In [101…

```
from scipy.optimize import linprog
import numpy as np
import string

# --- Setup securities (A, C, D, G–AA) ---
securities = ['A', 'C', 'D'] + list(string.ascii_uppercase[6:]) + ['AA']
n = len(securities)  # 24 securities total

# --- YTW values (remove B, E, F) ---
ytw = np.array([
    0.0675, 0.0510, 0.0743,             # A, C, D
    0.0611, 0.0616,                     # G–H
    0.06303, 0.06648, 0.0637106, 0.0716519,  # I–L
    0.0652835, 0.0679686, 0.0617869, 0.0761157, 0.055307, 0.0555836,  # M–R
    0.055484, 0.0602328, 0.0561343, 0.060865, 0.0638948, 0.0666737,
    0.0644262, 0.0610212, 0.0658001  # S–AA
])

# --- Objective: maximize expected return ---
c = -ytw  # minimize −ytw → maximize ytw

# --- Equality constraint: sum(weights) = 1 ---
A_eq = np.ones((1, n))
b_eq = np.array([1.0])

# --- Inequality constraints (in ≤ form) ---
A_ub_rows = []
b_ub = []

# 1️⃣ Expected return ≥ 6% →  −ytw·weights <= -0.06
row = -ytw
A_ub_rows.append(row)
b_ub.append(-0.06)

# 2️⃣ Verizon & Deutsche (G–H) ≥ 0.10 → −(G+H) <= -0.10
row = np.zeros(n)
row[3:5] = -1.0
A_ub_rows.append(row)
b_ub.append(-0.10)

# 3️⃣ ETFs (A–D, i.e. first 3) ≤ 0.10 → (A,C,D) <= 0.10
row = np.zeros(n)
row[0:3] = 1.0
A_ub_rows.append(row)
b_ub.append(0.1)

# 4️⃣ Utilities at least 15% (last 4 securities: W, X, Y, Z, AA → last 4 = X, Y, Z,
row = np.zeros(n)
row[-4:] = -1.0
A_ub_rows.append(row)
b_ub.append(-0.15)

# 5️⃣ Healthcare at least 15% (securities 12–16 after reindexing)
row = np.zeros(n)
row[12:17] = -1.0
A_ub_rows.append(row)
b_ub.append(-0.15)

# 6️⃣ I–AA ≤ 0.80 → (from index 5 onward)
row = np.zeros(n)
row[5:] = 1.0
A_ub_rows.append(row)
b_ub.append(0.80)

# 7️⃣ K–AA ≥ 0.02 → −(indices 7 onward) <= -0.02
```

```
row = np.zeros(n)
row[7:] = -1.0
A_ub_rows.append(row)
b_ub.append(-0.02)

# Stack all constraints
A_ub = np.vstack(A_ub_rows)
b_ub = np.array(b_ub)

# --- Bounds for each weight ---
bounds = []
for i in range(n):
    if i < 3:               # A–D (ETFs)
        bounds.append((0.02, 0.06))
    elif 3 <= i < 5:        # G–H
        bounds.append((0.11, 0.12))
    else:                   # Others
        bounds.append((0.02, 0.08))

# --- Solve the linear program ---
result = linprog(
    c,
    A_ub=A_ub, b_ub=b_ub,
    A_eq=A_eq, b_eq=b_eq,
    bounds=bounds,
    method='highs'
)

# --- Display results ---
if result.success:
    weights = result.x
    portfolio_return = np.dot(ytw, weights)
    print("✅ Optimal Weights (A, C, D, G–AA):")
    for name, w in zip(securities, weights):
        print(f"  {name}: {w:.4f}")
    print()
    print(f"Expected Portfolio Return: {portfolio_return * 100:.2f}%")
    print(f"ETFs (A–D): {sum(weights[0:3]):.4f}")
    print("A: IBHI, C: BNDX, D: PGHY")
    print()
    print(f"Main Bonds (G–H): {sum(weights[3:5]):.4f}")
    print("G: Verizon, H: Deutsche")
    print()
    print(f"Remaining Bonds (I–AA): {sum(weights[5:]):.4f}")
    print("I: Anglian Water, J: Citigroup, K: Intesa, L: Santander, M: StanChart, N
    print("P: DENTSPLY, Q: Rogers memorial, R: Baglan Moor, S: Ochsner LSU, T: AT&T
    print("W: NBN, X: Energy Transfer, Y: Keenan Fort, Z: PacificCorp, AA: TechnipF
    print()
    print(f"Healthcare (O–S): {sum(weights[12:17]):.4f}")
    print(f"Utilities (X–AA): {sum(weights[-4:]):.4f}")

else:
    print("❌ No feasible solution found.")
    print("Solver message:", result.message)
```

```
✅ Optimal Weights (A, C, D, G–AA):
  A: 0.0200
  C: 0.0200
  D: 0.0600
  G: 0.1100
  H: 0.1100
  I: 0.0200
  J: 0.0700
  K: 0.0200
  L: 0.0800
  M: 0.0200
  N: 0.0800
  O: 0.0200
  P: 0.0800
  Q: 0.0200
  R: 0.0200
  S: 0.0200
  T: 0.0200
  U: 0.0200
  V: 0.0200
  W: 0.0200
  X: 0.0800
  Y: 0.0200
  Z: 0.0200
  AA: 0.0300

Expected Portfolio Return: 6.53%
ETFs (A–D): 0.1000
A: IBHI, C: BNDX, D: PGHY

Main Bonds (G–H): 0.2200
G: Verizon, H: Deutsche

Remaining Bonds (I–AA): 0.6800
I: Anglian Water, J: Citigroup, K: Intesa, L: Santander, M: StanChart, N: Bontobel,
O: CVS
P: DENTSPLY, Q: Rogers memorial, R: Baglan Moor, S: Ochsner LSU, T: AT&T, U: Bell C
anada, V: British Telecom
W: NBN, X: Energy Transfer, Y: Keenan Fort, Z: PacificCorp, AA: TechnipFMC

Healthcare (O–S): 0.1600
Utilities (X–AA): 0.1500
```