

Modellizzazione di una rete di trasporto e simulazione delle emissioni acustiche mediante il software di simulazione SUMO

Webinar: “Misuriamo il suono, monitoraggio punto zero”

Terzo Focus Group del progetto RUMBLE

Il software SUMO

Introduzione al software

SUMO = Simulation of Urban MObility: software di pianificazione e di simulazione dei sistemi di trasporto **open source**

Consente di realizzare:

- **Modellizzazione della rete** di trasporto; è possibile **importare** il modello di rete da **OpenStreetMap** o da **Vissim**.
- **Stima** della domanda di trasporto.
- **Assegnazione** , attraverso **una micro simulazione**, della domanda di trasporto alla rete (determinazione delle caratteristiche di circolazione sugli archi e sui nodi della rete)

Modellazione della rete

Per la realizzazione del **modello di rete di trasporto** si hanno i seguenti moduli:

- **NetEdit**: consente di **generare e modificare il modello di rete di trasporto**. E' caratterizzato da una **GUI (Graphical User Interface)** con molte funzionalità.
- **NetConvert**: Per poter **trasformare un file di rete generato con OpenStreetMap**, in un formato leggibile da Sumo. Netconvert non ha una GUI, ma è utilizzato **scrivendo comandi specifici nel prompt dei comandi di DOS**. L'importazione da **Vissim** è possibile, ma avvengono degli errori: alcuni archi e nodi non vengono importati. L'importazione da Visum è possibile solo se si utilizza la versione di **Visum** in tedesco.

Domanda di trasporto

A partire da, in alternativa:

- **matrici O/D aggiornate** per i vari periodi di simulazione
- **flussi misurati** sulle diverse **sezioni** stradali,
- **percentuale di utenti** su ciascuna **manovra** delle **intersezioni** stradali,

in ogni caso vengono determinati:

- **numero di veicoli** da simulare,
- **composizione del parco veicolare** (percentuale di veicoli pesanti, di autobus, di motocicli)
- **per ogni veicolo** da simulare, la cosiddetta “**trip**”, ossia: **origine**, **destinazione**, ed **istante** di partenza del veicolo dall'origine.

Determinazione delle trips dei veicoli.

Sono disponibili i pacchetti seguenti (utilizzati in alternativa); tutti i pacchetti ricevono in input la rete di trasporto:

- **OD2Trips**: Riceve in input da matrice O/D una domanda valida in un certo periodo temporale (per es. 7-9): all'interno del periodo l'istante di partenza viene determinato mediante estrazione da distribuzione uniforme.
- **DFRouter**: Determina le trips dei veicoli a partire da misurazioni di traffico nelle sezioni stradali.
- **JTRRouter**: Determina le trips a partire dalle percentuali di veicoli su ogni manovra di ciascuna intersezione della rete.

Questi pacchetti non possiedono una propria interfaccia ma devono essere utilizzati mediante specifici comandi in ambiente DOS.

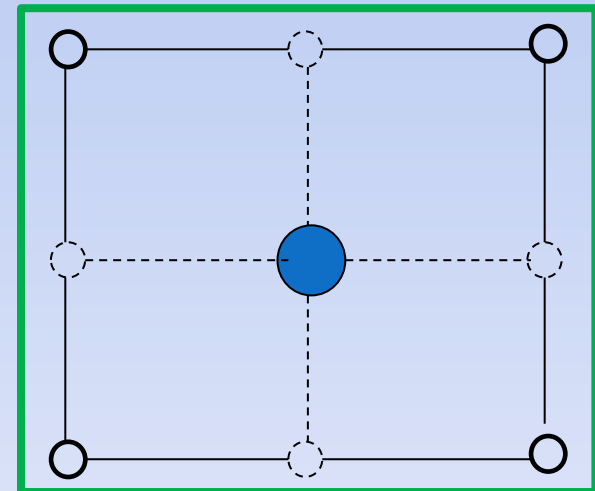
Dettagli su OD2Trips

In SUMO, la **domanda di trasporto** non ha **origine e destinazione** nei nodi centroidi ma in **archi** e, di preciso, nel **punto centrale dell'arco**.

DFRouter e **JTRRouter** non presentano problemi di determinazione degli **archi di origine e destinazione** della domanda.

Invece **OD2Trips** presenta il problema di **trasformare una domanda definita su centroidi in una domanda avente origini e destinazioni in archi**.

Per effettuare questa trasformazione **ci si avvale delle TAZ (Traffic Assignment Zones)**. Nell'esempio a fianco, ho un comparto ambientale. La **TAZ**, racchiusa all'interno del quadrato verde, **comprende: il centroide**, e gli **archi reali** che delimitano il comparto ambientale.



La domanda avente **origine o destinazione** in una data **TAZ** (ossia avente origine o destinazione nei **centroidi contenuti nella TAZ**) verrà da SUMO **assegnata agli archi contenuti nella TAZ**.

La domanda viene (da SUMO) **ripartita tra gli archi contenuti nella TAZ** in maniera proporzionale alla lunghezza e al numero di corsie di ciascun arco.

Le TAZ vengono **definite dall'analista**, che utilizza SUMO.

Per prima cosa, l'analista disegna i poligoni che delimitano le TAZ mediante Netedit. Poi, converte i poligoni in TAZ mediante i pacchetti (da utilizzare in successione) **polyconvert** ed **edgesInDistricts**.

Dettagli su DFRouter

DFRouter riceve in input:

- Il “**detector file**”, in cui è riportata la localizzazione delle sezioni stradali di rilievo. Nel detector file, per ogni sezione di rilievo, sono riportati: ID dell’arco, ID della corsia, posizione della sezione di rilievo lungo l’arco, ID della sezione di rilievo.
- Il “**measure file**”, in cui sono riportati, per ogni sezione di rilievo: i flussi rilevati, la composizione del parco veicolare, e gli istanti di inizio e fine del rilievo.

DFRouter determina: il numero di veicoli da simulare, la composizione del parco veicolare, e le trips dei veicoli.

Il numero dei veicoli viene determinato a partire dai rilievi nelle sezioni stradali. L'istante di ingresso nel sistema, per ciascun veicolo, viene determinato a partire dagli istanti di inizio e fine del rilievo, riportati nel **measure file**.

In questo caso, in cui la domanda è fornita a partire dai rilievi, **non si hanno TAZ**, e come **origine e destinazione delle trip dei veicoli** viene fornito **soltanto l'arco** (il veicolo appare nel **punto centrale dell'arco**).

Dettagli su JTRRouter

JTRRouter funziona in modo simile a DFRouter.

Anche JTRRouter riceve in input:

- Il “**detector file**”, in cui è riportata la localizzazione delle intersezioni stradali di rilievo. Nel detector file, per ogni intersezione di rilievo, sono riportati: ID dell’intersezione, posizione della manovra all’interno dell’intersezione, ID della manovra.
- Il “**measure file**”, in cui sono riportati, per ogni manovra dell’intersezione: i flussi rilevati, la composizione del parco veicolare, e gli istanti di inizio e fine del rilievo.

Anche JTRRouter determina trips aventi origine e destinazione in corrispondenza degli archi.

Inoltre **nemmeno** applicando JTRRouter **ci sono le TAZ**, e come **origine e destinazione** delle trip dei veicoli viene fornito **soltanto l’arco**.

Assegnazione della domanda

Per ogni veicolo, a partire dalla trip, viene determinata la route, ossia il percorso seguito dal veicolo.

Nella route del veicolo sono riportati: origine, destinazione, istante di partenza, ed archi che compongono la route.

Di default la route viene calcolata mediante assegnazione tutto o niente (**pacchetto Duarouter**), ma può essere determinata anche mediante Assegnazione Dinamica Intraperiodale (che, però, nella guida di Sumo è chiamata Dynamic User Equilibrium) (**pacchetto Duaiterate**).

Se la route viene determinata mediante Assegnazione Dinamica Intraperiodale, ciascun veicolo viene assegnato all'itinerario di minimo costo, che però viene aggiornato passo passo sulle condizioni di traffico (**algoritmo di Gawron**), oppure si utilizza il **modello Logit** (con costi aggiornati passo passo).

Assegnazione della domanda

In SUMO è inoltre possibile effettuare un'assegnazione di equilibrio mediante il pacchetto **Marouter**.

Marouter effettua una **assegnazione di equilibrio stocastico** essendo uno “Stochastic Equilibrium (SUE) model”. L'assegnazione di carico stocastico della rete, svolta in ciascuna iterazione del modello iterativo di assegnazione, è basato su un **modello logit modificato (proposto da Lohse)**.

Marouter fornisce come output:

- File **netload.xml**: contiene flusso, densità, tempo di percorrenza medio e grado di saturazione di ogni arco del grafo
- File delle **route**: riporta tutti gli itinerari utilizzati per congiungere ciascuna coppia O/D, il numero di veicoli/ora che percorre ogni itinerario, e gli archi percorsi da ogni itinerario.

Risultati della simulazione (simulation GUI)

Una volta calcolate le route, vengono determinati: i flussi sugli archi della rete, la densità, l'occupazione (periodo di tempo in cui un arco è occupato da almeno un veicolo), ma anche le emissioni atmosferiche e acustiche.

Inoltre è possibile avere una visualizzazione, sulla GUI, dei veicoli che si muovono nella rete di trasporto.

La GUI della simulazione può essere avviata se il file delle route è stato determinato mediante Duarouter e Dualterate. Se invece il file delle route è stato determinato mediante Marouter, questo non può essere utilizzato per la GUI perché non fornisce il percorso ottimo per singolo veicolo, ma per gruppi di veicoli.

Esempio applicativo con utilizzo di OD2Trips

Esempio di assegnazione e simulazione utilizzando SUMO su di una rete test, utilizzando una domanda arbitraria e considerando un parco veicolare omogeneo, composto da sole autovetture.

La procedura è farraginoso e si compone delle seguenti fasi:

Fase 1. Importazione della rete da Openstreetmap e correzione della rete mediante NetEdit

Fase 2. Creazione del file “taz” relativo alle zone di traffico

Fase 3. Creazione del file di domanda

Fase 4. Determinazione del file delle trips: numero di veicoli da simulare e, per ogni veicolo, origine, destinazione ed istante di partenza.

Fase 5. Determinazione delle route mediante Duaiterate

Fase 6. Output della simulazione.

Fase 1. Importazione e correzione della rete

Da **Openstreetmap** seleziono un'area da considerare come rete test (in questo caso una zona di Lucca) che viene automaticamente salvata come file `.osm`. Cambio il nome di questo file in `Lucca.osm`.

Successivamente nel **prompt dei comandi** del dos scrivo:

```
netconvert --osm Lucca.osm -o Lucca.net.xml.
```

Otterrò un file `.net.xml`, che rinominerò come `Lucca.net.xml`.

Apro il file `Lucca.net.xml` mediante la GUI di NetEdit e procedo a correggerlo. Infatti:

- Spesso sono sbagliati i sensi unici e il numero di corsie di ogni arco.
- La velocità di percorrenza dell'arco è spesso sbagliata (di solito OpenStreetMap considera un valore di velocità prefissato a seconda del tipo di strada, ad es. un arco urbano ha sempre velocità pari a 50 km/h indipendentemente dalle sue caratteristiche geometriche).
- Nelle intersezioni semaforizzate, il tempo di ciclo è impostato pari a 90 secondi e l'intersezione è organizzata sempre (o quasi) con 2 fasi, di ugual durata, indipendentemente dall'importanza dei rami.

Fase 2. Creazione del file relativo alle TAZ

La procedura per la creazione del file delle TAZ è molto lunga e farraginoso e richiede l'utilizzo di diverse funzioni di SUMO.

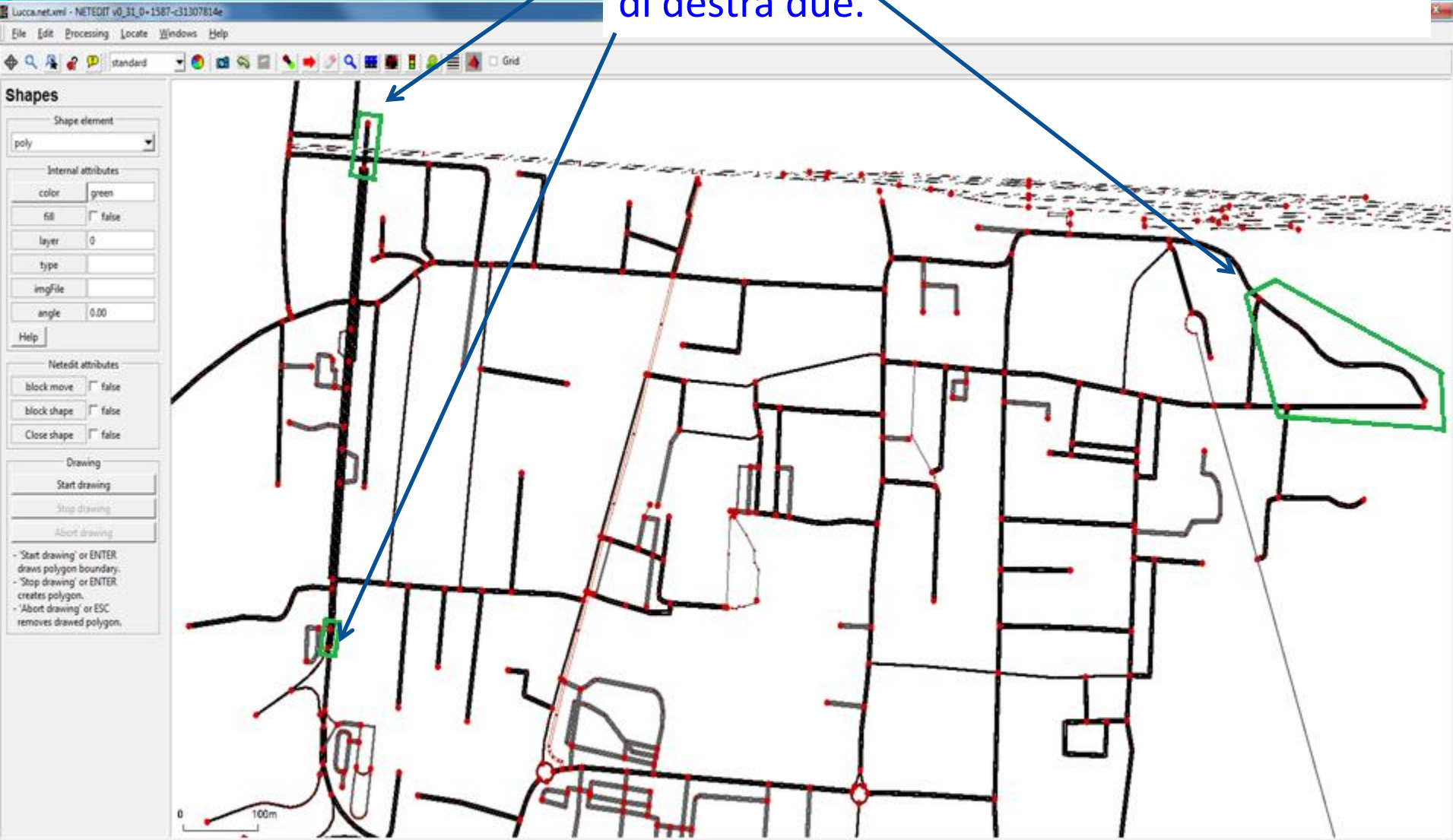
Per prima cosa, nel menu di Netedit clicco su “polygons”



Questa voce del menu mi consente di **disegnare sulla rete dei poligoni**: ogni **poligono identifica una TAZ** e deve comprendere al suo interno almeno un arco. La domanda avente origine o destinazione in un dato poligono (ossia, in una data TAZ) verrà da SUMO assegnata agli archi contenuti all'interno del poligono.

Per esempio, nella zona di Lucca considerata considero tre zone di traffico e quindi disegno tre poligoni (tre TAZ). Poi salvo la rete di trasporto con disegnati i poligoni (relativi alle TAZ) nel file Lucca.shapes.xml

Disegno dei poligoni (in verde): quelli di sinistra comprendono un solo arco, quello di destra due.



Import done:
314 nodes loaded.
33 types loaded.
670 edges loaded.

Quindi apro il prompt dei comandi del dos ed entro nella cartella in cui si trova il file di rete (Lucca.net.xml) e il file dei poligoni (Lucca.shapes.xml).

Poi scrivo:

```
polyconvert -n Lucca.net.xml --xml-files Lucca.shapes.xml -o Lucca.poi.xml
```

(-n sta per network, -o per output (definisco come chiamare il file di output))

In questo modo Polyconvert mi genera il file Lucca.poi.xml.

Per creare il file taz finale devo utilizzare il pacchetto edgesInDistricts, scrivendo, **sempre sul prompt dei comandi**, quanto segue:

```
edgesInDistricts.py -n Lucca.net.xml -t Lucca.poi.xml -o Lucca.taz.xml --verbose
```

(-n sta per network, -t sta per taz, con -o invece definisco come chiamare il file di output, --verbose significa che il programma scrive tutto quello che sta facendo: è cosa buona per controllare che non ci siano errori)

Viene così generato automaticamente il file Lucca.taz.xml.

Fase 3. Creazione del file di domanda

La matrice di domanda data deve essere riportata nel **file di domanda** secondo una **modalità precisa**.

Esempio di file di domanda, per 3 sole TAZ e per un'ora di simulazione:

0.00 è l'istante di inizio e 1.00 quello di fine, in ore, del periodo della simulazione a cui la domanda si riferisce. Ciò significa che tutti i veicoli qui riportati inizieranno il loro spostamento nella prima ora di simulazione.

1 è la zona (TAZ) di origine; 2 la zona (TAZ) di destinazione, 500 i veicoli che hanno origine in 1 e destinazione in 2.

Chiamo il file di domanda: **domanda.txt**.

```
$OR;D2
* From-Time To-Time
0.00 1.00
* Factor
1.00
* some
* additional
* comments
1 2 500.00
1 3 2500.00
2 1 500.00
2 3 500.00
3 1 2500.00
3 2 500.00
```

Fase 4. Determinazione del file delle trips

Prima di determinare il file delle trips occorre fare una **piccola modifica al file taz**. Il file taz che avevo ottenuto dalla procedura di cui al punto 2 si mostra nel modo seguente:

```
<tazs>
```

```
<taz id="poly_0" edges="-320013565 -32605882 320013565 32605882"/>
```

```
<taz id="poly_1" edges="-279107021#0 -75267770#0 -75267770#1 279107021#0  
75267770#0 75267770#1"/>
```

```
<taz id="poly_2" edges="-156531216#10 -156531216#11 -156644184#0 -  
156644184#1 -156644214 -156644221 156531216#10 156531216#11 156644184#0  
156644184#1 156644214 156644221"/>
```

```
</tazs>
```

taz id = riporta l'identificativo della TAZ

edges = riporta l'identificativo degli archi compresi all'interno della TAZ.

Il problema è che qui le tre TAZ sono denominate poly_0 , poly_1 , poly_2 , mentre nel file di domanda sono denominate 1, 2 e 3.

Procedo a modificare il file taz cambiando gli identificativi delle zone:

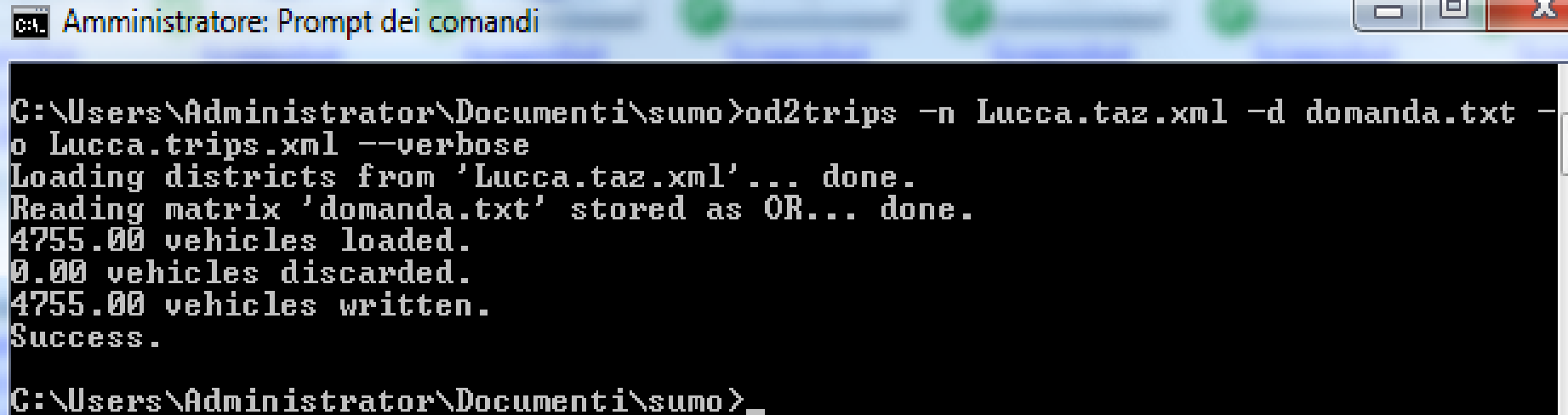
```
<tazs>  
  <taz id="1" edges="-32605882 32605882"/>  
  <taz id="2" edges=" -156395874 156395874"/>  
  <taz id="3" edges="-156531216#10 -156531216#11 -156644184#0 -156644184#1 -  
156644214 -156644221 156531216#10 156531216#11 156644184#0 156644184#1  
156644214 156644221"/>  
</tazs>
```

A questo punto posso **generare il file delle trips**, scrivendo sul prompt dei comandi:

```
od2trips -n Lucca.taz.xml -d domanda.txt -o Lucca.trips.xml --verbose
```

- n sta per network (Lucca.taz.xml riporta non solo le TAZ ma anche la rete)
- d sta per demand (domanda.txt è il file di domanda)
- o viene utilizzato per assegnare al file di output il nome che voglio io (altrimenti viene chiamato di default trips.trips.xml)
- verbose è il comando affinché SUMO scriva passo passo tutto quello che sta facendo (così se ci sono problemi me ne accorgo).

La schermata del prompt dei comandi su cui si è operato è la seguente:



```
C:\Users\Administrator\Documenti\sumo>od2trips -n Lucca.taz.xml -d domanda.txt -o Lucca.trips.xml --verbose
Loading districts from 'Lucca.taz.xml'... done.
Reading matrix 'domanda.txt' stored as OR... done.
4755.00 vehicles loaded.
0.00 vehicles discarded.
4755.00 vehicles written.
Success.

C:\Users\Administrator\Documenti\sumo>_
```

Estratto del file Lucca.trips.xml:

```
<trip id="63" depart="0.79" from="-32605882" to="156395855#0" fromTaz="1" toTaz="2"/>  
<trip id="3193" depart="1.88" from="156395855#0" to="-32605882" fromTaz="2" toTaz="1"/>  
<trip id="283" depart="2.83" from="32605882" to="-156395855#0" fromTaz="1" toTaz="2"/>  
<trip id="3028" depart="4.33" from="156395855#0" to="-32605882" fromTaz="2" toTaz="1"/>
```

trip id= id della trip,

depart= istante di partenza

from= id dell'arco di origine

to= id dell'arco di destinazione

fromTaz= id della TAZ di origine

toTaz= id della TAZ di destinazione

Fase 5. Determinazione delle route mediante Duaiterate

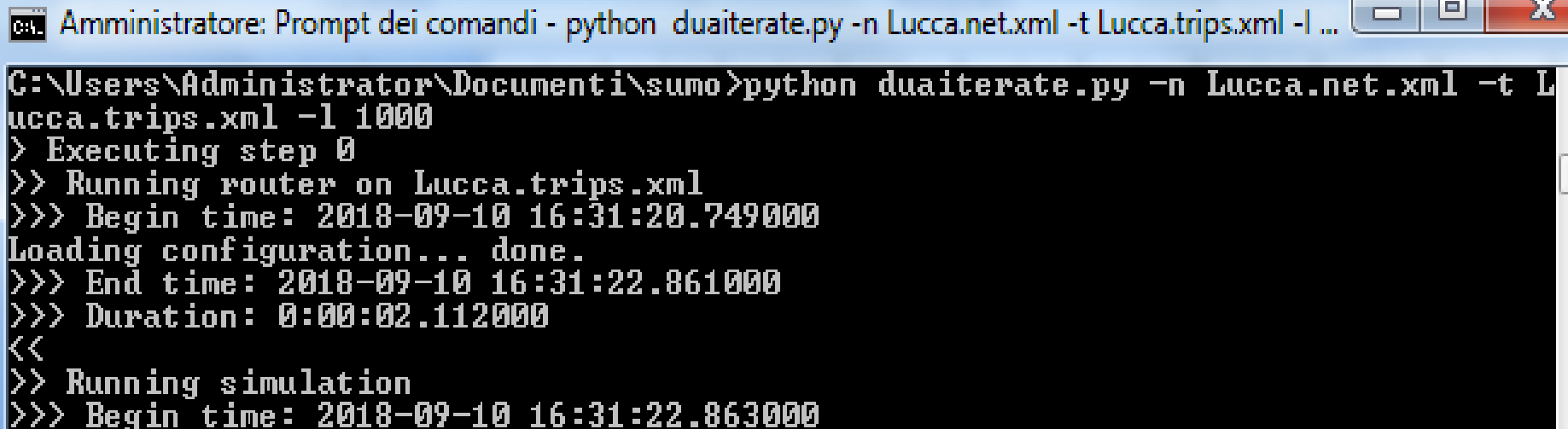
A questo punto procedo a determinare, per ogni veicolo, le route.

Scelgo di utilizzare l'assegnazione dinamica intraperiodale, applicando l'algoritmo di Gawron (assegnazione di default).

Scriverò quindi sul prompt dei comandi del dos:

```
python duaiterate.py -n Lucca.net.xml -t Lucca.trips.xml -l 1000
```

-n è il file di rete, -t il file delle trips; -l il numero di iterazioni



```
Amministratore: Prompt dei comandi - python duaiterate.py -n Lucca.net.xml -t Lucca.trips.xml -l ...  
C:\Users\Administrator\Documenti\sumo>python duaiterate.py -n Lucca.net.xml -t L  
ucca.trips.xml -l 1000  
> Executing step 0  
>> Running router on Lucca.trips.xml  
>>> Begin time: 2018-09-10 16:31:20.749000  
Loading configuration... done.  
>>> End time: 2018-09-10 16:31:22.861000  
>>> Duration: 0:00:02.112000  
<<  
>> Running simulation  
>>> Begin time: 2018-09-10 16:31:22.863000  
-
```


Dualterate fornisce in output il file delle route (Lucca.rou.xml). Un estratto del file è il seguente:

```
<vehicle id="519" depart="4.77" fromTaz="1" toTaz="3">
```

```
<route edges="-32605882 -320013565 -156395891#1 -156395891#0 -156395866#6 -  
156395866#5 -156395866#4 -156395866#3 -156395866#2 -156395866#1 -  
156395866#0 156531238#0 156531238#1 156531238#2 518897175 542722309#0  
542722309#1 542722309#2 542722309#3 518897173 156644184#0"/>
```

```
</vehicle>
```

```
<vehicle id="3886" depart="4.80" fromTaz="2" toTaz="3">
```

```
<route edges="-156395855#0 320013568#0 320013568#1 320013568#2 320013568#3  
320013568#4 320013563#0 320013563#1 -156395891#1 -156395891#0 -156395866#6  
-156395866#5 -156395866#4 -156395866#3 -156395866#2 -156395866#1 -  
156395866#0 156531238#0 156531238#1 156531238#2 518897175 542722309#0  
542722309#1 542722309#2 542722309#3 518897173 156644184#0"/>
```

```
</vehicle>
```

Il file delle route riporta: id del veicolo, istante di partenza, zone di origine e destinazione, e id degli archi (edges) percorsi dalla route.

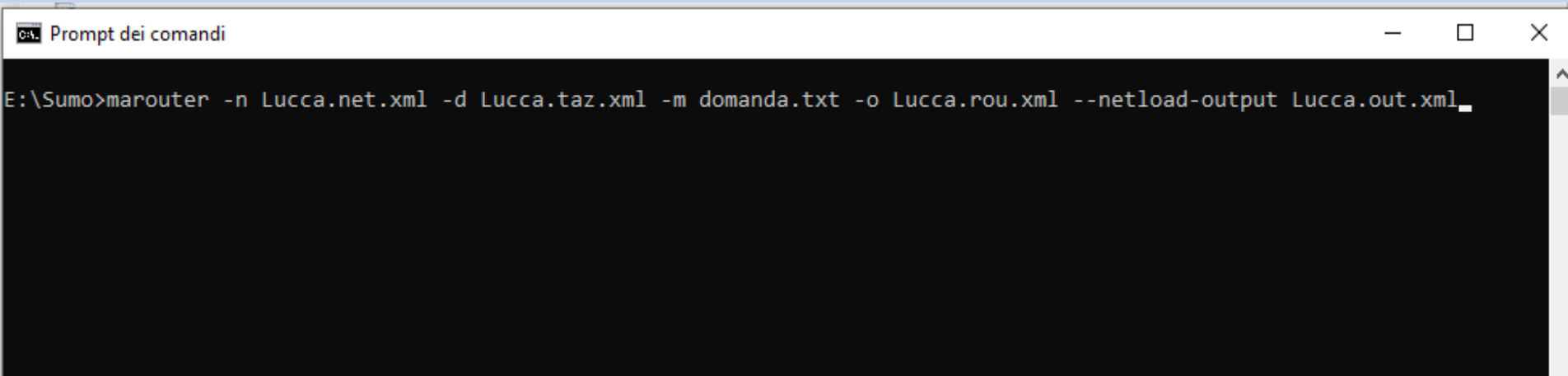
Fase 5b. Assegnazione di equilibrio stocastico mediante Marouter

In alternativa si può utilizzare il comando Marouter, che consente di effettuare una assegnazione di equilibrio stocastico.

Si scrive sul prompt dei comandi del dos:

```
marouter -n Lucca.net.xml -d Lucca.taz.xml -m domanda.txt -o  
Lucca.rou.xml --netload-output Lucca.out.xml
```

-n è il file di rete, -d il file delle taz; -m è il file di domanda -o il file delle route, output di marouter; --netload-output fornisce i parametri di traffico sugli archi della rete (flusso, densità, velocità)



```
cmd Prompt dei comandi  
E:\Sumo>marouter -n Lucca.net.xml -d Lucca.taz.xml -m domanda.txt -o Lucca.rou.xml --netload-output Lucca.out.xml
```

Marouter fornisce in output il file delle route (Lucca.rou.xml). Un estratto del file è il seguente:

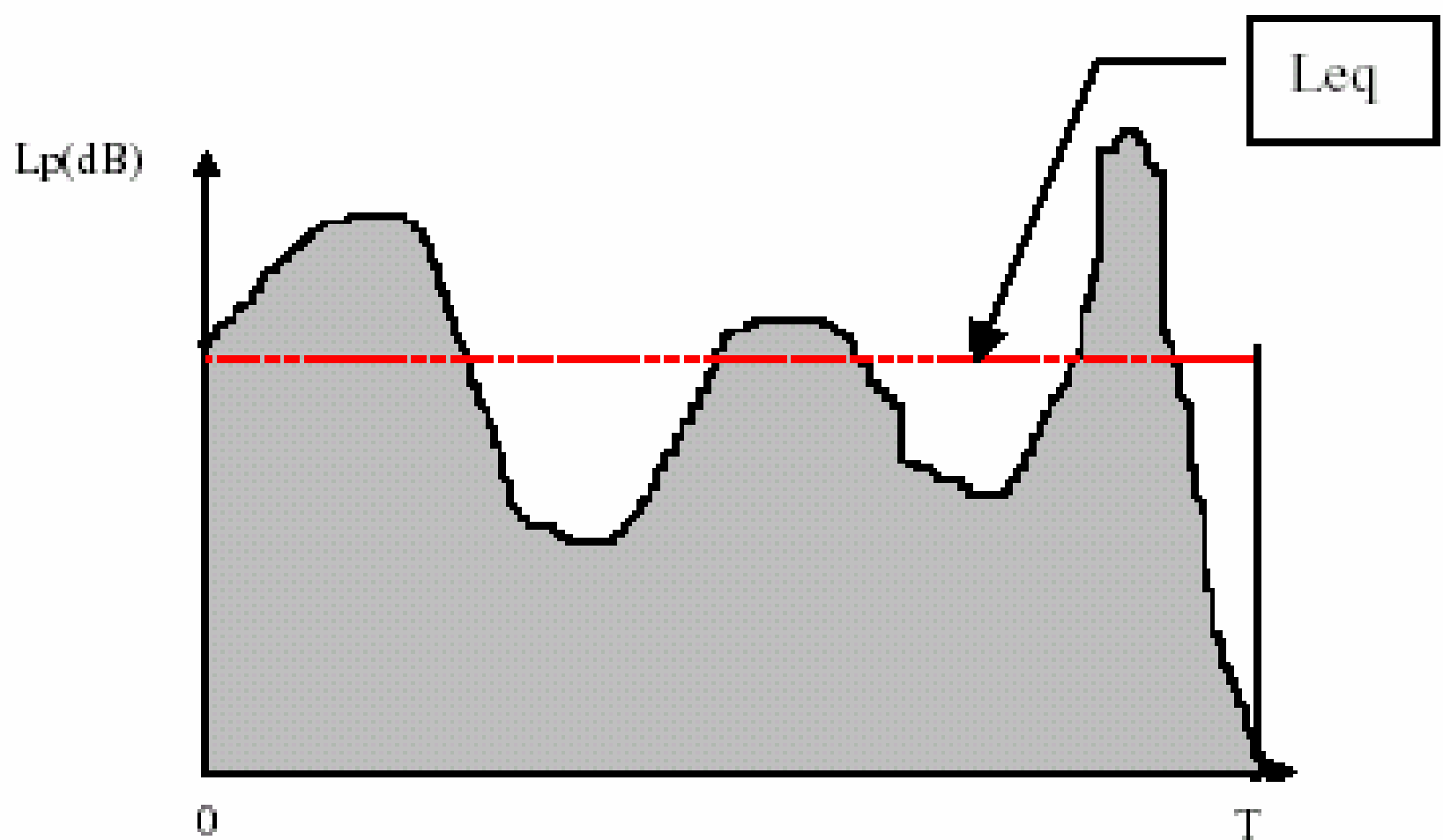
```
<flow id="4" begin="0.00" end="5400.00" number="3" fromTaz="1" toTaz="104"
departLane="free" departSpeed="max">
  <routeDistribution>
    <route cost="1383.86" probability="3.00000000" edges="247891972
247891973 247891968 247891974 421682019 421682018#0 421682018#1
421682018#2 166822766 164927792#0 166822763 -117221767#1 -117221767#0 -
421681433 -117221770 -421681429 -117222152 23939206 23939206-
AddedOffRampEdge 38506874 24359967-AddedOnRampEdge 24359967 24359960 -
421680786 -24359924 -421680785 -402114702 -127958737 -380388368 129447164
425496838 380390705#4 380390705#5 380390705#6 380390705#7 380390705#0
380390705#1 380390705#2 380388364 445006074 115006682#0 gneE15 gneE17
gneE18 gneE19 gneE20 gneE21 gneE22 gneE23 4045109#0.363 -364407383
364407384#0 297518614#0 297518614#1 297518614#2 511978479 493014849
493014848 368743152 493014089 495470609 510256602"/>
  </routeDistribution>
</flow>
```

Il file delle route riporta: id dell'itinerario, istante di inizio e fine del periodo di analisi, zone di origine e destinazione, tempo di percorrenza, n° di veicoli che percorrono la route, e id degli archi (edges) percorsi dalla route.

Fase 6. Output della simulazione

- **Dati di traffico** relativi ad ogni arco della rete: densità, occupazione, velocità, medie sull'intera simulazione;
- **Emissioni di inquinanti atmosferici** medie (sull'intera simulazione) su ogni arco della rete
- **Emissioni acustiche** medie: si fa riferimento al **livello sonoro equivalente (in dBA)**.

Il livello equivalente di un suono variabile in un dato periodo di tempo è quel livello di pressione sonora che avrebbe un suono costante con la medesima energia acustica, nello stesso intervallo di tempo, del suono variabile. Corrisponde, in pratica, ad una media energetica.



Fonte: A. Cocchi, Lucidi delle lezioni di Fisica Tecnica,
Università di Bologna

$$Leq = 10 \log \left[\frac{1}{T} \left(\int_0^T \frac{p_t^2}{p_0^2} dt \right) \right] \text{ dB}$$

p_t = pressione sonora misurata, variabile nel tempo

T = periodo di tempo di analisi

p_0 = pressione sonora di riferimento = $2 \cdot 10^{-5}$ Pa

Per ottenere i risultati della simulazione occorre redigere un **file di configurazione**. Un esempio è riportato qui di seguito:

```
<configuration>
  <input>
    <net-file value="Lucca.net.xml"/>
    <route-files value="Lucca.rou.xml"/>
    <additional-files value="example.add.xml"/>
  </input>
  <output>
    <summary-output value="results.out.xml"/>
    <emission-output value="emissions.out.xml"/>
  </output>
    <edgeData type="harmonoise"
file="acustico.out.xml"/>
  </output>
</output>
  <processing>
    <ignore-route-errors value="true"/>
  </processing>
</configuration>
```

Spiegazione del file di configurazione (solo i comandi significativi)

`<net-file value="Lucca.net.xml"/>` → File della rete

`<route-files value="Lucca.rou.xml"/>` → File delle route

`<additional-files value="example.add.xml"/>` → File “aggiuntivo”, che riporta quali dati di output voglio avere dalla simulazione (verrà spiegato di seguito)

`<summary-output value="results.out.xml"/>` → nome del file di output “principale”

`<emission-output value="emissions.out.xml"/>` → nome di un file di output in cui si riportano tutti i valori relativi alle emissioni.

`<edgeData type="harmonoise" file="rumore.out.xml"/>` → genera file di output che riporta i valori di emissione sonora, calcolati per ogni arco secondo la metodologia del **progetto Harmonoise** (che è attualmente un punto di riferimento per la stima delle emissioni sonore da traffico)

Esempio di file **aggiuntale**:

```
<additional>
```

```
  <edgeData id="all" type="emissions" file="inquinanti.out.xml"/>
```

```
  <edgeData id="all" type="harmonoise" file="rumore.out.xml"/>
```

```
  <edgeData id="all" file="congestione_archi.out.xml"/>
```

```
</additional>
```

edgeData id= specifico su quali archi voglio avere informazioni

Type= specifico il tipo di informazioni che desidero avere: emissions sono le emissioni atmosferiche, harmonoise quelle acustiche; di default invece ho i dati di traffico su ogni arco.

file= specifico il nome del file su cui riportare le informazioni richieste.

Esempio di output di simulazione - Dati di traffico su ogni arco (estratto)

	begin	end	id	entered	left	traveltime	density	occupancy	waitingTime	speed
8	0	2361	-156395855#2	19	18	3,37	0,82	0,37	0	11,94
9	0	2361	-156395855#3	19	21	1,12	1,07	0,4	0	11,34
10	0	2361	-156395855#4	34	34	1,97	1,71	0,71	0	11,09
11	0	2361	-156395855#5	33	34	2,5	1,42	0,6	1	11,76
12	0	2361	-156395855#6	33	34	7,06	1,23	0,58	0	12,64
13	0	2361	-156395860	5	3	0,94	0,56	0,15	0	6,89
14	0	2361	-156395866#0	86	86	22,28	14,7	7,16	1224	2,75
15	0	2361	-156395866#1	87	86	11,33	4,11	1,98	190	9,69

- begin, end: istanti di inizio e fine simulazione (in secondi)
- id: identificativo dell'arco
- entered, left: n° veicoli che sono entrati/usciti dall'arco durante l'intera simulazione
- travel time, waiting time: tempo di percorrenza, o di attesa, medio in secondi
- density: densità media in veicoli/km
- speed: velocità di percorrenza media dell'arco in m/s
- occupancy: occupazione dell'arco: ossia la quantità di tempo sul totale della simulazione, in cui sull'arco si trova almeno un veicolo

Esempio di output di simulazione - Emissioni atmosferiche (estratto)

	begin	end	id	CO_abs	CO2_abs	HC_abs	PMx_abs	NOx_abs	fuel_abs	electricity_abs
2	0	1152	-120694318	951,219778	50056,7919	5,510298	0,965901	21,151582	21,517287	0
3	0	1152	-120694328	1023,98297	52131,1121	5,894379	0,995765	21,921779	22,408998	0
4	0	1152	-120694330#0	624,597438	38141,4183	3,752304	0,744482	16,130869	16,39528	0
5	0	1152	-120694330#1	1954,32684	106547,048	11,400832	1,978402	44,32207	45,800126	0
6	0	1152	-129773489	200,948815	21387,646	1,415217	0,330876	8,282854	9,193613	0
7	0	1152	-156395845	2493,18717	142897,269	14,742792	2,737975	59,967426	61,425333	0
8	0	1152	-156395848#0	5889,37459	392794,552	36,06123	6,988848	160,487903	168,845413	0
9	0	1152	-156395848#1	975,376867	81148,2068	6,356573	1,405674	32,735895	34,88198	0
10	0	1152	-156395850#0	25024,7372	537522,826	126,632808	12,492936	240,673929	231,067671	0

CO_normed	CO2_normed	HC_normed	PMx_normed	NOx_normed	fuel_normed	electricity_normed
22,005936	1158,035791	0,127478	0,022346	0,48933	0,49779	0
28,136347	1432,425264	0,161962	0,027361	0,602353	0,61574	0
49,364365	3014,464646	0,296559	0,058839	1,274885	1,295783	0
46,688108	2545,367527	0,272361	0,047263	1,058837	1,094147	0
7,367887	784,188594	0,05189	0,012132	0,303695	0,337088	0
37,841614	2168,895844	0,223766	0,041557	0,910186	0,932314	0
103,070652	6874,344614	0,631112	0,122313	2,808718	2,954984	0
16,994997	1413,928889	0,110757	0,024493	0,570391	0,607785	0
1582,401938	33989,4543	8,007437	0,789972	15,218657	14,61122	0

Esempio di output di simulazione - Emissioni atmosferiche (estratto)

Output della simulazione relativo alle emissioni atmosferiche per ciascun arco della rete.

Siccome la schermata è molto ampia è stata riportata su due righe.

- begin = istante di inizio simulazione
- end = istante di fine simulazione
- id = identificativo dell'arco
- CO_abs, CO2_abs, HC_abs, PMx_abs, NOx_abs (mg): milligrammi totali di inquinanti emessi sul dato arco durante l'intera simulazione
- fuel_abs (ml), electricity_abs (Wh): consumo totale di carburante (in ml) o elettricità (in Wh) sul dato arco durante l'intera simulazione
- CO_normed, CO2_normed, HC_normed, PMx_normed, NOx_normed (g/km/h): emissioni medie di inquinanti su un dato arco, in g/km/h
- fuel_normed (l/km/h), electricity_normed (W/km): consumo medio di carburante, di energia elettrica, sul dato arco.

Esempio di output di simulazione - Emissioni acustiche (estratto)

	A	B	C	D
1	begin	end	id	noise
2	0	1152	-120694318	47,75
3	0	1152	-120694328	46,81
4	0	1152	-120694330#0	47,74
5	0	1152	-120694330#1	50,47
6	0	1152	-129773489	45,32
7	0	1152	-156395845	52,16
8	0	1152	-156395848#0	56,9
9	0	1152	-156395848#1	50,06
10	0	1152	-156395850#0	54,33
11	0	1152	-156395850#1	45,46
12	0	1152	-156395855#0	53,36
13	0	1152	-156395855#1	49,33
14	0	1152	-156395855#0	50,00

Esempio di output di simulazione - Emissioni acustiche (estratto)

Output della simulazione per ogni arco relativo alle emissioni acustiche:

- begin: istante di inizio simulazione
- end: istante di fine simulazione
- id: identificativo arco
- noise (dBA): si considera il **valore equivalente**, (ossia il livello sonoro che, costante nel tempo, possiede lo **stesso contenuto energetico** del livello sonoro variabile nel tempo)

Le emissioni acustiche sono state calcolate mediante il **modello di emissione di Harmonoise**, implementato in SUMO.

SUMO fornisce le **emissioni acustiche arco per arco**. Per confrontare i valori forniti da SUMO con i valori limite da normativa, in un tronco stradale a doppio senso di circolazione, occorre sommare (somma logaritmica) le emissioni acustiche dei due archi nelle due direzioni.