GLP - Grundlagen der Lineare Programmierung

Applied Discrete Optimization

Operations Research

Aufgabe 1A

```python
import gurobipy as gp

model = gp.Model("GLP")

x = model.addVar(name = "x")
y = model.addVar(name = "y")

model.setObjective(0.4*x + 0.2*y, gp.GRB.MINIMIZE)

model.addConstr(60*x + 7*y >= 320, "Constraint_1")
model.addConstr(8*x + 2*y >= 60, "Constraint_2")
model.addConstr(1*x + 0*y >= 60, "Constraint_3")
model.addConstr(x >= 0, "non-negativity_constraint")
model.addConstr(y >= 0, "non-negativity_constraint")
model.optimize()
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (linux64)

CPU model: 13th Gen Intel(R) Core(TM) i7-1355U
Thread count: 12 physical cores, 12 logical processors

Optimize a model with 5 rows, 2 columns and 7 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 6e+01]
  Objective range  [2e-01, 4e-01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [6e+01, 3e+02]
Presolve removed 5 rows and 2 columns
Presolve time: 0.01s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.4000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.02 seconds
Optimal objective  2.400000000e+01
```

Aufgabe 1B

```python
import gurobipy as gp

model = gp.Model("GLP")

x = model.addVar(name = "x")
y = model.addVar(name = "y")

model.setObjective(0.4*x + 1.5*y, gp.GRB.MINIMIZE)

model.addConstr(60*x + 0*y >= 320, "Constraint_1")
model.addConstr(8*x + 0*y >= 60, "Constraint_2")
model.addConstr(1*x + 80*y >= 60, "Constraint_3")
model.addConstr(x >= 0, "non-negativity_constraint")
model.addConstr(y >= 0, "non-negativity_constraint")
model.optimize()
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (linux64)

Optimize a model with 5 rows, 2 columns and 6 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 8e+01]
  Objective range  [4e-01, 2e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [6e+01, 3e+02]
Presolve removed 5 rows and 2 columns
Presolve time: 0.01s
Solved in 0 iterations and 0.02 seconds
Optimal objective  3.984375000e+00
```

Aufgabe 1C

```python
import gurobipy as gp

model = gp.Model("GLP")

x = model.addVar(name = "x")
y = model.addVar(name = "y")

model.setObjective(0.2*x + 0.2*y, gp.GRB.MINIMIZE)

model.addConstr(7*x + 40*y >= 320, "Constraint_1")
model.addConstr(2*x + 2*y >= 60, "Constraint_2")
model.addConstr(0*x + 0*y >= 60, "Constraint_3")
model.addConstr(x >= 0, "non-negativity_constraint")
model.addConstr(y >= 0, "non-negativity_constraint")
```

```
model.optimize()
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (linux64)

Optimize a model with 5 rows, 2 columns and 6 nonzeros
Coefficient statistics:
  Matrix range      [1e+00, 4e+01]
  Objective range   [2e-01, 2e-01]
  Bounds range      [0e+00, 0e+00]
  RHS range         [6e+01, 3e+02]
Presolve time: 0.02s

Solved in 0 iterations and 0.02 seconds
Infeasible model
```

Aufgabe 1D

```python
import gurobipy as gp

model = gp.Model("GLP")

x = model.addVar(name = "x")
y = model.addVar(name = "y")

model.setObjective(2*x + 0.2*y, gp.GRB.MINIMIZE)

model.addConstr(40*x + 0*y >= 320, "Constraint_1")
model.addConstr(26*x + 2*y >= 60, "Constraint_2")
model.addConstr(15*x + 0*y >= 60, "Constraint_3")
model.addConstr(x >= 0, "non-negativity_constraint")
model.addConstr(y >= 0, "non-negativity_constraint")
model.optimize()
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (linux64)

Optimize a model with 5 rows, 2 columns and 6 nonzeros
Coefficient statistics:
  Matrix range      [1e+00, 4e+01]
  Objective range   [2e-01, 2e+00]
  Bounds range      [0e+00, 0e+00]
  RHS range         [6e+01, 3e+02]
Presolve removed 5 rows and 2 columns
Presolve time: 0.02s
Solved in 0 iterations and 0.03 seconds
Optimal objective  1.600000000e+01
```

Aufgabe 2

```python
import gurobipy as gp

model = gp.Model("GLP")

x = model.addVar(name = "x")
y = model.addVar(name = "y")
w = model.addVar(name = "w")
z = model.addVar(name = "z")

model.setObjective(3*w + 20*x + 13*y + 3*z, gp.GRB.MINIMIZE)

model.addConstr(3*w + 1*x >= 1, "Constraint_1")
model.addConstr(2*w + 3*x + 3*y >= 2, "Constraint_2")
model.addConstr(4*x + 1*y + 1*z >= 1, "Constraint_3")
model.addConstr(1*w + 2*x + 1*y + 1*z >= 1, "constraint_4")
model.addConstr(x >= 0, "non-negativity_constraint")
model.addConstr(y >= 0, "non-negativity_constraint")
model.addConstr(w >= 0, "non-negativity_constraint")
model.addConstr(z >= 0, "non-negativity_constraint")

model.optimize()
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (linux64)

Optimize a model with 8 rows, 4 columns and 16 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 4e+00]
  Objective range  [3e+00, 2e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+00, 2e+00]
Presolve removed 4 rows and 1 columns
Presolve time: 0.01s
Presolved: 4 rows, 3 columns, 9 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   1.500000e+00   0.000000e+00      0s
       2    6.0000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.03 seconds
Optimal objective  6.000000000e+00
```

Aufgabe 3

```python
import gurobipy as gp

model = gp.Model("GLP")

x = model.addVar(name = "x")
y = model.addVar(name = "y")
z = model.addVar(name = "z")

model.setObjective(x + y + z, gp.GRB.MAXIMIZE)

model.addConstr(4*x + 2*y + 1*z <= 2, "Constraint_1")
model.addConstr(1*x + 3*y + 1*z <= 5, "Constraint_2")

model.addConstr(x >= 0, "non-negativity_constraint")
model.addConstr(y >= 0, "non-negativity_constraint")
model.addConstr(z >= 0, "non-negativity_constraint")

model.optimize()
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (linux64)

Optimize a model with 5 rows, 3 columns and 9 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 4e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+00, 5e+00]
Presolve removed 3 rows and 0 columns
Presolve time: 0.02s
Presolved: 2 rows, 3 columns, 6 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    2.0000000e+00   0.000000e+00   0.000000e+00      0s
       0    2.0000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.03 seconds
Optimal objective  2.000000000e+00
```