GAMSPY

COMFORTABLE AND PERFORMANT MODELING IN PYTHON

GAMSPy empowers you to integrate your preferred Python libraries (such as Numpy, Pandas, Networkx) into your optimization pipeline/modeling for seamless manipulation and visualization of data. Additionally, it streamlines the import and export of data with modeling and optimization of results in various data formats.

With GAMSPy, we present a novel approach to simplify the entire optimization process, commencing with data input and preprocessing, followed by the execution of the mathematical model, and concluding with data post processing and visualization – all within a unified and user-friendly Python environment.

KEY FEATURES AND ADVANTAGES

STREAMLINED OPTIMIZATION PIPELINE:

- Streamline your optimizition pipeline, manage data preprocessing and optimization tasks in a single intuitive Python environment.
- Integration with GAMS MIRO and GAMS Engine for local, server, or cloud-based optimization.

MATHEMATICAL MODELING IN PYTHON:

- Write intricate mathematical models directly in Python. Create robust, readable models with data/model separation.
- Rely on established modeling concepts that are now transferred to Python.

UNCOMPROMISED PERFORMANCE:

- Solver independence for customizable integration with diverse solvers.
- GAMSPy offloads to GAMS backend, maintaining performance.

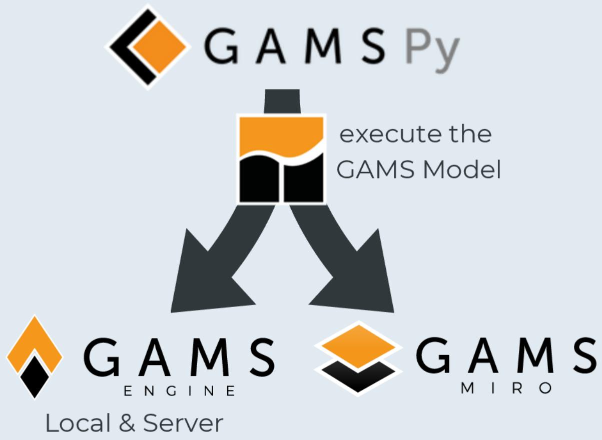
DIFFERENCE FROM GAMS:

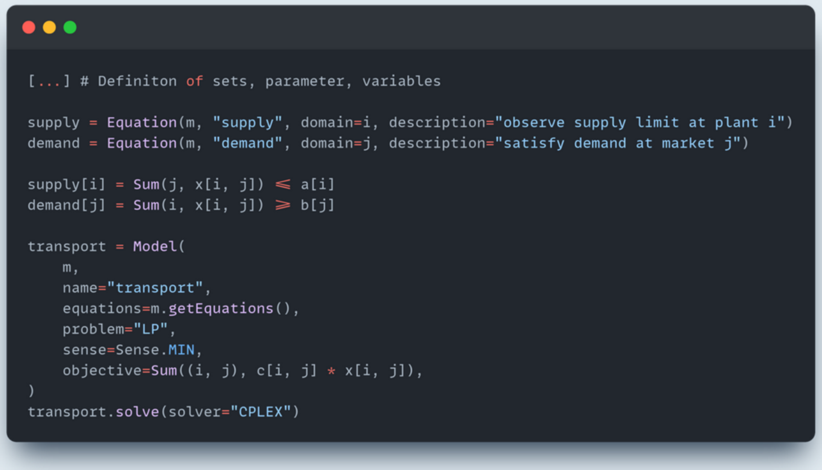
- GAMSPy brings GAMS's specialized modeling capabilities into Python, combining it with the flexibility of the language.
- Enables features like indexed assignment statements and concise equation definition in Python.

gams.com/gamspy

Seamless Integration with GAMS MIRO and GAMS Engine in GAMSPy

GAMSPy effortlessly integrates with GAMS MIRO and GAMS Engine, enabling you to execute your GAMSPy optimization on either your local machine, your dedicated server hardware (GAMS Engine One) or through GAMS Engine SaaS. With the latter, there's no need for you to manage a server, ensuring access to the necessary resources at any given time.





```
[...] # Definiton of sets, parameter, variables

supply = Equation(m, "supply", domain=i, description="observe supply limit at plant i")
demand = Equation(m, "demand", domain=j, description="satisfy demand at market j")

supply[i] = Sum(j, x[i, j]) <= a[i]
demand[j] = Sum(i, x[i, j]) >= b[j]

transport = Model(
    m,
    name="transport",
    equations=m.getEquations(),
    problem="LP",
    sense=Sense.MIN,
    objective=Sum((i, j), c[i, j] * x[i, j]),
)
transport.solve(solver="CPLEX")
```

GAMSPy comes with a preinstalled demo license.
Removing the model size limitation requires a GAMS 45 license or newer.

EFFICIENT MATHEMATICAL MODELING AND OPTIMIZATION

TAKE CLOSER LOOK
AT GAMSPY!

Mathematical Modeling in Python

Write complex mathematical models directly in Python.
Create robust, readable and maintainable mathematical models.
Leverage the power of separating instance data and model notation.

Uncompromised Performance

- Solver Independence: Choose and integrate different solvers based on your specific requirements.
- No loss in performance: GAMSPy offloads the heavy lifting to the efficient GAMS backend.
- Sparsity Handling: Let GAMSPy take care of handling sparse data cubes and focus on the formulation of the model.

Seamless Pipeline Management

- No switching of environments: Manage data preprocessing and optimization tasks within a single, intuitive environment.
- Leverage Python libraries to comfortably manipulate and visualize data.
  Import and export data and optimization results to many data formats.
  Leverage the seamless integration of GAMS MIRO and GAMS Engine.

U.S.: (+1) 202 342-0180
Europe: (+49) 221 949-9170

gams.com/gamspy

sales@gams.com