广州市城市规划勘测设计研究院
GUANGZHOU URBAN PLANNING & DESIGN SURVEY RESEARCH INSTITUTE

An Auto Calibration Extension for Plan Parameters in MATSim

Ming Lü, Zhang Xiaoming

06.2024

CONTENT

- purpose & goal
    - why develop this extension
    - goal: parameter adaptation according to different scales
- framework & structure
    - Module framework, related mechanism and pipelines
- CMA-ES algorithm
    - How CMA-ES works and its formulation
- scenario test
    - parameter-estimation results
    - proof that it is auto-adaptive to different population scales
- conclusions and future work

purpose & goal

It is easy to simulate a city using commuting trips.
It is NOT easy to simulate a city scenario with uncertainties.

Questions:
- For irregular/random/less frequent trips made in a city, how many percentages respectively are contributing to the transportation system, in terms of total kilometers traveled per day?
- What kinda data do we need to get these numbers? either from a survey or from LBS data
- are these uncertain trips they taken into considerations for travellers’ choices?

GOAL :
With limited data at hand(at least some trip diary data), try to estimate parameters for travel choices automatically.

purpose & goal

When calibrating a scenario there are three things to do:

config parameters

- running config parameters such as random seed, threads to make it running more efficient
- Hyper parameters setting up by experience
- designated parameters such as coordination system ids, output format
- running parameters that have influence on travel choices, mainly works on planCalcScore module

revise inputs esp. plans

- use data mining technique to get more precise daily plans from a massive data source
- conduct a detailed RP & SP combined survey to get a better understanding
- combine multi-source trips, network files, transit data etc.
- scale link capacity and storage factor for fast simulations

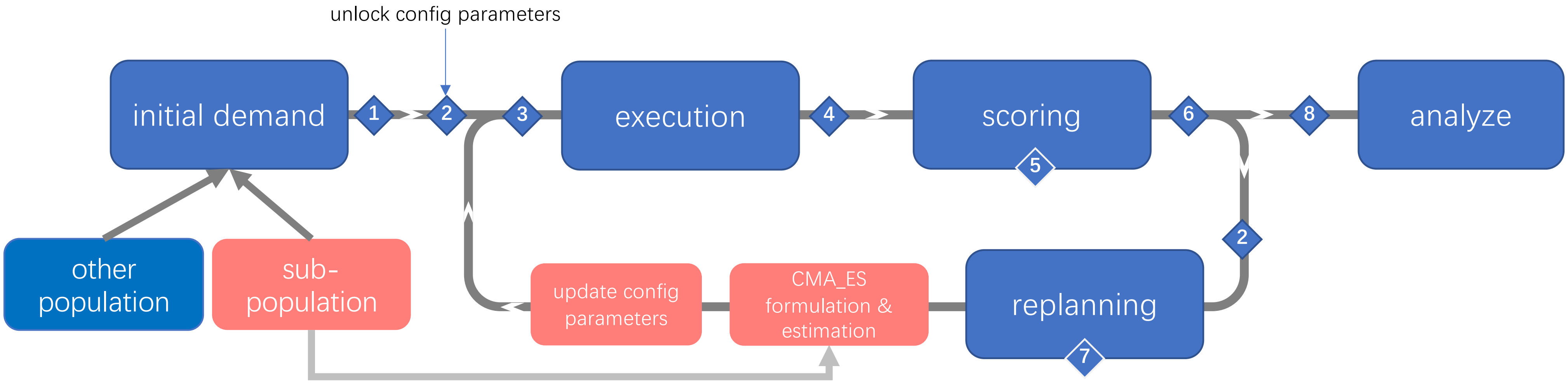
Algorithms interact with actual data

- CADYTS and other algorithm to improve routing or other choices regarding to counts data
- Simple strategy to make designated link volume towards ground truth data

CONTENT

- purpose & goal
    - why develop this extension
    - goal: parameter adaptation according to different scales
- framework & structure
    - Module framework, related mechanism and pipelines
- CMA-ES algorithm
    - How CMA-ES works and its formulation
- scenario test
    - Parameter-estimation results
    - proof that it is auto-adaptive to different population scales
- conclusion and future work

Framework of the extension



ground truth decisions as expected maximum utilities among alternatives

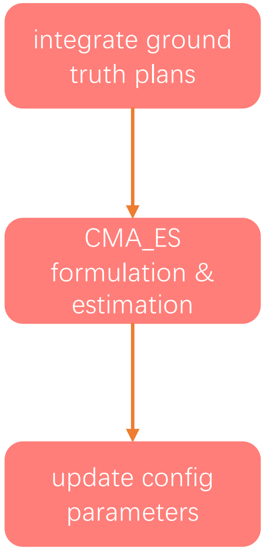
sub-population -at least have some ground truth data to calibrate for

update config parameters -parameters will be updated and used for next iteration

CMA_ES formulation & estimation -a logit like maximum log-likelihood is used as goal function

Controller Events: 1. simulation starts 2. Iteration Starts 3. Before Mobsim 4. After Mobsim
5. Scoring 6. Iteration Ends 7. Replanning 8. Simulation Ends(Shutdown)

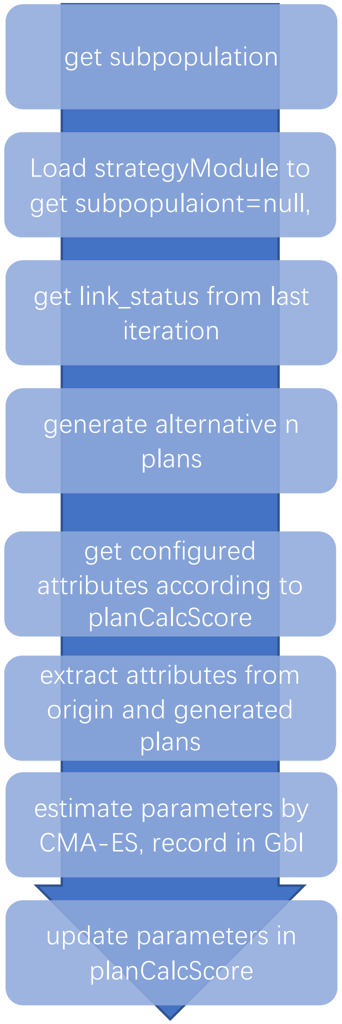
Extension structure – pipe line



integrate ground truth plans

CMA_ES formulation & estimation

update config parameters



get subpopulation

get the persons with subpopulation attribute equals designated string, and in strategy configs, strategyName is set to KeepLastSelected so these plans are set as ground truth and used for calibration.

Load strategyModule to get subpopulaiont=null,

Get the strategy update configs for population other than ground truth persons, for the moment only get those with subpopulation=designed for simplification reason.

get link_status from last iteration

~~Check how it records the link volumes and how it is used in the routings, integrate CADYTS to update routing selecting as well.~~

generate alternative n plans

According so strategy such as SelectRandom BestScore KeepLastSelected ChangeExpBeta SelectExpBeta SelectPathSizeLogit for subpopulation=null, generate groundtruth-subpopulation,
SelectPathSizeLogit might have potential problems as groundtruth-subpopulation only have one plan

get configured attributes according to planCalcScore

Check in planCalcScore config module, find out which attributes are set up, customized, or used as default

extract attributes from origin and generated plans

Check in planCalcScore config module, find out which attributes are set up, customized, or used as default. Mapping these attribute variables from groundtruth-subpopulation with parameters and form maximum likelihood functions(-log sum of selected utlilityfunctions)

estimate parameters by CMA-ES, record in Gbl

Estimate parameters using Covariance Matrix Adaptation Evolution Strategy till reach convergence or designated iteration

update parameters in planCalcScore

Update all parameters for customized, o in planCalcScore

config of the extension

1.add override module:planParamCalibration

```
<module name=”planParamCalibration" >
    <!-- fixed subpopulation addresses initial plans’ subpopulation attribute. This module uses these subpopulations’ selected plan to adjust some act and trip parameters in planCalcScore module -->
    <param name="fixedSubPopulation" value="surveyPeople" />
    <!- This parameter specifys when to stop update the paramters before iteration starts, options: designatedIteration, reachConvergence, default:designatedIteration(default maxItertaion=1). By default, the module will update parameters in planCalcScore till min(designatedIteration, iteration). If reachCovergence is set, updating parameters stops when average estimated difference is smaller than 0.001-->
 <param name="calibrationEndCriteria" value=”designatedIteration" />
<param name=”maxIteration" value=”1" />
<!- maximumLiklihood (MLE)method for parameter estimation, option:  -- >
    <param name=”mleOptimizer" value="1" />
    <param name="maximumLikelihood" value="1" />
     <param name="maxAlternativePlans" value=”5" />
    <param name=”mleIteration" value="1000" />
    <param name=”usePersonalParameter" value=”true" />
    <param name=”gbdtEnabled" value=”true" />

</module>
```

2. Population file:

```
<person id=”surveyPeople_0"> <attributes>
<attribute name="subpopulation"
class="java.lang.String" >surveyPeople</attribute> </attributes>
<plan selected="yes"> ...
```

3. Strategy module setup:

```
<module name="strategy" >
...
<parameterset type="strategysettings" >
    <param name="disableAfterIteration" value="-1" />
    <param name="executionPath" value="null" />
    <param name="strategyName" value="KeepLastSelected" />
    <param name="subpopulation" value="surveyPeople" />
    <param name="weight" value="0.2" />
  </parameterset>
...
</module>
```

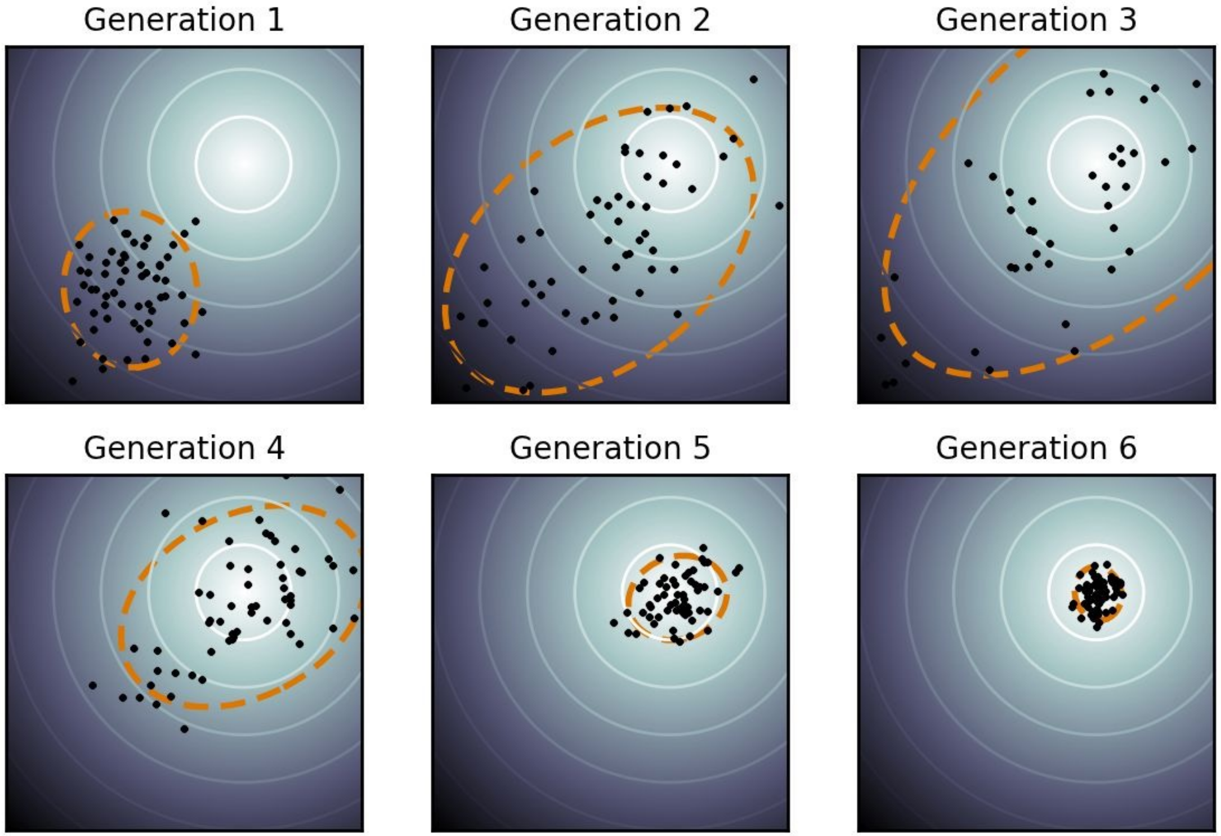
4. They share other parameters together!!

CONTENT

- purpose & goal
    - why develop this extension
    - goal: parameter adaptation according to different scales
- framework & structure
    - Module framework, related mechanism and pipelines
- CMA-ES algorithm
    - How CMA-ES works and its formulation
- scenario test
    - Parameter-estimation results
    - proof that it is auto-adaptive to different population scales
- conclusion and future work

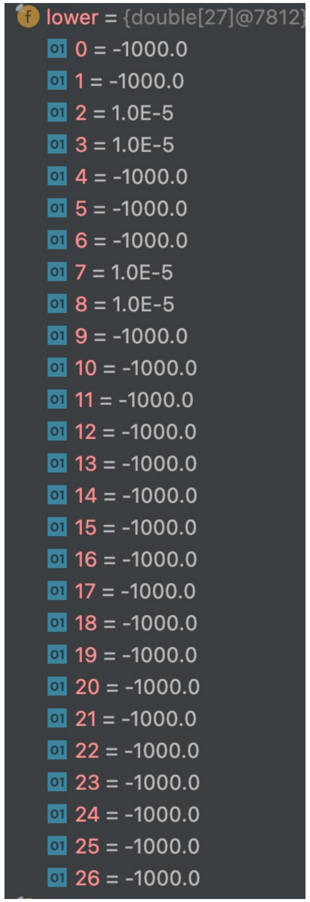
CMA-ES algorithm for parameter estimation

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a powerful optimization algorithm used primarily for solving non-linear, non-convex optimization problems in continuous domains. It is especially effective in high-dimensional spaces and scenarios where the objective function is complex, noisy, or lacks smoothness.



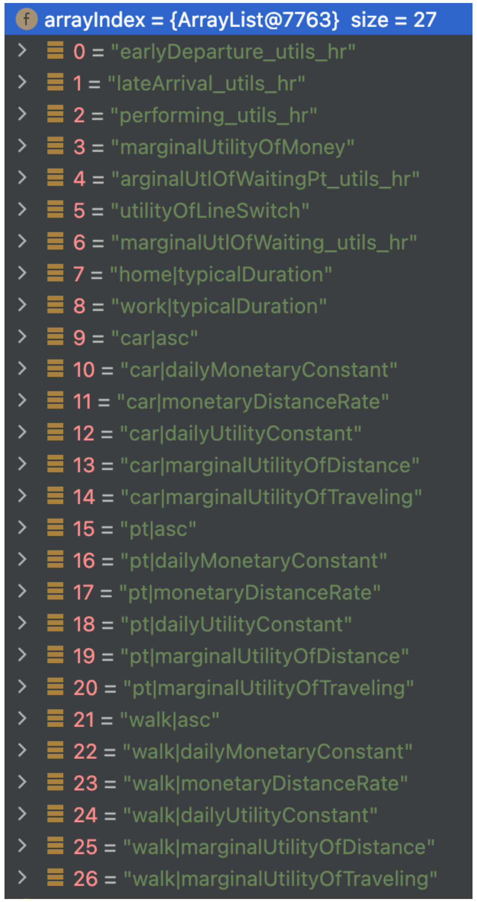
CMA-ES algorithm for parameter estimation

[lower bounds



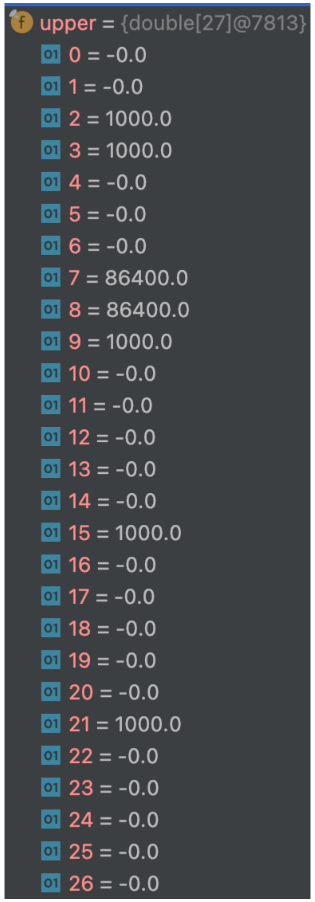
```
lower = {double[27]@7812}
  0 = -1000.0
  1 = -1000.0
  2 = 1.0E-5
  3 = 1.0E-5
  4 = -1000.0
  5 = -1000.0
  6 = -1000.0
  7 = 1.0E-5
  8 = 1.0E-5
  9 = -1000.0
  10 = -1000.0
  11 = -1000.0
  12 = -1000.0
  13 = -1000.0
  14 = -1000.0
  15 = -1000.0
  16 = -1000.0
  17 = -1000.0
  18 = -1000.0
  19 = -1000.0
  20 = -1000.0
  21 = -1000.0
  22 = -1000.0
  23 = -1000.0
  24 = -1000.0
  25 = -1000.0
  26 = -1000.0
```

Params to be estimated



```
arrayIndex = {ArrayList@7763}  size = 27
>  0 = "earlyDeparture_utils_hr"
>  1 = "lateArrival_utils_hr"
>  2 = "performing_utils_hr"
>  3 = "marginalUtilityOfMoney"
>  4 = "arginalUtlOfWaitingPt_utils_hr"
>  5 = "utilityOfLineSwitch"
>  6 = "marginalUtlOfWaiting_utils_hr"
>  7 = "home|typicalDuration"
>  8 = "work|typicalDuration"
>  9 = "car|asc"
>  10 = "car|dailyMonetaryConstant"
>  11 = "car|monetaryDistanceRate"
>  12 = "car|dailyUtilityConstant"
>  13 = "car|marginalUtilityOfDistance"
>  14 = "car|marginalUtilityOfTraveling"
>  15 = "pt|asc"
>  16 = "pt|dailyMonetaryConstant"
>  17 = "pt|monetaryDistanceRate"
>  18 = "pt|dailyUtilityConstant"
>  19 = "pt|marginalUtilityOfDistance"
>  20 = "pt|marginalUtilityOfTraveling"
>  21 = "walk|asc"
>  22 = "walk|dailyMonetaryConstant"
>  23 = "walk|monetaryDistanceRate"
>  24 = "walk|dailyUtilityConstant"
>  25 = "walk|marginalUtilityOfDistance"
>  26 = "walk|marginalUtilityOfTraveling"
```

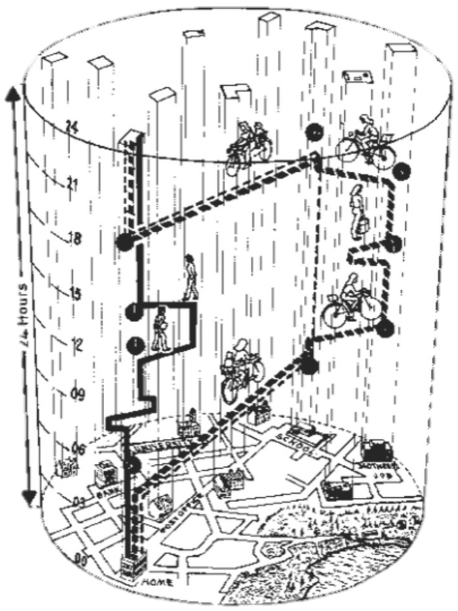
upper bounds]



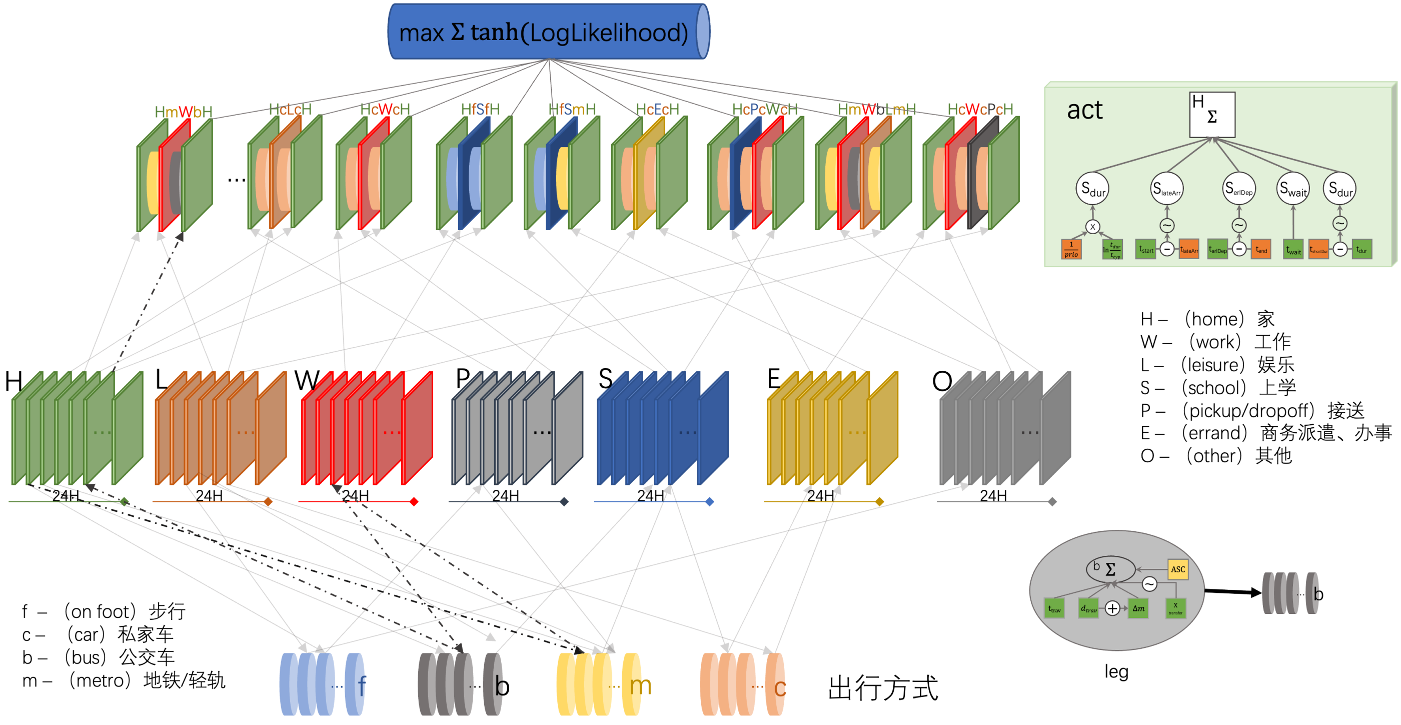
```
upper = {double[27]@7813}
  0 = -0.0
  1 = -0.0
  2 = 1000.0
  3 = 1000.0
  4 = -0.0
  5 = -0.0
  6 = -0.0
  7 = 86400.0
  8 = 86400.0
  9 = 1000.0
  10 = -0.0
  11 = -0.0
  12 = -0.0
  13 = -0.0
  14 = -0.0
  15 = 1000.0
  16 = -0.0
  17 = -0.0
  18 = -0.0
  19 = -0.0
  20 = -0.0
  21 = 1000.0
  22 = -0.0
  23 = -0.0
  24 = -0.0
  25 = -0.0
  26 = -0.0
```

1. **Initialize parameters**
2. Sample from the distribution
3. **Evaluate the objective function values of the samples**
4. Select the top samples with the best function values
5. **Calculate the evolution paths and update the covariance matrix and step size**
6. Check if stopping condition is met, terminate the algorithm or to the next generation

CMA-ES for plans – duplicates from CharyparNagelAct&LegScoring



typical activity trip chain



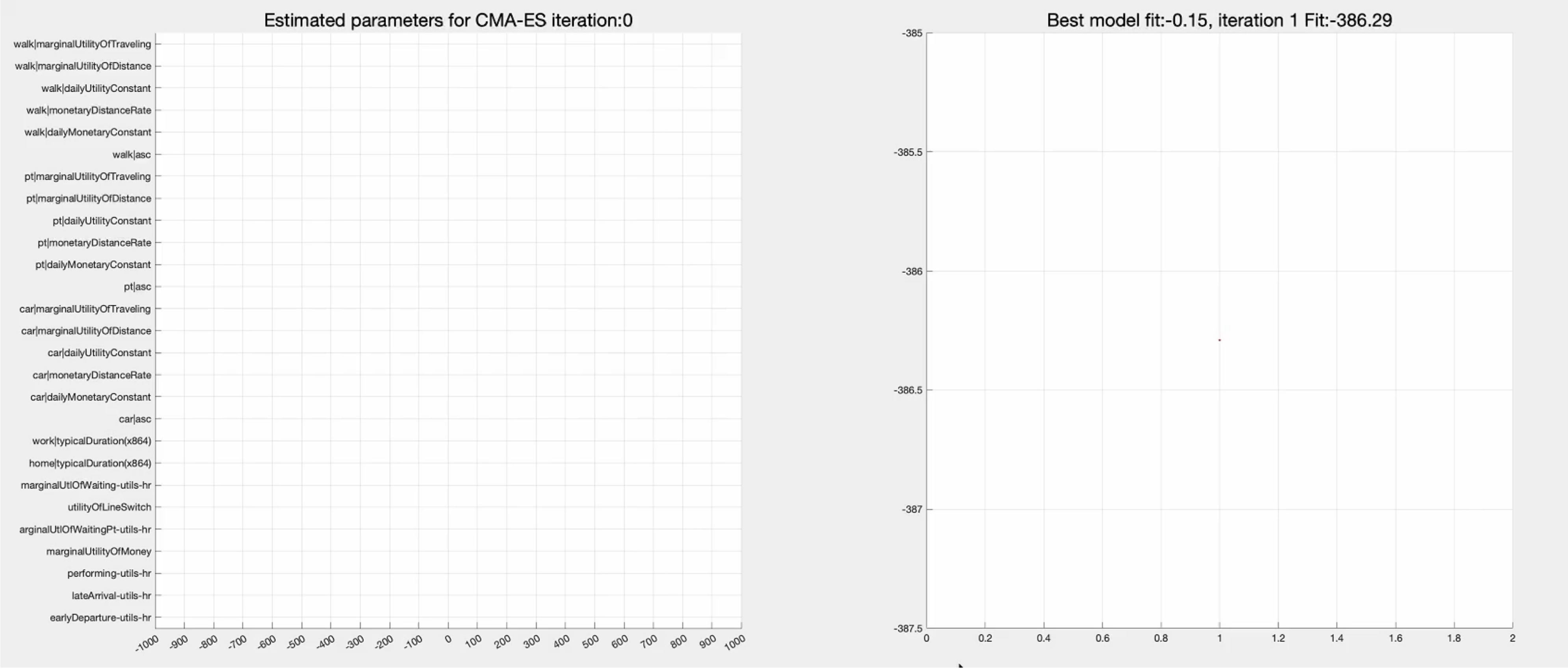
H – （home）家
W – （work）工作
L – （leisure）娱乐
S – （school）上学
P – （pickup/dropoff）接送
E – （errand）商务派遣、办事
O – （other）其他

f – （on foot）步行
c – （car）私家车
b – （bus）公交车
m – （metro）地铁/轻轨

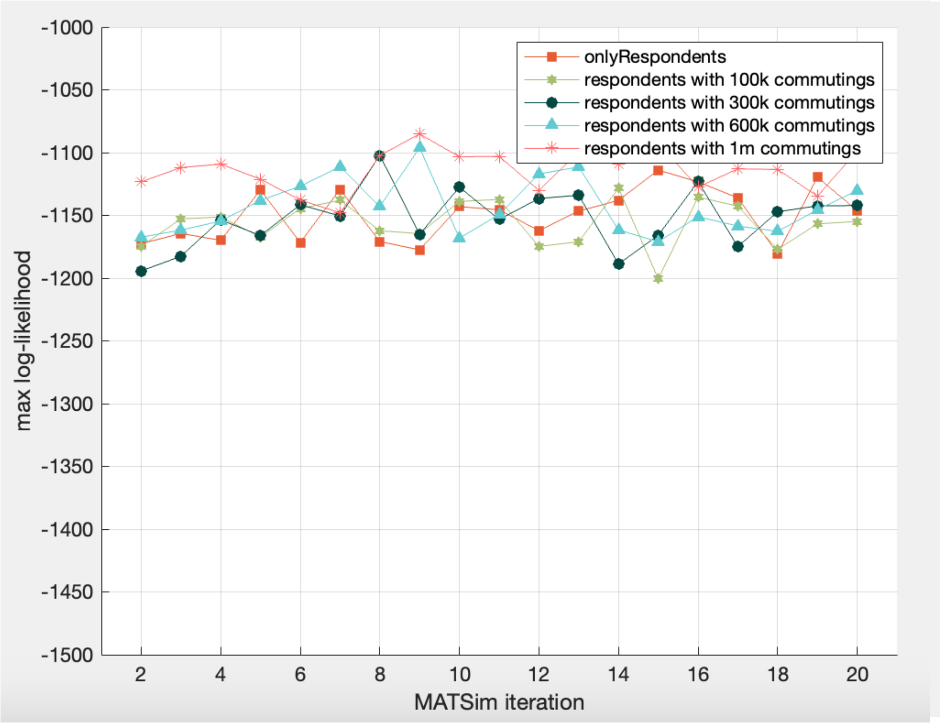
CONTENT

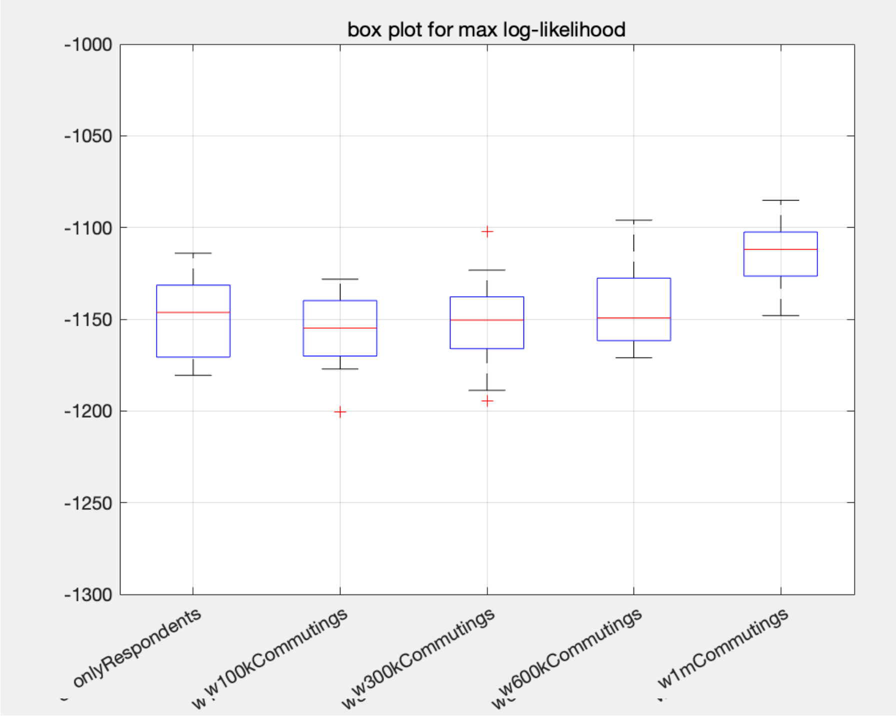
- purpose & goal
    - why develop this extension
    - goal: parameter adaptation according to different scales
- framework & structure
    - Module framework, related mechanism and pipelines
- CMA-ES algorithm
    - How CMA-ES works and its formulation
- scenario test
    - Parameter-estimation results
    - proof that it is auto-adaptive to different population scales
- conclusion and future work

Scenario test –
CMA-ES estimation inside a MATSim iteration



Scenario test –
Estimated model fit score with different scale of population



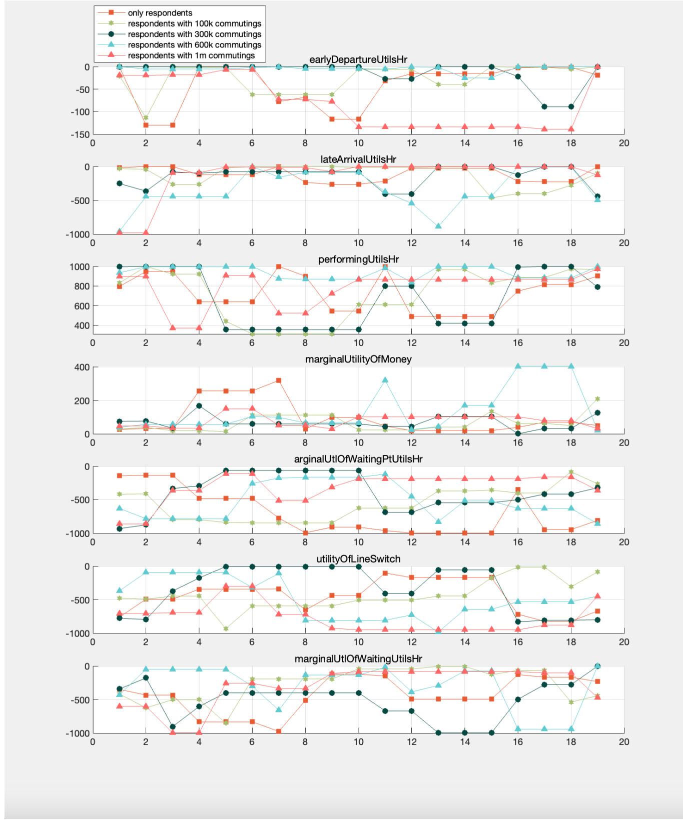


A street interview trip diary survey made at 09-10. 2023 in Caton

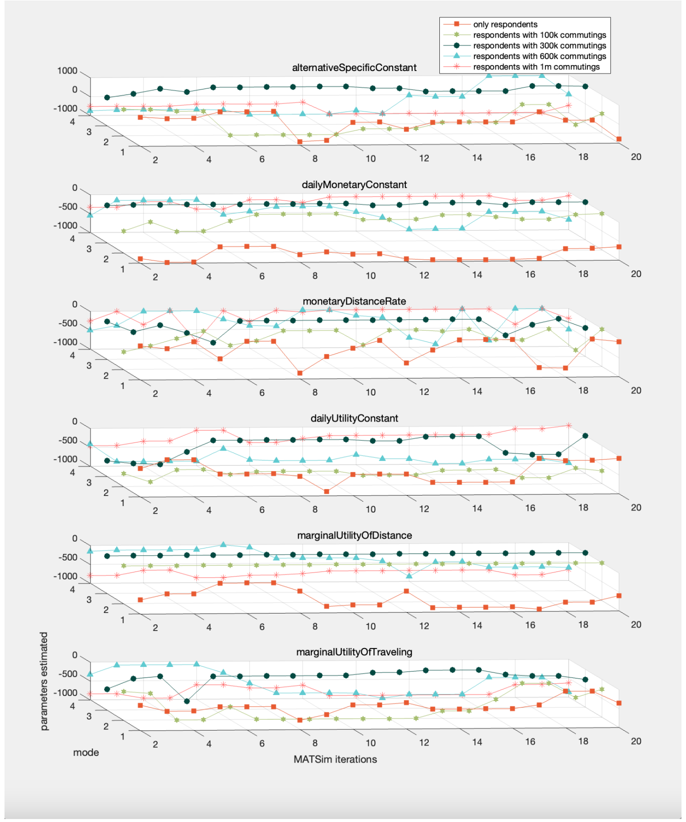
Commuting array is from LBS data from Gaode, with 100m grids accuracy

Scenario test –
Estimated parameters with different scale of population, different iterations

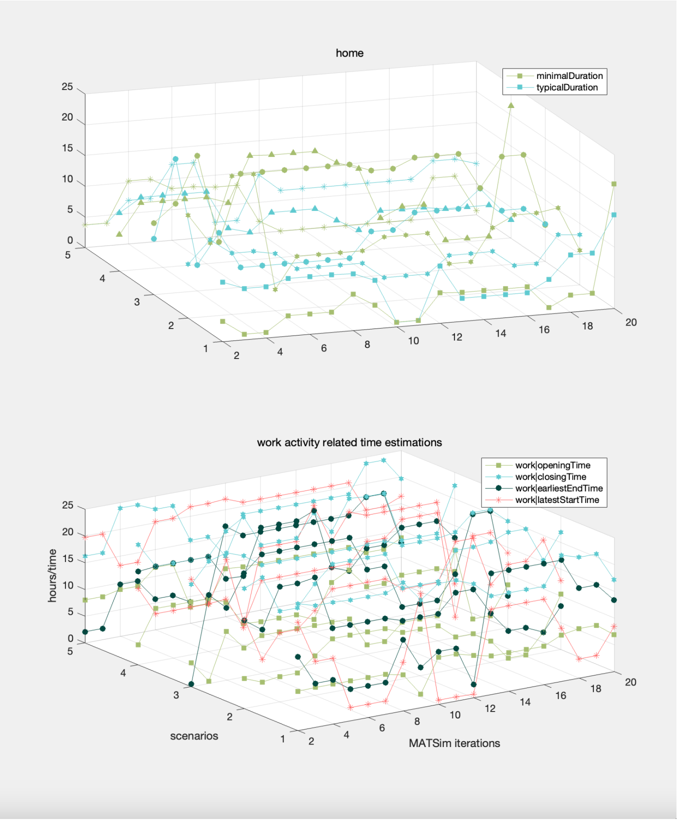
MATSim



for general parameters

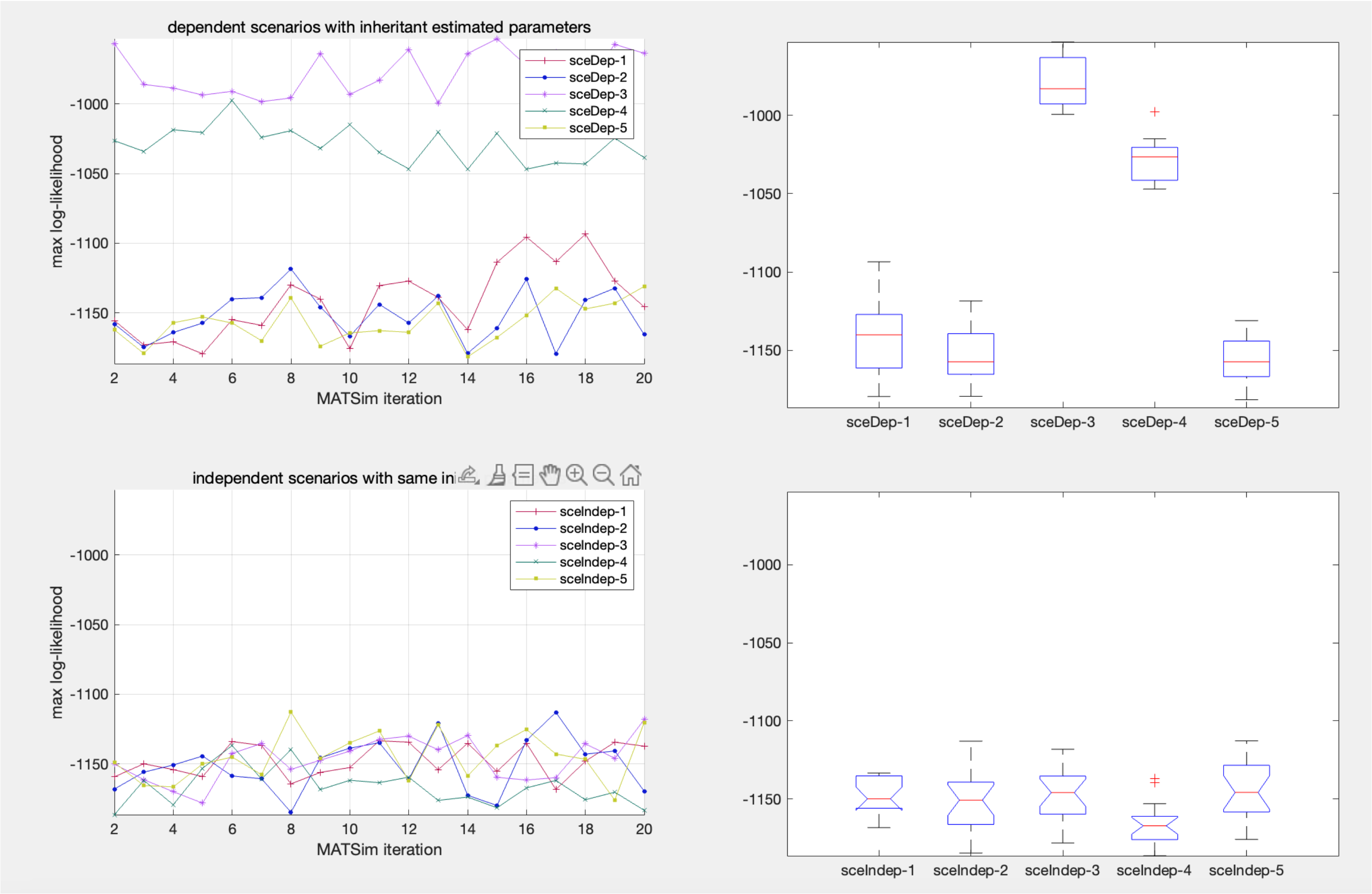


for leg/mode parameters



for act parameters

Scenario test –
CMAES best fitness with load from previous scenarios vs dependent parameters



Scenario test – future test comparison biogeme vs cma-es

Use biogeme & cma-es to estimate parameters(show)
Apply parameters in config files
run a couple iterations , check if the survey plan is still the select show: hit ratio comparison

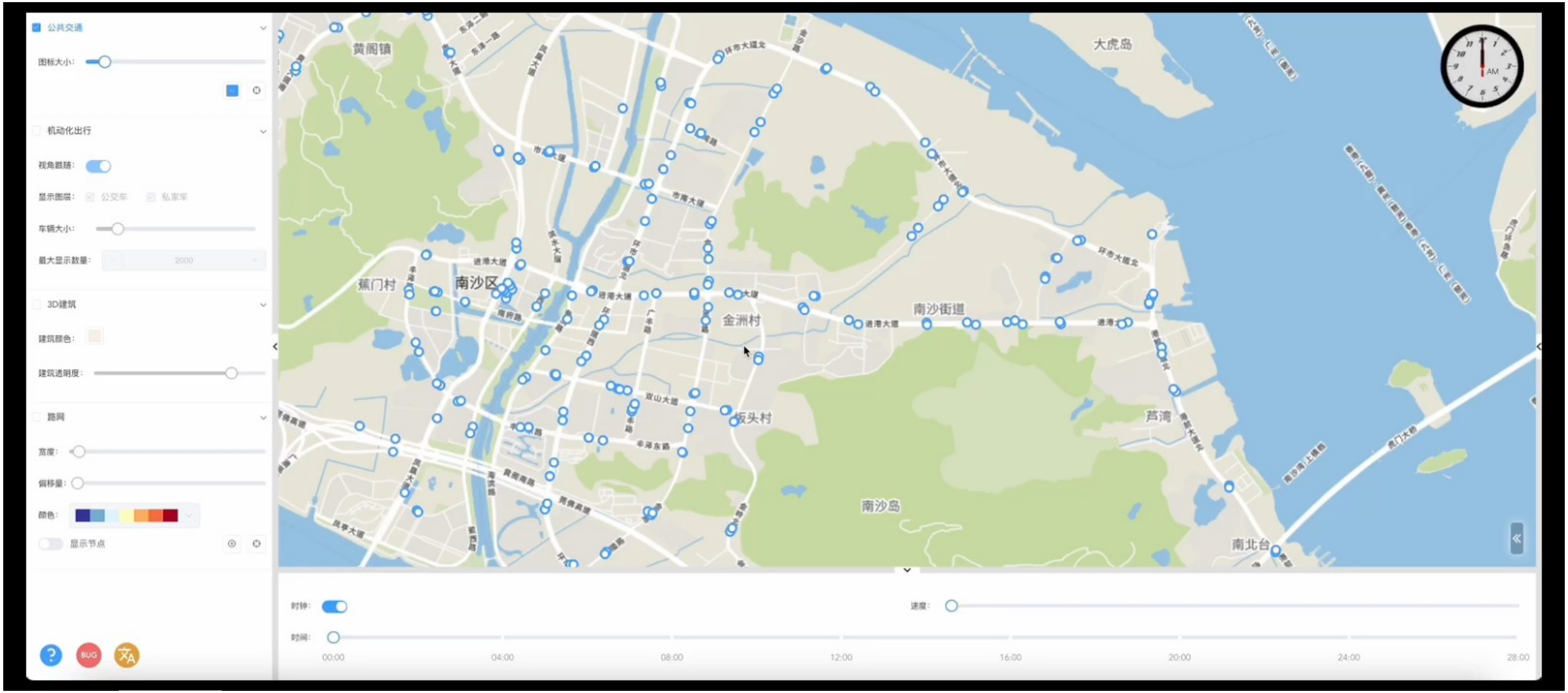
Self compare:
Run keep selected and get estimated parameters
Run without cali and check hit ratio

CONTENT

- purpose & goal
  - why develop this extension
  - goal: parameter adaptation according to different scales
- framework & structure
  - Module framework, related mechanism and pipelines
- CMA-ES algorithm
  - How CMA-ES works and its formulation
- scenario test
  - Parameter-estimation results
  - proof that it is auto-adaptive to different population scales
- conclusion and future work

Conclusion & future work

- The project is still under development and lots of debugging to do
- This extension provides a way to simplify parameter setups for modelling under lack of data
- Trip diary or other ground truth data is a must have, and it is designed to dimensionless of input scales, and rather focusing on decision makings mimicking the ground truth data.
- It will definitely slowdown MATSim running time for the time being. Apache common math3 library is used and parallel computing is not implemented
- Other solvers for the fitness/goal function might be tested later, like traditional c-Newton method, BP, etc.
- Local minimum might be a very tricky problem for CMA-ES …



☑ 公共交通

图标大小：

☐ 机动化出行

视角跟随：

显示图层： ☑ 公交车 ☑ 私家车

车辆大小：

最大显示数量： 2000

☐ 3D建筑

建筑颜色：

建筑透明度：

☐ 路网

宽度：

偏移量：

颜色： 🟦🟦⬜🟨🟧🟧🟥

显示节点

时钟：

速度：

时间：

Milu

convel@163.com