
Icopt Documentation

P. James Joyce

May 31, 2018

1	lcopt - An interactive tool for creating fully parameterised Life Cycle Assessment (LCA) foreground models	1
2	Features	3

lcopt - An interactive tool for creating fully parameterised Life Cycle Assessment (LCA) foreground models

Lcopt is a python module for creating fully parameterised LCA foreground models using a Flask based interactive GUI developed by [James Joyce](#).

- Intuitive GUI based on drawing of flow sheets
- Flexible model setup
- Integration with SimaPro and Brightway2
- Run LCA analyses and explore the results from inside the GUI

2.1 Introduction

2.1.1 What does it do?

LCOPT is designed to provide a simple and intuitive way to create ‘fully parameterised’ Life Cycle Inventory foreground models.

2.1.2 What’s a Life Cycle Inventory foreground model?

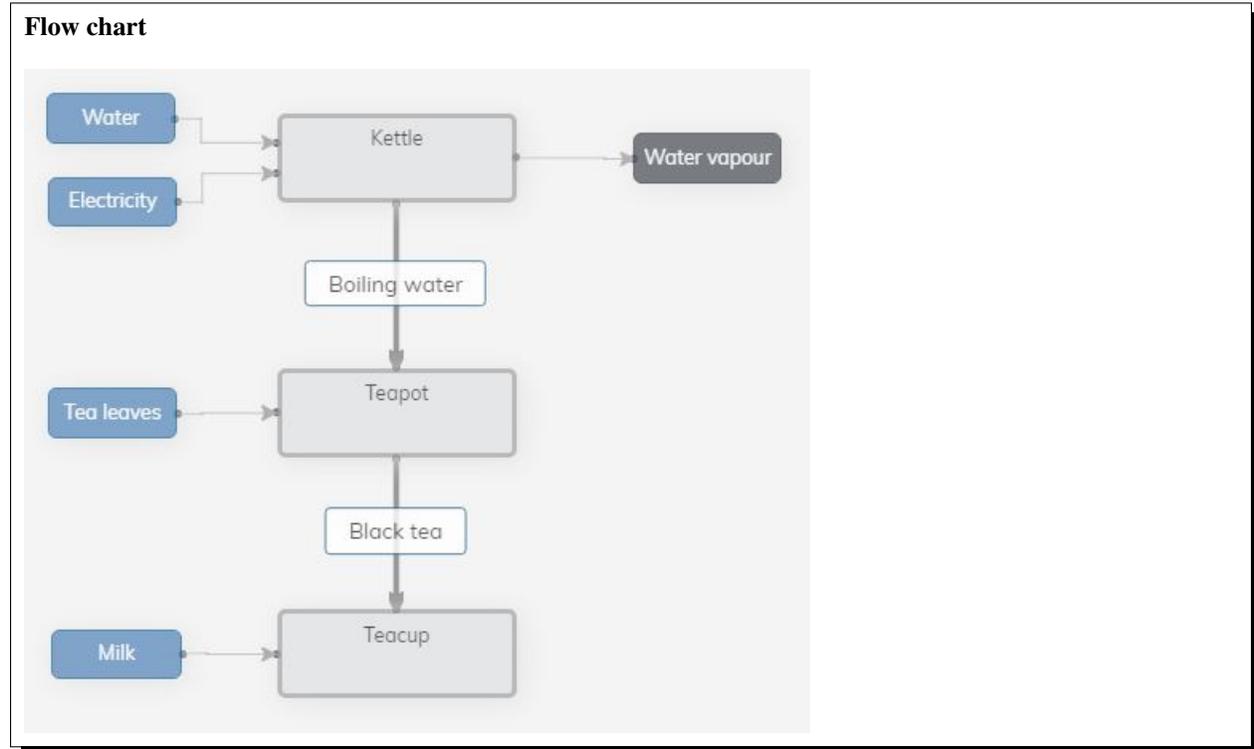
To conduct a Life Cycle Assessment (LCA) we need to create a model of the life cycle of a given product or process which we can then use to estimate the potential environmental impact which results. This includes extraction of raw materials, processing, transport, use, disposal etc. etc.

Because each item used in a life cycle has it’s own set upstream processes the LCA model tends to be split into the *foreground system* - the processes under the control of the person doing the LCA - and the *background system* - the rest of the industrial economy which provides the materials and services required by the foreground system.

Background system models are either determined by data providers (e.g. *ecoinvent*) or can be customised using *Ocelot*.

The foreground model is a set of linked *unit processes*. A unit process describes a stage in the life cycle. It is basically a list of the *exchanges* required to produce one unit of an intermediate (or final) product. These exchanges can either be inputs from the *technosphere* - the human constructed industrial economy - or resources from or emissions to the *biosphere*. Technosphere exchanges can either come from the background system or other processes in the foreground system.

For example, when you make a cup of tea you *make* some boiling water, add this to some tea leaves, add some milk, then use/drink it.



The boiling water is created in the unit process *kettle* which takes inputs from the background system of *water* and *electricity* and emits a small amount of *water vapour* to the biosphere.

The next unit process is *teapot*. This takes an input of *boiling water* from our first process, and *tea leaves* from the background system to produce *black tea*.

Finally, *teacup* takes an input of *black tea* from the *teapot* process and *milk* from the background system *Note : we're making a proper British cup of tea!*

We now have a foreground model for our cup of tea LCA.

2.1.3 Why use LCOPT for creating foreground models?

How you create your foreground model is up to you.

Simple models can be constructed in Excel. More complex models can be created in commercially available softwares such as [SimaPro](#) or [GaBi](#), or freeware such as [OpenLCA](#).

These fully featured software packages have a steep learning curve, can cost a lot of money, or require administrator rights to run (which can be an issue with university or company owned computers).

Lcopt is free, open source and written in Python. You can install it without admin rights and it runs in a normal web browser. It's designed to be simple and intuitive, but powerful enough to conduct *proper* LCAs.

2.1.4 How do you create foreground models in LCOPT?

When you start an LCA, it's usually a good idea to draw out a flow diagram of the system you're modelling on a piece of paper. This helps you identify the unit processes involved, the inputs they use and the extent of the *foreground system* you might be able to get primary data for.

Creating foreground models in Lcopt is as simple as drawing out your flow chart.

Instead of having to write out lists of exchanges to describe each unit process, you simply draw out a flow sheet of your process. You can add processes, link them together and add inputs and outputs. In the background Lcopt compiles these exchange lists, and gives each link a parameter. When you've finished your drawing you have a *fully parameterised* foreground model.

2.1.5 What does *fully parameterised* mean?

In order to analyse your LCA model, you need to know how much of each exchange is required by each unit process. In most softwares the default is to *hard code* these values in the unit process itself. This is great for a one-off model, but can be a real hassle if there are multiple variants of your system which you want to assess and compare.

Taking the example from above, some people like their tea with milk, some like it black. Some people like strong tea (with more tea leaves) others like it weaker. If you want to compare weak milky tea with strong black tea you'd need to create two different *teapot* processes (one for weak, one for strong) and two different *teacup* processes (one for milky, one for black), with the correct exchange amounts, and links to the version of *teapot* with the correct output.

If you then decide you want to model strong milky tea or weak black tea you need to create more unit processes. . .

Most LCA softwares allow you to replace the values in the unit processes with parameters, which you can then vary externally to create scenarios. Choosing, typing out and remembering parameters can be a boring and time consuming task though. With Lcopt it's all done automatically in the background. Every link in the model is given a parameter instead of a value. By default, the model is 'fully parameterised'.

This makes creating the model a lot quicker and the modelling and analysis a lot more flexible.

2.1.6 How do you go from a flow sheet to a model?

To go from an unquantified flow sheet to a meaningful LCA model, the first thing you need to do is create a *parameter set*. This provides a value for each link in a given scenario.

You can either enter an explicit value, or you can tell the model to generate its own value based on a function/formula. To help this you can also create *global parameters* to use in these functions.

For example, the amount of electricity (kWh) a kettle uses to boil a litre of water can be determined by its power rating (kW) and the amount of time it takes to boil (minutes).

You can create a global parameter for *power rating* and another one for *boil minutes*. You can then tell the model to generate a value for the input of *electricity* to *kettle* using the function `power_rating * boil_mins / 60`.

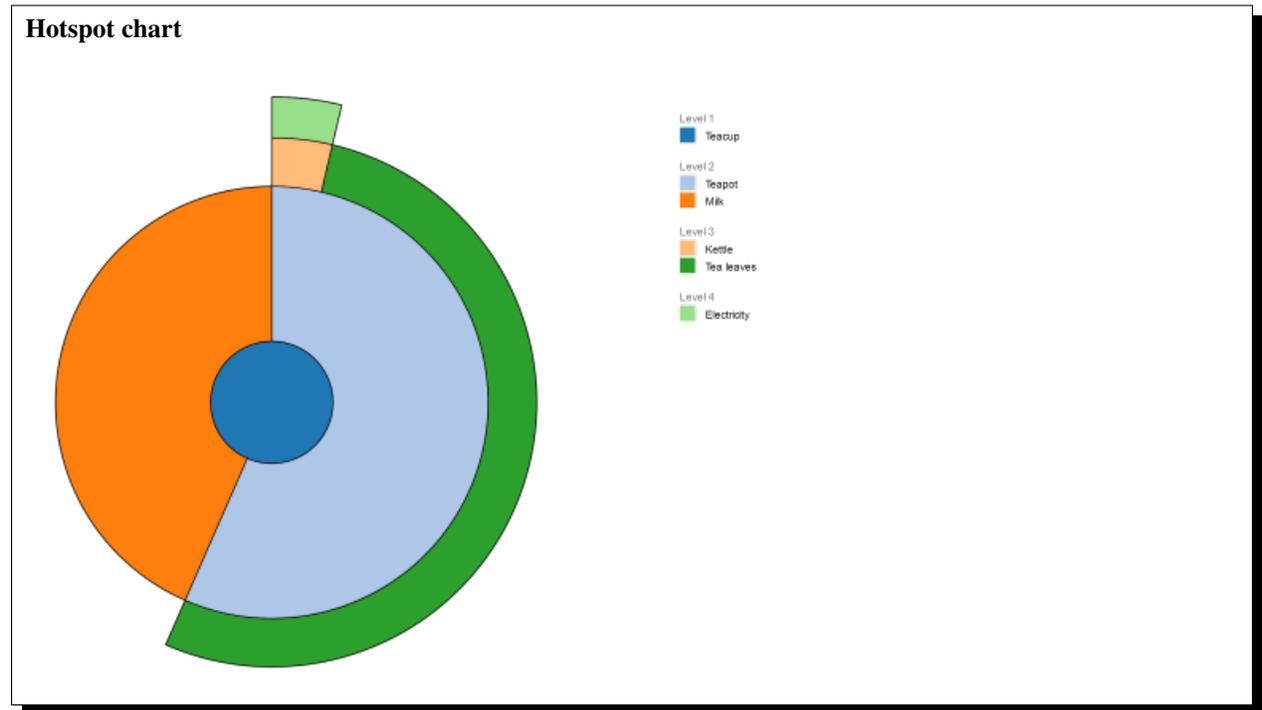
Now you can easily compare different kettles in your cup of tea LCA.

You can use any parameter in a function, including inputs to any process. For example, if you want to add transport of an input (in units of kg.km) you could create a 'transport distance' parameter, and tell Lcopt to work out the amount of 'transport' required by multiplying the amount of your input (kg) by the distance (km). Now whenever you change the amount of your input, the transport of that input automatically changes too.

Once you have a 'default' parameter set you can duplicate it and edit whichever parameters you need to describe a new scenario.

For example you can change the parameter for the input of *milk* to *teacup* to zero to model a black tea scenario, or increase the input of *tea leaves* to *teapot* to model a strong tea scenario.

2.1.7 How do you analyse a model?



To analyse the model you have two options.

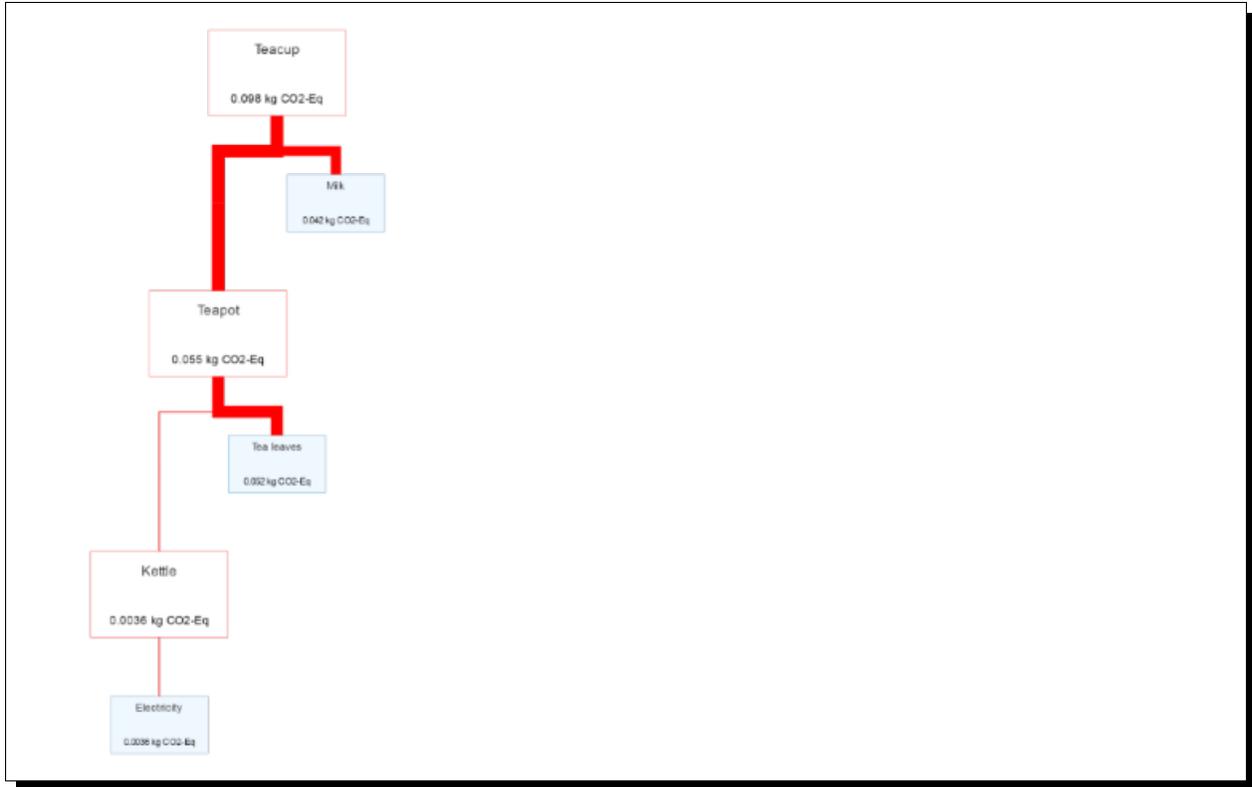
If you have a copy of SimaPro and are using Icopt as an easy way to generate a fully parameterised foreground model, you can export the model as a SimaPro .csv file and the parameter sets as an .xlsx file.

The second option is to run the analysis directly from Lcopt. This uses [Brightway](#) to do the calculations. Brightway is an open source LCA software also written in Python.

When you analyse a model from within Lcopt for the first time it creates a Brightway project. The foreground model is stored as it's own separate database within the project, with links to the biosphere and technosphere databases.

Lcopt calculates all of the parameters for each parameter set and then modifies the exchanges in the Brightway database before running the LCA calculations for that scenario.

Tree diagram



You can choose from over 700 different impact assessment methods (don't worry, there is a search function), and use as many as you like in one go. The default set is climate change (IPCC 2013 100a) and human toxicity (USEtox).

Lcopt's results screens contain bar charts, pie charts, tree diagrams, hotspot charts and tables. Charts can be exported to image files (300 dpi .png with transparent backgrounds) and the tables exported as Excel files.

Once the analysis has been run you can open the model in Brightway as you would any other project if you need to conduct a more detailed analysis. Note however that Brightway will only 'remember' the last scenario you ran so to do this it might be best to create your model with only one parameter set.

2.1.8 How do I get started?

To get started, follow the [installation instructions](#) on the next page, learn how to launch lcopt in your browser in the usage instructions then check out the video runthrough to see how to make the *cup of tea* model using Lcopt.

2.2 Installation

2.2.1 Basic Installation

Recommended Installation

The easiest way to install lcopt is via conda. This ensures the version of Python (3.6) is correct, all of the dependencies are correct and there are no conflicts. You can also do it in only 3 commands.

1. On the command line/console, create a new environment called lcopt:

```
conda create -n lcopt python=3.6
```

2. Activate the lcopt environment using one of these:

```
# Mac/Linux
source activate lcopt
# Windows
activate lcopt
```

3. Install lcopt:

```
conda install -y -q -c conda-forge -c cmutel -c haasad -c pjamesjoyce lcopt
```

2.2.2 Setting up brightway2 for lcopt

To use the interactive analysis feature, you need to set up brightway2 to play nicely with lcopt.

Lcopt needs to set up its own brightway project template so that it can create a new brightway project for each of your models using this template.

Step 1. Download the Ecoinvent 3.3 cutoff database (Ecoinvent license required)

Note: Note - if you don't have an ecoinvent license but want to try out lcopt, it can also be used with the freely available FORWAST database *Alternative setup - FORWAST*.

Log into ecoinvent.org and go to the Files tab

Download the file called `ecoinvent_3.3_cutoff_ecoSpold02.7z`

Extract the file somewhere sensible on your machine, you might need to download [7-zip](#) to extract the files.

Make a note of the path of the folder that contains the `.ecospold` files, its probably `<path/extracted/to>/datasets/`

Step 2a. lcopt-bw2-setup at the command line

At the command line type:

```
lcopt-bw2-setup path/to/ecospold/files # use "" if there are spaces in your path
```

Step 2b. Or run the setup utility in a jupyter notebook/python shell

Fire up your chosen python shell, then use:

```
from lcopt.utils import lcopt_bw2_setup
ecospold_path = r'path/to/ecospold/files' # put your own path in here
lcopt_bw2_setup(ecospold_path)
```

It'll take a while, but once its done it'll return `True` if it worked properly.

Running the setup utility in a python shell also allows you to overwrite your existing configuration if something has gone wrong by using `lcopt_bw2_setup(ecospold_path, overwrite=True)`

2.2.3 Alternative setup - FORWAST

Lcopt can be used without an ecoinvent license by using the [FORWAST](#) database instead.

Option 1. Download `lcopt_bw2_setup.py` and use that instead

At the command line type:

```
lcopt-bw2-setup-forward
```

Option 2. Run the setup utility in a jupyter notebook/python shell

Fire up your chosen python shell, then use:

```
from lcopt.utils import lcopt_bw2_forward_setup
lcopt_bw2_forward_setup()
```

It'll take a while, but once its done it'll return `True` if it worked properly.

As above, you can overwrite an existing configuration using `overwrite=True`

2.3 Getting Started

Set up the background databases

Set up with ecoinvent

Set up with FORWAST

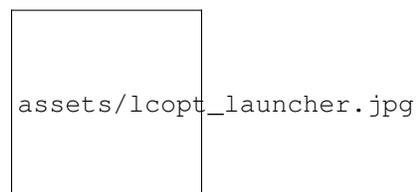
The best way to get started with Lcopt is to explore one of the example models.

First make sure you've set up either ecoinvent or FORWAST as a background database.

Then activate your lcopt environment, and use the command line command `lcopt-launcher` to launch Lcopt

```
activate lcopt
cd C:\Users\pjjoyce\Documents\01_Lcopt_models
lcopt-launcher
```

This dialog will appear:



Choose *Open Example Model*

The video below runs through the steps required to make the example model.

2.3.1 Video

2.4 Using Lcopt

2.4.1 Creating and loading your own models

There are 2 options for using lcopt's GUI (LCOPT INTERACT) to create and view your own models

Option 1. lcopt-launcher command line entry point (*Simpler, only access to GUI*)

Using this option, you can launch lcopt from the command line/console and use the GUI from then on.

Open the console/command line.

Activate your *lcopt* environment, `cd` into the folder you want to store your model in then simply use `lcopt-launcher`

```
activate lcopt
cd C:\Users\pjjoyce\Documents\01_Lcopt_models
lcopt-launcher
```

You'll get an option to either create a new model or load an existing one (or to view the example model).

If you choose to create a model you will be prompted for a name - this will be the model's name and the default filename.

If you choose to load a model a file dialog will open and allow you to choose an existing `.lcopt` file.

Once you've made your choice, the GUI will open in your default browser.

Option 2. Jupyter notebooks (*More complicated, more powerful*)

Probably the best way to use lcopt is via a jupyter notebook. This gives you more access to the inner workings if you need to get at something you can't see in the GUI.

`cd` into whatever folder you want your notebooks and lcopt models to be stored in, then start jupyter e.g.:

```
activate lcopt
cd C:\Users\pjjoyce\Documents\01_Lcopt_models
jupyter notebook
```

This will fire up the jupyter notebook server in your browser. Create a new notebook, give it a meaningful name.

Then in the first cell import lcopt:

```
from lcopt import *
```

Next create your LcoptModel:

```
model = LcoptModel('MyFirstModel')
```

or load an existing one:

```
model = LcoptModel(load = 'MyFirstModel')
```

Note: If you are using FORWAST instead of ecoinvent, you need to add the argument `useForwast=True` e.g. `model = LcoptModel(load = 'MyFirstModel', useForwast=True)`

Then launch the interactive model creator/analyser:

```
model.launch_interact()
```

2.4.2 LCOPT INTERACT - the GUI

Running the GUI via each of the options above launches a Flask server that gives you a nice UI to interact with the models. You can add processes, link them together, add biosphere and technosphere exchanges, and create parameter sets and functions using your parameters. It should be pretty intuitive, if you get stuck, try the 'more info...' buttons.

See the *video* in the *Getting Started* page for a runthrough of the functionality of the GUI.

When your model's ready you can export it to SimaPro as a .csv file and the parameter sets you've created as an Excel file (Note: you need SimaPro developer to import the parameter sets from the Excel file).

To run the analyses interactively using brightway2, make sure you've completed the additional setup step (*Setting up brightway2 for lcopt*).

The 'QUIT' button in the top right hand corner will shut down the Flask server and tell you to close the window.

If you're running from a jupyter notebook, this frees up the notebook again so you can run any commands you need to.

One useful command is `model.save()` which will save any unsaved changes (you can also save by clicking on the save button in LcoptInteract, but in case you forget you can use `model.save()`)

The model is saved as a .lcopt file in your working directory (its really a .pickle file, but the .lcopt extension makes it easier to filter on in the lcopt_launcher file picker)

NOTE: The next time you run the GUI from a notebook you need to use:

```
model = LcoptModel(load = 'MyFirstModel')
```

If you don't it'll create a new blank model called 'MyFirstModel'. If you do do this by accident fear not - it won't overwrite your .lcopt file until you save it. Quit interact by hitting the QUIT button and go back and change your command (just don't click the save button or run `model.save()` in the meantime)

2.5 Technical Documentation

2.6 Contact

2.6.1 Get in touch

If you have any problems or questions, please raise an issue on the [github page](#).

2.6.2 Contribute

If you want to contribute to Lcopt, please fork the [github repository](#) and open a pull request. Please accompany any new features with corresponding tests (written for `py.test`).