

```
In [2]: from gamspy import (Container, Variable, Equation, Model, Set, Parameter, Sum)
import numpy as np
```

```
In [3]: m = Container()

#SETS
materials = Set(m, 'materials', records=['Fe1', 'Fe2', 'Fe3', 'Cu1', 'Cu2',
elements = Set(m, 'elements', records = ['C', 'Cu', 'Mn'])
pct = Parameter(m, 'pct', domain = [elements, materials], records = np.array
                                [0, 0, 0
                                [1.3, 0.

price = Parameter(m, 'price', domain=[materials], records=np.array([200, 250
availability = Parameter(m, name='availability', domain=materials, records=r

lower_rb = Parameter(container=m, name="lower_rb", domain=elements, records=
upper_rb = Parameter(container=m, name="upper_rb", domain=elements, records=

v = Variable(m, 'v', domain=materials, type='Positive')

low_pc = Equation(m, 'low_pc', domain=elements)
low_pc[elements] = Sum(materials, pct[elements, materials] * v[materials]) >
high_pc = Equation(m, 'high_pc', domain=elements)
high_pc[elements] = Sum(materials, pct[elements, materials] * v[materials])
resources = Equation(m, 'resources', domain=materials)
resources[materials] = v[materials] <= availability[materials]
mb = Equation(m, 'mb')
mb[:] = Sum(materials, v[materials]) == 500

# Model setup
steel_model = Model(m,
    name='steel_model',
    equations=m.getEquations(),
    problem=Problem.LP,
    sense=Sense.MIN,
    objective=Sum(materials, price[materials] * v[materials]))
```

```
In [5]: steel_model.solve(options = Options(equation_listing_limit=100))
print("Objective Function Value: ", round(steel_model.objective_value,4), "\n")
print("materials: \n", v.records)
print("status: ", steel_model.status)
print("solver status: ", steel_model.solve_status)
```

Objective Function Value: 98121.6358

materials:

	materials	level	marginal	lower	upper	scale
0	Fe1	400.000000	0.000000	0.0	inf	1.0
1	Fe2	0.000000	66.371898	0.0	inf	1.0
2	Fe3	39.776302	0.000000	0.0	inf	1.0
3	Cu1	0.000000	142.111150	0.0	inf	1.0
4	Cu2	2.761272	0.000000	0.0	inf	1.0
5	Al1	57.462426	0.000000	0.0	inf	1.0
6	Al2	0.000000	15.241174	0.0	inf	1.0

status: ModelStatus.OptimalGlobal
solver status: SolveStatus.NormalCompletion

In []: