

Components for grouped optimization

Link - ElectricalLine

Link

<https://oemof-solph.readthedocs.io/en/latest/reference/oemof.solph.html#oemof.solph.custom.link.Link>

- In-development component
- One link can connect two buses in both directions



- Possibility to convert the buses, different factors considering the direction
- ```
>>> link = solph.custom.Link(
... label="transshipment_link",
... inputs={bel0: solph.Flow(), bel1: solph.Flow()},
... outputs={bel0: solph.Flow(), bel1: solph.Flow()},
... conversion_factors={(bel0, bel1): 0.92, (bel1, bel0): 0.99})
```

# ElectricalLine

[https://oemof-solph.readthedocs.io/en/latest/reference/oemof.solph.html#oemof.solph.custom.electrical\\_line.ElectricalLine](https://oemof-solph.readthedocs.io/en/latest/reference/oemof.solph.html#oemof.solph.custom.electrical_line.ElectricalLine)

- In-development component
- Use of **ElectricalBus** object essential
- Mainly used in linear optimal power flow calculations
- **Reactance** of the line as a parameter
  - $flow(n, t) = \frac{1}{reactance(n, t)} * (voltage_{angle}(i(n), t) - voltage_{angle}(o(n), t))$

$\forall t \text{ in timesteps}, \forall n \text{ in ElectricalLines}$

# ElectricalLine

[https://oemof-solph.readthedocs.io/en/latest/reference/oemof.solph.html#oemof.solph.custom.electrical\\_line.ElectricalLine](https://oemof-solph.readthedocs.io/en/latest/reference/oemof.solph.html#oemof.solph.custom.electrical_line.ElectricalLine)

```

b_el0 = custom.ElectricalBus(label="b_0", v_min=-1, v_max=1)

b_el1 = custom.ElectricalBus(label="b_1", v_min=-1, v_max=1)

b_el2 = custom.ElectricalBus(label="b_2", v_min=-1, v_max=1)

es.add(b_el0, b_el1, b_el2)

es.add(custom.ElectricalLine(input=b_el0, output=b_el1,
 reactance=0.0001,
 investment=Investment(ep_costs=10),
 min=-1, max=1,))

```

# Use examples

- Link:
  - [oemof-moea/liboemof.py at master · matpri/oemof-moea \(github.com\)](#)
  - [OSeEM-DE/base-NDE-SDE.py at master · znes/OSeEM-DE \(github.com\)](#)
- ElectricalLine:
  - [oemof-examples/lopf.py at master · oemof/oemof-examples \(github.com\)](#)

# More parameters ?

- If a more detailed model is required, the OEMOF.TABULAR package offers costs in relation with the link created: capacity costs and activity costs
- [oemof.tabular package — oemof.tabular 0.0.2dev documentation \(oemof-tabular.readthedocs.io\)](https://oemof-tabular.readthedocs.io)
- Capacity min, max, fixed?
- Losses ? Environmental impact?
- Applied to investment mode? (not possible with oemof.tabular.facades)

# Application to the project

```
1 import oemof.solph as solph
2 from tkinter import *
3 import oemof.tabular.facades as fc
4
5 a = [[1, 2], [2, 3]] # batiments liés entre eux, doivent représenter des bus
6 root = Tk()
7
8 for i in range(len(a)):
9 lab = Label(root, text="SLink" + str(i))
10 lab.pack()
11 link = solph.custom.Link(
12 label=lab,
13 inputs={a[i][0]: solph.Flow(), a[i][1]: solph.Flow()},
14 outputs={a[i][0]: solph.Flow(), a[i][1]: solph.Flow()},
15 conversion_factors={(a[i][0], a[i][1]): 1, (a[i][1], a[i][0]): 1},
16)
```

```
18 # utilisation de oemof.tabular
19 for i in range(len(a)):
20 lab = Label(root, text="TLinka" + str(i))
21 lab.pack()
22 link = fc.Link(
23 label=lab,
24 from_bus=a[i][0],
25 to_bus=a[i][1],
26 capacity=100,
27 loss=0.01,
28 capacity_cost=10,
29 marginal_cost=0,
30)
31
32 lab = Label(root, text="TLinkb" + str(i))
33 lab.pack()
34 link = fc.Link(
35 label=lab,
36 from_bus=a[i][1],
37 to_bus=a[i][0],
38 capacity=100,
39 loss=0.01,
40 capacity_cost=10,
41 marginal_cost=0,
42)
```