lca_algebraic

Bringing **symbolic calculus** to **Brightway2**

Raphaël Jolivet - Research Engineer / IT

Observation Impact Energy - MINES ParisTech

INCER-ACV project

MINES ParisTech | PSL

ADEME

AGENCE DE LA TRANSITION ÉCOLOGIQUE

ARMINES

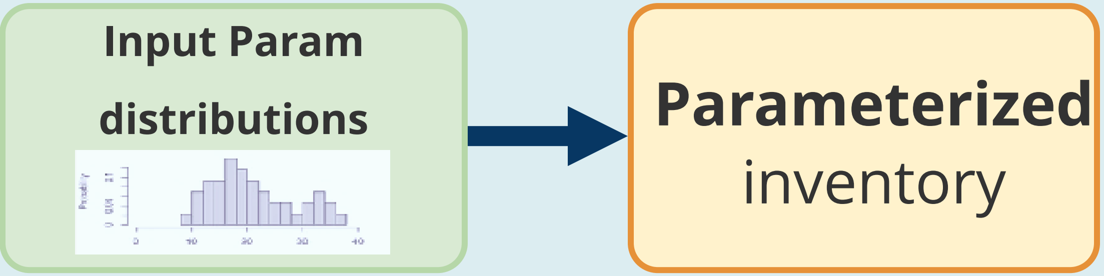
engie

INCER-ACV project

INCER-ACV project

Parameterized
inventory

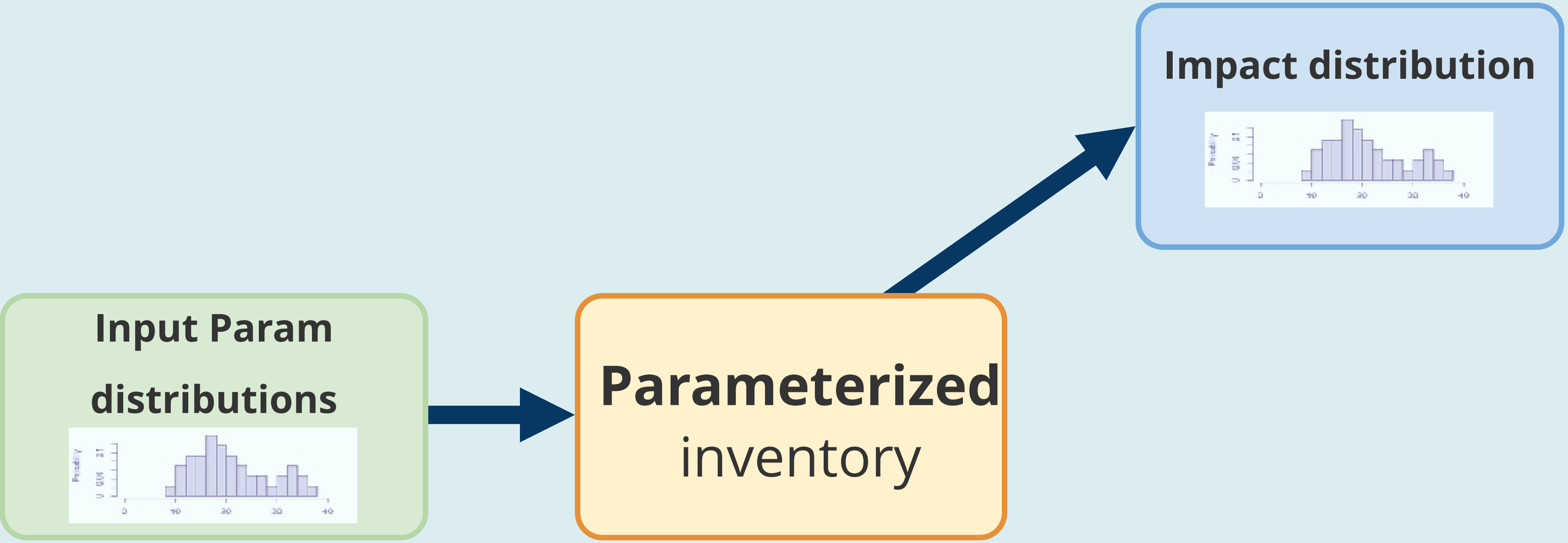
INCER-ACV project



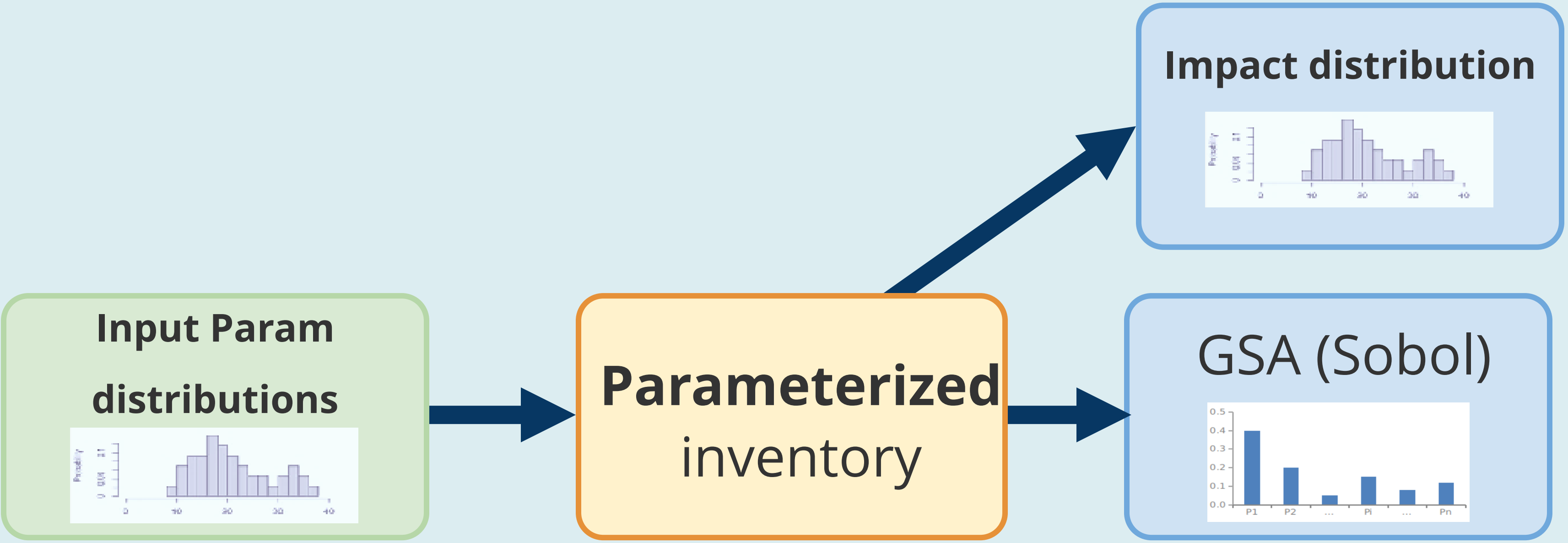
Input Param distributions

Parameterized inventory

INCER-ACV project



INCER-ACV project



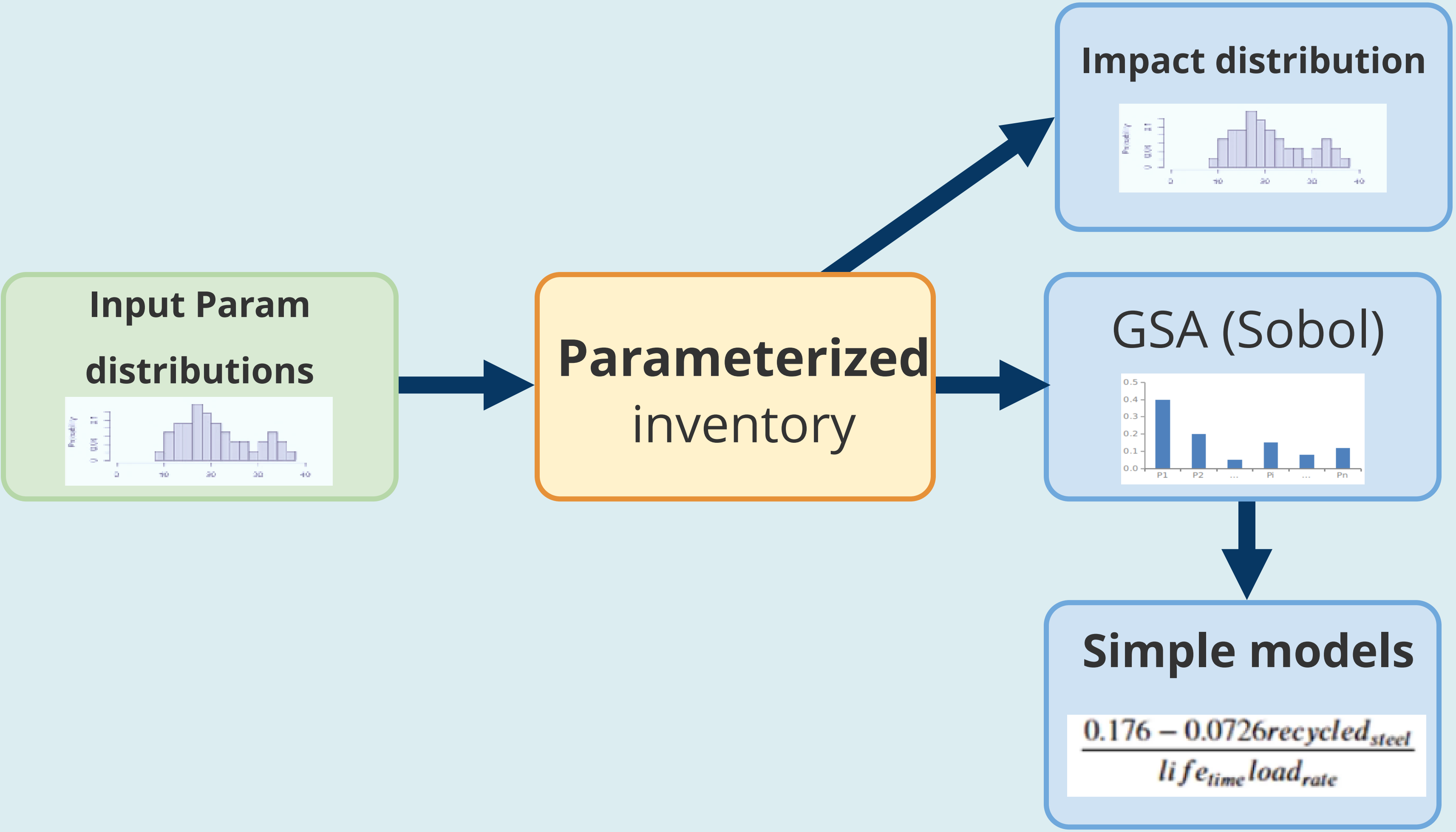
Input Param distributions

Parameterized inventory

Impact distribution

GSA (Sobol)

INCER-ACV project



Input Param distributions

Parameterized inventory

Impact distribution

GSA (Sobol)

Simple models

$$\frac{0.176 - 0.0726 recycled_{steel}}{life_{time} load_{rate}}$$

lca_algebraic

lca_algebraic

Open Source library based on ***Brightway2*** & ***Sympy***

lca_algebraic

Open Source library based on ***Brightway2*** & ***Sympy***

Helper functions for **compact** & **declarative** definition of **parametric inventories**

lca_algebraic

Open Source library based on ***Brightway2*** & ***Sympy***

Helper functions for **compact** & **declarative** definition of **parametric inventories**

Symbolic calculus (**Sympy**) brings:

lca_algebraic

Open Source library based on ***Brightway2*** & ***Sympy***

Helper functions for **compact** & **declarative** definition of **parametric inventories**

Symbolic calculus (**Sympy**) brings:

- Parametric amounts in **pure Python** (vs strings)

lca_algebraic

Open Source library based on ***Brightway2*** & ***Sympy***

Helper functions for **compact** & **declarative** definition of **parametric inventories**

Symbolic calculus (**Sympy**) brings:

- Parametric amounts in **pure Python** (vs strings)
- **Factorize** background activities
  ➡️ super fast Monte Carlo (~ **1 million** / sec vs **100**)
  ➡️ **GSA** (*Sobol* indices)

lca_algebraic

Open Source library based on ***Brightway2*** & ***Sympy***

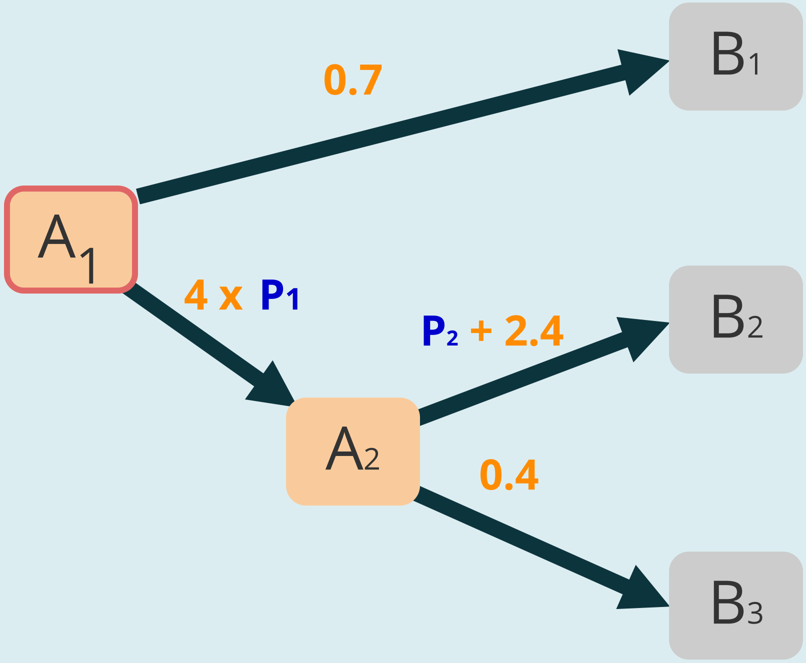
Helper functions for **compact** & **declarative** definition of **parametric inventories**

Symbolic calculus (**Sympy**) brings:

- Parametric amounts in **pure Python** (vs strings)
- **Factorize** background activities
  ➡️ super fast Monte Carlo (~ **1 million** / sec vs **100**)
  ➡️ **GSA** (*Sobol* indices)
- Automatic generation of **simplified models**

lca_algebraic

Internals



A_1

B_1

B_2

A_2

B_3

0.7

$4 \times P_1$

$P_2 + 2.4$

0.4

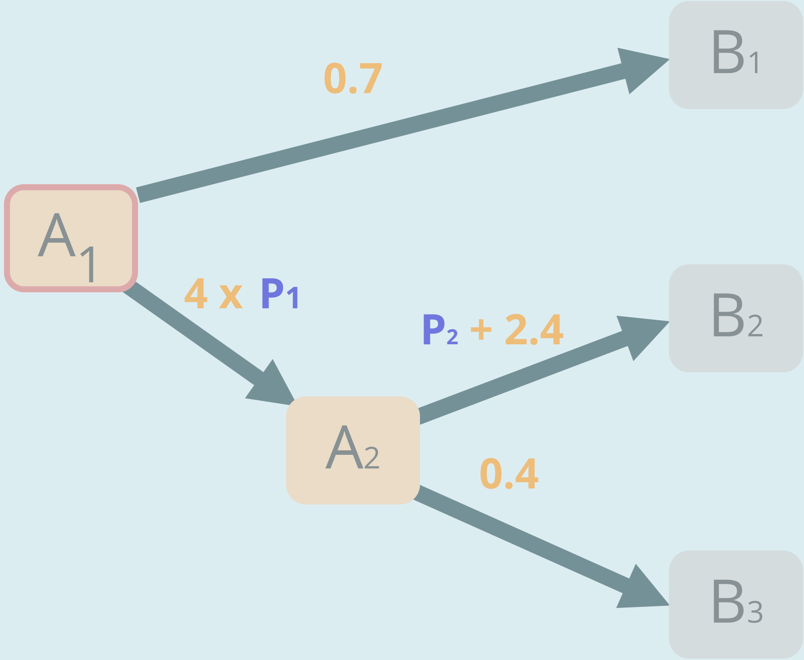
A_i Foreground activities

B_i Background activities

P_i Parameters

lca_algebraic

Internals



to **Sympy** expression :

$$A_1 = f(B_i, P_i)$$

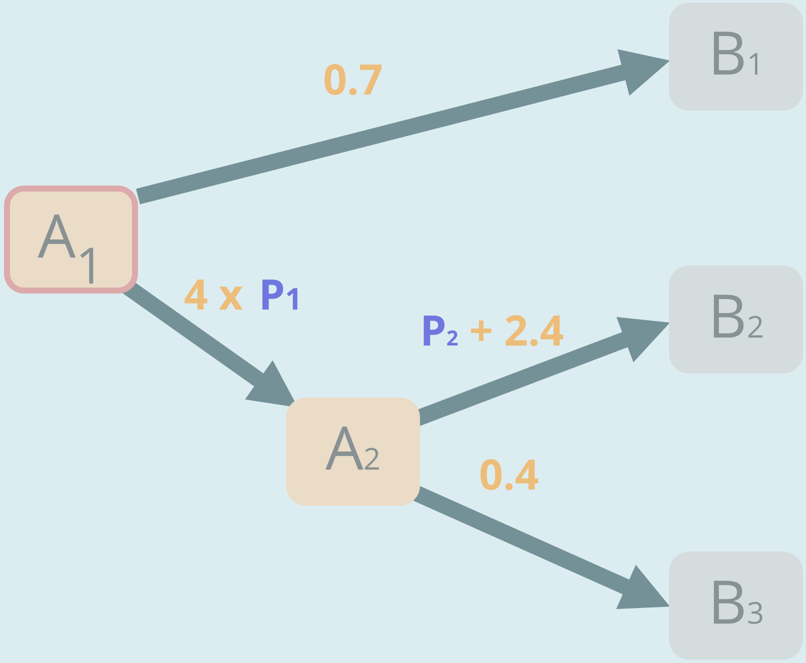
A_i Foreground activities

B_i Background activities

P_i Parameters

lca_algebraic

Internals



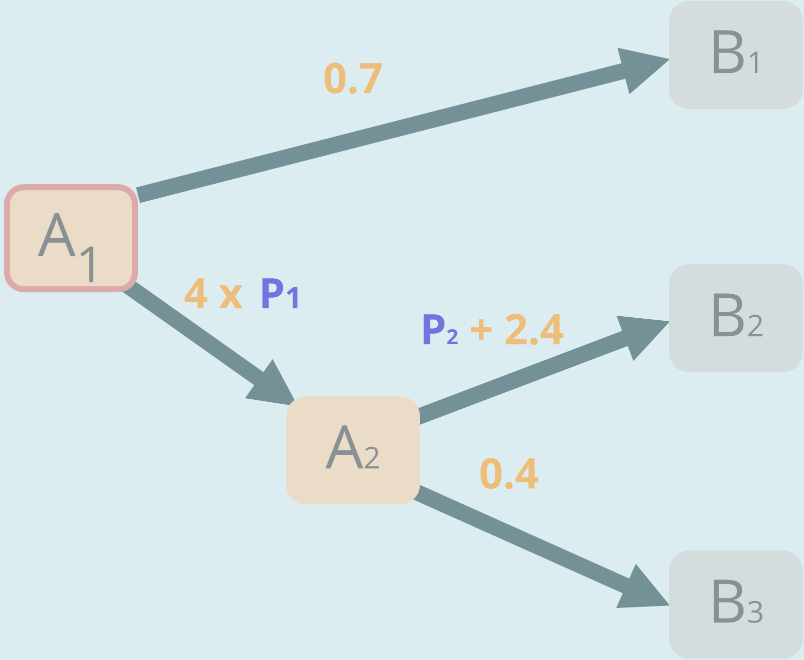
LCIA once, replace **Bi** by values :

$$A_1 = f(P_i)$$

- A_i Foreground activities
- B_i Background activities
- P_i Parameters

lca_algebraic

Internals



Compiles to **fast** **numpy** function

$$A_1 = f(P_i)$$ ⚡

A$_i$ Foreground activities

B$_i$ Background activities

P$_i$ Parameters

Demo time !