

交通工程学

Introduction to Traffic Engineering

第 6 节 交通仿真简介

葛乾

西南交通大学 交通工程系
西南交通大学 系统科学与系统工程研究所

本节目录

- ① 简单跟驰模型的仿真实现
- ② 商业软件
- ③ 开源微观仿真软件 SUMO
- ④ 开源宏观仿真软件 MATSim

跟驰模型回顾

公式

$$\ddot{x}_{n+1}(t + T) = \lambda[\dot{x}_n(t) - \dot{x}_{n+1}(t)]$$

原理

驾驶员反应 $(T + t)$ = 灵敏度 (λ) × 驾驶员接受的刺激 (t)

灵敏度

驾驶员对刺激的反应系数，量纲是 $1/s$

刺激

引导车加、减速引起的两车速度差或车间距变化

反应

驾驶员根据引导车的状态对后车进行操纵及效果

注意点

- 后车从了解到前车速度变化到做出反应需要一个反应时间
- 在每个时间间隔（也是我们仿真的时间单位），所有车辆的位置、速度和加速度均会更新。时间间隔越小，仿真的精度越高
- 车辆的速度和位置变化遵循牛顿第二定律，加速度的变化遵循跟驰模型

怎么写程序

- 假设仿真的总时长为 T_m , 仿真的时间间隔为 Δt , 则我们需要执行 $\lceil \frac{T_m}{\Delta t} \rceil^1$ 次循环
- 在每个仿真时间间隔, 对于每辆车 n , 我们可知

$$v_n(t+1) = v_n(t) + a_n(t) \times \Delta t \quad (1)$$

$$x_n(t+1) = x_n(t) + v_n(t) \times \Delta t + \frac{1}{2} a_n(t) \Delta t^2 \quad (2)$$

$$a_{n+1}(t+1) = \lambda [v_n(t) - v_{n+1}(t)] \quad (3)$$

式 (1-2) 来自牛顿动力学定律, 式 (3) 来自跟驰模型

- 下一时刻的状态 (加速度、速度、位置) 仅受上一仿真时刻的状态影响
- 所有车辆的起始状态、前车的制动行为可由随机生成, 也可读取自输入文件

¹此处的符号表示向上取整

伪代码

- 初始化：确定好输入参数，如 $\Delta t, v_n(0), x_n(0), a_n(0)$
- 循环：判断是否达到仿真结束时间，更新现在的时刻
 - 随机生成前车的制动状态（加速？减速？）
 - 分别由公式（1-3）更新速度、位置和加速度
 - 保存每辆车的当前状态

本节目录

① 简单跟驰模型的仿真实现

② 商业软件

③ 开源微观仿真软件 SUMO

④ 开源宏观仿真软件 MATSim

主流商业软件

- PTV Vissim
- AIMSUM
- Paramics

AIMSUN 的跟驰模型

- 基于 Gipps 模型

$$v_n(t + T) = v_n(t) + 3.6 \left[2.5 a_n T \left(1 - \frac{v_n(t)}{V_n} \right) \sqrt{0.025 + \frac{v_n(t)}{V_n}} \right]$$

式中 a_n 是最大的理想加速度； V_n 表示 n 的理想速度（自由流速度）

Vissim 的跟驰模型

- 从 Wiedemann 的两个模型 (Weidemann74 与 99) 修改而来，也被称为心理物理模型 (Psychophysical) 或动作点 (action-point) 模型
- 这类模型假设驾驶员仅仅在触发到某个距离或者相对速度的阈值时，才会变更驾驶行为
- 这些阈值由一个表格定义

本节目录

① 简单跟驰模型的仿真实现

② 商业软件

③ 开源微观仿真软件 SUMO

④ 开源宏观仿真软件 MATSim

为什么我们需要关注开源软件？

- 商业软件成本高昂
- 开源软件已经能满足通用要求，而且更容易定制
- 技术自主可控

为什么我们需要关注开源软件？

哈工大、哈工程被禁用MATLAB

时间: 2020年06月12日 | 作者: Admin | 来源: 机器之心

哈工大、哈工程的老师和学生们最近无法使用 MATLAB 了，这一消息迅速成为了人们关注的热点。



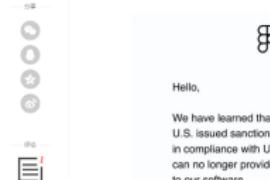
哈尔滨工业大学

近日，在知乎等社交媒体上，有哈工大学生表示收到了正版软件和润科技的通知，而在与 MATLAB 开发公司 MathWorks 交涉之后，人们得知因为美国政府实体名单的原因，相关授权已被中止。

大疆遭Figma断供，多家国产软件公司发声“可替代”

2022
03/14
22:21
爆料有奖
南航深媒爆料电话: 0755-82121212

南都讯 记者柳洋 近日，多方消息称，大疆遭美国软件公司Figma封停、断供。其中，即时设计iDesign发布公告称，3月12日在网络上传播的“Figma 封停所有被美国制裁名单的公司账号”一事，公司已经查证属实。



SUMO 能用来做什么

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

SUMO 能用来做什么

- Includes all applications needed to prepare and perform a traffic simulation (network and routes import, DUA, simulation)
- 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

使用范围

- traffic lights evaluation
- route choice and re-routing
- evaluation of traffic surveillance methods
- simulation of vehicular communications
- traffic forecast
- etc.

功能模块简介

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.

支持的跟驰模型

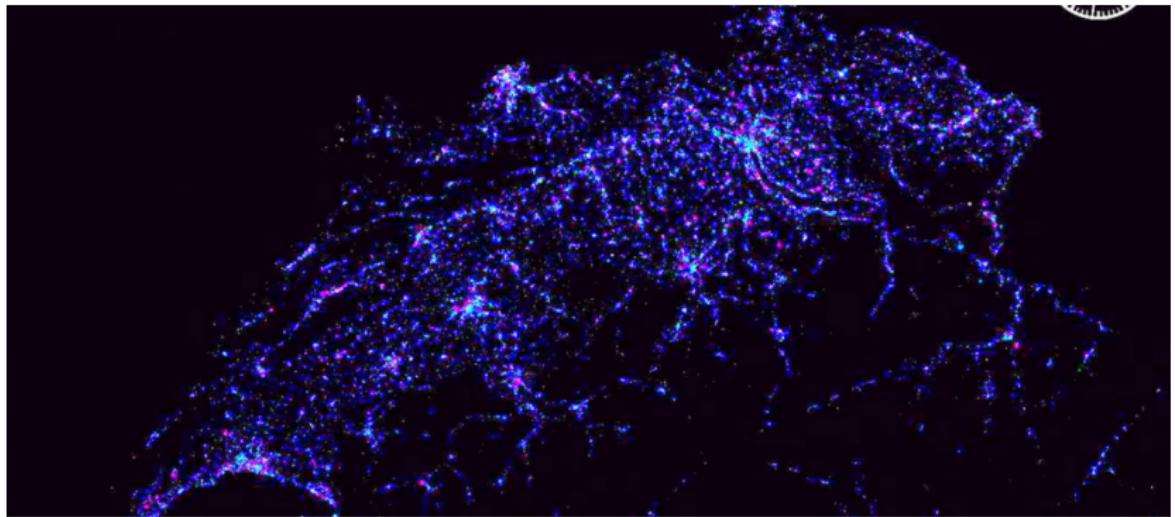
MSCFModel.cpp	MSCFModel_Krauss.cpp
MSCFModel.h	MSCFModel_Krauss.h
MSCFModel_ACC.cpp	MSCFModel_KraussOrig1.cpp
MSCFModel_ACC.h	MSCFModel_KraussOrig1.h
MSCFModel_CACC.cpp	MSCFModel_KraussPS.cpp
MSCFModel_CACC.h	MSCFModel_KraussPS.h
MSCFModel_CC.cpp	MSCFModel_KraussX.cpp
MSCFModel_CC.h	MSCFModel_KraussX.h
MSCFModel_Daniel1.cpp	MSCFModel_PWag2009.cpp
MSCFModel_Daniel1.h	MSCFModel_PWag2009.h
<u>MSCFModel_EIDM.cpp</u>	MSCFModel_Rail.cpp
└── MSCFModel_EIDM.cpp	MSCFModel_Rail.h
└── MSCFModel_EIDM.h	MSCFModel_SmartSK.cpp
MSCFModel_IDM.cpp	MSCFModel_SmartSK.h
MSCFModel_IDM.h	MSCFModel_W99.cpp
MSCFModel_Kerner.cpp	MSCFModel_W99.h
MSCFModel_Kerner.h	MSCFModel_Wiedemann.cpp
MSCFModel_Krauss.cpp	MSCFModel_Wiedemann.h
MSCFModel_Krauss.h	

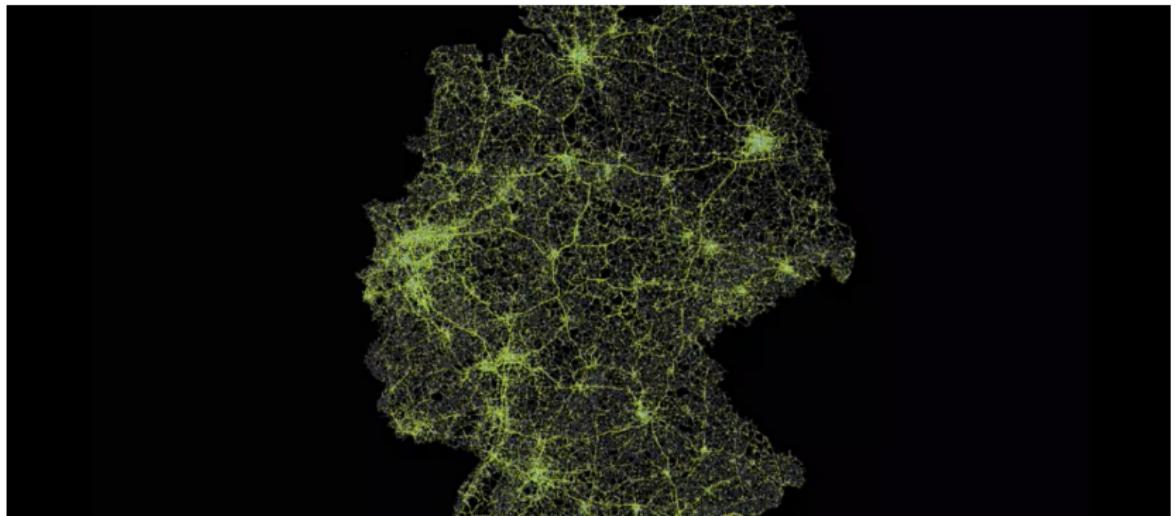
SUMO 的使用

- 手动编辑交通网络（观看视频）
- 从 openstreetmap 等导入（观看视频）

本节目录

- ① 简单跟驰模型的仿真实现
- ② 商业软件
- ③ 开源微观仿真软件 SUMO
- ④ 开源宏观仿真软件 MATSim











历史

- TRANSIMS: 由洛斯阿拉莫斯国家实验室研发; C/C++; 基于元胞自动机对道路交通进行仿真
- TRANSIMS 的开发者之一 Kai Nagel 于 1999 年换工作的到 ETH Zurich, 因为美国出口限制无法继续使用 TRANSIMS, 而开发了新的仿真软件
 - 基于 C/C++
 - 用了排队模型而非 CA
 - 使用 xml 存储数据
 - 数据库放于内存中, 以加快仿真速度
 - 刚开始的版本只考虑了私家车的仿真
 - 运行在集群上

发展

- Kai Nagel 于 2004 年再次换工作到 TU Berlin, Kay Axhausen (ETH) 开始加入开发
- 2006 年用 Java 重写了代码, 设计为单线程的架构 (64 位 CPU, 减少了 I/O)
- 2007 年之后, 加入更多新的特性
 - 交通方式选择
 - 换乘
 - 公交
 - 路段收费
 - 排放
 - 车辆计数
 - 灾害避险
 - 需求导向公交
 - 等等

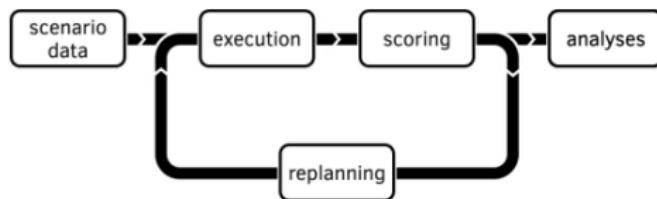
社区现状

- Kai Nagel 领导的团队，交通系统规划与交通信息研究组 (VSP, TU Berlin)
- Kay Axhausen 领导的团队，交通规划与系统研究所 (IVT, ETH Zurich)
- 两个团队合作创业：senozon（建模）/simunto（可视化）

基于主体的交通模型

- Agent: 一个有代表性的人，由群体合成（population synthesis）方法生成
- Plan: 一个主体（一天内）的活动计划，例如“早上 7:30 去工作，下午 17:00 回家，路上去购物，预计 17:45 到家”，每个主体至少应该有一个 Plan
- Score: 主体执行完 Plan 后所得到的效用
- Scenario, Model: 描述某个区域基础设施（供给）和需求的一些数据集和参数。在 MATSim 里，这两个词一般不做区分

MATSim 的仿真系统设计



scenario data: description of infrastructure (road network, transit schedule, ...) and agents.

execution: Agents' travel plans are executed in parallel.

A.k.a.: mobsim (mobility simulation), synthetic reality, network loading, traffic flow simulation

scoring: each executed plan obtains a score.

replanning: some agents change plans or create new ones.

analyses: traffic volumes, average speeds, utility changes, emissions, accessibility, ...

一些细节

- 迭代: execution, scoring 与 replanning 三个步骤循环执行
- 优化: 每个主体都希望优化他/她一天的出行活动所获得效用
- 进化算法: 每个主体都有自身的可选 Plan 集合 (选择集), 经过迭代后, 加入好的 Plan, 删去差的
- 共同进化: 每个 Plan 对主体来说是好是坏, 依赖于其他人的 Plan
- 纳什均衡: 每个主体做决策时仅最大化自身的 Plan

需要哪些场景数据 (Scenario Data)

- 最低限度
 - 交通网络数据 (network.xml)
 - 需求数据 (population.xml)
 - 配置数据 (config.xml)
- 其他数据
 - 公交数据 (transitSchedule.xml, transitVehicles.xml)
 - 设施数据 (facilities.xml)
 - 已有统计量 (counts.xml)
 - 路段收费 (roadpricing.xml)
 - 车道数据 (lanes.xml)
 - 信号配时 (在 SUMO 中可以自动生成, 但是 MATSim 本身不支持对信号配时的输入)
 - 等等

交通网络数据

- 描述车辆/主体所行驶的车道和交叉口
- 可以是多模式：车辆/步行/自行车/公交/铁路等
- 路网的参数
 - 长度
 - 限速
 - 车道数
 - 通行能力
- MATSim 中的道路是无向的



network.xml 示例

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE network SYSTEM "http://www.matsim.org/files/dtd/network_v2.dtd">
<network>
  <nodes>
    <node id="1" x="2000" y="1000" />
    <node id="2" x="4000" y="1500" />
    <node id="3" x="3000" y="3000" />
    ...
  </nodes>
  <links cpperiod="01:00:00">
    <link id="1" from="1" to="2" length="3000.0" capacity="1800" freespeed="13.88" permlanes="1" modes="car" />
    <link id="2" from="2" to="1" length="3000.0" capacity="1800" freespeed="13.88" permlanes="1" modes="car" />
    <link id="3" from="2" to="3" length="5000.0" capacity="3500" freespeed="22.22" permlanes="2" modes="car" />
    <link id="4" from="3" to="2" length="5000.0" capacity="1800" freespeed="22.22" permlanes="1" modes="car" />
    ...
  </links>
</network>
```

场景数据

- 描述主体所以以及他们的活动计划

- 活动类型
- 活动持续时长
- 活动地点
- 用到的交通模式
- 路线



population.xml 示例

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE population SYSTEM "http://www.matsim.org/files/dtd/population_v6.dtd">
<population>
    <person id="1">
        <attributes>
            <attribute name="employed" class="java.lang.Boolean" >true</attribute>
        </attributes>
        <plan score="140.00982053869416" selected="yes">
            <activity type="home" link="1112" x="890.0" y="685.0" end_time="07:43:56" />
            <leg mode="car" dep_time="07:18:16" trav_time="00:01:40">
                <route type="links" start_link="1112" end_link="2223" trav_time="00:31:40" distance="14403.0">1112 1222 2223</r
            </leg>
            <activity type="work" link="2223" x="1802.0" y="2782.0" end_time="17:09:02" />
            <leg mode="car" dep_time="17:43:36" trav_time="00:05:00">
                <route type="links" start_link="2223" end_link="1312" trav_time="00:27:32" distance="12573.0">2223 2313 1312</r
            </leg>
            <activity type="shopping" link="1312" x="1802.0" y="2782.0" end_time="17:42:02" />
            <leg mode="car" dep_time="17:43:36" trav_time="00:05:00">
                <route type="links" start_link="1312" end_link="1112" trav_time="00:04:15" distance="3127.0">1312 1211 1112</r
            </leg>
            <activity type="home" link="1112" x="890.0" y="685.0" />
        </plan>
        ...
    </person>
    ...
</population>
```

population.xml 示例 (cont.)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE population SYSTEM "http://www.matsim.org/files/dtd/population_v6.dtd">
<population>
  <person id="1">
    <attributes>
      <attribute name="employed" class="java.lang.Boolean" >true</attribute>
    </attributes>
    <plan score="140.00982053869416" selected="yes">
      <activity type="home" link="1112" x="890.0" y="685.0" end_time="07:43:56" />
      <leg mode="car" dep_time="07:18:16" trav_time="00:01:40">
        <route type="links" start_link="1112" end_link="2223" trav_time="00:31:40" distance="14403.0">1112 1222 2223</r
      </leg>
      <activity type="work" link="2223" x="1802.0" y="2782.0" end_time="17:09:02" />
      <leg mode="car" dep_time="17:43:36" trav_time="00:05:00">
        <route type="links" start_link="2223" end_link="1312" trav_time="00:27:32" distance="12573.0">2223 2313 1312</r
      </leg>
      <activity type="shopping" link="1312" x="1802.0" y="2782.0" end_time="17:42:02" />
      <leg mode="car" dep_time="17:43:36" trav_time="00:05:00">
        <route type="links" start_link="1312" end_link="1112" trav_time="00:04:15" distance="3127.0">1312 1211 1112</r
      </leg>
      <activity type="home" link="1112" x="890.0" y="685.0" />
    </plan>
    <plan ...>
      ...
    </person>
    ...
  </population>
```

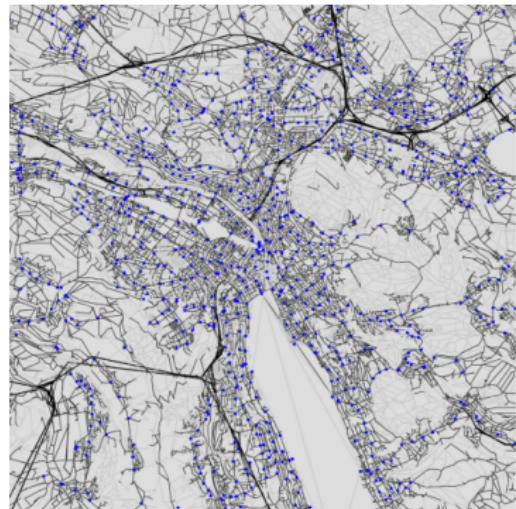
population.xml 示例 (cont.)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE population SYSTEM "http://www.matsim.org/files/dtd/population_v6.dtd">
<population>
  <person id="1">
    <attributes>
      <attribute name="employed" class="java.lang.Boolean" >true</attribute>
    </attributes>
    <plan score="140.00982053869416" selected="yes">
      <activity type="home" link="1112" x="890.0" y="685.0" end_time="07:43:56" />
      <leg mode="car" dep_time="07:18:16" trav_time="00:01:40">
        <route type="links" start_link="1112" end_link="2223" trav_time="00:31:40" distance="14403.0">1112 1222 2223</route>
      </leg>
      <activity type="work" link="2223" x="1802.0" y="2782.0" end_time="17:09:02" />
      <leg mode="car" dep_time="17:43:36" trav_time="00:05:00">
        <route type="links" start_link="2223" end_link="1312" trav_time="00:27:32" distance="12573.0">2223 2313 1312</route>
      </leg>
      <activity type="shopping" link="1312" x="1802.0" y="2782.0" end_time="17:42:02" />
      <leg mode="car" dep_time="17:43:36" trav_time="00:05:00">
        <route type="links" start_link="1312" end_link="1112" trav_time="00:04:15" distance="3127.0">1312 1211 1112</route>
      </leg>
      <activity type="home" link="1112" x="890.0" y="685.0" />
    </plan>
    <plan ...>
      ...
    </person>
  ...
</population>
```

公交数据

- 描述公交服务

- 公交站点的位置
- 公交线路编号
- 公交线路经行停靠点
- 路径中停靠点的出发时间
- 公交车辆的容量



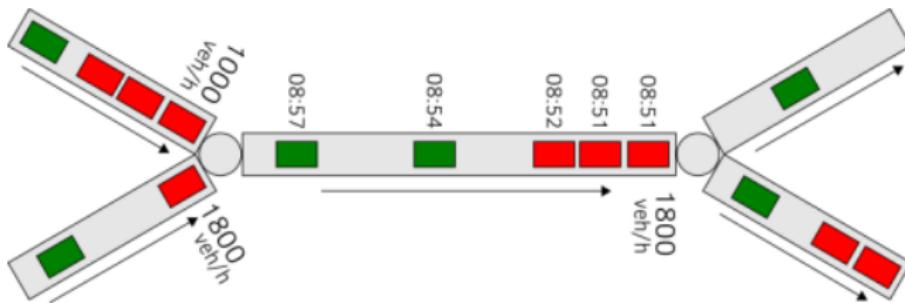
transitSchedule.xml 示例

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE transitSchedule SYSTEM "http://www.matsim.org/files/dtd/transitSchedule_v1.dtd">
<transitSchedule>
  <transitStops>
    <stopFacility id="1" x="1050" y="1050" linkRefId= "11" name="Aadorf" />
    <stopFacility id="2" x="2050" y="2940" linkRefId= "12" name="Beweiler"/>
  ...
  </transitStops>
  <transitLine id="Blue Line">
    <transitRoute id="outbound">
      <transportMode>train</transportMode>
      <routeProfile>
        <stop refId="1" departureOffset="00:00:00"/>
        <stop refId="2" arrivalOffset="00:03:20" departureOffset="00:04:00"/>
        <stop refId="3" arrivalOffset="00:09:00" />
      </routeProfile>
      <route>
        <link refId="11"/>
        <link refId="12"/>
        <link refId="23"/>
        <link refId="33"/>
      </route>
      <departures>
        <departure id="01" departureTime="06:00:00" vehicleRefId="tr_1" />
        <departure id="02" departureTime="06:15:00" vehicleRefId="tr_2" />
        <departure id="03" departureTime="06:30:00" vehicleRefId="tr_1" />
      </departures>
    </transitRoute>
  </transitLine>
</transitSchedule>
```

执行程序 MobSim

- 所有主体的所有活动计划同时执行，以模拟现实
- 仿真的开始时间为 00:00:00，逐步执行，每个循环对活动计划执行一下操作
 - 主体结束活动
 - 主体离开活动地点
 - 车辆进入交通流
 - 车辆穿过交叉口
 - 车辆因遭遇阻塞而落后其他车辆
 - 公交车到达停靠点
 - 主体到达活动地点
 - 执行新的活动

排队模型

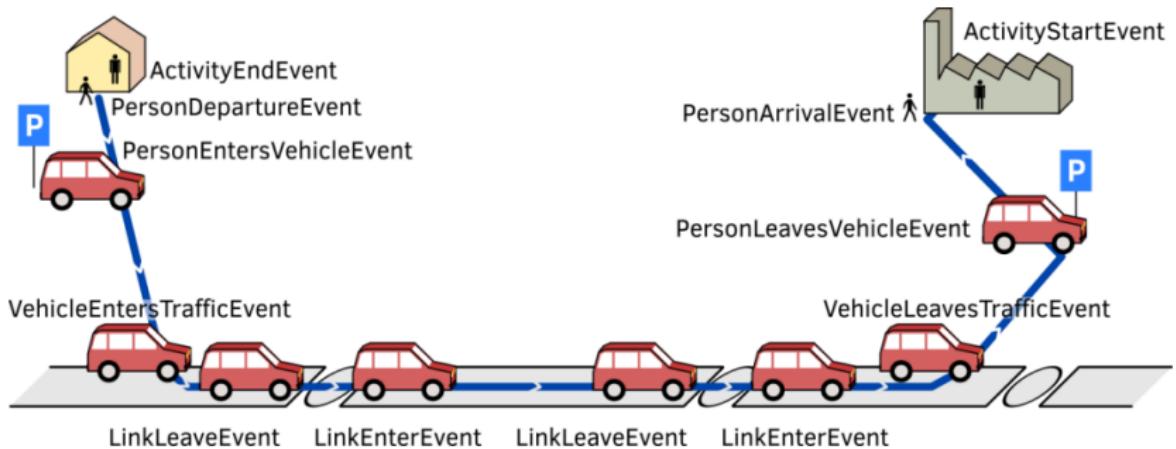


- QSim: 对排队的简单描述
- 排队规则: FIFO
- 根据旅行时间计算何时驶离
- 驶离的流率等于路段的通行能力
- 路段可能会塞满车 (导致拥堵甚至溢出路段)

执行结果：事件

- Mobsim 模块会生成 Events 作为仿真记录
- 每个事件都有一个时间戳和类型，也可能会包含额外信息
- 常见的事件类型
 - 主体开始/结束活动行为 ('actstart', 'actend')
 - 主体开始/结束单个旅程 (leg)('departure', 'arrival')
 - 车辆驶入/驶离某个路段 ('enter link', 'left link')
- 大型的仿真场景会生成数百万计的事件，这也是我们分析的数据源

事件示例



单个主体的事件示例

```
<event time="25279.0" type="actend" person="188" link="1112" actType="home" />
<event time="25279.0" type="departure" person="188" link="1112" legMode="car" />
<event time="25279.0" type="PersonEntersVehicle" person="188" vehicle="188" />
<event time="25279.0" type="vehicle enters traffic" person="188" link="1112" vehicle="188" networkMode="car" />
<event time="25280.0" type="left link" vehicle="188" link="1112" />
<event time="25280.0" type="entered link" vehicle="188" link="1222" />
<event time="25381.0" type="left link" vehicle="188" link="1222" />
... more enter link / left link events
<event time="25482.0" type="entered link" vehicle="188" link="2324" />
<event time="25582.0" type="vehicle leaves traffic" person="188" link="2324" vehicle="188" networkMode="car" />
<event time="25582.0" type="PersonLeavesVehicle" person="188" vehicle="188" />
<event time="25582.0" type="arrival" person="188" link="2324" legMode="car" />
<event time="25582.0" type="actstart" person="188" link="2324" actType="work" />

<event time="64332.0" type="actend" person="188" link="2324" actType="work" />
<event time="64332.0" type="departure" person="188" link="2324" legMode="car" />
<event time="64332.0" type="PersonEntersVehicle" person="188" vehicle="188" />
<event time="64332.0" type="vehicle enters traffic" person="188" link="2324" vehicle="188" networkMode="car" />
<event time="64333.0" type="left link" vehicle="188" link="2324" />
<event time="64333.0" type="entered link" vehicle="188" link="2423" />
<event time="64434.0" type="left link" vehicle="188" link="2423" />
... more enter link / left link events
<event time="64737.0" type="entered link" vehicle="188" link="1112" />
<event time="64837.0" type="vehicle leaves traffic" person="188" link="1112" vehicle="188" networkMode="car" />
<event time="64837.0" type="PersonLeavesVehicle" person="188" vehicle="188" />
<event time="64837.0" type="arrival" person="188" link="1112" legMode="car" />
<event time="64837.0" type="actstart" person="188" link="1112" actType="home" />
```

公交车的事件示例

```
<event time="21600.0" type="TransitDriverStarts"
    driverId="pt_tr_1_1" vehicleId="tr_1" transitLineId="Blue Line" transitRouteId="1to3" departureId="01" />
<event time="21600.0" type="departure" person="pt_tr_1_1" link="11" legMode="car" />
<event time="21600.0" type="PersonEntersVehicle" person="pt_tr_1_1" vehicle="tr_1" />
<event time="21600.0" type="vehicle enters traffic" person="pt_tr_1_1" link="11" vehicle="tr_1" networkMode="car" />
<event time="21600.0" type="VehicleArrivesAtFacility" vehicle="tr_1" facility="1" delay="Infinity" />
<!-- vehicle is at first stop, a passenger enters -->
<event time="21601.0" type="PersonEntersVehicle" person="298" vehicle="tr_1" />
<event time="21610.0" type="VehicleDepartsAtFacility" vehicle="tr_1" facility="1" delay="0.0" />
<event time="21611.0" type="left link" vehicle="tr_1" link="11" />
<event time="21611.0" type="entered link" vehicle="tr_1" link="12" />
<event time="21811.0" type="VehicleArrivesAtFacility" vehicle="tr_1" facility="2a" delay="1.0" />
<!-- vehicle is at second stop, passenger exits -->
<event time="21812.0" type="PersonLeavesVehicle" person="298" vehicle="tr_1" />
<event time="21821.0" type="VehicleDepartsAtFacility" vehicle="tr_1" facility="2a" delay="-39.0" />
<event time="21822.0" type="left link" vehicle="tr_1" link="12" />
<event time="21822.0" type="entered link" vehicle="tr_1" link="23" />
<event time="22123.0" type="left link" vehicle="tr_1" link="23" />
<event time="22123.0" type="entered link" vehicle="tr_1" link="33" />
<event time="22124.0" type="VehicleArrivesAtFacility" vehicle="tr_1" facility="3" delay="-36.0" />
<!-- vehicle is at third (and last) stop -->
<event time="22124.0" type="VehicleDepartsAtFacility" vehicle="tr_1" facility="3" delay="-36.0" />
<event time="22124.0" type="vehicle leaves traffic" person="pt_tr_1_1" link="33" vehicle="tr_1" networkMode="car" />
<event time="22124.0" type="PersonLeavesVehicle" person="pt_tr_1_1" vehicle="tr_1" />
<event time="22124.0" type="arrival" person="pt_tr_1_1" link="33" legMode="car" />
```

Events 与 Plans 的不同

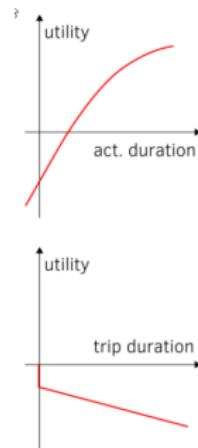
- Plans 是 mobsim 的输入，Events 是输出
- 两者经常不同
 - Plans 只是初始的假设，可能存在错误（如对路段旅行时间的假设）
 - 制定 Plans 时，很难预料到其他主体的行为
- 每次迭代都会生成事件数据，然而结果可能不同，因为主体会根据他人的计划而调整自身的计划

Events 与 Plans 的不同

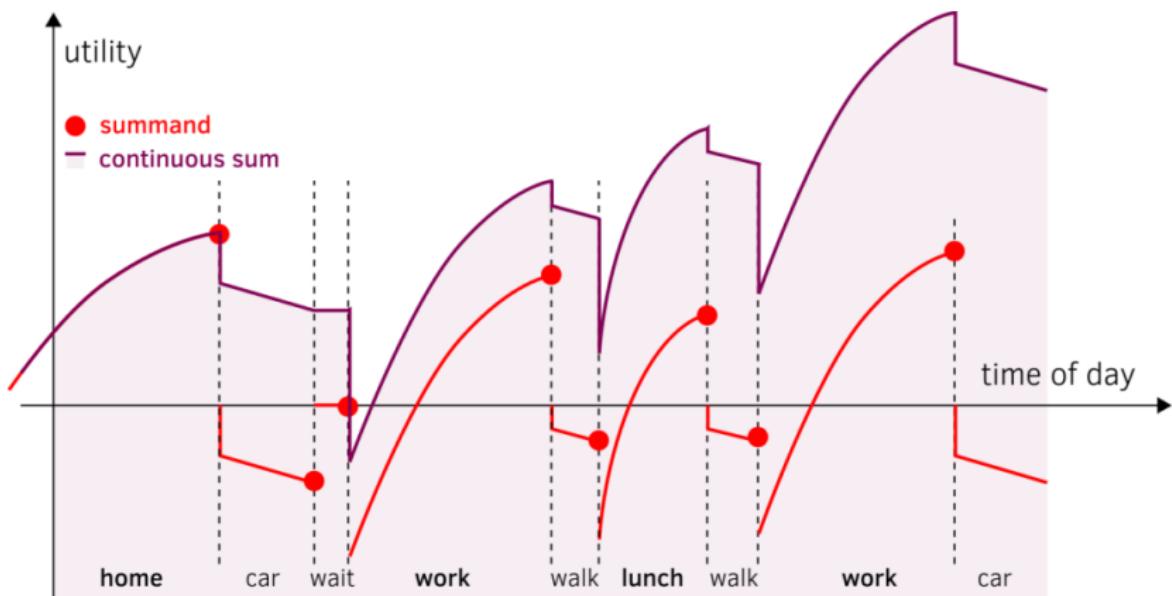
- Plans 是 mobsim 的输入，Events 是输出
- 两者经常不同
 - Plans 只是初始的假设，可能存在错误（如对路段旅行时间的假设）
 - 制定 Plans 时，很难预料到其他主体的行为
- 每次迭代都会生成事件数据，然而结果可能不同，因为主体会根据他人的计划而调整自身的计划

计划是如何改进的?

- 为比较不同计划的优劣，需要计算其能为主体带来的效用 → 评分函数需考虑
 - 参与活动带来的正效用
 - 旅行的负效用
 - 花钱的负效用（道路收费，车费等）
 - 迟到/早出发所带来的负效用
- 累加在一起，构成了主体对自己一天活动的评分
 - 评分函数 \approx 进化计算中的适应度函数
 - 评分函数 \approx 经济学中的效用函数



评分函数示例



重新计划

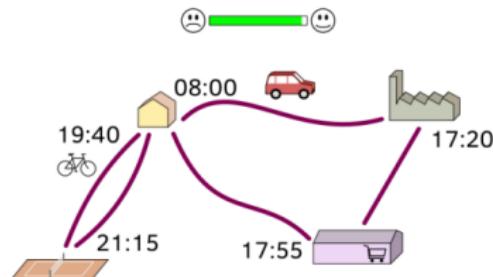
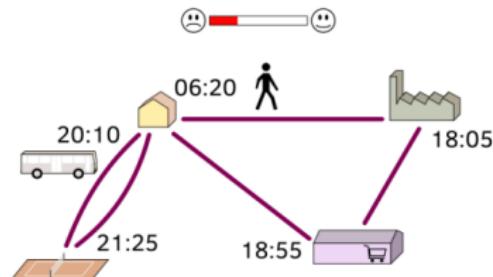
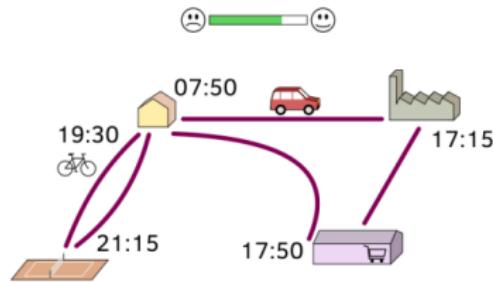
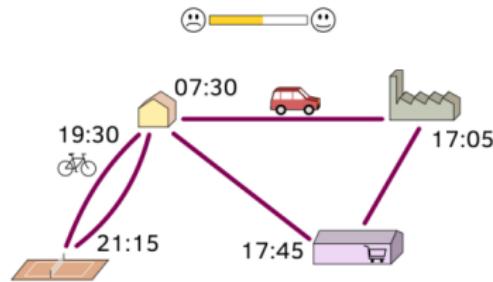
- 每个主体都有一个或者多个可选计划
 - 对应于选择集或者基因池
- 每次迭代，其中一个（被选择的）计划被执行并评分
- 部分主体创建（并选择）新的计划，其他主体会从已有的计划中选择一个
 - 对应于遗传算法中的基因变异
- 部分主体会移除较差的计划
 - 适者生存

重新计划：计划的修改

- 标准的选择维度（策略）
 - 出发时间选择
 - 交通方式选择
 - 路径选择
- 实验性策略
 - 次要活动²地点选择
- 目前不支持：活动模式选择

²主要指非通勤/通学之外的活动

重新计划示例



重新计划的注意事项

- 每次迭代，不需要每个主体都更新其计划
 - 否则易造成波动现象
- 经过若干次迭代，重新计划功能可被停用
 - 通常，一开始数次变动比较大
- 一般来说，重新计划会在总迭代数的一定百分比之后停用
 - 避免主体尝试不好的计划，如很少使用的交通方式

分析功能

- MATSim 自身有一些简单的分析功能，如行程的交通方式，每小时每个路段的平均旅行时间，每个路段的小时交通量，旅行时间统计等
- 但是多数分析跟问题有关，需要我们自己写代码
- 用来分析的数据源是 Events。例如统计路段交通量，可以从 Events 中为每个路段寻找车辆驶离路段的事件，然后求和
- 可视化？

课下训练

- 尝试自己动手写一个简单的跟驰模型
- 在自己电脑安装 Eclipse SUMO；探索一下 SUMO 的功能
- 在自己电脑安装 MATSim；探索一下 MATSim 的功能

谢谢！