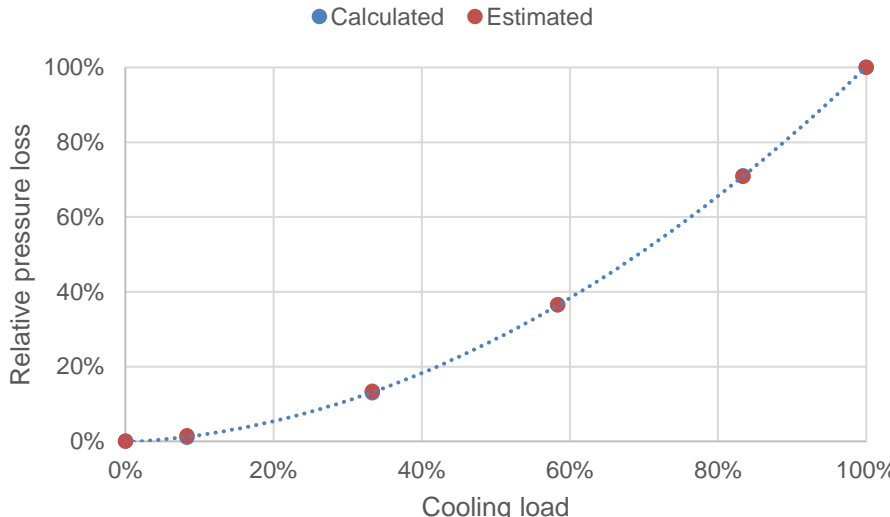
Table 11. Consumer class definition; variables.

class Consumer
Description <p>Class Consumer simply contains only the cooling demand and identification number.</p>
Variables <i>node_id (int)</i> <p>Identification number for identification purposes. Needs to be unique among specified consumers.</p> <i>demand (list, float)</i> <p>Cooling demand time series of the consumer, preferably 8760 (hours) long (kWh/h).</p>

Table 12. EnergyStorage class definition; variables.

class EnergyStorage
Description <p>Class EnergyStorage contains the specification for a energy storage. It is mainly used as data structure.</p>
Variables <i>capacity (float)</i> <p>Capacity (size) of the storage (kWh).</p> <i>losses (float)</i> <p>Loss factor for the storage, e.g. 0.02 means that 2 % of the energy content is lost in a hour.</p> <i>initCharge (float)</i> <p>Initial charge of the storage (in kWh).</p> <i>loadingCost (float)</i> <p>Loading cost for the storage (€/kWh).</p> <i>unloadingCost (float)</i> <p>Unloading cost for the storage (€/kWh).</p> <i>inv (float)</i> <p>Precalculated total investment for the storage (€).</p> <i>loadingEff (float)</i> <p>Loading efficiency, i.e. how much (0-1) of the loaded energy is lost.</p> <i>unloadingEff (float)</i> <p>Unloading efficiency, i.e. how much (0-1) of the unloaded energy is lost.</p> <i>loadingRatio (float)</i> <p>Ratio between the load capacity (kWh/h) and storage capacity (kWh).</p> <i>unloadingRatio (float)</i> <p>Ratio between the unload capacity (kWh/h) and storage capacity (kWh).</p> <i>lifetime (int)</i> <p>Lifetime of the storage, used in emissions calculation.</p> <i>dT (float)</i> <p>Temperature difference used in the storage (valid only for water based storages).</p>

Table 13. Network class definition; variables.

class Network
<p>Description</p> <p>Class Network contains the specification for a DC distribution network and function calculating the related losses, embedded emissions, costs and pumping electricity consumption.</p> <p>Embedded emissions and costs are calculated by using parameters in data file pipedata.txt (see Chapter 4.2). The heat loss calculation uses the same parameter. The method used is for insulated single steel pipes with a given DN size. The equations for heat loss calculation are based on work Bøhm et al (2005).</p> <p>The pumping electricity consumption calculation is a estimation and not a physical model. It calculates the consumption as a function of design pressure loss (pressure loss at maximum flow) and cooling demand. It does not take into account the location of the demand, i.e. flows in individual pipes, making the assumption that the cooling demand is distributed evenly within the system basically representing the network as a single pipe.</p> <p>A pressure drop (bar) is presented as a function of relative cooling load (currently cooling load / maximum cooling load) with the following equation:</p> $dp(L_r) = (0.9L_r^2 + 0.1L_r)dp_{design}$ <p>The coefficients are based case specific results of pressure drop calculation equations, see figure below.</p>  <p>The pump electricity consumption calculation itself uses the pressure drop calculated by the above equation with a volume flow corresponding to total cooling demand of each hour. An efficiency of 0.7 is assumed for the pump.</p> <p>Variables</p> <p><i>structure</i> (list, array[int, int, int, int, float]) Structure of the network, a list of arrays [pipe_id, start_node, end_node, dnsz, pipelen].</p> <p><i>node_roles</i> (list, array[int, int]) Roles for specific nodes in an array [node_id, role]. Roles; linking node = 0, consumer = 1, producer = 2.</p> <p><i>node_roles</i> (list, array[int, int]) Roles for specific nodes in an array [node_id, role]. Roles; linking node = 0, consumer = 1, producer = 2.</p> <p><i>losses</i> (float) Thermal losses of the full distribution network.</p>

investment (float)

Investment of the full distribution network.

om (float)

Operation and maintenance costs (0-1) as % of total investment.

lifetime (int)

Lifetime of the network (and/or pipes).

corr_coeff (float)

Correction coefficient for the heat loss calculations. Used if measurement data on losses is available.

net_params (dict, float)

Physical properties used in calculation of heat losses, {"lg": ground conductivity, "li": insulation material conductivity, "lc": plastic casing conductivity, "ls": pipe shell conductivity, "depth": pipe burying depth, "alfa": heat transfer coefficient on surface}

Tg (float)

Undisturbed group temperature used in heat loss calculation (°C).

Tf (float)

Supply (feed) temperature used in heat loss calculation (°C).

Tr (float)

Return temperature used in heat loss calculation (°C).

emissions (float)

Total embedded emissions (kgCO₂) of the network (pipes).