Linear Programming Example with `scipy.optimize.linprog`

This notebook demonstrates solving a **linear programming** (LP) problem using `scipy.optimize.linprog`. In LP, we maximize or minimize a **linear objective function** subject to **inequality constraints**.

Problem Setup

We want to **maximize** the objective function:

$$z = x_1 + 2x_2 + 3x_3$$

subject to the following constraints:

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 12, \\ 2x_1 + x_2 + 3x_3 &\leq 18, \end{aligned}$$

with non-negativity bounds:

$$x_1, x_2, x_3 \geq 0.$$

Code Explanation

1. **Objective Function Coefficients**: We define `obj = [-1, -2, -3]` since `linprog` minimizes by default, and maximizing (z) is equivalent to minimizing $-z$.
2. **Constraints**: The left-hand side matrix `lhs` represents the coefficients of x_1, x_2, and x_3 in the inequalities, and `rhs` is the vector of upper bounds for each inequality.
3. **Bounds**: We set each variable to be non-negative.

Solution

After setting up the parameters, we call `linprog()` to solve the LP problem and display the optimization results.

Παράδειγμα Γραμμικού Προγραμματισμού με την `scipy.optimize.linprog`

Εδω δείχνουμε πώς να λύσουμε ένα πρόβλημα **γραμμικού προγραμματισμού** (LP) χρησιμοποιώντας την `scipy.optimize.linprog`. Στον γραμμικό προγραμματισμό, επιδιώκουμε τη μεγιστοποίηση ή ελαχιστοποίηση μιας **γραμμικής αντικειμενικής συνάρτησης** υπό **γραμμικούς ανισοτικούς περιορισμούς**.

Ορισμός Προβλήματος

Θέλουμε να **μεγιστοποιήσουμε** την αντικειμενική συνάρτηση:

$$z = x_1 + 2x_2 + 3x_3$$

υπό τους εξής περιορισμούς:

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 12, \\ 2x_1 + x_2 + 3x_3 &\leq 18, \end{aligned}$$

με μεταβλητές με μη αρνητικές τιμές:

$$x_1, x_2, x_3 \geq 0.$$

Επεξήγηση Κώδικα

1. **Συντελεστές Αντικειμενικής Συνάρτησης**: Ορίζουμε `obj = [-1, -2, -3]` επειδή το `linprog` ελαχιστοποιεί, και η μεγιστοποίηση του z είναι ισοδύναμη με την ελαχιστοποίηση του $-z$.
2. **Περιορισμοί**: Ο πίνακας στην αριστερή πλευρά `lhs` περιγράφει τους συντελεστές των x_1, x_2 και x_3 στις ανισότητες, και το `rhs` είναι το διάνυσμα με τα άνω όρια για κάθε ανισότητα.
3. **Όρια**: Ορίζουμε κάθε μεταβλητή να είναι μη αρνητική.

Λύση

Αφού ορίσουμε τις παραμέτρους, καλούμε το `linprog()` για να λύσουμε το πρόβλημα LP και εμφανίζουμε τα αποτελέσματα της βελτιστοποίησης.

```
from scipy.optimize import linprog

# Objective function coefficients for maximization problem (converted to minimization by negating)
# Objective: Maximize z = x1 + 2*x2 + 3*x3 --> linprog minimizes, so we use -z = -x1 - 2*x2 - 3*x3
obj = [-1, -2, -3]

# Left-hand side of inequality constraints (Ax ≤ b format)
# Constraint 1: x1 + x2 + x3 ≤ 12
# Constraint 2: 2*x1 + x2 + 3*x3 ≤ 18
lhs = [
    [1, 1, 1],
    [2, 1, 3]
]
```

```python
# Right-hand side of inequality constraints
rhs = [12, 18]

# Variable bounds (all variables must be non-negative)
bnd = [
    (0, float('inf')),
    (0, float('inf')),
    (0, float('inf'))
]

# Solve the linear program
optimize = linprog(c=obj, A_ub=lhs, b_ub=rhs, bounds=bnd)

# Display the optimization results
optimize
```

```
        message: Optimization terminated successfully. (HiGHS Status 7: Optimal)
        success: True
         status: 0
            fun: -27.0
              x: [ 0.000e+00  9.000e+00  3.000e+00]
            nit: 2
          lower:  residual: [ 0.000e+00  9.000e+00  3.000e+00]
                 marginals: [ 1.500e+00  0.000e+00  0.000e+00]
          upper:  residual: [       inf        inf        inf]
                 marginals: [ 0.000e+00  0.000e+00  0.000e+00]
          eqlin:  residual: []
                 marginals: []
        ineqlin:  residual: [ 0.000e+00  0.000e+00]
                 marginals: [-1.500e+00 -5.000e-01]
 mip_node_count: 0
 mip_dual_bound: 0.0
        mip_gap: 0.0
```

Start coding or generate with AI.