

# MATSim: Parameters

```
<!-- ===== -->

<module name="network" >
  <param name="inputChangeEventsFile" value="null" />
  <param name="inputNetworkFile" value="&INBASE;/network.xml" />
  <param name="outputChangeEventsFile" value="null" />
  <param name="outputNetworkFile" value="&OUTBASE;/output_network.xml" />
  <param name="timeVariantNetwork" value="false" />
</module>

<!-- ===== -->

<module name="plans" >
  <param name="inputPlansFile" value="&INBASE;/plans.xml.gz" />
  <param name="outputPlansFile" value="&OUTBASE;/output_plans.xml.gz" />
  <param name="outputSample" value="1.0" />
  <param name="switchOffPlansStreaming" value="yes" />
</module>

<!-- ===== -->
```

*org.matsim.core.config.groups* (or output\_config)

22 modules

\* x parameters/module

= lots of parameters

Minimal configuration

+ frequent extensions

src/playground/anhorni/berlin/config.xml

(svn update)



module authors  
run experts

1. Minimal configuration
3. Some more scoring parameters
4. Using traffic count data
5. Using facilities
6. Making a sample run
7. Speed-up (multi-processor machine)
  - Replanning threads
  - Parallel event handling
  - JDEQSim
8. Calibration of Berlin scenario & Questions

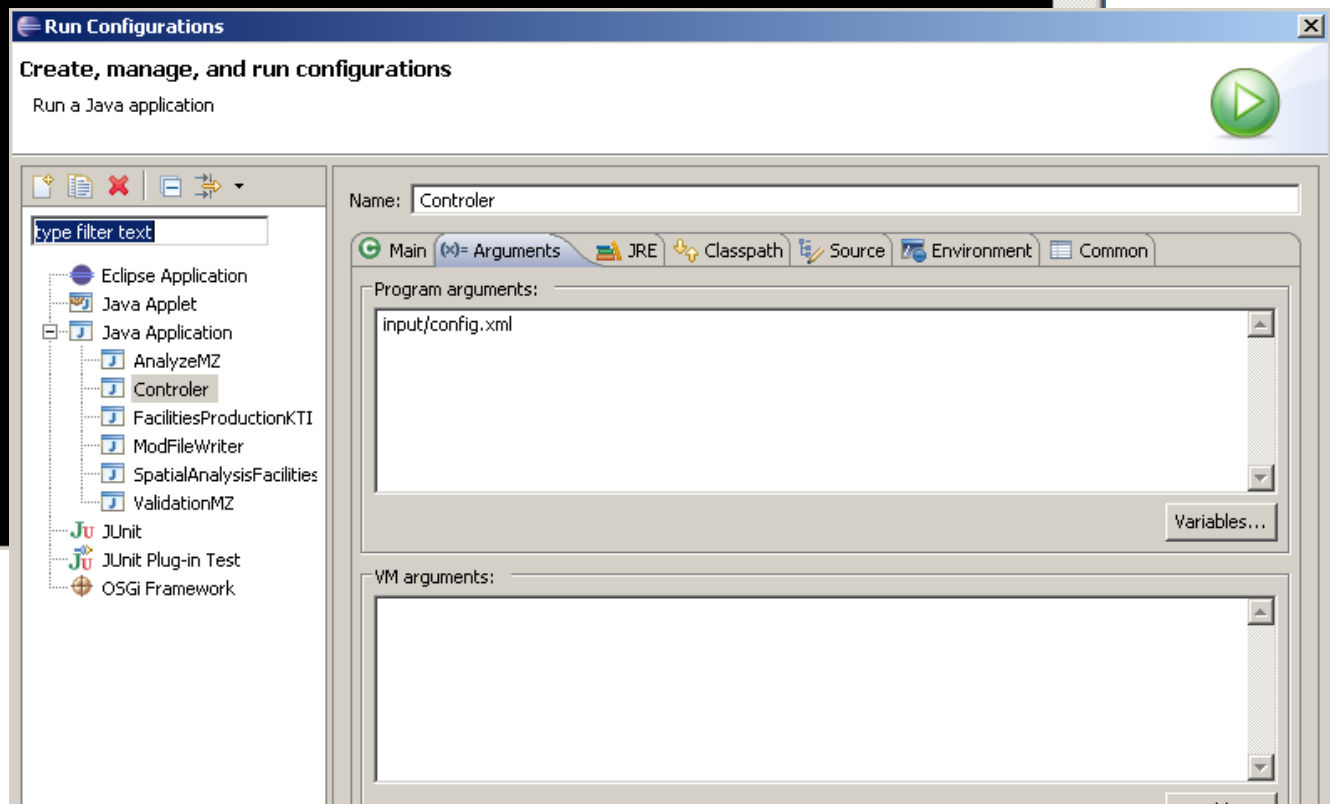
```
anhorni@satawal: ~
Using username "anhorni".
Authenticating with public key "labuse-ivt" from agent
Linux satawal 2.6.21-amd64-k8-smp #1 SMP Fri Jul 6 18:20:03 CEST 2007 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 14 11:11:20 2009 from surcouf.ethz.ch
anhorni@satawal:~$
```

make clean ; nice -n 10 make run CONFIG="input/config.xml"

"MEMORY=-Xms18g -Xmx18g" "MAINCLASS=../Controler" &> output/run.log



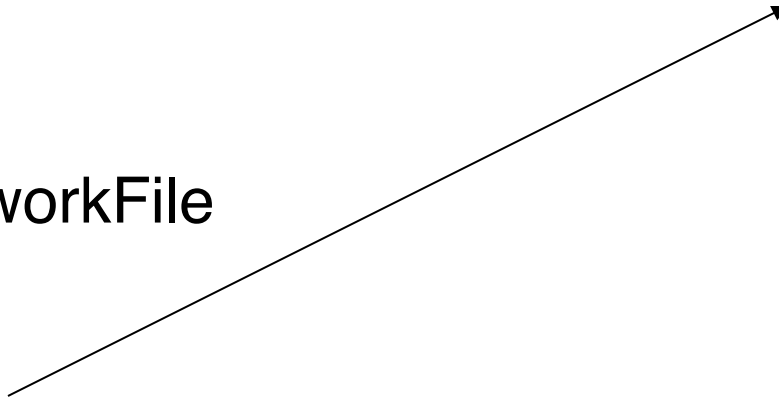
```
<?xml ... ?>  
<!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v1.dtd"  
...  
>
```

```
<config>  
  <module name="..." >  
    <param name="..." value="..." />  
    <param ...  
    ...  
  </module>  
</config>
```

```
<!ELEMENT      config (configincludel  
module)*>  
<!ATTLIST     config  
              xml:lang NMTOKEN "de-CH">  
<!ELEMENT     module (param)*>  
<!ATTLIST     module  
              name CDATA #REQUIRED>  
<!ELEMENT     param EMPTY>  
<!ATTLIST     param name CDATA #REQUIRED  
                  value CDATA #REQUIRED>
```

# 1. Minimal Configuration

- plans
  - inputPlansFile
- network
  - inputNetworkFile
- strategy
  - Module\_x
  - ModuleProbability\_x



E.g.:

- KeepLastSelected
- ReRoute
- ReRoute\_Dijkstra
- ReRoute\_Landmarks
- TimeAllocationMutator
- ExternalModule
- Planomat
- SelectExpBeta
- SelectRandom
- ChangeLegMode
- LocationChoice

- planCalcScore

- activityType\_x

Same types as in the plans file

- activityPriority\_x

t\*

- activityTypicalDuration\_x

E.g. shop different minimal duration than work

- activityMinimalDuration\_x

- activityLatestStartTime\_x

- activityEarliestEndTime\_x

E.g. different typical leisure durations (e.g. visiting friends vs. jogging)

- activityOpeningTime\_x

- activityClosingTime\_x

Project „Westumfahrung“:

101 activityTypes!

e.g.: w4, w8, ..., s1, s2, ...

- planCalcScore

- activityType\_x
- activityPriority\_x
- activityTypicalDuration\_x
- activityMinimalDuration\_x
- activityLatestStartTime\_x
- activityEarliestEndTime\_x

**plans**

person desires

- activityOpeningTime\_x
- activityClosingTime\_x

**facilities** open times

## 2. Some More Scoring Parameters

- planCalcScore

$$U_{plan} = \sum_{i=1}^n U_{act,i} + \sum_{i=2}^n U_{trav,i-1,i}$$

$$U_{act,i} = U_{dur,i} + U_{wait,i} + U_{late.ar,i} + U_{early.dp,i} + U_{short.dur,i}$$

performing

waiting

lateArrival

earlyDeparture

$$U_{trav,i-1,i} = U_{traveldur,i-1,i} + U_{traveldist,i-1,i}$$

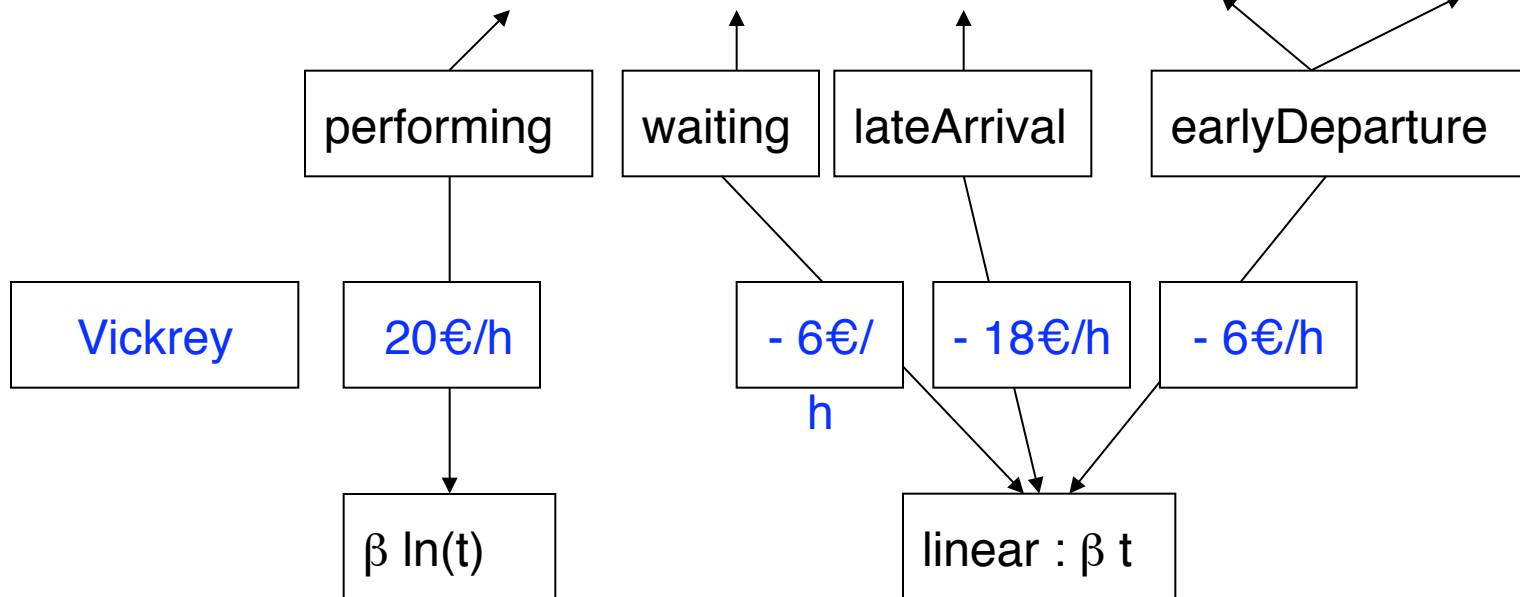
travel{... | Pt | Walk}

marginalUtilOfDistance{Car | Pt  
| Walk}

## 2. Some More Scoring Parameters

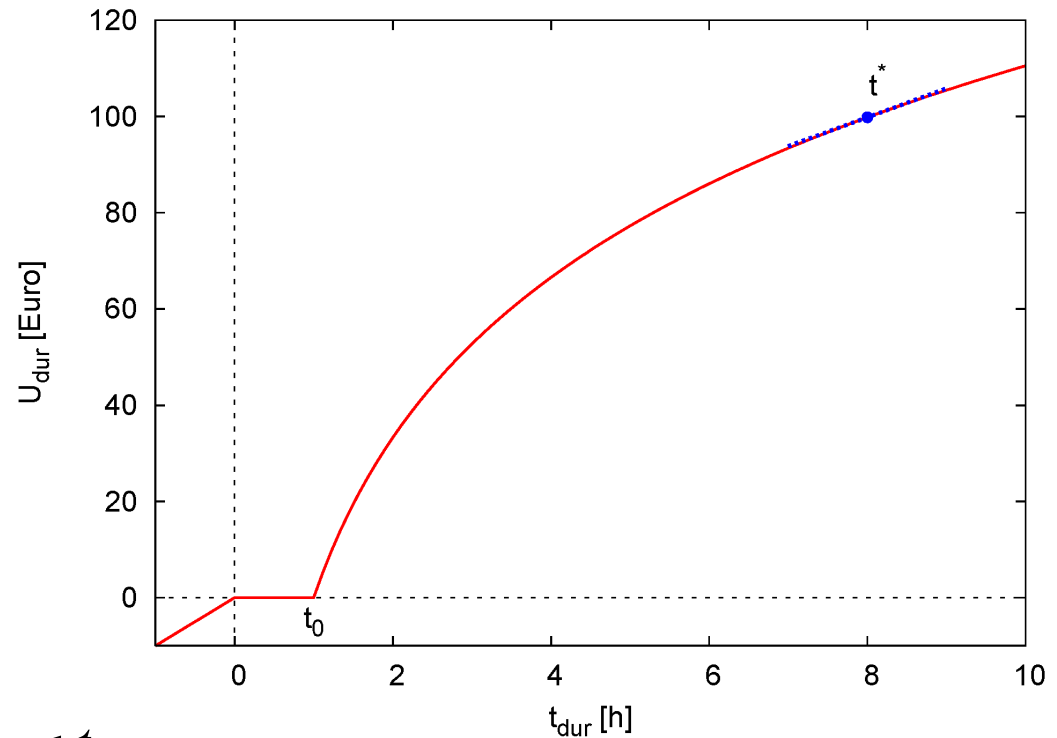
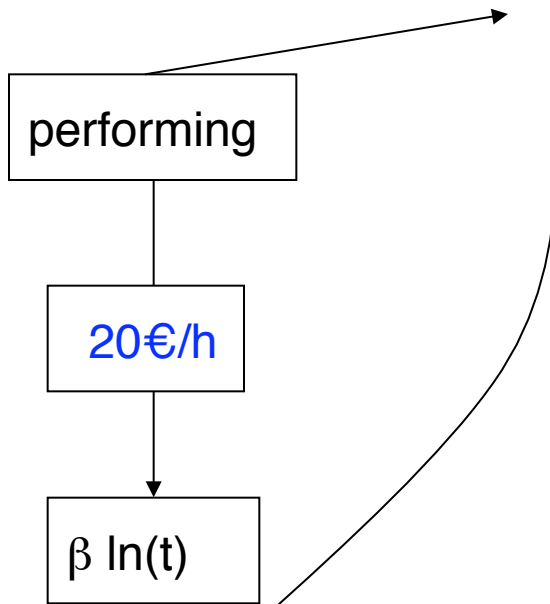
- planCalcScore

$$U_{act,i} = U_{dur,i} + U_{wait,i} + U_{late.ar,i} + U_{early.dp,i} + U_{short.dur,i}$$



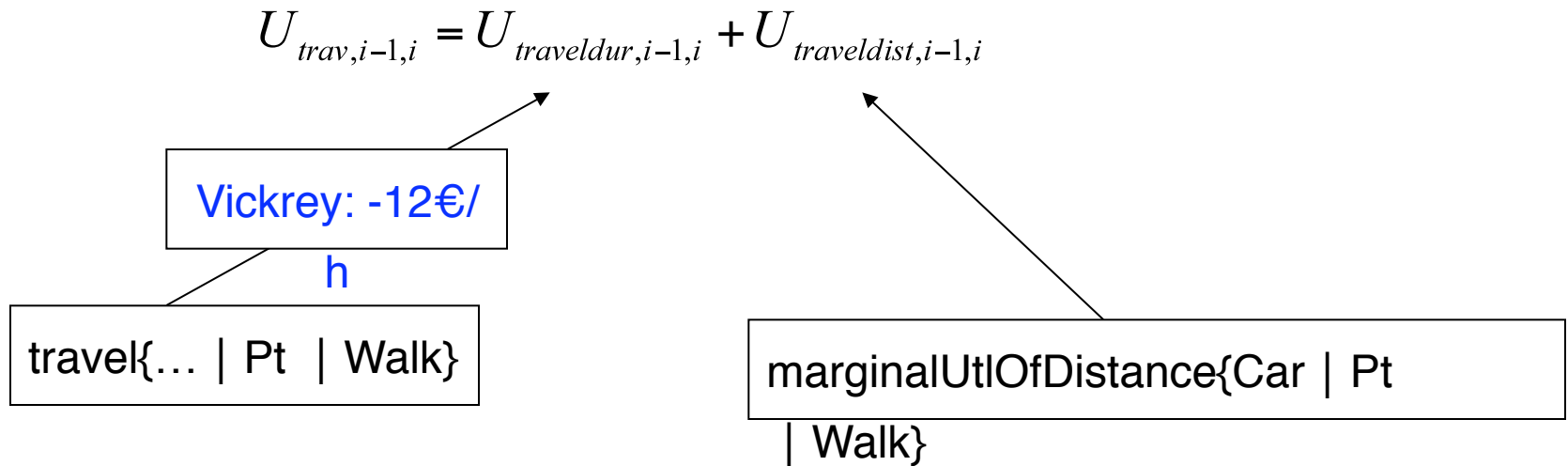
- planCalcScore

$$U_{act,i} = U_{dur,i} + U_{wait,i} + U_{late.ar,i} + U_{early.dp,i} + U_{short.dur,i}$$



$$U_{dur,i} = \begin{cases} \beta_{dur} \cdot t^* \cdot \ln(t_{dur,i} / t_{0,i}) & \text{if } t_{0,i} \leq t_{dur,i} \\ 0 & \text{if } 0 \leq t_{dur,i} < t_{0,i} \\ \alpha \cdot t_{dur,i} & \text{else } (\alpha > 0) \end{cases}$$

- planCalcScore



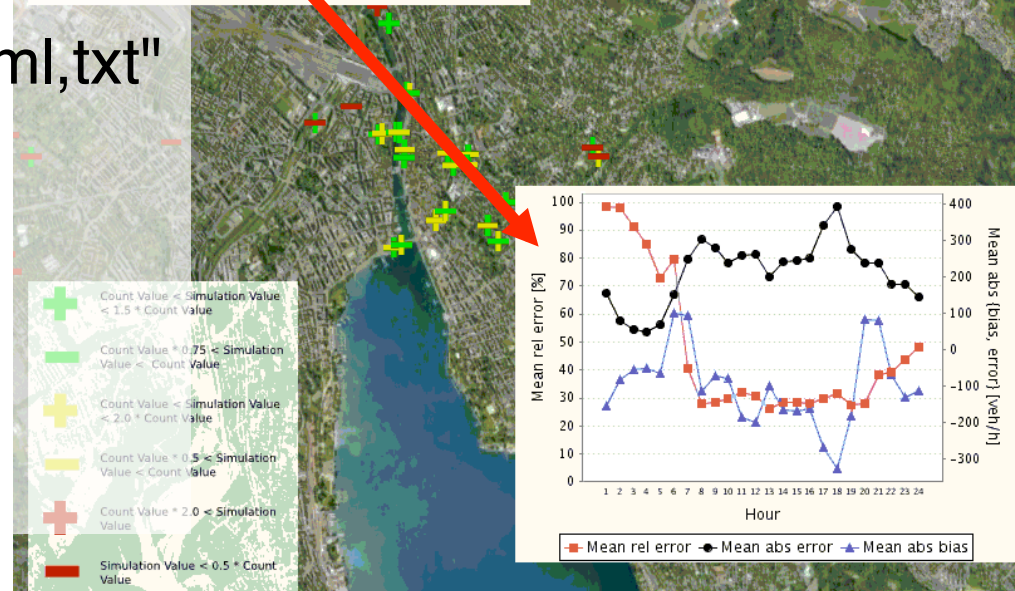
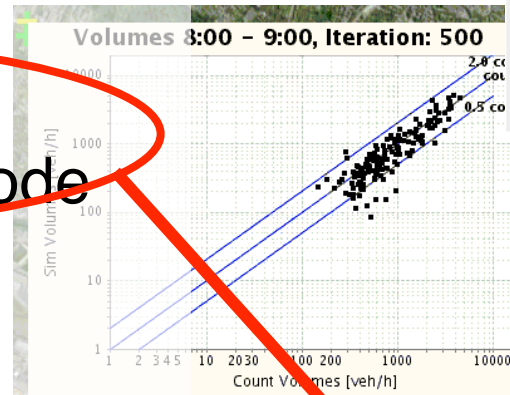
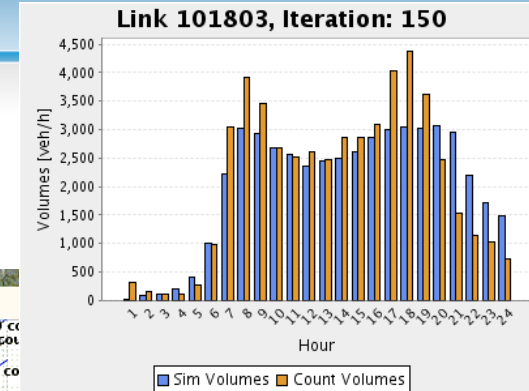
- `planCalcScore`
  - `learningRate`

$$\text{score} := \text{learningRate} * \text{new score} + (1 - \text{learningRate}) * \text{old score}$$

damping

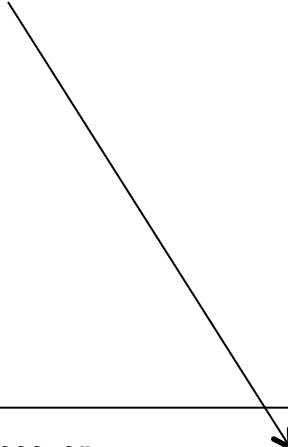
# 3. Using Count Data

- counts
  - inputCountsFile
  - distanceFilter
  - distanceFilterCenterNode
  - outputformat value="kml,txt"
  - countsScaleFactor



- facilities
  - inputFacilitiesFile

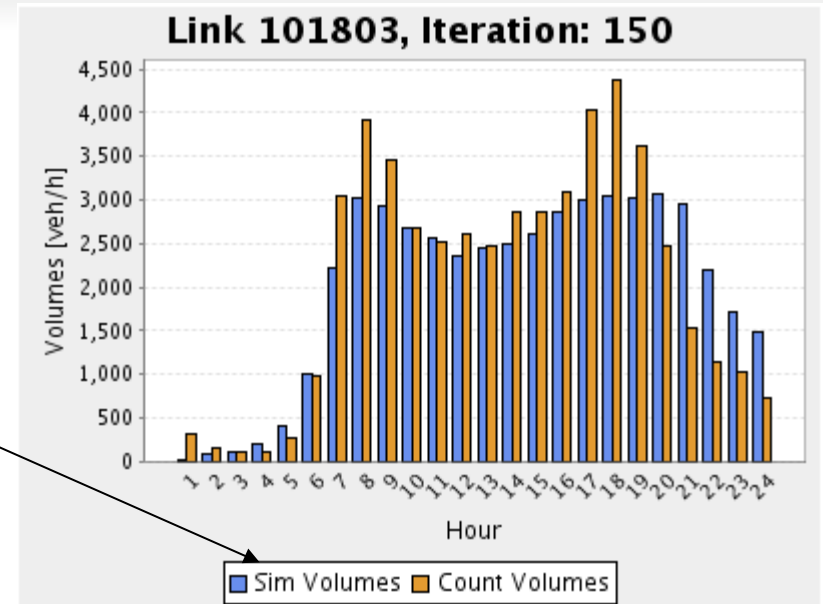
Example:



```
<facility id="1" x="486300.0" y="111000.0">  
  <activity type="shop">  
    <capacity value="150" />  
    <opentime day="wkday" start_time="09:00:00" end_time="18:30:00" />  
    <opentime day="sat" start_time="08:00:00" end_time="13:00:00" />  
  </activity>  
  <activity type="work_sector3">  
    <capacity value="10" />  
    <opentime day="wkday" start_time="08:00:00" end_time="19:30:00" />  
    <opentime day="sat" start_time="07:30:00" end_time="15:00:00" />  
  </activity>  
</facility>
```

xml

- counts
  - countsScaleFactor



- simulation
  - storageCapacityFactor
  - flowCapacityFactor

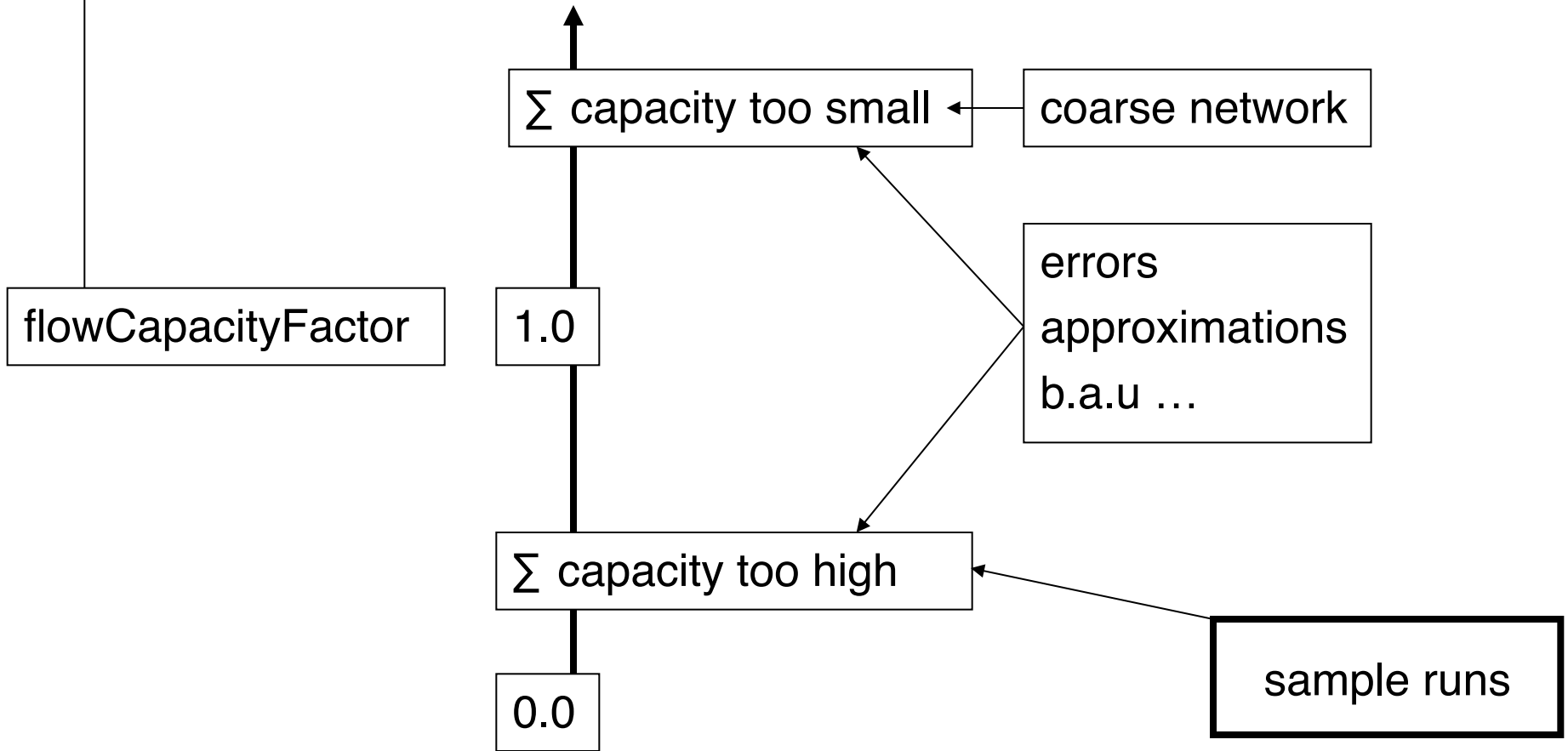
Calibration of Mobsim

Different data!

# 5. Making a Sample Run

network:

```
<link id="1" from="9" to="17" length="540.0" freespeed="12.5"  
capacity="13200.0" ... />
```



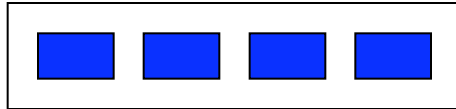
# 5. Making a Sample Run

network:

```
<link id="1" from="9" to="17" length="540.0" freespeed="12.5"  
capacity="13200.0" ... />
```

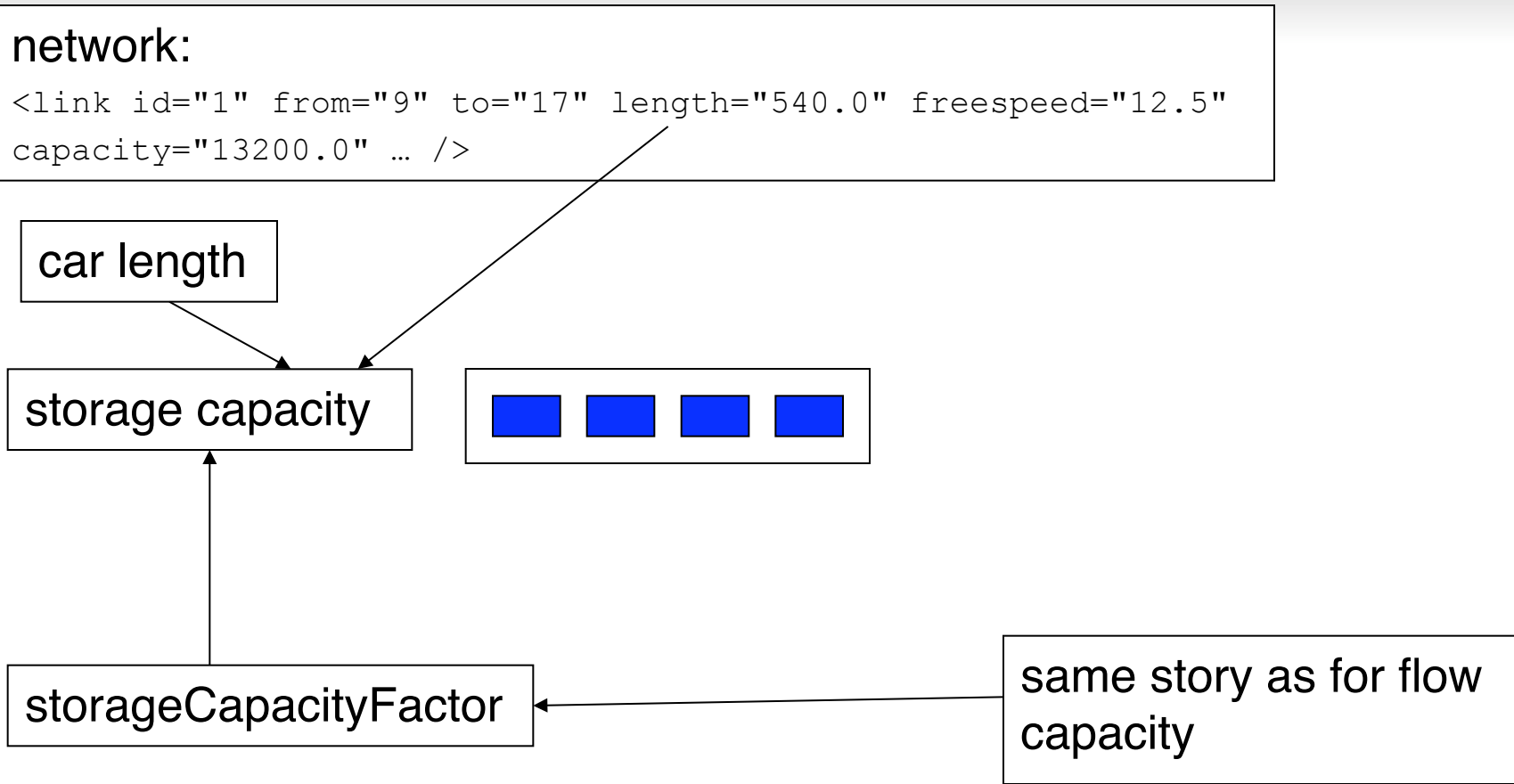
car length

storage capacity

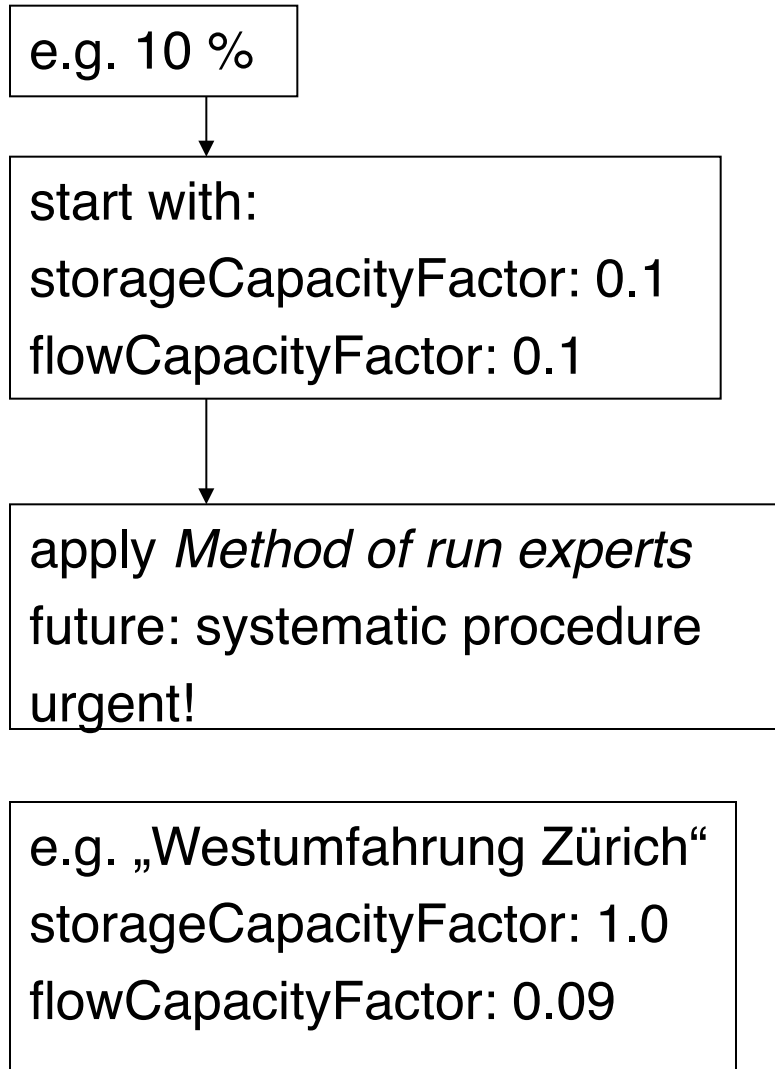


storageCapacityFactor

same story as for flow capacity



# 5. Making a Sample Run



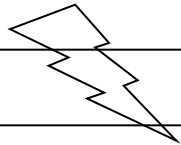
Pitfall for small sample runs  
(e.g. 1 %)

e.g.  
link length = 100 m  
car size = 3.5 m  
storage capacity link  $\approx$  20 cars  
( $\neq$  storageCapacityFactor)

1% sample:  
storage capacity = 0.2

min (storageCapacity) = 1.0

does not scale linearly!



- global
  - numberOfThreads (→ replanning threads)
- parallelEventHandling (+ JDEQSim, next slide)
  - numberOfThreads (mandatory)
  - estimatedNumberOfEvents

Documentation: <http://matsim.org/node/238>

## 6. Speed-up

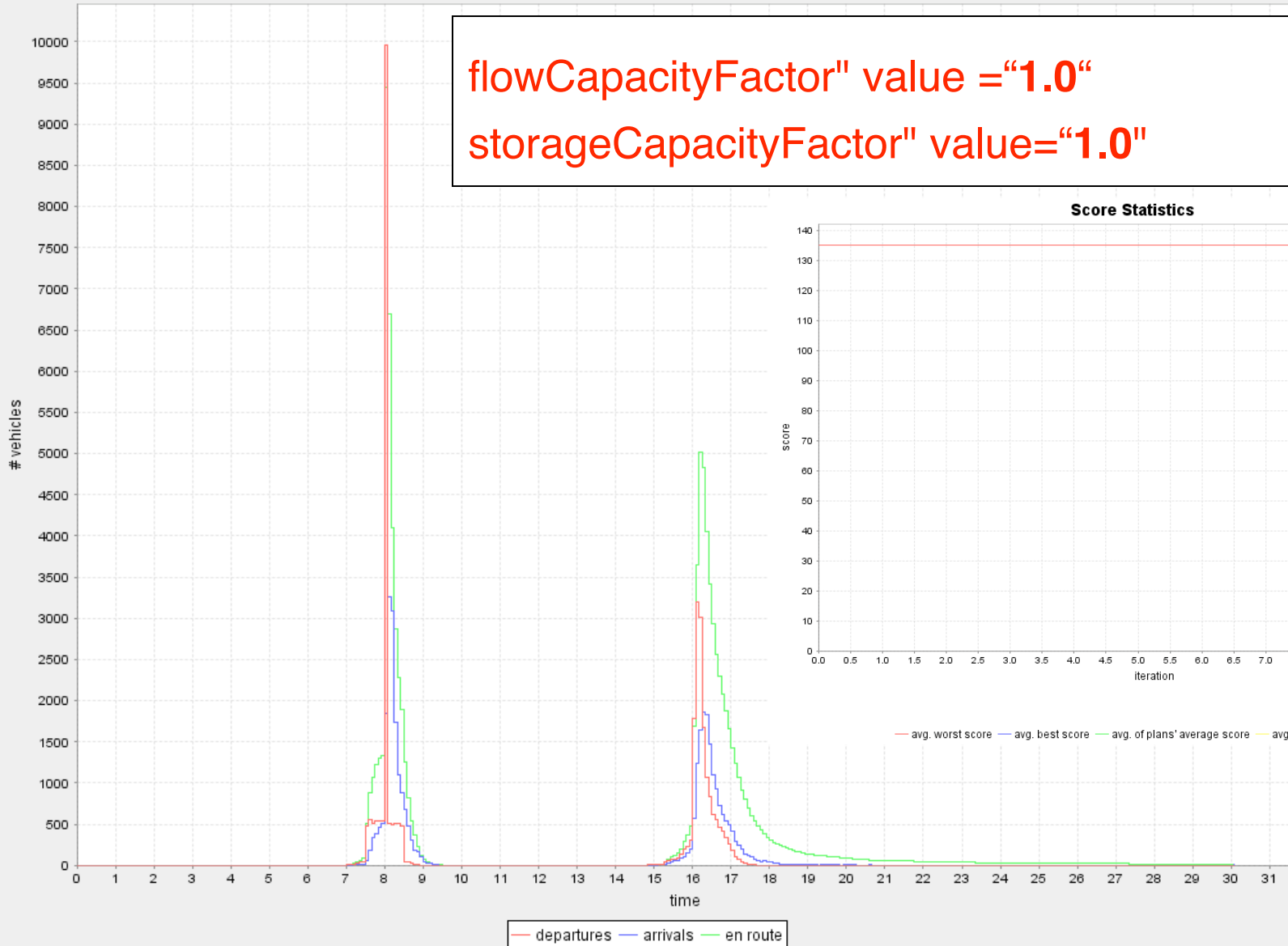
- JDEQSim
  - flowCapacityFactor
  - storageCapacityFactor
  - gapTravelSpeed
- Documentation: *<http://matsim.org/node/249>*

# 7. Calibration of Berlin Scenario

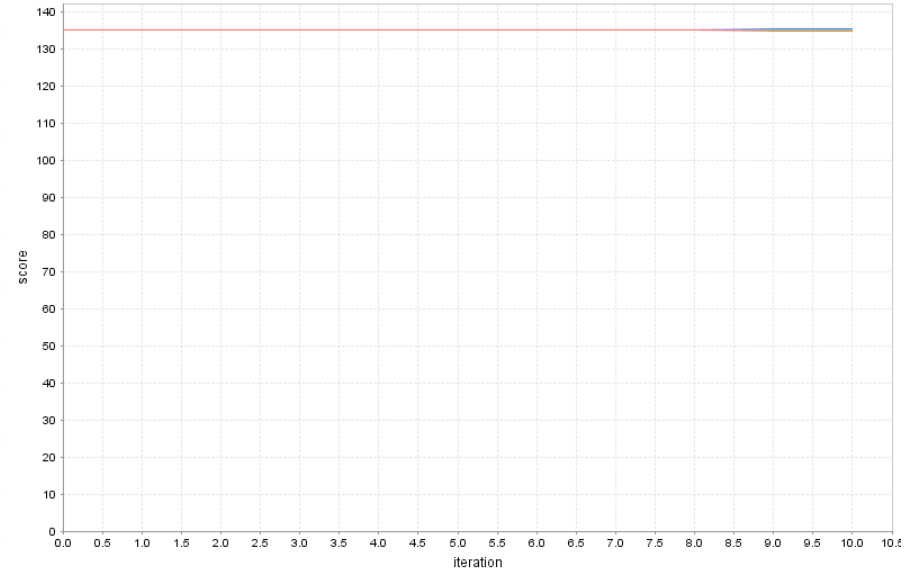
1 % sample!

22

Leg Histogram, all, it.10

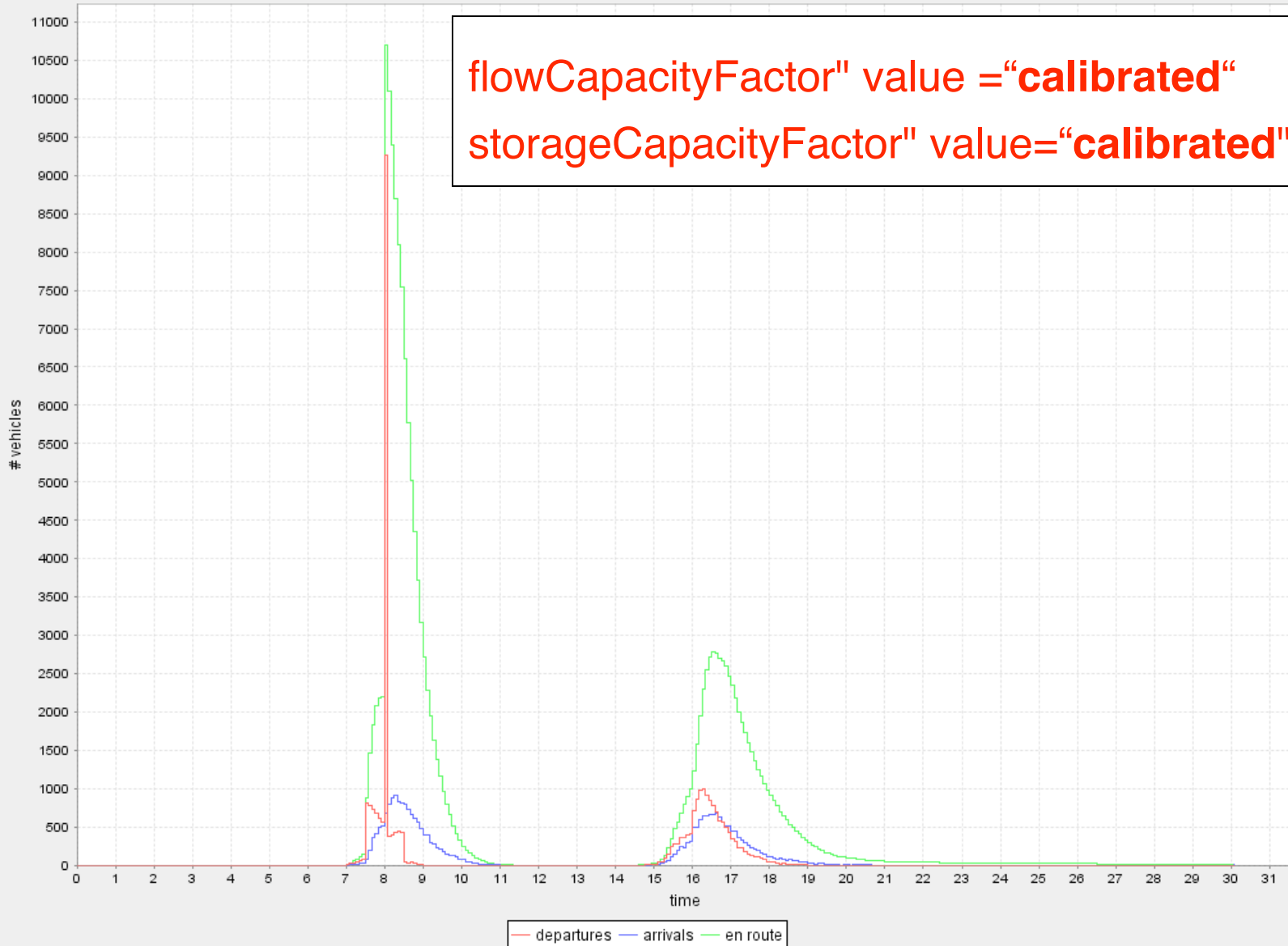


Score Statistics



MATSim

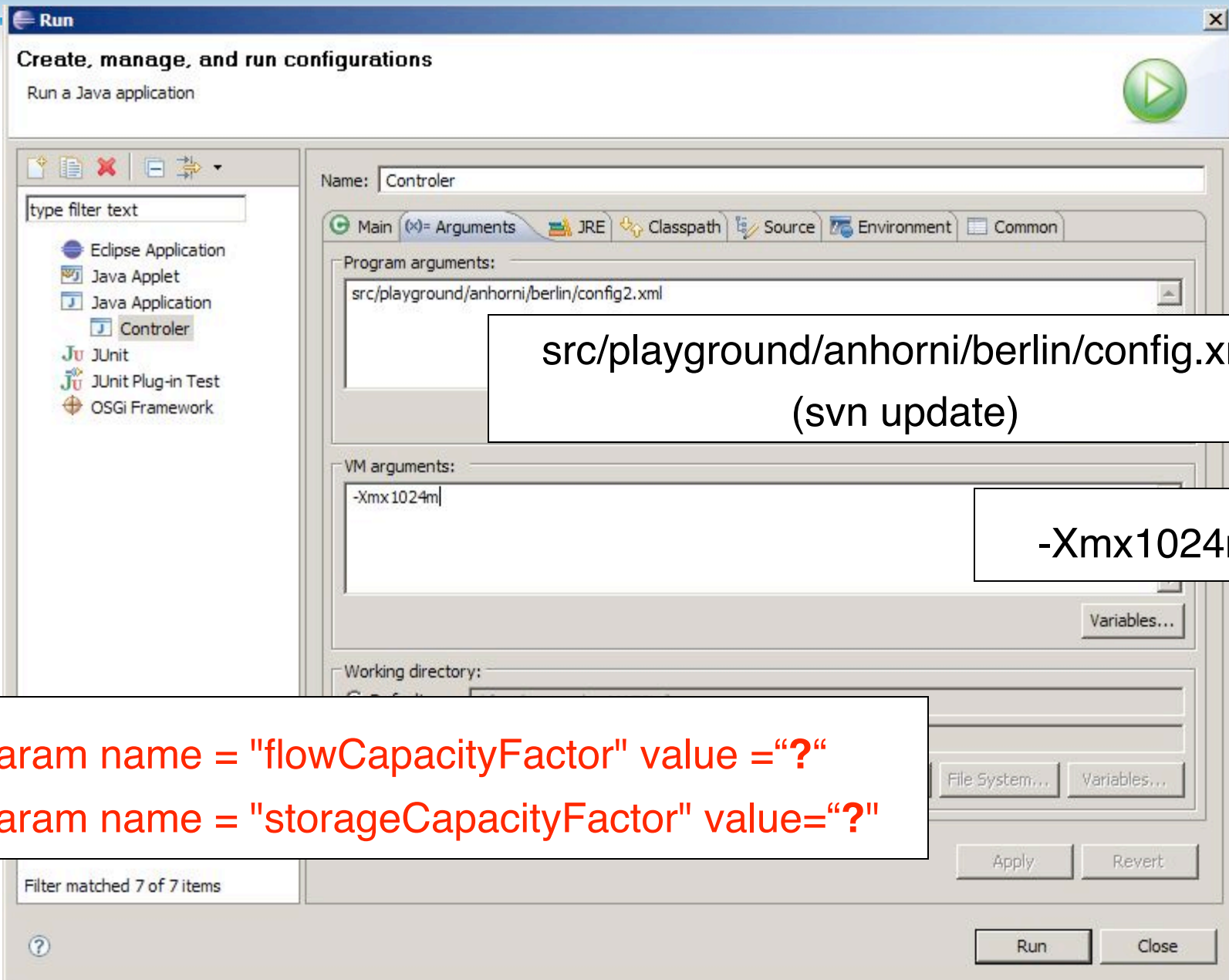
Leg Histogram, all, it.10



# 7. Calibration of Berlin Scenario

1 % sample!

24



src/playground/anhorni/berlin/config.xml  
(svn update)

-Xmx1024m

<param name = "flowCapacityFactor" value = "?"  
<param name = "storageCapacityFactor" value = "?"

- controler (M. Rieser / D. Grether)
- global (M. Rieser / M. Balmer)
- network (M. Rieser / M. Balmer)
- plans (M. Rieser / M. Balmer)
- planCalcScore (K. Nagel)
- strategy (M. Rieser)
- simulation (M. Rieser / D. Strippgen)
- counts (A. Horni / D. Grether)
- facilities (K. Meister / M. Balmer)
- JDEQSim (R. Waraich)
- parallelEventHandling (R. Waraich)