

Gurobi 11.0 升级全攻略

(版本 20231129)

一、	Gurobi 11.0 核心变化	2
二、	Gurobi 11.0 可以求解的问题类型	4
三、	Gurobi 11.0 安装路线图	5
四、	Gurobi 11.0 许可文件更新方法	6
	(4.1) 学术许可更新方法	6
	(4.2) 商业许可更新方法	6
	(4.3) 许可文件必备知识	7
	(4.4) 如何验证 Gurobi 激活成功	8
五、	Gurobi 11.0 Python 模块库安装方法	9
六、	Gurobi 11.0 机器硬件环境配置建议	11
七、	Gurobi 11.0 功能变化细节	12
	(7.1) 支持的操作系统	12
	(7.2) 对于其他软件的支持版本	12
	(7.3) 依赖库变化	13
	(7.4) 影响所有 API 的新功能	13
	(7.5) gurobipy 的变化	15
	(7.6) Java 包的更改	17
	(7.7) MATLAB 的更改	17
	(7.8) 运算服务器、集群管理器和即时云	18

Gurobi 11.0 升级全攻略

Gurobi 11.0 已经发布。我们希望这个文档可以回答关于 Gurobi 11.0 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。Gurobi 中国网站 www.gurobi.cn 邮箱：help@gurobi.cn

一、Gurobi 11.0 核心变化

（一）性能提升

所有问题类型求解速度进一步提升，混合整数二次模型（凸和非凸）、非线性模型速度提升更明显。

（二）重要功能更新

（1）进一步支持全局精确混合整数非线性优化。在分段线性逼近的方式上，引入全局精确非线性算法。设置 `FuncNonLinear = 1` 参数可以切换求解算法，让求解混合整数非线性模型速度更快、精度更高。

（2）对于二次非凸问题（双线性、二次等式、分母带变量等），`NonConvex = 2` 不再需要显性设置，Gurobi 会自动启用双线性算法。

（3）支持动态分布式调优功能。对于之前分布式调优的重要补充，允许调优工作在多个机器上临时开展并在完成后回收他用，提高调优效率和设备利用。

（4）支持非确定性并发 LP 算法。用于将 LP 求解性能进一步提升，用户可以设置非确定性的并发原始单纯形和对偶单纯形算法。

（5）支持非基解 LP 优化设置。通过设置 `SolutionTarget` 参数，用户可以选择是否获得 LP 的基最优解，还是更快地获得非基解的 LP 最优解。

（三）系统架构的提升

（1）集群管理器集成 SAML: 除了已有的对 LDAP 支持之外，用户可以通过集成基于 SAML 的身份管理系统来享受集中式单点登录功能。

（2）支持 Cosmos DB: 在部署集群架构时，除了对 MongoDB 和 Amazon DocumentDB 数据库支持之外，现在支持 Microsoft Azure Cosmos DB。

（3）使用性增强: 集群管理器增加对用户名称控制、日期格式改进、HTTP 安全头文件的支持。

(4) 时间和内存限制管理：运算服务器增加了新的用于时间和内存管理的优化参数。

(四) 其他接口变化

(1) Java 包支持由 Maven Central 分发。

(2) Gurobipy 的 Callback 回调函数可以接受矩阵变量和矩阵约束。允许 callback 可以是任何可调用的对象（类、函数等）。

(3) Gurobipy 的矩阵变量和矩阵约束支持基于 Numpy 规则的合并操作 (hstack, vstack, concatenate)。

(4) Gurobipy 自带类型提示辅助功能，用户不用再单独安装 gurobipy-stubs。

详细变化说明请参考后续章节。

二、 Gurobi 11.0 可以求解的问题类型

Gurobi 是全球综合能力领先的数学规划求解器。目前最新版本 11.0，可以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）

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 功能见安装目录/docs/refman.pdf 参考手册。

三、 Gurobi 11.0 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）的群文件中获得最新版本安装包，Windows/Mac 包直接双击安装，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，放置到合适位置，验证激活成功。见本文档后续章节
- 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 11.0 Python 模块库安装方法
 - Matlab: <https://support.gurobi.com/hc/en-us/articles/4533938303505#section:MATLAB>
 - ◆ Matlab 范例在安装目录/`examples/matlab` 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->
 - Java: <https://support.gurobi.com/hc/en-us/articles/360013193472-How-do-I-use-Gurobi-in-an-Eclipse-Java-project->
 - 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
- 随机手册和范例：Gurobi 安装目录/`examples/` 下的范例涵盖了 Gurobi 常用的知识点。参考手册在安装目录/`docs` 下的 `refman.pdf`，这是经常需要使用的手册。
- 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 11.0 许可文件更新方法

(4.1) 学术许可更新方法

【对于免 IP 验证学术许可，已有版本 10.0 许可且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 11，`VERSION=11`，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 11，`VERSION=11`，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 11.0，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；

(4.2) 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。

(4.3) 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: `opt/gurobi`

Mac: `/Library/gurobi`

如果没有这些目录，请创建一个。这是 Gurobi 默认搜索位置。

如果保存到其他位置，需要创建 `GRB_LICENSE_FILE` 系统环境变量（不是用户环境变量），指向保存的目录和文件名（例如 `GRB_LICENSE_FILE = C:\myfolder\gurobi.lic`）。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

(4.4) 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，Windows 下输入 `gurobi` 回车，Linux 下输入 `gurobi.sh` 回车，Mac 下输入 `gurobi.sh` 回车，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。

```
Gurobi Interactive Shell, Version 11.0.0
```

```
Copyright (c) 2023, Gurobi Optimization, LLC
```

```
Type "help()" for help
```

```
gurobi>
```

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。

五、 Gurobi 11.0 Python 模块库安装方法

- **Gurobi 11.0 支持的 Python 版本包括 3.8, 3.9, 3.10, 3.11, 3.12**
- **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。**

以下总结了在 Python 中安装 Gurobi 11.0 的二种在线方法和一种离线方法。如果需要更多指示，可以参考随机手册。

（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
```

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，帮助用户更轻松地管理支撑环境。

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。

六、 Gurobi 11.0 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。

七、 Gurobi 11.0 功能变化细节

(7.1) 支持的操作系统

Platform (port)	Operating System	Compiler	Notes
Windows 64-bit (win64)	Windows 10, 11, Windows Server 2016, 2019, 2022	Visual Studio 2017-2022	Use <code>gurobi_c++md2017.lib</code> (e.g.) for C++
Linux x86-64 64-bit (linux64)	Red Hat Enterprise Linux 8, 9	GCC >= 8.5	
	SUSE Enterprise Linux 15		
	Ubuntu 20.04, 22.04		
	Amazon Linux 2		
macOS 64-bit universal2 (<code>macos_universal2</code>)	12 (Monterey), 13 (Ventura), 14 (Sonoma)	Xcode 13/14	
Linux arm64 64-bit (<code>armlinux64</code>)	Red Hat Enterprise Linux 8, 9	GCC >= 8.5	
	SUSE Enterprise Linux 15		
	Ubuntu 20.04, 22.04		
	Amazon Linux 2		
AIX 64-bit (power64)	AIX 7.2, 7.3	XL C/C++ 9	Note that Gurobi 11 will be the last supported version for AIX. Due to limited Python support on AIX, this port does not include the Interactive Shell or the Python libraries.

(7.2) 对于其他软件的支持版本

Language	Version
Python	3.8, 3.9, 3.10, 3.11, 3.12
MATLAB	R2019a-R2023b
R	4.3
JDK	8, 11, 17
.NET	6.0

(7.3) 依赖库变化

Linux GNU C 库 (glibc) 依赖项

Linux x86_64: Gurobi 优化器现在需要的最低 glibc 版本为 2.17。对于版本 10.0, 所需的最低 glibc 版本为 2.3.4。

上面的支持平台列表考虑了这些更改。

(7.4) 影响所有 API 的新功能

- 混合整数非线性模型(MINLP)

Gurobi 现在采用空间分支法和外逼近法求解混合整数非线性问题，而不是静态的分段线性逼近方法。更多信息可以参见参考手册说明。

- 非凸参数的默认行为

NonConvex 参数的默认行为已更改。以前如果您要求解具有非凸二次目标或约束的模型，则必须显式将此参数设置为值 2。现在它将默认解决此类模型。要恢复到以前的行为，请将参数设置为 1。

- 在环境之间复制模型

现在可以使用以下用法将模型从一个环境复制到另一个环境：

C: GRBcopymodeltoenv

C++: GRBModel (const GRBModel& model, const GRBEnv& targetenv)

Java: GRBModel (GRBModel model, GRBEnv targetenv)

.NET: GRBModel (GRBModel model, GRBEnv targetenv)

Python: model.copy(env)

这个修改主要用于将同一个模型的不同版本放在多个线程中并发运行时便于处理。

Gurobi 不支持单一环境内多模型并发处理。

对于 Compute Server 用户，请注意，您可以将模型从客户端复制到 Compute Server 环境，但无法在两个 Compute Server 环境之间复制模型。

- 更好地控制并发 LP 优化

Method 参数的选项 5 已被弃用，取而代之的是新的 ConcurrentMethod 参数。后者

允许您控制哪些方法与确定性 (**Method=4**) 或非确定性 (**Method=3**) 并发优化器同时运行。现在可以通过选择 **Method=4** 和 **ConcurrentMethod=3** 来复制已弃用的设置 **Method=5**。举另一个例子, 可以使用 **Method=3** 和 **ConcurrentMethod=3** 来选择原始和对偶单纯形算法的非确定性并发执行。

- 恢复优化时更改 **Threads** 参数

如果优化被中断 (例如由于时间限制), 用户现在可以修改 **Threads** 参数, 以便恢复求解过程时可以使用新的线程数来优化。

- 调优

分布式调优可以动态地使用远程工作机。使用参数 **TuneDynamicJobs** 可以动态控制并行远程工作机集合的大小。这些工作机使用有限时间后会被回收, 以便他们可以从事其他远程工作。

从 **Gurobi 11** 开始, 我们在调优结果中包含用于求解模型的第一组参数 (基础参数设置)。该结果作为第一组调优结果。因此要获得最佳调优参数集, 需要在 **TuneResults** 中获得至少 2 个参数集, 并检索索引为 1 的参数集。

调优算法现在支持日志回调。您可以向环境添加日志回调并检索来自调优器的所有日志消息。

- 分段线性逼近

参数 **FuncPieces** (函数约束分段线性逼近的主要优化参数) 的默认行为已更改。**11.0** 版本的新默认设置是使用相对误差方法进行近似, 而 **10.0** 或更早版本主要用于设置函数约束的分段数量。

- 新参数

以下参数是 **Gurobi 11.0** 中的新增参数:

- ◆ **SolutionTarget** 指定 LP 的求解目标。
- ◆ **FuncNonlinear** 控制广义函数约束是否采用非线性函数处理还是通过分段线性逼近。
- ◆ **MixingCut** 用于控制混合切平面的生成。
- ◆ **TuneDynamicJobs** 设置动态工作机启用分布式调优。
- ◆ **TuneUseFilename** 设置调优时用文件名称做为模型名称 (仅限命令行)。

(7.5) gurobipy 的变化

- 安装变化
 - ◆ gurobipy 及其 setup.py 文件不再包含在 Gurobi 安装程序里。用户需要使用 pip 或 conda 将 gurobipy 安装到 Python 环境中。
 - ◆ gurobi conda 软件包已更新, pip 现在可以识别 gurobipy 已经通过 conda 安装到环境中。因此在 conda 环境中使用 pip 安装依赖于 gurobipy 的软件包 (例如 gurobi-machinelearning 或 gurobipy-pandas) 不再导致 gurobipy 的重复安装。
 - ◆ gurobipy 的类型提示和补全功能现在包含在 pip wheel 文件和 conda 包中。对于 Gurobi 版本 11, 用户不需要也不应该再安装 gurobipy-stubs 软件包, 因为它将覆盖 gurobipy 里的类型注释。gurobipy-stubs 软件包将逐步淘汰。
- 新的类、方法和属性
 - ◆ SOS 对象有一个新的 index 属性, 返回模型中 SOS 约束的索引。
 - ◆ MConstr 和 MQConstr 对象现在实现了 fromlist 类方法, 该方法与 MVar 类上的相应方法具有相同的行为。
 - ◆ 新方法 hstack、vstack 和 concatenate 允许矩阵对象以与相应 numpy 函数相同的方式连接在一起。
 - ◆ 添加了一个新的 MGenConstr 类来保存一组广义约束。对于版本 11, 只有 addGenConstrIndicator 方法能够返回此类型的对象。
 - ◆ Env 对象现在有一个新增方法 getParam, 它返回环境中给定参数的当前值。
- 对现有方法的增强
 - ◆ Model.optimize 和 Model.computeIIS 方法现在接受任何 Python 可调用类型作为回调函数。TSP 示例代码已更新, 以演示可调用类的使用。
 - ◆ 矩阵 API 现在与回调方法完全集成: Model.cbLazy、Model.cbCut 和 Model.cbSetSolution 现在接受矩阵对象。
 - ◆ 当使用 Model.addMConstr 或 Model.addConstr 与矩阵对象 API 构造一组线性约束时, 用户定义的约束名称可以作为列表或 ndarray 传递。
 - ◆ Model.addGenConstrIndicator 和指示约束的重载符号 >> 现在可以处理矩阵线

性表达式对象。这允许以数组化方式构建指示约束（包括广播行为）。

- ◆ 用户现在可以在启动 Python 程序之前通过将环境变量 `GUROBIPY_ALLOW_DEFAULTENV` 设置为 0 来显性禁用默认环境。通过此设置，必须使用用户代码中的 `Env` 对象显性管理环境。如果使用此设置，则在没有 `Env` 对象的情况下调用 `Model()` 以及调用 `setParam` 等全局函数将引发 `GurobiError`。

- 可能需要更新您的代码的更改

- ◆ 现在，如果传递给 `Model.getAttr` 和 `Model.setAttr` 的建模对象不属于模型，则会引发异常。以前的做法是为模型上具有与建模对象相同索引的变量返回或设置属性值，这通常不是想要的操作。
- ◆ 使用大小为 1 的矩阵对象调用 `Model.addGenConstrIndicator`，之前返回 `GenConstr` 对象，现在将返回大小为 1 的 `MGenConstr` 对象。约束名称也可能根据结果的形状而改变。要恢复到以前的行为，请在传递给此函数的矩阵对象上调用 `.item()`。
- ◆ `tupledict` 现在遵循 Python 3 字典语义：它的 `keys`、`values` 和 `items` 方法现在返回可迭代视图而不是列表。重要的是，`keys` 不再返回 `tuplelist`。如果您仍然需要，可以调用 `gp.tuplelist(td.keys())`。
- ◆ 如果用户指定的参数不存在，`Env.setParam` 现在会引发 `GurobiError`。在以前的版本中，此类调用会打印一条消息，然后失败但没有异常。
- ◆ Gurobi 日志消息现在转到“gurobipy”记录器而不是“gurobipy.gurobipy”。如果您使用 Python 的日志记录库来收集 Gurobi 日志消息，则应将处理程序附加到根记录器或 `logging.getLogger("gurobipy")`。
- ◆ 现在调用不包含元素的 `MVar`、`MLinearExpr` 或 `MQuadExpr` 的 `sum()` 方法将返回一个适当形状的全零数组，而不是引发异常。这与 `numpy` 数组的行为相匹配。
- ◆ 将太多键（超过所包含元组的长度）传递给任何模式匹配方法（`tuplelist.select`、`tupledict.select`、`tupledict.sum` 和 `tupledict.prod`）现在会引发 `IndexError`。此更改几乎只影响以前版本中返回空集的情况。例外情况是其他索引是通配符。即对于包含 2 元组的元组列表：`tl.select("*", 2, "*")` 现在引发异常，而在以前的版本中它给出与 `tl.select("*", 2)` 相同的（非

空) 结果。

- ◆ 模块库 `gurobi` 别名已从所有平台上的 `conda` 包中删除。`import gurobi` 的任何用法都必须替换为 `import gurobipy`。

- 已弃用的功能

如果您要从以前版本的 `Gurobi` 升级，我们建议您首先使用 `Gurobi 10` 运行代码，并启用警告以捕获 `gurobipy` 中的弃用内容。修复这些已弃用的用法将有助于保持 `Gurobi 11` 及更高版本的兼容性。可以通过使用 `-X dev` 或 `-W` 默认标志运行代码来启用警告。有关更多详细信息，请参阅 `Python` 开发模式或警告包文档。

在 `Gurobi 11` 中，以下用法已被弃用，并将在未来版本中删除：

- ◆ 将 `tupledict` 传递给 `quicksum` 函数将会失效，因为结果与将 `Python` 原生字典类型传递给 `quicksum` 不一致。应使用 `tupledict` 的 `sum` 方法来计算 `tupledict` 值的总和。
- ◆ `tupledict` 的 `iterkeys()`、`itervalues()` 和 `ititems()` 方法将会失效。这些方法是 `Python 2` 的遗留语法。相应的 `Python 3` 方法 `keys()`、`values()` 和 `items()` 现在返回可迭代视图，因此是推荐方法。
- ◆ `gurobipy.system()` 函数已弃用，应改用 `Python` 标准库中的 `os.system()`。
- ◆ `gurobipy.models()` 函数已弃用，不应再使用

(7.6) Java 包的更改

- 更新

我们对包结构进行了重要更改，以允许发布到 `Maven Central`。请注意以下更新：

- ◆ 包名称更改：Java 包名称已从 `gurobi` 更新为 `com.gurobi.gurobi`。如果您要集成 Java 包，请确保相应地更新您的 `import` 语句。

(7.7) MATLAB 的更改

在 `macOS` 平台上，`Gurobi mex` 接口现在也可用于原生 `Apple` 芯片 `MATLAB` (`mexmac64`)。

(7.8) 运算服务器、集群管理器和即时云

请参考运算服务器、集群管理器和即时云的详细发布说明