Laimonas Beniušis studento nr. 1410102
Kompiuterių mokslas 1 1gr
Naudota priemonė: SageMath
Optimizavimo metodai užduotis 3

Tiesinio programavimo uždavinys

$$\begin{aligned} min \quad & 2x_1 - 3x_2 - 5x_4 \\ -x_1 + x_2 - x_3 - x_4 &= 8 \\ 2x_1 + 4x_2 &= 10 \\ -x_3 + x_4 &\leq 3 \\ x_1, x_3 &\geq 0 \end{aligned}$$

Užrašome uždavinį matriciniu pavidalu.
Įvykdomi pertvarkymai:

$$\begin{aligned} min \quad & 2x_1 - 3x_2 - 5x_4 \\ -x_1 + x_2 - x_3 - x_4 &\geq 8 \\ x_1 - x_2 + x_3 + x_4 &\geq -8 \\ 2x_1 + 4x_2 &\geq 10 \\ -2x_1 - 4x_2 &\geq -10 \\ x_3 - x_4 &\geq -3 \\ x_1 &\geq 0 \\ x_3 &\geq 0 \end{aligned}$$

Gautos Matricos: $C = \begin{bmatrix} 2 \\ -3 \\ 0 \\ -5 \end{bmatrix}$ $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$ $A = \begin{bmatrix} -1 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 4 & 0 & 0 \\ -2 & -4 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ $B = \begin{bmatrix} 8 \\ -8 \\ 10 \\ -10 \\ -3 \\ 0 \\ 0 \end{bmatrix}$

Uždavinys matriciniu pavidalu:

$$\begin{aligned} min \quad & C^T X \\ AX = B, & X \geq 0 \end{aligned}$$

Uždavinio sprendiniai:

$x_1 = 0$ , $x_2 = 2.5$ , $x_3 = 0$ , $x_4 = -5.5$
Minimumas: 20

Apribojimai pakeisti į konstantas pagal studento numerį ir atlikti petvarkymai
a = 1, b = 0, c = 2

$$\begin{aligned} min \quad & 2x_1 - 3x_2 - 5x_4 \\ & -x_1 + x_2 - x_3 - x_4 = 1 \\ & 2x_1 + 4x_2 = 0 \\ & -x_3 + x_4 \leq 2 \\ & x_1, x_3 \geq 0 \end{aligned} \quad \rightarrow \quad \begin{aligned} min \quad & 2x_1 - 3x_2 - 5x_4 \\ & -x_1 + x_2 - x_3 - x_4 \geq 1 \\ & x_1 - x_2 + x_3 + x_4 \geq -1 \\ & 2x_1 + 4x_2 \geq 0 \\ & -2x_1 - 4x_2 \geq 0 \\ & x_3 - x_4 \geq -2 \\ & x_1 \geq 0 \\ & x_3 \geq 0 \end{aligned}$$

Uždavinio sprendiniai:

$x_1 = 0$ , $x_2 = 0$ , $x_3 = 0$ , $x_4 = -1$
Minimumas: 5

Formuluojamas Dualus uždavinys:

$$
\begin{aligned}
min \quad & 2x_1 - 3x_2 - 5x_4 \\
-x_1 + x_2 - x_3 - x_4 &\geq 8 \\
x_1 - x_2 + x_3 + x_4 &\geq -8 \\
2x_1 + 4x_2 &\geq 10 \\
-2x_1 - 4x_2 &\geq -10 \\
x_3 - x_4 &\geq -3 \\
x_1 &\geq 0 \\
x_3 &\geq 0
\end{aligned}
\quad \rightarrow \quad
\begin{aligned}
max \quad & 8y_1 - 8y_2 + 10y_3 - 10y_4 - 3y_5 \\
-y_1 + y_2 + y_3 - 2y_4 + y_6 &\leq 2 \\
y_1 - y_2 + 4y_3 - 4y_4 &\leq -3 \\
-y_1 + y_2 + y_5 + y_7 &\leq 0 \\
-y_1 + y_2 - y_5 &\leq -5
\end{aligned}
$$

Sprendinių nėra

Formuluojamas Dualus uždavinys (su studento nr. konstantomis) :

$$
\begin{aligned}
min \quad & 2x_1 - 3x_2 - 5x_4 \\
-x_1 + x_2 - x_3 - x_4 &\geq 1 \\
x_1 - x_2 + x_3 + x_4 &\geq -1 \\
2x_1 + 4x_2 &\geq 0 \\
-2x_1 - 4x_2 &\geq 0 \\
x_3 - x_4 &\geq -2 \\
x_1 &\geq 0 \\
x_3 &\geq 0
\end{aligned}
\quad \rightarrow \quad
\begin{aligned}
max \quad & 1y_1 - 1y_2 + 0y_3 - 0y_4 - 2y_5 \\
-y_1 + y_2 + y_3 - 2y_4 + y_6 &\leq 2 \\
y_1 - y_2 + 4y_3 - 4y_4 &\leq -3 \\
-y_1 + y_2 + y_5 + y_7 &\leq 0 \\
-y_1 + y_2 - y_5 &\leq -5
\end{aligned}
$$

Sprendinių nėra

Abu suformuoti dualūs uždaviniai neturi įvykdomo sprendinio

## Kodas, sėkminga optimizacija:

```
p = MixedIntegerLinearProgram(maximization=False)
x = p.new_variable(real=True)
p.set_objective(2*x[1] - 3*x[2] - 5*x[4])
p.add_constraint(- x[1] + x[2] - x[3] - x[4] == 8)
p.add_constraint(2*x[1] + 4*x[2] ==10)
p.add_constraint(-x[3]+x[4] <= 3)
p.add_constraint(x[1] >= 0)
p.add_constraint(x[3] >= 0)

p.show()
print("ATS.:"+str(round(p.solve(), 3)))
print(p.get_values(x[1],x[2],x[3],x[4]))


a,b,c = 1,0,2
p = MixedIntegerLinearProgram(maximization=False)
x = p.new_variable(real=True)
p.set_objective(2*x[1] - 3*x[2] - 5*x[4])
p.add_constraint(- x[1] + x[2] - x[3] - x[4] == a)
p.add_constraint(2*x[1] + 4*x[2] ==b)
p.add_constraint(-x[3]+x[4] <= c)
p.add_constraint(x[1] >= 0)
p.add_constraint(x[3] >= 0)

p.show()
print("ATS.:"+str(round(p.solve(), 3)))
print(p.get_values(x[1],x[2],x[3],x[4]))

p = MixedIntegerLinearProgram(maximization=False)
x = p.new_variable(real=True)
p.set_objective(2*x[1] - 3*x[2] - 5*x[4])
p.add_constraint(- x[1] + x[2] - x[3] - x[4] >=1)
p.add_constraint(x[1] - x[2] + x[3] + x[4] >= -1)
p.add_constraint(2*x[1] + 4*x[2] >=0)
p.add_constraint(-2*x[1] - 4*x[2] >=0)
p.add_constraint(x[3]-x[4] >= -2)
p.add_constraint(x[1] >= 0)
p.add_constraint(x[3] >= 0)

p.show()
print(str(round(p.solve(), 3)))
print(p.get_values(x[1],x[2],x[3],x[4]))
```

Kodas, nėra sprendinių:

```
p = MixedIntegerLinearProgram(maximization=True)
y = p.new_variable(real=True)
p.set_objective(8*y[1] - 8*y[2] + 10*y[3] - 10*y[4] - 3*y[5] )
p.add_constraint(- y[1] + y[2] + y[3] - 2*y[4] + y[6] <=2)
p.add_constraint(y[1] - y[2] + 4*y[3] - 4*y[4] <=-3)
p.add_constraint(-y[1] + y[2] +y[5] + y[7]<=0)
p.add_constraint(-y[1] + y[2] - y[5]<=-5)

p.show()
print(str(round(p.solve(), 3)))
print(p.get_values(y[1],y[2],y[3],y[4],y[5],y[6],y[7]))



p = MixedIntegerLinearProgram(maximization=True)
y = p.new_variable(real=True)
p.set_objective(1*y[1] - 1*y[2] + 0*y[3] - 0*y[4] - 2*y[5] )
p.add_constraint(- y[1] + y[2] + y[3] - 2*y[4] + y[6] <=2)
p.add_constraint(y[1] - y[2] + 4*y[3] - 4*y[4] <=-3)
p.add_constraint(-y[1] + y[2] +y[5] + y[7]<=0)
p.add_constraint(-y[1] + y[2] - y[5]<=-5)

p.show()
print(str(round(p.solve(), 3)))
print(p.get_values(y[1],y[2],y[3],y[4],y[5],y[6],y[7]))
```