



GUROBI

OPTIMIZATION

# Gurobi Optimizer – Get the Software

## Gurobi Optimizer

Gurobi Optimizer is the Gurobi optimization libraries. In addition to the software, the corresponding README file contains installation instructions. [Here is the list of bug fixes for each release.](#)

Current version		64-bit Windows	64-bit Linux	64-bit macOS	64-bit AIX
9.1.0	<a href="#">README</a>	<a href="#">Gurobi-9.1.0-win64.msi</a>	<a href="#">gurobi9.1.0_linux64.tar.gz</a>	<a href="#">gurobi9.1.0_mac64.pkg</a>	<a href="#">gurobi9.1.0_power64.tar.gz</a>
md5 Checksum		69488cc43d46b7398e90f5c334ae04da	628c4e2c6fc34193f9dd5852d34b3e1b	5a1e5dd8393e45f714780368d2be92ab	82414392941b7d58b3b4255b9a6995fb
Old versions					
9.0.3	<a href="#">README</a>	<a href="#">Gurobi-9.0.3-win64.msi</a>	<a href="#">gurobi9.0.3_linux64.tar.gz</a>	<a href="#">gurobi9.0.3_mac64.pkg</a>	<a href="#">gurobi9.0.3_power64.tar.gz</a>
md5 Checksum		5394eff3d8f5d8c16190f9ea5bc70020	832040cce622ba7f267e26645fcd200d	758713ea51b0981928f85d9bd81e6b27	948768b299de3d6c69653c7c0a0ed3a5
8.1.1	<a href="#">README</a>	<a href="#">Gurobi-8.1.1-win64.msi</a>	<a href="#">gurobi8.1.1_linux64.tar.gz</a>	<a href="#">gurobi8.1.1_mac64.pkg</a>	<a href="#">gurobi8.1.1_power64.tar.gz</a>
md5 Checksum		17dfc21f0ed64daaa4bdf7634eab705b	05cbb96072e393bd4ebb1d8b9526ce01	d05a73c0df6622851b4371dc1d292579	3d1a756695d52065eeefc15516d9aac6
8.0.1	<a href="#">README</a>	<a href="#">Gurobi-8.0.1-win64.msi</a>	<a href="#">gurobi8.0.1_linux64.tar.gz</a>	<a href="#">gurobi8.0.1_mac64.pkg</a>	<a href="#">gurobi8.0.1_power64.tar.gz</a>
md5 Checksum		d9363f13daa63b79c0cdaa37ad92e8b6	cfc595ddf9482734bdc0268749093cc4	a02d04ef884e64e7091ef7a7439cfe68	877f94a02e602346ee767b9894df4030

# License Details

Information and installation instructions

License ID	516013
Date issued	2020-10-28T22:25:41
Purpose	Trial
License Type	TRIAL
Key Type	TRIAL
Version	9
Expiration Date	2021-04-26
Distributed Limit	0
Host Name	
Host ID	

## Installation

To install this license on a computer where Gurobi Optimizer is installed, copy and paste the following command to the Start/Run menu (Windows only) or a command/terminal prompt (any system):

```
grbgetkey 83af988a-196c-11eb-865d-0a7c4f30bdbe
```

The **grbgetkey** command requires an active internet connection. If your computer has no internet access, or you get no response or an error message such as "Unable to contact key server", [Please click here for additional instructions](#).



grbgetkey bba60259-a126-e14f-  
dab2-580a56ac4d2e|



















# Anaconda Installers

## Windows

Python 3.8

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

## MacOS

Python 3.8

64-Bit Graphical Installer (462 MB)

64-Bit Command Line Installer (454 MB)

## Linux

Python 3.8

64-Bit (x86) Installer (550 MB)

64-Bit (Power8 and Power9) Installer (290 MB)

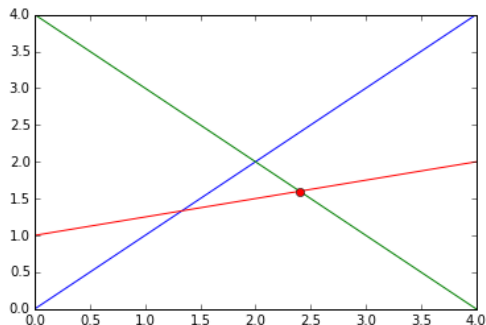
Supercharge your data science  
efforts with **Anaconda**.

Get Started

```
In [45]: from gurobipy import *
m = Model()
v0 = m.addVar()
v1 = m.addVar()
m.update()
m.addConstr(v0 - v1 <= 4) # Constraint 1
m.addConstr(v0 + v1 <= 4) # Constraint 2
m.addConstr(-0.25*v0 + v1 <= 1) # Constraint 3
m.setObjective(v1, GRB.MAXIMIZE) # Objective: maximize v1
m.params.outputflag = 0
m.optimize()
```

Plot the optimal solution...

```
In [46]: import matplotlib.pyplot as pyplot
pyplot.plot([0,4], [0,4]) # Constraint 1
pyplot.plot([4,0], [0,4]) # Constraint 2
pyplot.plot([0,4], [1,2]) # Constraint 3
pyplot.plot([v0.x], [v1.x], 'ro') # Plot the optimal vertex
pyplot.show()
```



In [ ]:

File Edit Search Source Run Debug Consoles Projects Tools View Help


 /home/spyder
 

/opt/gurobi910/linux64/examples/python/untitled1.py

Console 1/A X

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
```

Created on Mon Nov 2 13:07:00 2020

```
@author: gurobi
"""
```

Python 3.8.6 -- IPython 7.19.0

In [1]: import gurobipy as gp

```
In [2]: model = gp.read('/opt/gurobi910/linux64/examples/data/p0033')
Using license file /home/heinz/gurobi.lic
Read MPS format model from file /opt/gurobi910/linux64/examples/data/p0033.mps
Reading time = 0.00 seconds
P0033: 16 rows, 33 columns, 98 nonzeros
```

```
In [3]: model.optimize()
Gurobi Optimizer version 9.1.0 build v9.1.0rc0 (linux64)
Thread count: 4 physical cores, 4 logical processors, using up to 4 threads
Optimize a model with 16 rows, 33 columns and 98 nonzeros
Model fingerprint: 0xc84dd1e1
Variable types: 0 continuous, 33 integer (0 binary)
Coefficient statistics:
  Matrix range      [1e+00, 4e+02]
  Objective range   [5e+01, 5e+02]
  Bounds range      [1e+00, 1e+00]
  RHS range         [1e+00, 3e+03]
Found heuristic solution: objective 3828.0000000
Presolve removed 5 rows and 14 columns
Presolve time: 0.00s
Presolved: 11 rows, 19 columns, 71 nonzeros
Found heuristic solution: objective 3089.0000000
Variable types: 0 continuous, 19 integer (16 binary)
```

Root relaxation: objective 2.839492e+03, 10 iterations, 0.00 seconds

Nodes		Current Node		Objective Bounds		Work	
Expl	Unexpl	Obj	Depth IntInf	Incumbent	BestBd	Gap	It/Node Time
0	0	2839.49184	0	3	3089.00000	2839.49184	8.08% - 0s
0	0	2941.40000	0	1	3089.00000	2941.40000	4.78% - 0s
0	0	2952.00000	0	1	3089.00000	2952.00000	4.44% - 0s
0	0	3045.27500	0	5	3089.00000	3045.27500	1.42% - 0s
0	0	3089.00000	0	7	3089.00000	3089.00000	0.00% - 0s

Cutting planes:

```
Gomory: 3
MIR: 1
```

```
Explored 1 nodes (24 simplex iterations) in 0.05 seconds
Thread count was 4 (of 4 available processors)
```

Solution count 2: 3089 3828

```
Optimal solution found (tolerance 1.00e-04)
Best objective 3.089000000000e+03, best bound 3.089000000000e+03, gap 0.00000%
```

In [4]:

IPython console

Files

Help

Variable explorer

Plots

History

LSP Python: ready Line 8, Col 1 UTF-8 LF RW Mem 9%

File Edit Search Source Run Debug Consoles Projects Tools View Help


 /opt/gurobi910/linux64/examples/python/mip1.py
 

#!/usr/bin/env python3.7

# Copyright 2020, Gurobi Optimization, LLC

# This example formulates and solves the following simple MIP model:

# maximize

#  $x + y + 2z$ 

# subject to

#  $x + 2y + 3z \leq 4$ #  $x + y \leq 1$ #  $x, y, z$  binary

```
import gurobipy as gp
from gurobipy import GRB
```

try:

# Create a new model

m = gp.Model("mip1")

# Create variables

x = m.addVar(vtype=GRB.BINARY, name="x")

y = m.addVar(vtype=GRB.BINARY, name="y")

z = m.addVar(vtype=GRB.BINARY, name="z")

# Set objective

m.setObjective(x + y + 2 \* z, GRB.MAXIMIZE)

# Add constraint:  $x + 2y + 3z \leq 4$ 

m.addConstr(x + 2 \* y + 3 \* z &lt;= 4, "c0")

# Add constraint:  $x + y \leq 1$ 

m.addConstr(x + y &gt;= 1, "c1")

# Optimize model

m.optimize()

for v in m.getVars():

print('%s %g' % (v.varName, v.x))

print('Obj: %g' % m.objVal)

except gp.GurobiError as e:

print('Error code ' + str(e.errno) + ': ' + str(e))

except AttributeError:

print('Encountered an attribute error')

Console 1/A ×

Python 3.8.6 -- IPython 7.19.0

```
In [1]: runfile('/opt/gurobi910/linux64/examples/python/mip1.py',
            wdir='/opt/gurobi910/linux64/examples/python')
```

Using license file /home/heinz/gurobi.lic

Gurobi Optimizer version 9.1.0 build v9.1.0rc0 (linux64)

Thread count: 4 physical cores, 4 logical processors, using up to 4 threads

Optimize a model with 2 rows, 3 columns and 5 nonzeros

Model fingerprint: 0xf43f5bdf

Variable types: 0 continuous, 3 integer (3 binary)

Coefficient statistics:

Matrix range [1e+00, 3e+00]

Objective range [1e+00, 2e+00]

Bounds range [1e+00, 1e+00]

RHS range [1e+00, 4e+00]

Found heuristic solution: objective 2.0000000

Presolve removed 2 rows and 3 columns

Presolve time: 0.00s

Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.02 seconds

Thread count was 1 (of 4 available processors)

Solution count 2: 3

Optimal solution found (tolerance 1.00e-04)

Best objective 3.000000000000e+00, best bound 3.000000000000e+00, gap 0.0000%

x 1

y 0

z 1

Obj: 3

In [2]:

IPython console

Files

Help

Variable explorer

Plots

History

LSP Python: ready

Line 1, Col 1

ASCII

LF

R

Mem 9%



File Edit Search Source Run Debug Consoles Projects Tools View Help

/home/spyder

/opt/gurobi910/linux64/examples/python/sudoku.py

#!/usr/bin/env python3.7

# Copyright 2020, Gurobi Optimiz

# Sudoku example.

# The Sudoku board is a 9x9 grid  
 # of 3x3 grids. Each cell in the  
 # No two grid cells in the same  
 # same value.

# In the MIP formulation, binary  
 # cell <math>i,j</math> takes value 'v'. The  
 # 1. Each cell must take exactly  
 # 2. Each value is used exactly  
 # 3. Each value is used exactly  
 # 4. Each value is used exactly  
 # Input datasets for this example

```
import sys
import math
import gurobipy as gp
from gurobipy import GRB
```

```
if len(sys.argv) < 2:
    print('Usage: sudoku.py file')
    sys.exit(0)
```

f = open(sys.argv[1])

grid = f.read().split()

```
n = len(grid[0])
s = int(math.sqrt(n))
```

# Create our 3-D array of model

model = gp.Model('sudoku')

vars = model.addVars(n, n, n, vtype=gp.GRB.VAR\_INTEGER)

# Fix variables associated with

```
for i in range(n):
    for j in range(n):
        if grid[i][j] != '.':
            v = int(grid[i][j]) - 1
            vars[i, j, v].LB = 1
```

# Each cell must take one value

```
model.addConstrs((vars.sum(i, j, '*') == 1
                  for i in range(n)
```

Python 3.8.6 -- IPython 7.19.0

## Run configuration per file

Select a run configuration:

/opt/gurobi910/linux64/examples/python/sudoku.py

## Console

- ☒ Execute in current console
- ☐ Execute in a dedicated console
- ☐ Execute in an external system terminal

## General settings

- ☐ Remove all variables before execution
- ☐ Run in console's namespace instead of an empty one
- ☐ Directly enter debugging when errors appear
- ☒ Command line options:

## Working directory settings

- ☐ The directory of the file being executed
- ☐ The current working directory
- ☒ The following directory:

## External system terminal

- ☐ Interact with the Python console after execution
- ☐ Command line options:

☐ Always show this dialog on a first file run

Run

× Cancel

✓ OK

IPython cons...

F...

H...

Variable explo...

PL...

Hist...

LSP Python: ready

Line 1, Col 1

ASCII

LF

R

Mem 9%





# Run



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open:

grbgetkey bba60259-a126-e14f-dab2



OK

Cancel

Browse...

```
c:\gurobi910>gurobi_cl c:\gurobi910\win64\examples\data\coins.lp
```