```
import numpy as np
from scipy.optimize import minimize

sigma1 = 1
sigma2 = 1.5
sigma3 = 1.25

sigma12 = 0.1
sigma13 = -0.1
sigma23 = 0.2

m1 = 1
m2 = 2
m3 = 1.5

m0 = 1.35

# Αντικειμενική συνάρτηση -- Objective function
def objective(w):
    return sigma1*w[0]**2 + sigma2*w[1]**2 + sigma3*w[2]**2  \
     + 2*sigma12*w[0]*w[1] + 2*sigma13*w[0]*w[2] + 2*sigma23*w[1]*w[2]

# Περιορισμοί -- Constraints
constraints = [
  {'type': 'eq',   'fun': lambda w: w[0] + w[1] + w[2] - 1},  # w1 + w2 + w3 = 1
  {'type': 'ineq', 'fun': lambda w: m1*w[0] + m2*w[1] + m3*w[2] - m0}, # m1*w1 + m2*w2 + m3*w3 >= m0
  {'type': 'ineq', 'fun': lambda w: w[0]},           # w1 >= 0
  {'type': 'ineq', 'fun': lambda w: w[1]},           # w2 >= 0
  {'type': 'ineq', 'fun': lambda w: w[2]}            # w3 >= 0
  ]

w0 = np.array([1/3, 1/3, 1/3])

result = minimize(objective, w0, method='SLSQP', constraints=constraints)

# Αποτελέσματα -- Results

print("Optimal solution:", result.x)
print("Minimum standart deviation:", result.fun)
print("Mean value:", m1*result.x[0] + m2*result.x[1] + m3*result.x[2])
```

```
Optimal solution: [0.44340399 0.21093856 0.34565745]
Minimum standart deviation: 0.4299165686473888
Mean value: 1.3837672854497343
```

Start coding or generate with AI.