

# 1 Scipy 中的非线性规划：minimize 函数

在科学计算和优化问题中，`scipy` 库提供了强大的工具来解决非线性规划问题。其中，`scipy.optimize.minimize` 函数是一个非常实用的工具，它可以用于求解带约束和不带约束的非线性优化问题。

## 1.1 函数原型

`scipy.optimize.minimize` 函数的原型如下：

```
1 scipy.optimize.minimize(fun, x0, args=(), method=None, jac=None, hess=None,
```

各参数的含义如下：

- `fun`: 需要最小化的目标函数。
- `x0`: 初始猜测值，是一个数组。
- `args`: 传递给目标函数和约束函数的额外参数。
- `method`: 优化算法，可选值有 'Nelder - Mead' 单纯形法、'Powell'、'CG'、'BFGS'、'Newton - CG'、'L - BFGS - B'、'TNC'、'COBYLA'、'SLSQP' 等。
- `jac`: 目标函数的雅可比矩阵（一阶导数）。
- `hess`: 目标函数的海森矩阵（二阶导数）。
- `hessp`: 海森矩阵与向量的乘积。
- `bounds`: 变量的边界约束。
- `constraints`: 约束条件。
- `tol`: 收敛精度。
- `callback`: 每次迭代后调用的回调函数。
- `options`: 优化算法的选项。

## 1.2 示例问题

考虑一个简单的非线性规划问题：

最小化目标函数

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 2.5)^2$$

约束条件：

$$x_1 - 2x_2 + 2 \geq 0$$

$$-x_1 - 2x_2 + 6 \geq 0$$

$$-x_1 + 2x_2 + 2 \geq 0$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

以下是使用 `scipy.optimize.minimize` 函数解决该问题的 Python 代码：

```
1 import numpy as np
2 from scipy.optimize import minimize
3
4 # 定义目标函数
5 def objective(x):
6     return (x[0] - 1) ** 2 + (x[1] - 2.5) ** 2
7
8 # 定义约束条件
9 constraints = [
10     {'type': 'ineq', 'fun': lambda x: x[0] - 2 * x[1] + 2},
11     {'type': 'ineq', 'fun': lambda x: -x[0] - 2 * x[1] + 6},
12     {'type': 'ineq', 'fun': lambda x: -x[0] + 2 * x[1] + 2}
13 ]
14
15 # 定义变量的边界约束
16 bounds = [(0, None), (0, None)]
17
18 # 初始猜测值
```

```

19 x0 = np.array([2, 0])
20
21 # 使用 SLSQP 方法进行优化
22 result = minimize(objective, x0, method='SLSQP', bounds=bounds, constraints=
23
24 # 输出结果
25 if result.success:
26     print("最优解找到！")
27     print(f"最优解: {result.x}")
28     print(f"最优值: {result.fun}")
29 else:
30     print("优化过程失败。")

```

通过上述代码，我们可以利用 `scipy.optimize.minimize` 函数有效地解决非线性规划问题。不同的问题可能需要选择不同的优化算法和设置合适的参数。