

Lab Platform: SUMO

Official web: [Eclipse SUMO - Simulation of Urban MObility](#)

SUMO User Documentation

About

"Simulation of **Urban MObility**", or "SUMO" for short, is an open source, microscopic, multi-modal traffic simulation. It allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. The simulation allows to address a large set of traffic management topics. It is purely microscopic: each vehicle is modelled explicitly, has an own route, and moves individually through the network. Simulations are deterministic by default but there are various options for [introducing randomness](#).

If you download the SUMO package, you will note that it contains further applications besides SUMO. These applications are used to import/prepare road networks and demand data for being used in SUMO, see [Included Applications](#) for a more verbose list.

《城市机动能力模拟》，简称“SUMO”，是一种开源的、微观的、多模态的交通模拟。它允许模拟由单个车辆组成的给定交通需求如何通过给定的道路网络。模拟允许解决大量的交通管理主题。它纯粹是微观的：每辆车都是明确建模的，有自己的路线，并在网络中单独移动。默认情况下，模拟是确定性的，但有多种选择可以[引入随机性](#)。

如果您下载 SUMO 包，您会注意到它包含除 SUMO 之外的其他应用程序。这些应用程序用于导入/准备道路网络和需求数据，以便在 SUMO 中使用。

1. Introduction

Features

- Includes all applications needed to prepare and perform a traffic simulation (network and routes import, DUA, simulation)
- Simulation
 - Space-continuous and time-discrete vehicle movement
 - Different vehicle types
 - Multi-lane streets with lane changing
 - Different right-of-way rules, traffic lights
 - A fast OpenGL graphical user interface
 - Manages networks with several 10.000 edges (streets)
 - Fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine)
 - Interoperability with other application at run-time
 - Network-wide, edge-based, vehicle-based, and detector-based outputs
 - Supports person-based inter-modal trips
- Network Import
 - Imports VISUM, Vissim, Shapefiles, OSM, RoboCup, MATsim, OpenDRIVE, and XML-Descriptions
 - Missing values are determined via heuristics
- Routing
 - Microscopic routes - each vehicle has an own one
 - Different Dynamic User Assignment algorithms
- High portability
 - Only standard C++ and portable libraries are used
 - Packages for Windows main Linux distributions exist
- High interoperability through usage of XML-data only
- Open source ([EPL 2.0](#) [🔗](#))

特征

- 包括准备和执行交通模拟所需的所有应用程序（网络和路线导入、DUA、模拟）
- 模拟
 - 空间连续和时间离散的车辆运动
 - 不同的车辆类型
 - 可变换车道的多车道街道
 - 不同的通行权规则，红绿灯
 - 一个快速的 OpenGL 图形用户界面
 - 管理具有数个 10.000 条边（街道）的网络
 - 执行速度快（在 1GHz 机器上高达 100.000 次车辆更新/秒）
 - 在运行时与其他应用程序的互操作性
 - 全网络、基于边缘、基于车辆和基于检测器的输出
 - 支持以人为本的多式联运
- 网络导入
 - 导入 VISUM、Vissim、Shapefile、OSM、RoboCup、MATsim、OpenDRIVE 和 XML 描述
 - 缺失值是通过启发式确定的
- 路由
 - 微观路线——每辆车都有自己的路线
 - 不同的动态用户分配算法
- 高便携性
 - 仅使用标准 C++ 和可移植库
 - 存在适用于 Windows 主要 Linux 发行版的软件包
- 通过仅使用 XML 数据实现高互操作性
- 开源 ([EPL 2.0](#))

Application Name	Short Description
sumo	The microscopic simulation with no visualization; command line application
sumo-gui	The microscopic simulation with a graphical user interface
netconvert	Network importer and generator; reads road networks from different formats and converts them into the SUMO-format
netedit	A graphical network editor.
netgenerate	Generates abstract networks for the SUMO-simulation
duarouter	Computes the fastest routes through the network, importing different types of demand description. Performs the DUA
jtrrouter	Computes routes using junction turning percentages
dfrouter	Computes routes from induction loop measurements
marouter	Performs macroscopic assignment
od2trips	Decomposes O/D-matrices into single vehicle trips
polyconvert	Imports points of interest and polygons from different formats and translates them into a description that may be visualized by sumo-gui
activitygen	Generates a demand based on mobility wishes of a modeled population
emissionsMap	Generates an emission map
emissionsDrivingCycle	Calculates emission values based on a given driving cycle
Additional Tools	There are some tasks for which writing a large application is not necessary. Several solutions for different problems may be covered by these tools.

Application Name	应用名称	简短的介绍
sumo	相扑	没有可视化的微观模拟；命令行应用程序
sumo-gui	sumo-gui	具有图形用户界面的微观模拟
netconvert	网络转换	网络导入器和生成器；从不同格式读取道路网络并将其转换为 SUMO 格式
netedit	网络编辑	图形网络编辑器。
netgenerate	网络生成	为 SUMO 模拟生成抽象网络
duarouter	双路由器	计算通过网络的最快路线，导入不同类型的需求描述。执行 DUA
jtrrouter	路由器	使用路口转弯百分比计算路线
dfrouter	路由器	根据感应回路测量计算路线
marouter	路由器	进行宏观赋值
od2trips	od2trips	将 O/D 矩阵分解为单次车辆行程
polyconvert	多边形转换器	从不同格式导入兴趣点和多边形，并将它们转换为可由sumo-gui可视化的描述
	活性源	根据建模人群的出行意愿生成需求
activitygen	排放地图	生成发射贴图
emissionsMap	排放驾驶周期	根据给定的驾驶循环计算排放值
emissionsDrivingCycle	附加工具	有些任务不需要编写大型应用程序。这些工具可能涵盖针对不同问题的多种解决方案。
Additional Tools	几方在他们的工作中扩展了 SUMO 包并提交了他们的代码。这些贡献通常未经测试，可能已过时。 在此处 查找所有贡献的列表。	

某些应用程序可能需要关注一下：

- Polyconvert：从不同格式导入兴趣点和多边形
- Activitygen：根据建模人群的出行医院生成需求

2. Basic

2-1. Notation in this document

引用的数据类型

- **<BOOL>**：一个布尔值，使用“t”或“true”和“f”或“false”进行编码
- **<INT>**：一个整数值，可以是负数
- **<UINT>**：无符号整数值，必须 ≥ 0
- **<FLOAT>**：浮点数
- **<TIME>**：时间，以秒为单位；允许使用分数，例如“12.1”
- **<STRING>**：任何字符串，但仅使用 ASCII 字符
- **<ID>**：一个字符串，不能包含以下字符：'#'

警告
不允许使用的字符列表不完整

- **<FILE>**或**<FILENAME>**：文件的（相对或绝对）路径；另请参阅[#Referenced 文件类型](#)
- **<PATH>**：（相对或绝对）路径（通常指向文件夹）
- **<COLOR>**：由“”分隔的四组浮点数（**<FLOAT>**、**<FLOAT>**、**<FLOAT>**、**<FLOAT>**），描述了红色、绿色、蓝色和 alpha 分量，范围从 0.0 到 1.0（alpha 组件是可选的），或者列表可以包含 0-255 范围内的整数。请注意，分隔符必须是逗号，并且不允许有空格。颜色也可以使用带有[HTML 颜色代码](#)的单个字符串或基本颜色之一（“红色”、“绿色”、“蓝色”、“黄色”、“青色”、“品红色”、“黑色”、“白色”、“灰色”）。“random”的值将分配随机颜色。
- **<2D-POSITION>**：两个由“”分隔的浮点数（**<FLOAT>**、**<FLOAT>**），分别描述 x 和 y 偏移量。z 隐含地为 0
- **<3D-POSITION>**：由','分隔的三个浮点数（**<FLOAT>**、**<FLOAT>**、**<FLOAT>**），分别描述了 x-、y- 和 z-offset
- **<POSITION-VECTOR>**：2D 或 3D 位置列表，由 ' ' 分隔。即（**<2D-POSITION>** **<2D-POSITION>**，**<3D-POSITION>**）
- **<2D-BOUNDING_BOX>**：由“”分隔的四个浮点数（**<FLOAT>**、**<FLOAT>**、**<FLOAT>**、**<FLOAT>**），分别描述了 x-minimum、y-minimum、x-maximum 和 y-maximum
- **<PROJ_DEFINITION>**：一个字符串，包含 proj.4 使用的投影定义；请注意，您必须将定义字符串放在引号中

参考文件类型

- **<NETWORK_FILE>**：由netgenerate或 netconvert构建的 [SUMO 网络文件](#)
- **<ROUTES_FILE>**：由duarouter或 jtrrouter或手动 构建的[SUMO 路由文件](#)
- **<TYPE_FILE>**：SUMO 边缘类型文件，手工构建或下载
- **<OSM_FILE>**：从OpenStreetMap 导出的[OpenStreetMap 文件](#)

2-2. Computer-Skills

苹果系统

首先，您需要确定当前使用的是哪个 shell（**bash**或**zsh**）。在终端中，键入 `ps -p $$`（这应该让您在 CMD 下看到您使用的是哪个 shell）。macOS Catalina 中的默认 shell 是 **zsh**。

对于 ZSH

在终端中，执行以下步骤：

```
open ~/.zshrc
```

这将在 TextEdit 中打开 `.zshrc` 文件。将以下行添加到该文档：

```
export SUMO_HOME="/your/path/to/sumo"
```

确保替换 `/your/path/to/sumo` 为您的相扑**实际路径**。

将该文档保存在 TextEdit 中。现在您需要通过输入以下内容来应用更改：

```
source ~/.zshrc
```

在终端。你完成了！要测试环境变量是否设置正确，您可以将其打印到终端：

```
echo $SUMO_HOME
```

你应该能够在那里获得最近添加的路径。

2-3. Installing SUMO

[Installing - SUMO Documentation](#)

2-4. Using the Command Line Applications

Configuration File

```
1 <!-- test.sumocfg -->
2 <configuration>
3   <input>
4     <net-file value="test.net.xml"/>
5     <route-files value="test.rou.xml"/>
6     <additional-files value="test.add.xml"/>
7   </input>
8 </configuration>
9
10 <!-- sumo -c test.sumocfg -->
```

[Using the Command Line Applications - SUMO Documentation](#)

- 命令行配置选项

2-4. Tutorials

- Beginner tutorials

- [Hello World](#) - 使用 [netedit](#) 创建一个简单的网络和需求场景，并使用 [sumo-gui](#) 将其可视化
- [OSMWebWizard](#) - 使用 [osmWebWizard.py](#) 只需点击几下即可设置场景；从 [OpenStreetMap](#) 获取网络
- [快速入门](#)- 更复杂的 [netedit](#) 教程；相扑的第一步
- [Driving in Circles](#) - 与 [netedit](#) 一起工作；定义流程；让车辆使用重路由绕圈行驶
- [SUMOlympics](#) - 在 [netedit](#) 中创建特殊车道和简单的交通信号灯，更多关于流量和车辆类型，使用车辆
- [Autobahn](#) - 建造高速公路，创建混合高速公路流，可视化车速，保存视图设置
- [Manhattan](#) - 构建 [曼哈顿移动模型](#) [↗](#)
- [公共交通](#)- 从头开始构建公共交通场景
- [TaxiService](#) - 从头开始构建出租车服务

• SUMO User Conference Tutorial

SUMO 用户大会是由德国航空航天中心 (DLR) 在柏林组织的年度活动。自 2015 年以来，每次会议都以 SUMO 教程开始。您可以在下面找到教程材料（幻灯片和输入文件）。自 2019 年以来，教程也被录制成视频。

文件

- [SUMO 2015](#) [↗](#): 使用 xml 补丁进行网络编辑，人员，
- [SUMO 2016](#) [↗](#): 网络编辑、meso、容器、[2016 年新功能（幻灯片）](#) [↗](#)
- [SUMO 2017](#) [↗](#): 网络编辑、randomTrips、校准器（仅限 xml）、公共交通（已过时）
- [SUMO 2018](#) [↗](#): 修复多式联运路口、[netedit](#) 中的校准器、路口模型参数、编辑形状
- [SUMO 2019](#) [↗](#): 网络编辑、可视化交通数据、来自 OSM 的公共交通、停车、网络生成
- [SUMO 2020](#) [↗](#): 转向车道、routeSampler.py、在 [netedit](#)、出租车/DRT 中定义计数数据
- [SUMO 2021](#) [↗](#): 红绿灯布局，间接左转，TAZ，OD-traffic，GTFS
- [SUMO 2022](#) [↗](#): 网络编辑、流量、对向行驶、人行横道、停车位搜索

• Advanced Tutorials

- [你好 SUMO](#) - 最简单的网络和一辆“手工”设置的汽车
- [ScenarioGuide](#) - 构建模拟场景所需步骤的高级概述
- [HighwayDetector](#) - 如何根据感应回路数据创建高速公路场景
- [FundamentalDiagram](#) - 如何使用 SUMO 计算基本图
- [PT from OpenStreetMap](#) - [展示如何完全从 OpenStreetMap](#) [↗](#) 创建可运行的公共交通场景 [↗](#)
- [导入 GTFS](#) - 使用通用交通提要规范从公共资源导入公共交通时刻表
- [港口](#)- 创建一个场景，模拟来自 [OpenStreetMap](#) [↗](#) 的任意港口的港口物流 [↗](#)

• Traci Tutorials

这些教程使用 [Python-TraCI](#) 库将 python 脚本与正在运行的 [相扑](#) 模拟相连接。

- [TraCI4Traffic Lights](#) - 如何通过 TraCI 将外部应用程序连接到 SUMO 以进行交通灯控制的示例
- [TraCIPedCrossing](#) - 通过 TraCI 构建行人驱动的交通灯的示例
- [基于边行程时间模拟动态车辆绕行](#) [（外部）](#) [↗](#)

• Other

移动模拟课程

[Ednardo Ferreira](#) 提供的葡萄牙语Udemy 教程。 [↗](#)

ITSC 2015

- 在http://sumo.dlr.de/daily/ITSC2015_tutorial.zip [↗](#)下载教程文件 [↗](#)

进出口

- [跟踪文件生成](#) - 展示如何获取车辆跟踪文件，可用于模拟车辆通信

校准/验证

- [Calibration/San Pablo Dam](#) - 使用 NEARCTIS 暑期学校期间使用的观察点车辆通过时间校准跟车参数

杂项

- [铁路教程](#) [↗](#)

2-5. XML Validation

All SUMO applications support XML validation for their inputs. To enable this, the following options can be used:

Option	Description
<code>-X <STRING></code> <code>--xml-validation <STRING></code>	Set schema validation scheme of XML inputs ("never", "local", "auto" or "always"); <i>default: local</i>
<code>--xml-validation.net <STRING></code>	Set schema validation scheme of SUMO network inputs ("never", "local", "auto" or "always"); <i>default: never</i>
<code>--xml-validation.routes <STRING></code>	Set schema validation scheme of SUMO route inputs ("never", "local", "auto" or "always"); <i>default: local</i>

Validation is performed by activating [XML schema processing](#) [↗](#) in the XML parser. This catches many common input errors such as spelling mistakes and attributes which should have been placed within another element.

3. TraCI

- [TraCI - SUMO Documentation](#)
- [Interfacing TraCI from Python - SUMO Documentation](#)

TraCI

TraCI简介

TraCI是“交通控制接口”的简称。允许访问正在运行的道路交通模拟，它允许检索模拟对象的值并“在线”操纵它们的行为。

使用TraCI

相扑启动

TraCI 使用基于 TCP 的客户端/服务器架构来提供对 [sumo](#) 的访问。因此，[sumo](#) 充当服务器，使用额外的命令行选项启动：`--remote-port <INT>` 其中 `<INT>` 是 [sumo](#) 将监听传入连接的端口。

当使用 `--remote-port <INT>` 选项启动时，[sumo](#) 仅准备模拟并等待所有外部应用程序连接并接管控制。请注意，当 [sumo](#) 作为 TraCI 服务器运行时，`--end <TIME>` 选项将被忽略，[sumo](#) 一直运行到客户端要求模拟结束。

当使用 [sumo-gui](#) 作为服务器时，必须在处理 TraCI 命令之前使用 [播放 按钮](#) 或设置选项 `--start` 来启动模拟。

多个客户端

可以连接的客户端数量可以作为附加选项 `--num-clients <INT>` 给出，其中 1 是默认值。[请注意，在多客户端场景中，您必须使用 SetOrder 命令明确指定客户端的执行顺序。](#)

每个客户端必须指定一个唯一的（但在其他方面是任意的）整数值，并且客户端命令将在每个模拟步骤中按照从最低值到最高值的顺序进行处理。

客户端在每个模拟步骤后自动同步。这意味着，在所有客户端调用“simulationStep”命令之前，模拟不会进入下一步。此外，simulationStep 命令仅在模拟完成后才将控制权返回给客户端。

警告

只有在所有客户端都连接后，模拟才会开始。

协议规范

请参阅[TraCI 协议规范](#)（包括基本流程、消息、数据类型）。

关闭

使用 TraCI 时，忽略sumo的--end选项。[而是通过发出关闭命令](#)来关闭模拟。要检测是否所有路径文件都已用完，所有车辆是否都离开了模拟，可以检查命令[getMinExpectedNumber](#)是否 返回0。一旦所有客户端发送 关闭命令，模拟就会结束。

也可以使用 [load](#)命令用新的参数列表重新加载模拟。

TraCI 命令

- [控制相关命令](#)：执行模拟步骤，关闭连接，重新加载模拟。
- [通用参数](#)

对于以下 API，ID 等于 [sumo](#)的输入文件中定义的 ID。在这里，您可以找到它们的一般结构。

价值检索

- 交通对象
 - [Vehicle Value Retrieval](#)检索有关车辆的信息
 - [Person Value Retrieval](#)检索关于人的信息
 - [Vehicle Type Value Retrieval](#) 检索有关车辆类型的信息
 - [Route Value Retrieval](#) 检索有关路由的信息
- 探测器和输出
 - [Induction Loop Value Retrieval](#) 检索有关感应回路的信息
 - [车道区域检测器值检索](#) 检索有关车道区域检测器的信息
 - [Multi-Entry-Exit Detectors Value Retrieval](#) 检索有关多入口/多出口检测器的信息
 - [校准器值检索](#) 检索有关校准器的信息
 - [RouteProbe](#) 检索有关 RouteProbe 的信息
- 网络
 - [Junction Value Retrieval](#)检索有关路口的信息
 - [Edge Value Retrieval](#) 检索有关边的信息
 - [Lane Value Retrieval](#) 检索有关车道的信息
- 基础设施
 - [Traffic Lights Value Retrieval](#) 检索有关交通灯的信息
 - [BusStop Value Retrieval](#) 检索有关 BusStops 的信息
 - [Charging Station Value Retrieval](#) 检索有关充电站的信息
 - [Parking Area Value Retrieval](#) 检索有关停车场的信息
 - [Overhead Wire Value Retrieval](#) 检索有关架空电线的信息
 - [Rerouter](#) 检索有关 rerouter 的信息
- 杂项
 - [Simulation Value Retrieval](#)检索有关模拟的信息
 - [GUI 值检索](#) 检索有关模拟可视化的信息
 - [PoI 值检索](#) 检索有关兴趣点的信息
 - [Polygon Value Retrieval](#)检索有关多边形的信息

状态改变

- 交通对象
 - [Change Vehicle State](#) 改变车辆的状态
 - [Change Person State](#) 改变一个人的状态
 - [更改车辆类型状态](#)更改车辆类型的状态
 - [更改路线状态](#)更改路线的状态
- 探测器和输出
 - [更改校准器状态](#)更改校准器状态
 - [Change Inductionloop State](#)改变感应回路状态
 - [更改车道区域检测器状态](#)更改车道区域检测器状态
- 网络
 - [Change Edge State](#)改变边的状态
 - [Change Lane State](#)改变车道的状态
- 基础设施
 - [Change Traffic Lights State](#)改变交通灯的状态
- 杂项
 - [Change Simulation State](#)改变模拟
 - [更改 GUI 状态](#)更改模拟可视化
 - [更改 PoI 状态](#)更改兴趣点的状态（或添加/删除一个）
 - [更改多边形状态](#) 更改多边形的状态（或添加/删除一个）

订阅

- [TraCI/对象变量订阅](#)
- [TraCI/对象上下文订阅](#)

使用 SUMO 作为库

通常，TraCI 用于耦合多个进程：一个 SUMO 服务器进程和一个或多个 TraCI 客户端进程。或者，[Libsumo](#)可用于将 SUMO 作为库嵌入到客户端进程中。这允许使用与客户端库中相同的方法签名，但避免了套接字通信的开销。[Libsumo 支持使用SWIG](#) 生成客户端库，因此可以与大量编程语言一起使用。下载 sumo-build 时包含 C++、Java 和 Python 绑定。

使用示例

- 有一个[关于将 TraCI 用于自适应交通信号灯](#)（使用 Python）的教程。
- 教程/[CityMobil](#)教程使用 TraCI 为车辆分配新路线（使用 Python）。
- Tutorials /[TraCIPedCrossing](#) 教程使用 TraCI 构建带有行人触发交通灯的十字路口。

资源

编程语言的接口

- Python: [工具包/traci](#)允许使用 Python 与sumo交互（这个库每天测试并支持所有 TraCI 命令）。
- C++: [libtraci](#)是一个客户端库，它是sumo源代码树的一部分。它与libsumo完全 API 兼容。
- C++: [C++ TraCIAPI](#)是一个客户端库，它是sumo源代码树的一部分。（API 覆盖几乎完成，但此客户端不再更新。请改用 libtraci）
- C++: [Veins](#) 项目提供了一个用于将 [sumo](#)与 [OMNET++](#) [耦合](#)的中间件。作为基础设施的一部分，它为 TraCI API 提供了一个 C++ 客户端库（API 的完整性有点落后于 Python 客户端）。
- .NET: [TraCI.NET](#) [是一个客户端库](#)，具有几乎完整的 API 覆盖率。
- Matlab [TraCI4](#) [Matlab](#) 。客户端作为每个 SUMO 版本的一部分包含在 [<SUMO_HOME>/tools/contributed/traci4matlab](#) 并非所有 TraCI 命令都已实现。
- Java: [libtraci](#)是一个客户端库，它是sumo源代码树的一部分。[它与libsumo完全 API 兼容](#)，并且 sumo 版本提供了预编译的 Java 绑定（通过 SWIG）。
- Java: [TraaS](#)提供了一个客户端库，它是sumo源代码树的一部分（API 覆盖范围很大，但此客户端不再更新。改用 libtraci）
- 其他: SWIG 支持的任何语言原则上都可以使用 libsumo 或 libtraci 提供的绑定



V2X 仿真

TraCI 允许结合使用sumo和通信网络模拟器来模拟[车辆通信](#)。有关可用解决方案的列表，请参阅 [主题/V2X](#) 。

其他资源

- [sumo](#)的 TraCI 服务器是普通发行版的一部分。源代码位于文件夹中 `src/traci-server` 。

参考

- Axel Wegener、Michal Piorkowski、Maxim Raya、Horst Hellbrück、Stefan Fischer 和 Jean-Pierre Hubaux。TraCI：用于耦合道路和网络模拟器的接口。第 11 届通信和网络仿真研讨会论文集，2008 年 4 月。[可在 ACM 数字图书馆获取](#) [↗](#)
- Axel Wegener、Horst Hellbrück、Christian Wewetzer 和 Andreas Lübke：具有反馈回路的 VANET 仿真环境及其在交通灯辅助中的应用。第三届 IEEE 汽车网络和应用研讨会论文集，美国路易斯安那州新奥尔良，2008 年。[可在 IEEEExplore 获取](#) [↗](#)

表现

使用 TraCI 会减慢仿真速度。减速的程度取决于许多因素：

- 每个模拟步骤的 TraCI 函数调用数
- 被调用的 TraCI 函数的类型（有些比其他的更昂贵）
- TraCI 脚本中的计算
- 客户语言

例子

作为一个示例用例，考虑在每个模拟步骤中检索每辆车的 x、y 位置（使用 python 客户端）：

```
while traci.simulation.getMinExpectedNumber() > 0:
    for veh_id in traci.vehicle.getIDList():
        position = traci.vehicle.getPosition(veh_id)
    traci.simulationStep()
```

- 该脚本每秒能够处理大约 25000 辆汽车。
- [通过使用订阅](#)，同样的值检索也可以加快到每秒 50000 辆汽车：



```
while traci.simulation.getMinExpectedNumber() > 0:
    for veh_id in traci.simulation.getDepartedIDList():
        traci.vehicle.subscribe(veh_id, [traci.constants.VAR_POSITION])
    positions = traci.vehicle.getAllSubscriptionResults()
    traci.simulationStep()
```

[在博洛尼亚场景](#)（9000 辆汽车，5000 个模拟步骤）上使用此脚本时，记录了以下运行时间：

- 不含 TraCI 8s
- 简单位置检索 90 年代
- 使用订阅 42s 检索

C++ 客户端性能更高：

- 简单位置检索 80 年代
- 使用订阅检索 28s

当前和未来的发展

从历史上看，TraCI 对每个域（感应回路、车辆等）使用不同的（单字节）命令 ID，其中字节的较高半部分表示命令（get、set、subscribe...），而较不重要的是域本身。为了允许超过 16 个由此拆分产生的域，最高有效位（直到 1.7.0 版才使用，因为只有 7 个命令）现在也用于域（只有三个用于命令）。这允许 28 个域，因为四个通用命令（如 SIMSTEP）阻止了一些可用的组合。目前只剩下四个可能的域。

此外，在 libsumo 发明之后，TraCI 接口的某些部分是如此通用，以至于使用 Apache Kafka 或 Google 协议缓冲区发明一个包装器可能并不难，从长远来看，它可以取代所有字节摆弄和不同的需要手工制作的客户。

故障排除

输出文件未关闭。

如果客户端在模拟仍在关闭时尝试访问输出，则会出现此问题。这可以通过让客户端等待模拟关闭来解决。错误报告是[#524](#)。

过时的 API

TraCI命令曾经有两“代”。第一个主要使用相扑中使用的字符串类型 ID 与这些基于 int 的外部表示之间的内部映射。映射是在内部完成的（在TraCI中）。第二“一代”，当前一代使用的字符串 ID 等于那些相扑读数。如果您绑定到第一代 API（例如，如果您想使用 TraNS），您只能使用 0.12.3 版本以下的[sumo](#)。请参阅 [有关获取](#)

重点关注TraCI监控时如何对现有车辆状态进行改变

选择周围车辆位置作为自己的位置时,可将被选择对象的车辆改变其颜色作标记

4. sumolib

- [Sumolib - SUMO Documentation](#)
- 作用: **sumolib**是一组 python 模块，用于处理 sumo 网络、模拟输出和其他模拟工件
- 与TraCI对比

附加功能

使用 TraCI 时，有一些常见任务未包含在 traci 库中，例如

- 分析道路网络
- 解析模拟输出

对于此功能，建议使用 [Tools/Sumolib](#)

- 功能举例:
 - 导入网络并检索节点和边
 - 计算纯 xml 边缘文件中的平均边缘速度
 - 使用统计模块计算平均速度
 - 根据地理坐标定位附近的边缘
 - 解析路由文件中的所有边
 - 在路线文件中解析车辆及其路线边缘
 - 解析边缘数据(meanData) 文件中的所有边缘
 - 坐标转换
 - 操作和编写 xml

5. Additional tools

这些工具分为以下主题：

- [assign](#) - 流量分配工具
- [构建](#)- 用于代码样式和构建子系统的工具
- [detector](#) - 一些处理现实生活感应回路数据的工具
- [turn-data](#) - 一些具有转数和转数比数据的工具以及现实生活中的感应回路数据
- [district](#) - 用于处理交通分析区域 (TAZ) aka districts 的工具
- [import](#) - 用于从各种格式导入数据的附加工具
 - [GTFS](#) - 一些使用 GTFS 数据的助手
 - [MATSim](#) - 一些使用 MATSim 数据的助手
 - [OpenDRIVE](#) - 使用 OpenDRIVE 数据的一些助手
 - [OSM](#) - 一些访问/使用 OpenStreetMap 数据的助手
 - [Vissim](#) - 使用 VISSIM 数据的一些助手
 - [Visum](#) - 使用 VISUM 数据的一些助手
- [net](#) - 用于处理网络（主要是 SUMO 网络）的工具
- [projects](#) - 我们的项目使用的额外工具
- [route](#) - 用于处理路线的工具
- [shapes](#) - 用于处理形状 (PoI 和多边形) 的工具
- [tls](#) - 交通灯系统工具
- [traci](#) - TraCI 接口的 Python 实现
- [traceExporter](#) - 生成各种格式的移动轨迹/轨迹/轨道/gps 轨迹
- [trip](#) - 没有od2trips 的行程生成和修改
- [可视化](#)- SUMO 输出的图形评估
- [sumolib](#) - 用于处理 sumo 网络、模拟输出和其他实用函数的 python 库
- [output](#) - 用于处理 sumo 输出文件的 python 工具
- [emissions](#) - 用于计算和可视化排放的工具
- [railways](#) - 模拟火车的工具
- [drt](#) - 模拟需求响应运输的工具
- [xml](#) - 用于转换 xml 和 xml 模式相关实用程序的 python 工具
- [devel](#) - 开发者工具
- [杂项](#)- 不属于上述类别的工具

5-1. xml工具

- [Xml - SUMO Documentation](#)
- 作用: 实现SUMO 输出与 CSV/电子表格互相转换的工具
- 内容
 - [xml2csv.py](#)
 - [csv2xml.py](#)
 - [xml2protobuf.py](#)

- `protobuf2xml.py`
- `changeAttribute.py`
- `filterElements.py`

5-2. TraceExport.py

- [TraceExporter - SUMO Documentation](#)
- 作用: 将移动轨迹（FCD 输出）导出为不同的“轨迹文件”格式

`traceExporter.py` 将 SUMO `fcd` 输出转换和过滤为不同的“跟踪文件”格式：OMNET、Shawn、ns2/ns3、PHEM。

- **目的：**跟踪文件转换/生成
- **系统：**便携（Linux/Windows已测试）；在命令行上运行
- **输入（强制）：**SUMO `fcd-output`
- **输出：**一个或多个“跟踪文件”和其他补充文件
- **编程语言：**Python

- 不同的应用程序读取“车辆痕迹”，文件主要包含车辆随时间变化的位置。**traceExporter.py** 通过将 SUMO 的 `fcd` 输出转换为多种支持的格式来生成此类文件。可以生成以下文件，按应用程序排序：

- OMNET: mobility-traces (.xml)
- Shawn: snapshot-files (.xml)
- ns2/ns3: trace-files, activity-files, and mobility-files (all *.tcl)
- PHEM: .dri-files, .str-files, .fzp-files, .flt-files
- GPSDAT csv traces (id,date,x,y,status,speedKMH)
- GPX [↗](#)
- KML [↗](#)
- Kepler-JSON [↗](#)
- IPG Traces (experimental)
- SSAM trajectories [↗](#)
- POIs (can visualize GPS distortions and sampling in [sumo-gui](#))
- `fcd output` to restrict the file to certain regions or time intervals

- OMNET: 移动痕迹 (.xml)
- 肖恩: 快照文件 (.xml)
- ns2/ns3: 跟踪文件、活动文件和移动文件 (所有 *.tcl)
- PHEM: .dri 文件、.str 文件、.fzp 文件、.flt 文件
- GPSDAT csv traces (id,date,x,y,status,speedKMH)
- GPX [↗](#)
- 知识分子语言 [↗](#)
- 开普勒-JSON [↗](#)
- IPG 痕迹 (实验)
- SSAM 轨迹 [↗](#)
- POI (可以在 [sumo-gui](#) 中可视化 GPS 失真和采样)
- `fcd` 输出以将文件限制在特定区域或时间间隔

6. Visualization

推荐的可视化选项, 基于python的matplotlib库实现

- [Visualization - SUMO Documentation](#)
- 可能会用到的例子

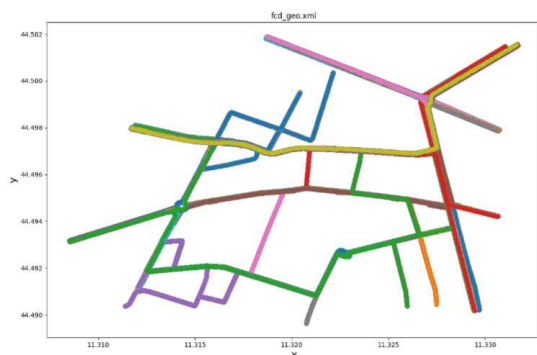
随时间变化的所有轨迹 1

输入是fcd_output。

调用以生成绘图：

```
python tools/visualization/plotXMLAttributes.py -x x -y y -s -o allXY_output.png fcd.xml --scatterplot
```

其中 -x 是 x 轴的属性；-y 是 y 轴的属性；-s 是显示剧情；-o 是输出文件名；--scatterplot 是制作散点图而不是线图..



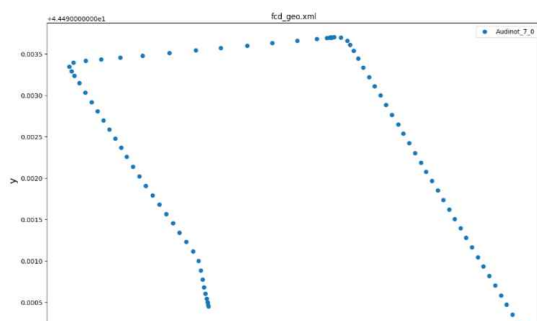
随时间选择的轨迹 1

输入是fcd_output。

调用以生成绘图：

```
python tools/visualization/plotXMLAttributes.py -x x -y y -s -o vehLocations_output.png fcd.xml -i id --filter-ids Audinot_7_0 --scatterplot --legend
```

其中 -x 是 x 轴的属性；-y 是 y 轴的属性；-s 是显示剧情；-o 是输出文件名；-i 是过滤后的属性名称（在本例中为车辆 ID）；--filter-ids 是过滤后的属性名称的值（本例中车辆id = Audinot_7_0）；--scatterplot 是做散点图而不是线图；--legend 是显示图例。



plot_trajectories.py

在给定的--fcd 输出文件中创建所有轨迹图。该工具特别位于<SUMO_HOME> /tools 中。

使用示例：

```
python tools/plot_trajectories.py fcd.xml -t td -o plot.png -s
```

选项-t (--trajectory-type)支持可以相互绘制的不同属性。参数是一个双字母代码，每个字母编码一个从 fcd 输入派生的属性。

笔记

plot_trajectories.py 类似于plotXMLAttributes.py但专用于fcd-output。它支持加载数据（行驶距离）中不存在的派生属性，还允许进行更多过滤（--filter-route、--filter-edges）。

可用属性

- t: 以秒为单位的时间
- d: 行驶距离（每辆车的第一个 fcd 数据点从 0 开始）。使用欧拉积分根据速度计算距离。设置选项--ballistic用于弹道集成。
- a: 加速度
- s: 速度（米/秒）
- i: 车辆角度（航行度数）
- x: 以米为单位的 X 位置
- y: 以米为单位的 Y 位置
- k: 公里数（需要--fcd-output.distance）
- g: 与领导者的差距（需要--fcd-output.max-leader-distance）

示例轨迹类型

- **td**: 时间与距离
- **ts**: 时间与速度
- **ta**: 时间与加速度
- **ds**: 距离与速度
- **da**: 距离与加速度
- **xy**: 行驶路径的空间图
- **kt**: 公里数与时间的关系（与选项`--invert-yaxis`结合以获得经典的铁路图）。



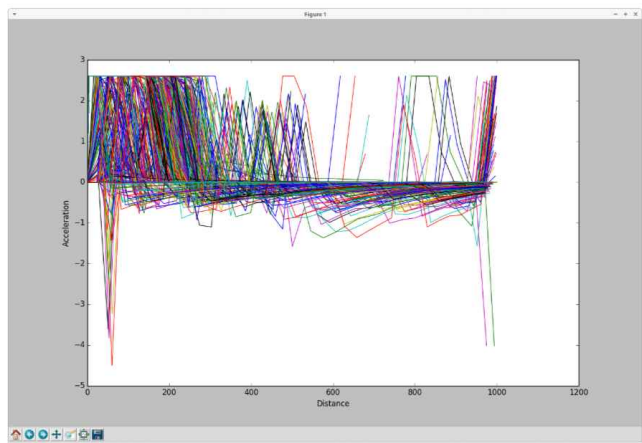
加速度与距离

输入是fcd-output。

调用以生成绘图：

```
python tools/plot_trajectories.py -t da -o Plot_trajectories.png fcd.xml
```

其中 -t 是上述轨迹类型；-o 是输出文件名。



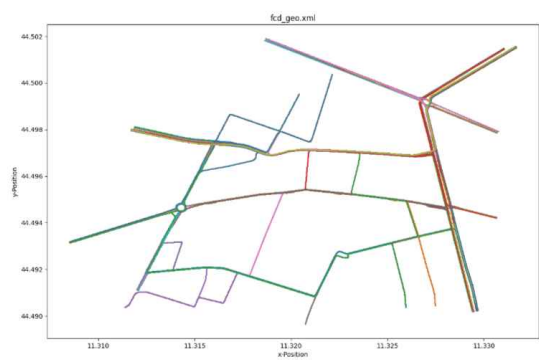
随时间变化的所有轨迹 2

输入是fcd-output。

调用以生成绘图：

```
python tools/plot_trajectories.py -t xy -o allLocations_output.png fcd.xml
```

其中 -t 是上述轨迹类型；-o 是输出文件名。



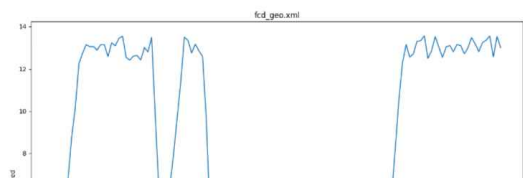
随时间变化的选定轨迹 2

输入是fcd-output。

调用以生成绘图：

```
python tools/plot_trajectories.py -t xy -o selectXY_output.png fcd.xml --filter-ids Audinot_7_0
```

其中 -t 是上述轨迹类型；-o 是输出文件名；--filter-ids 是过滤属性的值（在本例中为 id = Audinot_7_0）。



如果您使用 SUMO，请告诉我们您的 [出版物](#)。

笔记

一般引用 SUMO 时，请使用我们当前的参考出版物：[“使用 SUMO 进行微观交通模拟”](#) [↗](#)；Pablo Alvarez Lopez、Michael Behrisch、Laura Bieker-Walz、Jakob Erdmann、Yun-Pang Flötteröd、Robert Hilbrich、Leonhard Lüken、Johannes Rummel、Peter Wagner 和 Evamarie Wießner。IEEE 智能交通系统会议 (ITSC)，2018 年。

本文档的内容可自由编辑。要编辑某个页面，请单击右上角的“在 GitHub 上编辑”按钮并提交 Pull Request。[这是关于编辑文章的简短帮助](#)。您还可以在本地[构建文档](#)，或[下载](#) [↗](#)副本。

本文档不断更新，始终参考最新的开发版本。SUMO 特定发行版本的文档包含在该版本的下载中，可以通过打开[<SUMO_HOME> /docs/userdoc/index.html](#) 查看。

介绍

- [交通模拟SUMO](#)

基本用法

- 本文档中的符号
- [需要](#)，基本的计算机技能
- [安装](#) [相扑](#)
- [使用 SUMO 命令行应用程序](#)
- [教程](#)
- [验证应用程序输入](#)

网络建设

- [SUMO路网介绍](#)
- [抽象网络生成](#)
- [使用netconvert导入网络](#)
 - [使用 XML 定义自己的网络](#)
 - [导入非 SUMO 网络](#)
 - [来自 OpenStreetMap](#)
 - [三键式场景生成器](#)
 - [来自 VISUM](#)
 - [来自 Vissim](#)
 - [来自 OpenDrive](#)
 - [来自 MATsim](#)
 - [来自 ArcView（形状文件）](#)
 - [来自 DlrNavTeq](#)
 - [来自 Robocup 模拟联赛](#)
 - [导入 SUMO 网络和修补](#)
 - [为高速公路模拟构建网络](#)
 - [构建行人模拟网络](#)
 - [更多网络转换选项](#)
 - [附加输出](#)
- [使用 netedit 创建和修改网络](#)
- [包括高程数据](#)
- [地理坐标](#)

需求建模

- [SUMO 需求建模简介](#)
- [车辆、车辆类型和路线的定义](#)
- [使用 netedit 定义流量需求](#)
- [公共交通模拟](#)
- [模拟个人和旅行链](#)
- [物流模拟](#)
- [最短或最佳路径路由](#)
- [多式联运路线](#)
- [仿真中的路由](#)
- [计算动态用户分配](#)
- [产生行人交通需求](#)
- [生成车辆类型分布以对车队建模](#)

需求生成的数据源

- [导入 O/D 矩阵](#)
 - [其他 VISUM 需求进口商](#)
 - [其他 Vissim 需求进口商](#)
- [来自计数数据的路线（道路计数、转弯计数）](#)
- [按转弯概率路由](#)
- [基于活动的需求生成](#)
- [随机旅行](#)
- [多模态随机交通](#)
- [GTFS数据](#)



模拟

- 基本定义
- 保存和加载模拟状态

输出

- 仿真输出概览

TraCI（在线互动）

- TraCI概述-交通控制接口_ _
- Libsumo - 使用 sumo 作为库

交通管理和其他结构

- 红绿灯
- 公共交通
- 变速标志
- 路由器/替代路线标牌
- 汽化器（已弃用，请改用校准器）
- 流量和速度和类型的动态校准
- 停车场
- 转机

交通模式

- 行人模拟
- 自行车模拟
- 铁路模拟
- 航道模拟

附加功能

- 排放量
- 电动车
- 电动混合动力汽车、架空线路、变电站
- 后勤
- 通用参数
- 形状可视化
- 无线设备检测
- 紧急车辆
- 简单队列行驶 (Simpla)
- 需求响应交通 (DRT) / 出租车
- 绿灯最佳速度咨询 (GLOSA)

型号详情

- 车速
- 车辆插入
- 车辆权限（访问限制）
- 道路容量
- 路口动态
- 随机性
- 路由和重新路由
- 子车道模型
- 逆向行驶
- 安全
- 细观模型
- 长度和距离

常见问题

- 为什么车辆要传送
- 意外干扰
- 转机次数太多
- 意想不到的变道操作？
- 如何获得高流量？

附加工具

除了主要应用程序（sumo、sumo-gui、netedit、netconvert 等）外，还有 150 多个附加工具。它们涵盖了从交通网络分析、需求生成、需求修改到输出分析的主题。其中大部分是用python 编写的。所有工具都可以在<SUMO_HOME> /tools下的 SUMO 发行版中找到。

有关所有工具的索引，请参见：

- 刀具索引

以下是一些最重要/最常用工具的链接：

- [osmWebWizard](#) - 创建一个简单的场景，只需点击几下并使用您的网络浏览器
- [从 Python 连接 TraCI](#) - 使用 Python 访问正在运行的 SUMO 模拟
- [sumolib](#) - 用于处理 SUMO 网络和一般 sumo xml 文件的 Python 模块
- [Xml 工具](#)- 将 SUMO 输出转换为 CSV/电子表格的工具，反之亦然
- [traceExporter.py](#) - 将移动轨迹（FCD 输出）导出为不同的“轨迹文件”格式
- [netdiff.py](#) - 确定两个网络之间的差异
- [可视化工具](#)- 以图形化和友好的方式可视化各种模拟输出

理论

- [一般交通模拟](#)

应用手册

- [相扑](#)
- [sumo-gui](#)
- [网络转换](#)
- [网络编辑](#)
- [网络生成](#)
- [od2trips](#)
- [双路由器](#)
- [路由器](#)
- [路由器](#)
- [路由器](#)
- [多边形转换器](#)
- [活性源](#)
- [排放地图](#)
- [排放驾驶周期](#)

软件贡献

有些人扩展了 SUMO 或构建了工具以使其更有用。并非所有这些扩展都是“SUMO 核心”的一部分。

- [包含在发行版中](#)
- [外部的](#)

附录

- [变更日志](#)
- [词汇表](#)
- [FAQ](#)