

Gurobi 13 升级全攻略

(版本 20251114)

〇、	本次更新说明.....	2
一、	Gurobi 13 核心变化.....	3
二、	Gurobi 13 可以求解的问题类型.....	5
三、	Gurobi 13 安装路线图.....	6
四、	Gurobi 13 使用 GPU 加速功能.....	8
五、	Gurobi 13 学习资料.....	9
六、	Gurobi 13 许可文件更新方法.....	11
	(1) 学术许可更新方法.....	11
	(2) 商业许可更新方法.....	11
	(3) 许可文件必备知识.....	12
	(4) 如何验证 Gurobi 激活成功.....	13
	(5) 原厂关于许可升级的指南.....	13
七、	Gurobi 13 Python 模块库安装方法.....	14
八、	Gurobi 13 机器硬件环境配置建议.....	17
九、	Gurobi 13 支持的操作系统和其他软件版本.....	18
	(1) 支持的操作系统.....	18
	(2) 支持的其他软件版本.....	18
	(3) 支持的 GPU 平台.....	18
十、	Gurobi 13 详细功能变化.....	20
十一、	Gurobi 13 交互界面.....	21
十二、	Gurobi 13 快速模型优化和测试指南.....	22
十三、	Gurobi 非线性函数表达方式和优化方式.....	25
十四、	Gurobi 基本概念之变量、约束、目标和表达式.....	30
十五、	如何快速判断模型可能存在数值问题.....	32
十六、	Gurobi 默认变量为非负变量.....	33

○、 本次更新说明

本文档于 2025 年 11 月 14 日做了部分更新，主要包括：

(1) Gurobi 13.0.0 版本发布

Gurobi 13.0.0 详细变化见

<https://docs.gurobi.com/projects/optimizer/en/13.0/reference/releasenotes.html>

Gurobi 13 升级全攻略

Gurobi 13.0 已经发布。我们希望这个文档可以回答关于 Gurobi 13.0 安装和升级的主要问题。如果用户有任何问题，以下是联系方式：

Gurobi QQ 技术群：一群 251135672；二群 705288945；三群 681080916；四群 491711468；五群 663846679；六群 740757471。群信息的更新见

<http://www.gurobi.cn/about.asp?id=2>

Gurobi 中国网站 www.gurobi.cn 邮箱: help@gurobi.cn

一、Gurobi 13 核心变化

（一）性能提升

在连续问题、整数问题、线性和二次凸问题、二次非凸问题、广义非线性问题等所有问题类型求解速度进一步提升。

Gurobi 13.0 比 12.0 的速度提升比例		
类型	整体提升	大于 100 秒的复杂问题提升
LP	0.6%	3.9%
MIP	8.2%	15.9%
MIQP	5.1%	7.3%
MIQCP	13.2%	25.8%
Nonconvex MIQCP	54.7%	2.68x
MINLP	2.52x	(6.34x)*
IIS	67.8%	(2.01x)*
*测试数据量有限		

（二）重要功能更新

（1）增加基于 GPU 架构的 PDHG 算法，可以加快特定大规模 LP 和 MIP 模型的求解速度。用户可以在普通 CPU 架构中启动 PDHG 算法，也可以在具备 NVIDIA 先进显卡的环境中启动基于 GPU 的 PDHG 算法，通过 Method 和 PDHGGPU 参数控制算法启动。

（2）增加针对纯连续广义非线性模型的 NLBarrier 非线性内点法，可以更快速度获得局部最优解。这个算法可以补充 Gurobi 已有的针对纯连续广义非线性采用的全局优化算法，而提供给一些需要更快获得局部最优解的场景使用。用户可以设置 OptimalityTarget 参数为 1 来启动这个算法。

(3) 丰富了 NoRel 启发算法功能。用户可以通过设置 NoRelHeurSolutions 来控制 NoRel 算法的终止, 以及通过设置 VarHintVal 变量属性值, 来给 NoRel 提供引导值。

(4) 用户可以在 Callback 回调函数中, 指定哪些 where 事件可以触发 Callback 的响应。那些没有指定的 where 事件则会被忽略。这个设定可以提升含有 Callback 函数的优化模型的求解速度。

(5) 对于 Threads 参数增加了新的选项。通过设定 Threads 参数等于-1, 可以让 Gurobi 调用全部可用的线程对问题进行求解。在之前 Threads=0 的默认设定中, Gurobi 只会使用不超过 32 个线程的最高限制。

(6) 支持的广义非线性函数增加了二个新的函数, 一个是双曲正切函数, 另外一个带符号的幂函数。

(7) 日志增加了更多有参考意义的输出信息。

(8) 调优时用户可以设置需要忽略的参数。

(9) 调优时用户可以像正常优化那样设置分支优先级(通过变量 BranchPriority 属性), 以及设置多组初始解。

(10) 增加了多目标优化属性信息。用户可以在完成多目标优化之后, 获得每个目标优化之后的更多有用参数。

详细变化说明请参考

<https://docs.gurobi.com/projects/optimizer/en/13.0/reference/releasenotes/changes.html>

二、 Gurobi 13 可以求解的问题类型

Gurobi 是全球综合能力领先的全局精确算法数学规划求解器。目前最新版本 13.0, 可以采用全局精确算法（获得全局最优解）求解的问题类型包括：

- （1）线性约束和目标模型（连续变量、混合整数，全局精确最优）、
- （2）二阶锥模型（连续变量、混合整数，全局精确最优）
- （3）二次凸约束和目标模型（连续变量、混合整数，全局精确最优）
- （4）二次非凸（双线性、二次等式约束、分母带变量、高阶多项式等）约束和目标模型（连续变量、混合整数，全局精确最优）
- （5）非线性模型（除式、高阶多项式、指数、对数、三角函数、范数、逻辑函数等）（连续变量、混合整数，全局精确最优）（纯连续变量，局部最优）
- （6）基于 GPU 加速的 PDHG 算法用于求解大规模连续变量线性模型

对于以上模型，可以叠加的功能包括但不限于：

- （1）约束和目标中带有最大、最小、绝对值等数学函数，或者带有 AND、OR、INDICATOR 逻辑条件的模型（连续变量、混合整数）
- （2）多目标优化
- （3）需要获得部分或者全部可行解或者最优解的模型
- （4）不可行约束冲突分析
- （5）优化参数自动调优功能
- （6）分布式计算和多并发计算

更多 Gurobi 功能见参考手册 (<https://docs.gurobi.com/13.0/>)。

三、 Gurobi 13 安装路线图

- 旧版本可以保留也可以卸载。如果不再使用，建议卸载。
- 用任何邮箱注册英文官网 www.gurobi.com，以便未来获得更多第一手资料。注册是否成功不影响后续操作。
- 从英文官网或者 Gurobi QQ 技术群（一群 251135672；二群 705288945；三群 681080916；四群 491711468；五群 663846679；六群 740757471。群信息更新见 <http://www.gurobi.cn/about.asp?id=2>）的群文件中获得最新版本安装包。

安装包区分支持 GPU 和 不支持 GPU 二种，用户需要关注安装包文件名中是否带有 gpu 字样。只有 Linux 和 ARM Linux 操作系统支持 GPU 加速功能，Windows 和 Mac 操作系统没有 GPU 加速的安装包。对于支持 GPU 加速的硬件和软件要求请参见本文档的《Gurobi 13 支持的操作系统和其他软件版本》章节。用户根据实际需要，只需要选择其中一个安装包安装。

安装包名称	适用的操作系统	是否支持 GPU 加速
Gurobi-13.0.0-win64.msi	Windows	否
gurobi13.0.0_macos_universal2.pkg	Mac	否
gurobi13.0.0_linux64.tar.gz	Linux	否
gurobi13.0.0_armlinux64.tar.gz	ARM Linux	否
gurobi13.0.0_linux64gpu.tar.gz	Linux	是
gurobi13.0.0_armlinux64gpu.tar.gz	ARM Linux	是

Windows/Mac 包直接双击安装，Linux/ARM Linux 包用 `tar xvfz` 安装并设置环境变量，在线指南 <https://support.gurobi.com/hc/en-us/articles/14799677517585>

- 获取许可文件：注册 portal.gurobi.com 自助申请 Named-User Academic 许可，需要验证学校 IP 地址；或者去 <http://www.gurobi.cn/NewsView1.Asp?id=4> 提交资料申请免 IP 学术许可。获得激活码之后，打开命令行窗口（通过 `cmd` 命令或者打开终端窗口），用 `cd` 命令进入到 gurobi 安装目录/bin 目录下（不熟悉 `cd` 命令的用户可以百度一下），输入 `grbgetkey xxx` 完整激活码后回车，在提示信息中输入保存许可文件的目录名称（请看本文档“许可文件必备知识”中许可文件建议存放目录，并事先创建好这些目录），并回车，在联网状态下会自动产生许可文件并放置到合适位置。请看本文档“如何验

证 Gurobi 激活成功”，确保激活成功后再进行后续操作。

- 在安装和激活之后，再进行各种开发编程语言包的安装
 - Python: 见后续章节 Gurobi 13.0 Python 模块库安装方法
如果能顺利运行 gurobi 安装目录/examples/python/mip1.py 范例，说明安装成功。
 - Matlab: <https://support.gurobi.com/hc/en-us/articles/4533938303505#section:MATLAB>
如果能顺利运行 gurobi 安装目录/examples/matlab/mip1.m 范例，说明安装成功。其他范例在相同目录下。
 - Visual Studio C++: <https://support.gurobi.com/hc/en-us/articles/360013194392-How-do-I-configure-a-new-Gurobi-C-project-with-Microsoft-Visual-Studio-2017->
如果能顺利运行 gurobi 安装目录/examples/c++/mip1_c++.cpp 范例，说明安装成功。其他范例在相同目录下。
 - Java: <https://support.gurobi.com/hc/en-us/articles/360013193472-How-do-I-use-Gurobi-in-an-Eclipse-Java-project->
如果能顺利运行 gurobi 安装目录/examples/java/Mip1.java 范例，说明安装成功。其他范例在相同目录下。
 - .NET: <https://support.gurobi.com/hc/en-us/articles/17307672758417-Tutorial-Getting-Started-with-the-Gurobi-NET-API>
如果能顺利运行 gurobi 安装目录/examples/c#/mip1_cs.cs 范例，说明安装成功。其他范例在相同目录下。
 - 第三方建模工具（AMPL, GAMS, CVXPY, PYOMO, PULP, YALMIP 等等）看该工具配置 Gurobi 的指南
- 手册和随机范例：Gurobi 安装目录/examples/ 下的范例涵盖了 Gurobi 常用的知识点。在线参考手册地址：<https://docs.gurobi.com/13.0/>，用户可以下载离线版的 HTML 参考手册以及 PDF 参考手册。Gurobi 13.0 所有帮助信息入口为本地安装目录下的 ReleaseNotes.html 文件。
- 每次大版本升级，建议用户对之前版本模型中设置的优化参数进行重新评估，重新进行参数调优，确定是否保留、删除还是修改。

四、 Gurobi 13 使用 GPU 加速功能

Gurobi 13 增加了基于 GPU 架构的 PDHG 算法，可以加快特定大规模 LP 和 MIP 模型的求解速度。用户可以在普通 CPU 架构中启动 PDHG 算法，也可以在具备 NVIDIA 先进显卡的环境中启动基于 GPU 的 PDHG 算法，通过 `Method` 和 `PDHGGPU` 参数控制算法启动。启动的方法如下：

- (1) 请看本文档《Gurobi 13 支持的操作系统和其他软件版本》章节，配置合适的操作系统、显卡和显卡驱动程序。
- (2) 如果使用 Python 以外的编程语言，请看本文档《Gurobi 13 安装路线图》章节，下载和安装带有 GPU 功能的安装包。
- (3) 如果使用 Python 编程语言，请看本文档《Gurobi 13 Python 模块库安装方法》章节，离线安装支持 GPU 功能的 `gurobipy` 包。
- (4) 在使用过程中，需要设置参数 `Method=6, PDHGGPU=1` 二个参数，来自动启动 GPU 算法。

注意 Gurobi 13 中的 GPU 加速功能仍然处于实验和开发过程中，并非常规成熟功能之一，原厂只提供有限度技术支持。用户如果发现问题，可以向原厂反馈。

五、 Gurobi 13 学习资料

Gurobi 英文和中文网站提供了丰富的学习资源和互动论坛，收藏这些链接，加速 Gurobi 学习和进阶过程。

手册和随机范例：Gurobi 安装目录/examples/ 下的范例涵盖了 Gurobi 常用的知识点。在线参考手册地址：<https://docs.gurobi.com/13.0/>，用户可以下载离线版的 HTML 参考手册以及 PDF 参考手册。Gurobi 13.0 所有帮助信息入口为本地安装目录下的 ReleaseNotes.html 文件。

【1】数学规划基本概念视频集合。从最简单的案例出发，快速熟悉数学规划的各种概念，包括可行域，对偶原理，敏感度分析，分支定界等等。Gurobi 提供了通俗易懂的短视频集合。

线性规划概念介绍：<https://www.gurobi.com/resource/mathematical-programming-tutorial-linear-programming/>

混合整数规划概念介绍：<https://www.gurobi.com/resource/tutorial-mixed-integer-linear-programming/>

【2】Gurobi 代码示范程序。Gurobi 提供了众多经典小案例和示范程序代码，可以让学习人员快速掌握建模和编码方法。

功能代码示范案例：这些程序都保存在本地的 Gurobi 安装目录下，覆盖不同的 Gurobi 功能设计。<https://www.gurobi.com/resource/functional-code-examples/>

行业应用程序示范案例：源自多个应用领域、具有代表性的应用示范案例。有问题背景，有程序代码，有输出结果，统一在 Jupyter Notebook 中。<https://www.gurobi.com/resource/modeling-examples-using-the-gurobi-python-api-in-jupyter-notebook/>

【3】技术论坛。Gurobi 技术论坛是学术许可用户获得技术支持的最好途径，同时也是发现 Gurobi 各种使用经验的最佳入口。

<https://support.gurobi.com/hc/en-us/community/topics>

【4】知识库。这些提供了各种常见问题的技术解答。

<https://support.gurobi.com/hc/en-us/categories/360000840331-Knowledge-Base>

【5】希望从其他开源优化器(GLPK, LP_Solve 等)或者其他商业优化器(IBM Cplex, FICO Xpress)转入到 Gurobi? 非常容易, 跟着这些转换文档操作就可以。

<https://www.gurobi.com/resource/switching-to-gurobi/>

【6】中文培训视频。Gurobi 中文网站提供了全面而系统的 Gurobi Python 培训视频, 特别包含了实战案例, 可以直接应用在企业业务中。

<http://www.gurobi.cn/picexhview.asp?id=90>

【7】Gurobi 相关的酷应用

Gurobi 发布或更新了很多非常酷的应用, 包括:

(1) 更新了“模型诊断器” Gurobi Model Analyzer, 帮助用户更好发现模型本身的问题。网址: <https://gurobi-modelanalyzer.readthedocs.io/>

(2) 更新了最优潮流 ACOPF 和 DCOPF 开源模型 Gurobi OPF OptiMod, 更好更快发挥应用价值。网址: <https://gurobi-optimods.readthedocs.io/>

(3) 提供了高质量 Gurobi 人工智能答疑助手 GuroBot, 基于 ChatGPT, 可以高质量解答 Gurobi 技术问题, 达到专家水准。网址: <https://www.gurobi.com/resources/how-well-can-an-llm-answer-technical-questions-about-gurobi>

(4) 提供了 Gurobi AI 建模工具 Gurobi AI Modeling, 根据用户交互方式, 把实际问题转换成高质量模型。网址: <https://gurobi-ai-modeling.readthedocs.io/>

六、 Gurobi 13 许可文件更新方法

(1) 学术许可更新方法

【对于免 IP 验证学术许可，仍在有效期内的用户无需申请新激活码，只需要重新运行原激活码，产生并覆盖原许可文件即可。在官网自助获取的 IP 验证学术许可，需要重新自助申请新的激活码】

- **【查验】** 如果已经有正在使用的 `gurobi.lic` 许可文件，打开查看，如果版本号已经更新为 13，`VERSION=13`，并且 `EXPIRATION` 到期日期仍然有效，那么许可更新工作已经完成。
- **【更新】** 如果许可仍然在有效期内，找到当初产生许可文件的激活码 `grbgetkey` XXXXX（或者从 www.gurobi.com 个人账户中获得，或者通过邮件 help@gurobi.cn 获得），在保持激活的机器和用户名不变的情况下，重新联网运行此激活码，产生新的许可文件，替换掉之前的许可文件，就可以正常使用。
- **【注意】** 如果找不到激活码，请发邮件到 help@gurobi.cn 中文邮箱，发送邮件时请提供 `gurobi.lic` 许可文件中的序列号（License ID）。
- **【检验】** 打开更新后的 `gurobi.lic` 许可文件，版本号已经更新为 13，`VERSION=13`，并且 `EXPIRATION` 到期日期仍然有效。
- **【重新申请】** 如果更新许可文件后仍然无法运行版本 13.0，或者之前没有申请过激活码，则需要重新申请。学术许可有二个申请途径，（1）通过英文官网自助获取，这样获取的激活码在激活时需要验证用户的 IP 地址是否标注为学术机构；（2）通过中文官网 <http://www.gurobi.cn/NewsView1.Asp?id=4> 提交申请资料，这样的激活码在激活时不进行学术 IP 的验证；

(2) 商业许可更新方法

对于在维保期内的商业用户，我们会逐个联系，安排合适的升级时间。如果需要试用商业许可，请按照 <http://www.gurobi.cn/NewsView1.Asp?id=9> 的要求填写表格并保存后

发到 help@gurobi.cn 邮箱。

(3) 许可文件必备知识

(1) 先安装软件，再通过联网运行 `grbgetkey + 激活码` 获得 `gurobi.lic` 许可文件并保存到本地。每台机器上只需要保存一个 `gurobi.lic` 文件。请删除多余。

(2) `gurobi.lic` 许可文件的建议保存目录

Windows: `c:\gurobi`

Linux: 根目录/`opt/gurobi`

Mac: 根目录/`Library/gurobi`

如果没有这些目录，请创建一个。这是 Gurobi 默认搜索位置。

如果一台机器上有多个用户账户，每个账户都申请了不同的许可文件，那么请把这些许可文件放到每个账户的 `home` 目录下（请百度一下 Linux, Windows, Mac 系统中登录账户的 `home` 目录是哪里）。

如果保存到其他位置，需要创建 `GRB_LICENSE_FILE` 环境变量，指向保存的目录和文件名（例如 `GRB_LICENSE_FILE = C:\myfolder\gurobi.lic`）。如果设置的是系统环境变量，则需要重启电脑使之生效。

(3) 打开 `gurobi.lic` 文件，显示许可过期日期（`EXPIRATION=`），以及支持的最大版本号（`VERSION=`）。许可文件不能编辑。许可到期后，重新申请获得新激活码，产生新的 `gurobi.lic` 许可文件替换掉旧许可文件。

(4) 如果既放置在 (2) 中的位置，又设置了 `GRB_LICENSE_FILE` 系统环境变量，Gurobi 会以 `GRB_LICENSE_FILE` 的设置为准。

(5) 激活码一旦激活，就和机器捆绑，无法转移、无法注销。更换机器、重装系统、更换用户账号都需要申请新激活码。如果需要新的激活码，请重新提交资料再次申请。每个人的免 IP 许可激活码的数量是有限的，需要珍惜使用。

(6) 个人学术许可不支持安装到 WSL2 系统中，或者 Docker 容器机器内（例如超算客户端，K8s 客户端等）。如果需要使用 WSL2 或者容器机，需要申请学术院系许可。

(7) 打开 `gurobi.lic` 许可文件，如果显示 `TYPE=TRIAL` 或者 `TYPE=PIP`，那么这些许可仅供培训或者个人学习使用，有 2000 个变量和约束的限制（如果是二次模型，则变

量限制为不超过 200 个)。如果超过规模, 运行时会产生 `Model too large for size-limited license` 信息。

(8) 免 IP 学术许可、商业测试许可的申请网站: www.gurobi.cn

(4) 如何验证 Gurobi 激活成功

产生并放置正确的许可文件之后, 打开命令行窗口, 通过 `cd` 命令进入到 `gurobi` 安装目录\bin 目录下, 输入

```
gurobi_cl --license
```

出现以下信息就表明软件已经安装成功, 并且许可文件已经配置成功。同时, 这个信息也指出了生效的许可文件的路径。如果生效的许可文件路径不是用户期望的, 可以进行调整和修改。

```
Set parameter LicenseID to value XXXXX  
Set parameter LogFile to value "gurobi.log"  
Using license file c:\gurobi\gurobi.lic
```

在验证成功之后, 用户可以再根据自己开发语言的需要, 安装不同的环境, 例如 `Eclipse`, `Visual Studio`, `R`, `Matlab` 等。详见本文档安装路线图部分。

(5) 原厂关于许可升级的指南

用户也可以参考原厂关于许可升级的指南

<https://support.gurobi.com/hc/en-us/articles/360038212992-How-do-I-update-my-Gurobi-installation-to-a-newer-version>

七、 Gurobi 13 Python 模块库安装方法

- **Gurobi 13.0.0 支持的 Python 版本包括 3.10 - 3.13**
- **用户先安装 Python IDE 编辑工具（例如 PyCharm, Anaconda/Spyder 等）之后，再按照本指南中的步骤操作。**
- **不论使用何种 IDE 或者 Python 环境，用户需要明确正在使用的 Python 执行文件的位置和目录。gurobipy 将会被安装到该 Python 执行程序所在目录下的 \Lib\site-packages 目录下。**

以下总结了在 Python 中安装 Gurobi 13.0 的安装方法。如果需要更多指示，可以参考随机手册。

gurobipy 包区分支持 GPU 加速功能和不支持 GPU 加速功能二个包。不支持 GPU 加速功能（也就是常规功能）的 gurobipy 包可以通过在线和离线二种方式安装，而支持 GPU 加速功能的 gurobipy 包因为还处于实验发展阶段，目前只支持离线安装方式。对于支持 GPU 加速的硬件和软件要求请参见本文档的《Gurobi 13 支持的操作系统和其他软件版本》章节。

第一部分：不带 GPU 加速功能（常规功能）的 gurobipy 包安装方式

（一）在线安装：用户电脑需保持连接互联网

第一种方法：pip 安装

进入到激活环境的 Python 命令行窗口（如果安装到 Anaconda 里，以管理员权限进入到 Anaconda Prompt 窗口并进入到激活环境；如果安装到 PyCharm 里，则可以点击 IDE 左下角 Terminal 打开命令行窗口；如果是其他 Python IDE，则进入到对应激活环境的命令行窗口），联网运行

```
pip install gurobipy 或者（如果镜像源没有及时更新）
pip install -i https://pypi.org/simple gurobipy
```

则将 Gurobi 模块（非 Gurobi 完整安装包）安装到当前激活的 Python 环境中。

如果在 pip 中升级 Gurobi，则输入

```
pip install gurobipy --upgrade
```

或者指定版本

```
pip install gurobipy==9.1.2
```

如果需要卸载旧版本，输入

```
pip uninstall gurobipy
```

第二种方法：Anaconda 安装

如果你使用 Anaconda，以管理员权限进入到 Anaconda Prompt 窗口，通过 conda 命令运行

```
conda install -c gurobi gurobi
```

如果在 Anaconda 中升级 Gurobi，则输入

```
conda update gurobi
```

或者指定版本

```
conda install gurobi=9.1.2
```

如果删除旧版本，输入

```
conda remove gurobi
```

获取 Gurobi 许可文件：

第一种 pip 安装方法在安装 gurobipy 时会自动产生一个有时效限制的 2000 个变量和约束限制的培训许可（许可文件 gurobi.lic 中显示 TYPE=PIP），用户只能在 Python 环境中使用。这个许可受到版权限制，不能应用于论文发表、商业部署、项目实施等非培训和学习场合。

如果用户需要产生更高级别的许可（例如无限制的学术许可，院系许可，商业许可、离线许可等），需要运行 Gurobi 专门的激活工具（例如 grbgetkey 和 grbprobe）。这些激活工具不包含在 gurobipy 模块库中，需要安装 Gurobi 独立安装包，或者单独从以下地址下载并解压缩后使用：

<https://support.gurobi.com/hc/en-us/articles/360059842732>

申请 Gurobi 许可请参见本文档前述章节。

（二）离线安装

我们强烈建议用户采用在线安装的方式，因为 `pip` 和 `conda` 都提供了良好的管理功能，支持自动下载依赖包，帮助用户更轻松的管理支撑环境。

如果用户电脑不联网，可以通过 `pip download` 或者 <https://pypi.org/simple/gurobipy/> 下载和保存与操作系统、Python 版本相匹配的 `wheel` 文件，然后按照在线安装中第一种 `pip` 安装方法，进入到 `wheel` 文件保存的目录，运行 `pip install` 命令。例如

```
cd d:\myfolder

pip install gurobiwheelfile.whl
```

对于复杂的 `pip` 离线安装，特别是存在依赖库的安装，请用户自行百度搜索方法。

第二部分：带 GPU 加速功能的 `gurobipy` 包安装方式

(1) 登录到官网软件下载和操作系统、Python 版本匹配的 `gurobipy wheel` 包

操作系统	Python 版本	下载地址
Linux	3.10	https://packages.gurobi.com/13.0/wheels/linux64gpu/gurobipy-13.0.0+cu129-cp310-cp310-manylinux_2_24_x86_64.whl
Linux	3.11	https://packages.gurobi.com/13.0/wheels/linux64gpu/gurobipy-13.0.0+cu129-cp311-cp311-manylinux_2_24_x86_64.whl
Linux	3.12	https://packages.gurobi.com/13.0/wheels/linux64gpu/gurobipy-13.0.0+cu129-cp312-cp312-manylinux_2_24_x86_64.whl
Linux	3.13	https://packages.gurobi.com/13.0/wheels/linux64gpu/gurobipy-13.0.0+cu129-cp313-cp313-manylinux_2_24_x86_64.whl
ARM Linux	3.10	https://packages.gurobi.com/13.0/wheels/armlinux64gpu/gurobipy-13.0.0+cu129-cp310-cp310-manylinux_2_26_aarch64.whl
ARM Linux	3.11	https://packages.gurobi.com/13.0/wheels/armlinux64gpu/gurobipy-13.0.0+cu129-cp311-cp311-manylinux_2_26_aarch64.whl
ARM Linux	3.12	https://packages.gurobi.com/13.0/wheels/armlinux64gpu/gurobipy-13.0.0+cu129-cp312-cp312-manylinux_2_26_aarch64.whl
ARM Linux	3.13	https://packages.gurobi.com/13.0/wheels/armlinux64gpu/gurobipy-13.0.0+cu129-cp313-cp313-manylinux_2_26_aarch64.whl

(2) 按照本章第一部分的离线安装方式，将 `gurobipy` 安装到对应的 Python 目录下。

(3) 需要申请合适的许可（学习许可/学术许可/商业许可）才能使用。

八、 Gurobi 13 机器硬件环境配置建议

好马配好鞍。良好的机器配置能够充分发挥 Gurobi 的优秀性能。机器硬件配置几个关键因素包括：

（一）CPU 主频（CLOCK RATE）和内存通道数（MEMORY CHANNEL）

CPU 的单线程主频速度对于充分发挥 Gurobi 的性能非常重要。作为参考，第三方测试平台在评测优化器性能时采用的主频速度在 3.5 G hz - 4.4 G hz。如何选择最新的芯片？可以参考 CPU 单线程性能排行榜（见链接），帮助选择合适的机器配置。

<https://www.cpubenchmark.net/singleThread.html>

同时，如果 CPU 可以支持更大的内存通道数，则可以增加数据处理容量，有助于加快速度。高性能服务器 CPU 可以支持的最大内存通道数一般在 6-8 个左右。

较高的 L1/L2 CPU 缓存对于性能也有很大帮助。

（二）高性能随机存取存储器（RAM）

高性能体现在

（1） 充足的容量，避免因为内存不足导致无法求解大规模问题。Gurobi 软件本身对于问题规模没有限制，因此问题规模仅受制于内存容量。

（2） 选择高带宽、低延迟，例如 DDR4 内存。

（3） 可以参考 RAM 性能排行榜（见链接），帮助选择合适的内存。

<https://www.memorybenchmark.net/>

（三）物理核心数（CORES）

较多的核心数对于求解 MIP 问题可能会有帮助，但高度依赖于问题的类型。当并发模型数量较多时，较多的核心数可以有帮助。

九、 Gurobi 13 支持的操作系统和其他软件版本

13.0.0 版本支持的操作系统和其他软件环境

(1) 支持的操作系统

Platform (port)	Operating System	Compiler
Windows 64-bit (win64)	Windows 10 (LTSC), 11, Windows Server 2016, 2019, 2022, 2025	Visual Studio 2017-2022 [1]
Linux x86-64 64-bit (linux64)	Red Hat Enterprise Linux 8, 9, 10	GCC >= 8.5
	SUSE Enterprise Linux 15	
	Ubuntu 22.04, 24.04	
	Amazon Linux 2023	
macOS 64-bit universal2 (macos_universal2)	13 (Ventura), 14 (Sonoma), 15 (Sequoia)	Xcode 14/15/16
Linux arm64 64-bit (armlinux64)	Red Hat Enterprise Linux 8, 9, 10	GCC >= 8.5
	SUSE Enterprise Linux 15	
	Ubuntu 22.04, 24.04	
	Amazon Linux 2023	
Gurobi 也支持国产芯片和国产操作系统，详见 http://www.gurobi.cn/NewsView.Asp?id=125		

(2) 支持的其他软件版本

Language	Version
Python	3.10-3.13
MATLAB	R2019a-R2025a
R	4.5
JDK	8, 11, 17, 21
.NET	8.0

(3) 支持的 GPU 平台

支持 Gurobi PDHG 算法的平台仅限 armlinux64 和 linux64 操作系统。仅支持 NVIDIA 硬件。Gurobi 推荐使用 NVIDIA H100 或者更高配置显卡，但只要适配 CUDA 计算能力 ≥ 8.0 , ≤ 12.0 的都支持。需要安装 NVIDIA 驱动程序，不必安装完整的 CUDA。驱动程序版本需要 ≥ 525 , ≤ 580 .

十、 Gurobi 13 详细功能变化

用户可以通过以下链接获得全面和完整的 Gurobi 13 功能变化说明

<https://docs.gurobi.com/projects/optimizer/en/13.0/reference/releasenotes.html>

十一、 Gurobi 13 交互界面

从 Gurobi 11.0.2 开始，Gurobi 逐步取消通过运行 `gurobi` 或者 `gurobi.sh` 命令启动 Gurobi 交互界面的功能。对于仍然希望使用 Gurobi 交互界面的用户，可以通过以下方式重现：

(1) 在命令行下，找到并进入安装了 `gurobipy` 模块库的 Python 执行程序的目录，输入并运行 `python` 命令，或者在命令行下输入完整路径的 Python 执行程序。这样启动 Python 自身的交互界面

(2) 在 `>>>` 提示下，输入（可以直接拷贝粘贴）

```
import os
from gurobipy import *
setParam("LogFile", "mylog.log")
```

如果提示找不到 `gurobipy`，那么请参考本文档的 Python 安装方法，安装 `gurobipy` 模块库。做为良好习惯，用户首先设置日志文件，这样在后续运行过程中都可以保留日志。安装 `os` 模块库的目的是为了在交互环境中，可以方便的使用一些系统命令，例如变换到某个特定的目录下，可以用以下命令

```
os.chdir(r"C:\gurobi1300\win64\examples\data")
```

可以用以下命令查看当前目录

```
os.system("cd")
```

一旦导入了 `gurobipy` 库，那么 Python 的交互环境就变成了 Gurobi 的交互环境。之前 Gurobi 交互环境中的所有命令，可以保持不变地应用到这里。例如用户可以读入模型文件再优化

```
model=read("mymodel.mps")
model.optimize()
```

用户可以使用交互环境快速测试模型的性能，验证语法和调试等。可以参考本文档中的《Gurobi 13 快速模型优化和测试指南》。

对于取消原有的 Gurobi 交互环境有任何疑问，可以参考以下的技术帖子

<https://support.gurobi.com/hc/en-us/articles/24811784733585-Deprecation-of-the-Gurobi-Interactive-Shell>

十二、 Gurobi 13 快速模型优化和测试指南

本指南是为快速运行 Gurobi, 并在已有模型文件基础上 (LP/MPS/REW 格式), 对 Gurobi 优化参数进行调整和性能测试而制作。这个快速指南并不能代替详细的 Gurobi 使用手册。

假设许可文件已经正确放置, Gurobi 可以正常运行。

(1) 启动 DOS 行命令方式。在左下角开始菜单中, 在运行 (RUN) 或者提示输入框中, 输入 `cmd`, 启动 DOS 命令行方式。

(2) 通过 `cd` 命令, 进入到模型数据文件所在的目录。Gurobi 支持的模型数据格式包括 MPS, LP, REW 等。

(3) 按照本文档《Gurobi 13 交互界面》章节的操作方法, 启动了 Gurobi 交互界面如下

```
C:\gurobi1200\win64\examples\data>C:\Python312\python
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> from gurobipy import *
>>> _
```

(4) 读入数据模型文件, 输入

```
m=read('abc.mps')
```

(5) 这样数据就读入到 `m` 变量中。用户如果需要日志输出的话, 需要设置

```
m.Params.LogFile="mylog.log"
```

这样就会在相同目录下产生日志文件。如果采用所有默认优化参数, 那么可以直接运行优化

```
m.optimize()
```

(6) 用 `ctrl-C` 中断优化运行。

(7) 输出优化结果, 可以将当前找到的可行解或者最优解输出, 可以输入

```
m.write('output.sol')
```

那么在当前目录下, 输出一个 `output.sol` 文件, 可以用文本编辑器打开。文件名称可以是任何名称, 只要后缀是 `sol` 就可以。

(8) 清除当前的最优解或者可行解, 可以输入

```
m.reset()
```

这样再输入 `m.optimize()` 时, 会从头开始运行; 否则, 会从当前可行解再继续进行优化。

(9) Gurobi 默认优化参数已经可以取得较好的结果，但也提供了优化参数调整的功能，更好地针对具体问题提升优化性能。常用的优化参数有 `MIPFocus`, `Presolve`, `Method`, `ImproveStartGap`, `ImproveStartTime` 等等。我们并不建议过度调整参数，因为 Gurobi 已经将可调整的参数大大缩减。这些参数的使用方法都是一样的，就是在运行优化之前，设定这些参数，设置方法为

```
m.Params.XXXX=Y
```

例如：如果希望以取得可行解为优化策略，那么可以设定为

```
m.Params.MIPFocus=1
```

然后再运行

```
m.optimize()
```

如果希望将优化参数的数值恢复到默认数值，可以采用

```
m.Params.XXXX='default'
```

如果希望清除所有优化参数的数值，而恢复到默认数值，可以采用

```
m.resetParams()
```

手册中详细提供了这些参数的取值范围，默认数值和功能。

(10) 一些常见情况的优化参数建议：

MIPFocus: 默认值 0，试图在最优值和可行解之间取得平衡。1：以可行解为目标；2：以证明最优性为目标；3：以优化边界为目标。

Presolve: 默认值：-1，自动决定预优化力度。0：关闭预优化；1：保守；2：激进。

Method: 默认值：-1，自动决定优化方法。0：原始单纯型；1：对偶；2：Barrier；3：随机并行；4：确定并行。如果模型巨大，可以考虑 2。

ImproveStartTime 和 **ImproveStartGap:** 都是确定从什么条件开始，优化转向对可行解质量的提升上。优化从 **ImproveStartTime** 设定的时间之后（以秒为单位），或者 **ImproveStartGap** 设定的收敛 gap 达到之后开始进行可行解的提升。

MinRelNodes: 如果模型在根节点没有找到可行解，可以考虑设置这个参数，例如 10000。

NoRelHeurTime / NoRelHeurWork: 对于很难找到可行解的复杂混合整数模型，可以考虑设置这二个参数启动 NoRel 启发式算法。单位为秒。

TimeLimit: 时间终止条件。当达到规定的运行时间后，优化终止。单位为秒。

MipGap: 偏差终止条件。当整数规划的偏差下降到设定值后，优化终止。默认为 0，一般可以设定为 0.05 或者 0.01

(11) 自动参数调优

Gurobi 提供了自动参数调优工具，可以自动寻找更好的优化参数。读入模型后，清除已设置的参数 `m.resetParams()`

然后设置自动调优时间 `m.Params.TuneTimeLimit=XXX` （以秒为单位）

运行调优工具 `m.tune()`

Gurobi 在调优结束时，显示较好的参数集合。具体说明请参考参考手册。

(12) 不可行或者无界模型

如果使用 Gurobi 时碰到“Model is infeasible or unbounded”信息时，首先可以设置参数 `DualReductions = 0`，然后再运行优化，就会得到明确的信息。如果不可行 (infeasible)，可以运行 `m.computeIIS()` 获得冲突的约束条件，通过 `m.write('abc.ilp')` 输出这些冲突的约束条件到 LP 文件浏览。如果是无界的，那么可以检查变量和约束的上下界设置，是否有可能出现无界情况。对于连续模型，可以设置参数 `InfUnbdInfo = 1` 来获得连续模型的 `UnbdRay`。

更多 Gurobi 信息，参见参考手册。

十三、 Gurobi 非线性函数表达方式和优化方式

Gurobi 从 10.0 版本开始支持指数、对数、三角函数等混合整数非线性模型。随着版本的不断更迭，非线性函数的表达方式和优化方式都在不断发生变化，现总结如下，更多详细资料请查看 Gurobi 的参考手册。注意这里指的非线性不包括二次表达式。对于二次表达式，无论是凸还是非凸（二个连续变量相乘），Gurobi 都支持直接表达，不要有特殊表示方式。

版本	表达方式	优化方式	是否全局 精确优化	需要设置的参数
10	(1) 单变量非线性函数组合方法	静态分段线性近似	否	FuncPieceError, FuncPieceLength, FuncPieceRatio, FuncPieces 等
11	(1) 单变量非线性函数组合方法	静态分段线性近似	否	FuncPieceError, FuncPieceLength, FuncPieceRatio, FuncPieces 等
		动态外逼近和空间 分支	是	FuncNonlinear = 1 (非默认)
12/13	(1) 单变量非线性函数组合方法	静态分段线性近似	否	FuncPieceError, FuncPieceLength, FuncPieceRatio, FuncPieces 等
		动态外逼近和空间 分支	是	FuncNonlinear = 1 (默认)
	(2) 操作符树状结构方法	动态外逼近和空间 分支	是	FuncNonlinear = 1 (默认)
	(3) (仅限 Python) 多变量非线性函数直接表示方法	动态外逼近和空间 分支	是	FuncNonlinear = 1 (默认)

在这篇文档中，我们重点介绍表达方式的区别，而对于优化方式的区别，有兴趣的用户可以通过参考手册详细了解。

我们举例说明不同表达方法之间的区别。

例如我们需要表示非线性函数约束 $z = \sin(2.5 * x_1) + x_2$ ，这是一个多变量非线性函数，其中 x_1, x_2 是二个变量。

(方法 1) 单变量非线性函数组合方法

这种方法需要引入多个中间变量，将多变量非线性函数拆分成多个单变量

非线性函数，然后通过引入 `addGenConstrSin()`, `addGenConstrExp()`, `addGenConstrLog()` 等单变量非线性函数组合而成。

针对我们给出的例子，我们需要定义 y_1, y_2 二个额外的变量，其中

$$y_1 = 2.5 * x_1$$

$$y_2 = \sin y_1$$

最后我们的非线性函数可以用 $z = y_2 + x_2$ 来表示。用 Python 语言书写如下（假设 x_1, x_2, y_1, y_2, z 已经通过 `model.addVar()` 定义为变量）

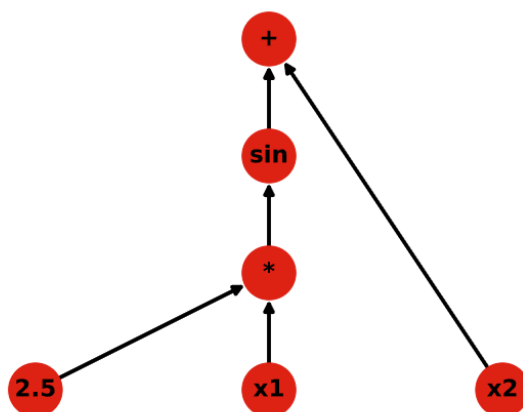
```
model.addConstr(y1 == 2.5 * x1)
model.addGenConstrSin(y1, y2)
model.addConstr(z == y2 + x2)
```

因为 Gurobi 的 `addGenConstrSin()` 等非线性函数只能接受单变量做为参数，不接受多变量和表达式，因此用户需要通过引入中间变量，不断化繁为简，变成多个单变量非线性函数再组合到一起。

（方法 2）操作符树状结构方法

Gurobi 12/13 引入了操作符树状结构来描述一个复杂的多变量非线性函数。任何多变量复合非线性函数都可以通过一个操作符树状结构来描述。

例如对于本文档中的例子 $\sin(2.5 * x_1) + x_2$ 包括了加法、乘法和 `sin` 三个操作符。依赖关系和运算次序如下：`sin(2.5 *  $x_1$ )` 和 x_2 的加法是最顶层。第一个部分是 `sin` 函数，依赖于常数 2.5 和变量 x_1 的乘积。按照数据流动的方向，我们可以得到如下的关系图



为了把这个树状结构的每个节点传递给 Gurobi API 函数，需要进行编码。编码由三个数组组成，每个数组的第 i 个元素代表了第 i 个节点。因此节点 i 由三个数组中的第 i 个元素的信息来确定。

这三个数组分别是

- 操作符数组 (*opcode array*)，它包含了每个节点的基本操作。完整的操作符列表见下面表格。
- 数据数组 (*data array*)。它包含了节点的辅助信息，例如常数值或者变量。如果节点不需要提供辅助信息，填写为-1
- 父节点索引数组 (*parent array*)。它通过提供父节点的索引编号来构造树状结构。根节点的父节点索引用-1 表示。

节点在树状结构中的表示顺序需要遵循一定的次序，也就是每个节点后面需要紧跟着所有子节点，并且连续排列。

本文档中的例子 $\sin(2.5 * x_1) + x_2$ 的编码表如下

array index	opcode	data	parent
0	plus	-1	-1
1	sin	-1	0
2	multiply	-1	1
3	constant	2.5	2
4	variable	x1	2
5	variable	x2	0

注意：只有后三列数组需要传递给 Gurobi API 接口，而第一列数据默认是从 0 开始递增的索引，不用传递给 Gurobi API 接口。

Gurobi 支持的操作符

注意这里的操作符可以直接表示在非线性函数中。特别是用户可以直接使用除法“/”符号，来表示分母带变量的非线性；直接用“**”符号来表示幂指数。

CONSTANT	VARIABLE	PLUS	MINUS	MULTIPLY	DIVIDE	UMINUS
常数	变量	加	减	乘	除	取反
SQUARE	SQRT	SIN	COS	TAN	POW	EXP
平方	平方根	正弦	余弦	正切	幂	指数
LOG	LOG2	LOG10	LOGISTIC			
自然对数	2-对数	10-对数	逻辑函数			

用 Python 语言书写如下（假设 x_1, x_2, z 已经通过 `model.addVar()` 定义为变量）

```
opcode = [GRB.OPCODE_PLUS, GRB.OPCODE_SIN,
GRB.OPCODE_MULTIPLY, GRB.OPCODE_CONSTANT,
GRB.OPCODE_VARIABLE, GRB.OPCODE_VARIABLE]
data=[-1,-1,-1,2.5,x1,x2]
parent =[-1,0,1,2,2,0]
model.addGenConstrNLAdv(z,opcode,data,parent)
```

如果用其他编程语言，需要采用类似的方法，定义好三个数组的数据，然后调用各自语言的 `model.addGenConstrNL()` 函数进行表达。

（3）（仅限 Python）多变量非线性函数直接表示方法

如果使用 Python 编程，Gurobi 提供了一个更方便、更便捷地表达多变量复合非线性函数的方法。`gurobipy` 提供了非线性函数库 `nlfunc`，支持直接表达多变量、复合非线性函数。也支持用户使用除法“/”符号，来表示分母带变量的非线性；直接用“**”符号来表示幂指数。

假设 x_1, x_2, z 已经通过 `model.addVar()` 定义为变量，并且在程序开始部分已经用 `from gurobipy import nlfunc` 导入了 `nlfunc` 库，那么用户只需要用一句话就可以完成非线性函数的表达

```
m.addGenConstrNL(z, nlfunc.sin(2.5*x1)+x2)
```

从以上说明中可以看出，**Gurobi** 支持多种非线性函数的表达方式。如果用户使用 **Python** 编程语言的话，采用 **nlfunc** 库是最方便的方法。

(4) 复杂范例

为了检验用户已经对这三种方法完全了解，我们给出如下优化问题，完整的 **Python** 优化代码可以从 http://www.gurobi.cn/download/grb_nlp_example_v12.py 下载

$$\text{maximize } \sqrt{\frac{(1 - \cos(x^2)) * y^{0.8} + 2xy}{1 + x + y}} - x$$

s.t. $0 \leq x \leq 10, 0 \leq y \leq 30, x, y$ 是整数

十四、 Gurobi 基本概念之变量、约束、目标和表达式

(1) 变量、约束和目标是构成 Gurobi 模型三个核心要素。而表达式只是构成这些要素的重要中间环节。

(2) 变量：Gurobi 只有变量的概念，不区分决策变量、辅助变量、中间变量、松弛变量等。所有变量在 Gurobi 中都需要通过 `model.addVar()`, `model.addVars()`, `model.addMVar()` 等来定义为变量。

(3) 约束是由变量通过表达式构成的，表达式本身只是构造约束的一个环节。常数*变量、常数*变量*变量（也就是线性项和二次项）可以任何加总构成线性或者二次表达式。不论变量的类型（Gurobi 支持二次凸和非凸模型），只要最终展开形式不超过二次，就可以直接写入到约束中。

常规的赋值语句并不能成为模型的约束，不具有约束力，只是构造表达式的中间环节。例如 `x`、`y`、`z` 是通过 `model.addVar()` 定义的变量，那么

`w=x+y` 并没有产生具有模型约束力的约束，而只是构成了一个线性表达式类型的 `w`

`w=(x+y)*x` 并没有产生具有模型约束力的约束，而只是构成了一个二次表达式类型的 `w`

只有 `model.addConstr()`, `model.addConstrs()` 等约束构造函数才能产生等式或者不等式约束，所以上面定义好 `w` 表达式之后，需要成为约束的一部分，例如

`model.addConstr(w <= 10)` 等价于 `model.addConstr(x+y <= 10)` 或者 `model.addConstr((x+y)*x <= 10)`

如果表达式中的阶次超过二次，需要降阶处理，才能成为约束。例如

`w=x**2*(x+y)` 这个表达式超过二次，需要通过 `model.addVar` 引入中间变量，例如 `z`，进行降阶，变成 `w=z*(x+y)`, `z=x**2`，最终写法是

`w=z*(x+y)`

`model.addConstr(w <= 10)`

`model.addConstr(z==x**2)` 不能直接写为 `z=x**2`，不但没有约束力，而且还会把 `z` 的类型由变量变为二次表达式类型。

约束是否超过二次，以展开后的已定义变量的阶次为准。如果上面约束写

`model.addConstr(w*w<=10)` 那么因为 w 是表达式，不是已定义变量，那么需要展开，以展开后的 x, y 的最高阶次来判断，如果没有超过二次，就没有问题。如果 w 已经定义为变量，那么这个约束不需要展开，只有二次，没有问题。

线性和二次表达式中不能混入 SOS 和广义函数表达式，例如 $x+y+x^{0.2}+\max(x,y)$ 是不允许的。 $x^{0.2}$ 和 $\max(x,y)$ 需要引入中间变量，配合约束条件，进行替换，请看下面的说明。

(4) 约束分为线性约束，二次约束，SOS 约束，广义约束四大类别，每个类别有各自构造方法、属性、和获取方法。定义中间变量，并配合约束条件，往往是进行降阶、消除非线性项的常用方法。例如目标函数是 `minimize (x^0.2+max(x,y))` 这样混合指数函数和取变量最大值函数的复杂表达式，而 Gurobi 要求单目标只能是线性或者二次表达式，那么可以定义二个额外的变量

```
u=model.addVar()
```

```
v=model.addVar()
```

将 $u=x^{0.2}$ 和 $v=\max(x,y)$ 通过广义约束的方式定义为

```
model.addGenConstrPow (x,u,0.2)
```

```
model.addConstr (v==gurobipy.max_(x,y))
```

这样目标函数就可以变成 `minimize (u+v)` 这样的线性表达式。用户可以参阅参考手册的详细说明。

(5) Gurobi 支持单目标和多目标。一个模型的目标不是必须的。如果没有定义，那么默认目标为 0，模型只要获得一个可行解就会终止。

单目标既可以通过 `model.setObjective()` 定义，也可以在定义变量时，在 `model.addVar/model.addVars` 函数中的 `obj` 参数定义。单目标只能是线性或者二次表达式，其他形式参考 (4) 进行转换。具体使用请看参考手册。

Gurobi 提供了强大的多目标优化功能，用户可以以混合方式或者分层方式来优化多个目标。但 Gurobi 要求每个多目标的目标函数需要是线性表达式。如果用户的目标中出现了非线性项，用户可以参考 (4) 中的方法。

十五、 如何快速判断模型可能存在数值问题

做为科学计算软件，求解器优化时都设定了特定的计算精度和理想的数值范围。如果用户的模型能够遵循这些精度设置，那么对于求解速度和求解结果的精度都有很大帮助，反之则可能会造成求解速度变慢，结果精度变差的现象。

为了获得最大数值稳定性和运算效率，Gurobi 建议：

(1) 变量、约束条件、目标的系数取值合适的范围，使得变量的优化值尽量范围在【-1e+4,1e+4】之间，约束取值在【-1e+4,1e+4】之间，目标取值【-1e+4,1e+4】之间

(2) 查看日志，最大系数/最小系数 的比值，对于线性模型控制在 1e+9 之内，二次和非线性模型控制在 1e+6 之内，见图片

Optimize a model with 45616 rows, 38073 columns and 266705 nonzeros			
Model fingerprint: 0xafed6844			
Model has 10164 quadratic constraints			
Model has 567 general constraints			
Variable types: 35301 continuous, 2772 integer (0 binary)			
Coefficient statistics:			
Matrix range	1e-03,	9e+03]	—— 选出最大值 最大值/最小值 比值尽量控制在以下范围内:
QMatrix range	1e+00,	2e+03]	
QLMatrix range	6e-01,	1e+03]	—— 选出最小值 线性模型 1e9 之内 二次模型 1e6 之内
Objective range	5e-01,	5e-01]	
Bounds range	8e-03,	8e+03]	
RHS range	1e-03,	5e+03]	
QRHS range	1e+00,	1e+00]	

(3) 详细信息请阅读参考手册第七章“GUIDELINES FOR NUMERICAL ISSUES”

(4) 如果数值问题造成日志结尾出现了警告信息，用户可以尝试设置参数

Presolve=0, NumericFocus=2

十六、 Gurobi 默认变量为非负变量

Gurobi 新定义变量均默认为非负变量，下界是 0，不是负无穷大。这是为了方便实际应用的设置（实际应用中大部分场景的变量都是非负变量）。如果变量可以取负值，在定义变量时，需要显性设置 `lb=-GRB.INFINITY`。

特别在以下情况时，用户需要注意设置中间变量（辅助变量）的下界可以取负值

- (1) 绝对值 $y=\text{abs}(x)$ 中的 x 变量
- (2) Log 函数、三角函数等可能会取负值的结果变量
- (3) 其他任何可以导致负值的中间变量