Mosek wrapper

最优化 最优化 Mosek

线性规划

Problem:

$$
\begin{aligned}
&\max 3x_0 + 1x_1 + 5x_2 + 1x_3 \\
&\text{s.t. } 3x_0 + 1x_1 + 2x_2 = 30 \\
&2x_0 + 1x_1 + 3x_2 + 1x_3 \geq 15 \\
&2x_1 + 3x_3 \leq 25 \\
&0 \leq x_0 \leq \infty \\
&0 \leq x_1 \leq 10 \\
&0 \leq x_2 \leq \infty \\
&0 \leq x_3 \leq \infty
\end{aligned}
$$

Ans:

$$
x = [0.0, 0.0, 15.0, 8.333333333333334]
$$

Code:

```
def main():
    params = {
              # 目标函数向量
              "C_obj"  : [3, 1, 5, 1],
              # 约束矩阵
              "A_con"  : [[3, 1, 2, 0],
                          [2, 1, 3, 1],
                          [0, 2, 0, 3]],
              # 约束矩阵中每个约束的下界向量，mosek_g.INF是正无穷
              "blc"  : [30, 15, -mosek_g.INF],
              # 约束矩阵中每个约束的上界向量
              "buc"  : [30, mosek_g.INF, 25],
              # 每个变量的下界约束向量
              "blx"  : [0, 0, 0, 0],
              # 每个变量的上界约束向量
              "bux"  : [mosek_g.INF, 10, mosek_g.INF, mosek_g.INF],
              # 是否为最小化问题
              "minimize" : False,
              # 是否输出全部求解信息
              "silent" : False
```

```
            }
    pro = mosek_linearp(params)
    pro.fit()
    code, result = pro.fit()
    # code为0表示求解成功，result为字典
    if code == 0:
        print(result["x"])
```

混合整数线性规划

Problem:

$$\begin{aligned}&\max 7x_0 + 10x_1 + 1x_2 + 5x_3\\&\text{s.t. } x_0 + x_1 + x_2 + x_3 \leq 2.5\\&x_0, x_1, x_2 \in \mathbb{Z}\\&x_0, x_1, x_2, x_3 \geq 0\end{aligned}$$

Ans:

$$x = [0.0, 2.0, 0.0, 0.5]$$

Code:

```
def main():
    params = {"C_obj"  : [7, 10, 1, 5],
              "A_con"  : [[1, 1, 1, 1]],
              "blc"  : [-mosek_g.INF],
              "buc"  : [2.5],
              "blx"  : [0,0,0,0],
              "bux"  : [mosek_g.INF, mosek_g.INF, mosek_g.INF, mosek_g.IN
F],
              "minimize" :F alse,
              # 第0,1,2个变量是整数变量
              "integ_flag" :[1, 1, 1, 0],
              "silent": True,
             }
    pro = mosek_integerp(params)
    code, result = pro.fit()
    if code == 0:
        print(result["x"])
```

二次优化

Problem:

$$\min \frac{1}{2} x^T Q^{obj} x + c^T x$$
$$\text{s.t. } \frac{1}{2} x^T Q^{con0} x + Ax \geq b,$$
$$x \geq 0$$

where

$$Q^{obj} = \begin{bmatrix} 2 & 0 & -1 \\ 0 & 0.2 & 0 \\ -1 & 0 & 2 \end{bmatrix}, c = [0, -1, 0]^T, A = [1, 1, 1], b = 1$$
$$Q^{con0} = \begin{bmatrix} -2 & 0 & 0.2 \\ 0 & -2 & 0 \\ 0.2 & 0 & -0.2 \end{bmatrix}$$

Ans:

$$x = [0.4488485199618974, 0.9319361480448437, 0.6741131920778094]$$

Code:

```
def main():
    # 目标函数中的二次项矩阵
    Q_obj = [[2, 0, -1], [0, 0.2, 0], [-1, 0, 2]]
    # 约束中的第一个二次项矩阵
    Q_con_0 = [[-2, 0, 0.2], [0, -2, 0], [0.2, 0, -0.2]]
    # 约束中的所有二次项矩阵组合到一个列表中
    Q_con = [Q_con_0]
    params = {
              # 目标函数中的线性部分
              "C_obj"  : [0, -1, 0],
              # 目标函数中的二次项部分
              "Q_obj"  : Q_obj,
              # 约束中的线性部分
              "A_con"  : [
                          [1, 1, 1]
                         ],
              # 约束中的二次项部分
              "Q_con"  : Q_con,
              "blc"  : [1],
              "buc"  : [mosek_g.INF],
              "blx"  : [0, 0, 0],
              "bux"  : [mosek_g.INF, mosek_g.INF, mosek_g.INF],
              "minimize" :True,
              "silent": False
            }
    pro = mosek_quadraticp(params)
    code, result = pro.fit()
    if code == 0:
        print(result["x"])
```