

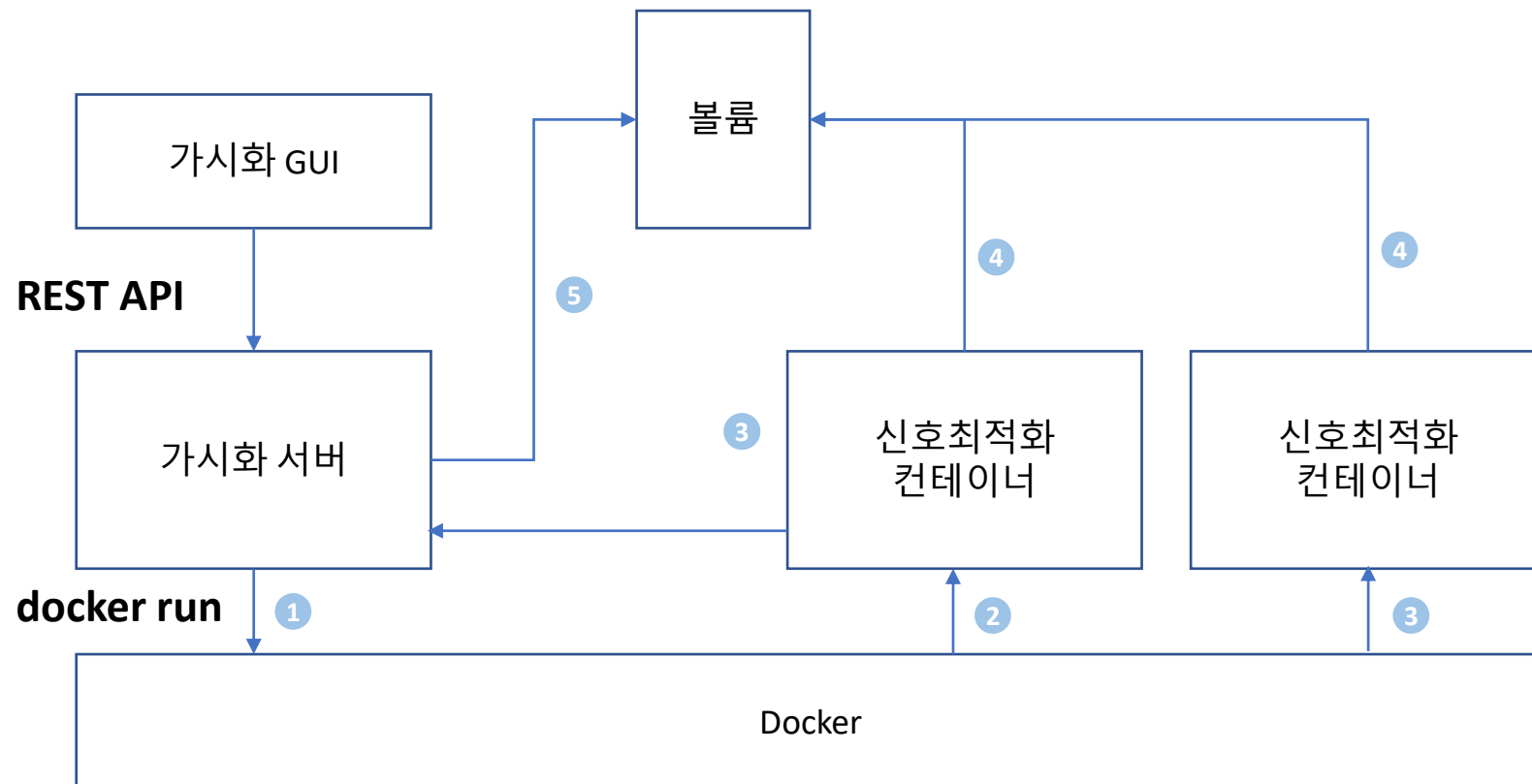
UNIQ-VIS

신호최적화 연동 및 가시화

2023

Modutech

도커 기반 실행제어 구조



UNIQ-VIS VM 접속 환경

- 가시화 서버
 - ubuntu@101.79.1.124
 - SALT 및 신호최적화 모듈 연동을 위한 볼륨 경로
 - /home/ubuntu/uniq-simv2
- 가시화 서버 사용자 인터페이스
 - <http://101.79.1.124:8080>

신호 최적화 실험

1. 신호최적화 등록

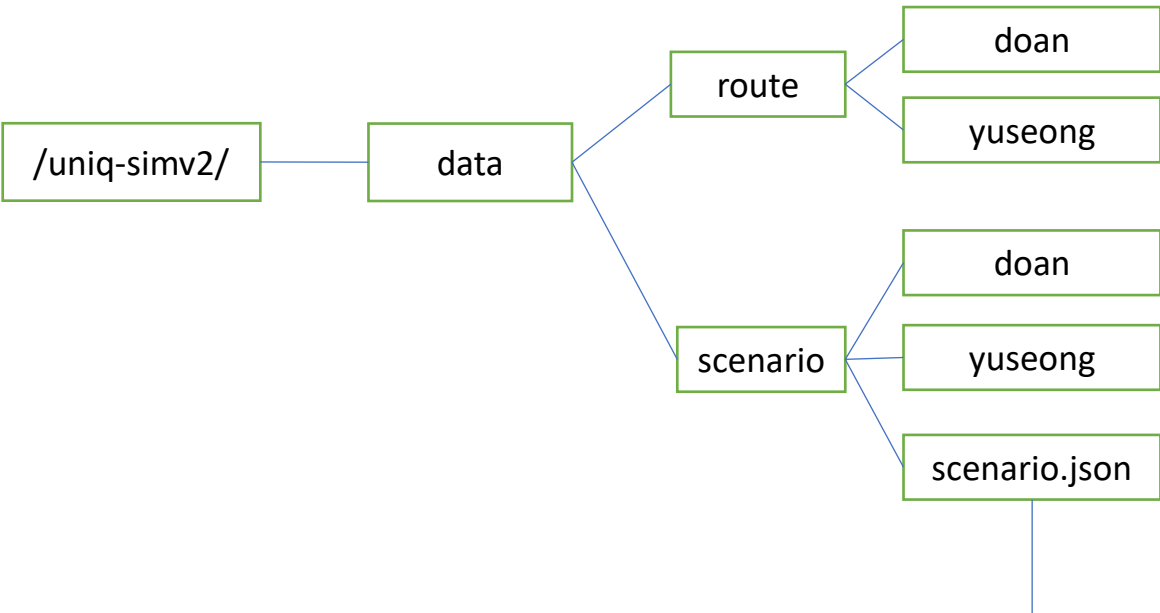
- 신호 최적화 등록시 아래 경로에 신호 최적화를 위한 시나리오 파일이 생성 되어야 함
 - /uniq-simv2/opt/{opt-xxx}/input

2. 학습

- 신호 최적화 학습 단계에서는 아래 경로에 모델과 학습 결과가 생성됨
 - /uniq-simv2/opt/{opt-xxx}/model
 - /uniq-simv2/opt/{opt-xxx}/output/train

3. 신호적용

- 신호적용 단계에서는 기존 신호 및 선택한 모델에 대한 2개의 결과물이 아래 경로에 생성 됨
 - /uniq-simv2/opt/{opt-xxx}/output/simulate
 - /uniq-simv2/opt/{opt-xxx}/output/test



```
[
  {
    "region": "doan",
    "description": "도안지역",
    "tls": [
      "SA 101",
      "SA 102",
      "SA 103"
    ]
  },
  {
    "region": "yuseong",
    "description": "둔산,월평 지역",
    "tls": [
      "SA 28"
    ]
  }
]
```

GUI 에서 신호최적화 실험 등록시 지역 선택에 사용

둔산, 월평지역을 yuseong.tss.xml 파일을 사용하고 있어
지도상에 yuseong.tss.xml 파일에 포함된 신호그룹이
모두 표시됨

uniq-simv2/

data

grid_.geojson

route

doan

doan_20220601.rou.xml

doan_20220602.rou.xml

yuseong

yuseong_20220601.rou.xml

yuseong_20220602.rou.xml

route_sim

ddg.rou.xml

dg.rou.xml

dj.rou.xml

doan.rou.xml

jg.rou.xml

seogu.rou.xml

yuseonggu.rou.xml

scenario

doan

doan.connection.xml

doan.edge.xml

doan.node.xml

doan.tss.xml

doan_simulate.scenario.json

doan_test.scenario.json

doan_train.scenario.json

scenario.json

yuseong

yuseong.connection.xml

yuseong.edge.xml

yuseong.node.xml

yuseong.tss.xml

yuseong_simulate.scenario.json

yuseong_test.scenario.json

yuseong_train.scenario.json

scenario_sim

scenario_dj_ddg.zip

scenario_dj_dg.zip

scenario_dj_dj.zip

scenario_dj_doan.zip

scenario_dj_jg.zip

scenario_dj_seogu.zip

scenario_dj_yuseonggu.zip

opt

OPTI_202309_00180

input

yuseong_simulate.scenario.json

yuseong_test.scenario.json

yuseong_train.scenario.json

model

sappo

SAPPO-_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5-trial_0_SA_28_actor.h5

SAPPO-_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5-trial_80_SA_28_critic_2.h5

SAPPO-_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5-trial_80_SA_28_critic_3.h5

SAPPO-_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5-trial_80_SA_28_critic_4.h5

output

simulate

S-271001771538-1_PeriodicOutput.csv

ft_phase_reward_output.txt

progress.txt

test

S-166593626023-0_PeriodicOutput.csv

_PeriodicOutput.csv

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_0.csv

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_100.csv

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_120.csv

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_200.csv

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_300.csv

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_380.csv

progress.txt

rl_phase_reward_output.txt

train

0

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_0.csv

1

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_0.csv

99

_state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mlen_16_control_cycle_5_s600_0.csv

OPTI_202309_00057_PeriodicOutput.csv

_PeriodicOutput.csv

progress.txt

train_epoch_tl_reward.txt

train_epoch_total_reward.txt

신호최적화 실행 결과

신호최적화 simulate 모드 실행 결과

ft_phase_reward_output.txt
progress.txt
S-927708787942-1_PeriodicOutput.csv

simulate 모드로 실행 중에 동적으로 N 번 업데이트 됨
마지막에 선택한 모델의 값임

- test 모드가 완료될 때 1번 생성됨
- 선택한 모델별로 결과 파일이 생성 됨
- 개별시뮬레이션 종료 후에도 해당 값을 생성하느라 실제 컨테이너가 종료되는 데는 시간이 걸림

신호최적화 train 모드 실행 결과

_state_vd_action_ga_reward_wq_alr_0.005_clr_0.005_mLen_16_control_cycle_5_s600_0.csv
_state_vd_action_ga_reward_wq_alr_0.005_clr_0.005_mLen_16_control_cycle_5_s600_1.csv
_state_vd_action_ga_reward_wq_alr_0.005_clr_0.005_mLen_16_control_cycle_5_s600_2.csv
_state_vd_action_ga_reward_wq_alr_0.005_clr_0.005_mLen_16_control_cycle_5_s600_300.csv
progress.txt
rl_phase_reward_output.txt
S-245338779744-0_PeriodicOutput.csv

Test 모드로 실행 중에 동적으로 N 번 업데이트 됨
마지막에 실행한 모델의 값임

두 값을 사용해
신호최적화 화면의
통과시간 향상률 테이블 및 차트 출력

환경 설정 사항 (변경 완료)

☐ 지역

- ☐ 도안: doan;
- ☐ 유성: yuseong;

☐ 모델

- ☐ SAPPO: sappo

☐ 상태

- ☐ 차량 수: v; 차량 밀도: d; 차량 수와 차량 밀도: vd

☐ 액션

- ☐ 현시 최소최대 만족: gt; 현시 주기 만족: ga;

☐ 보상함수

- ☐ 통과 차량 수: pn; 대기 큐 길이: wq; 통과 시간: wt;

☐ 교통 수요 (route)

- ☐ route의 경우, scenario.json 로 각각 만들어야 함!

☐ Learning Rate 추가

☐ MemoryLength 추가

환경 설정 사항 (변경 x)

- 시뮬레이션 시간
- 대상 교차로
- 도커 이미지 (최적화 코드 버전)
- 통계 주기
- 가시화 주기
- 에포크
- 모델 저장 주기

데모 고려사항

state, action, reward, alr, clr, mLen 값에 따라
파일 이름이 생성됨

신호최적화 등록시 사용한 값과 동일한 설정으로
사용해야 함

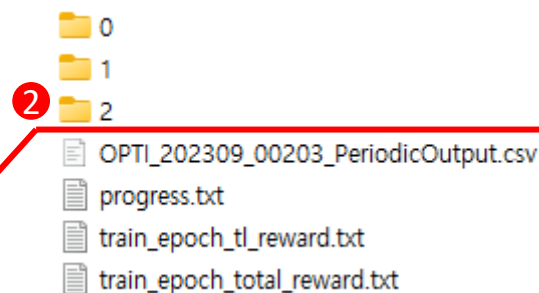
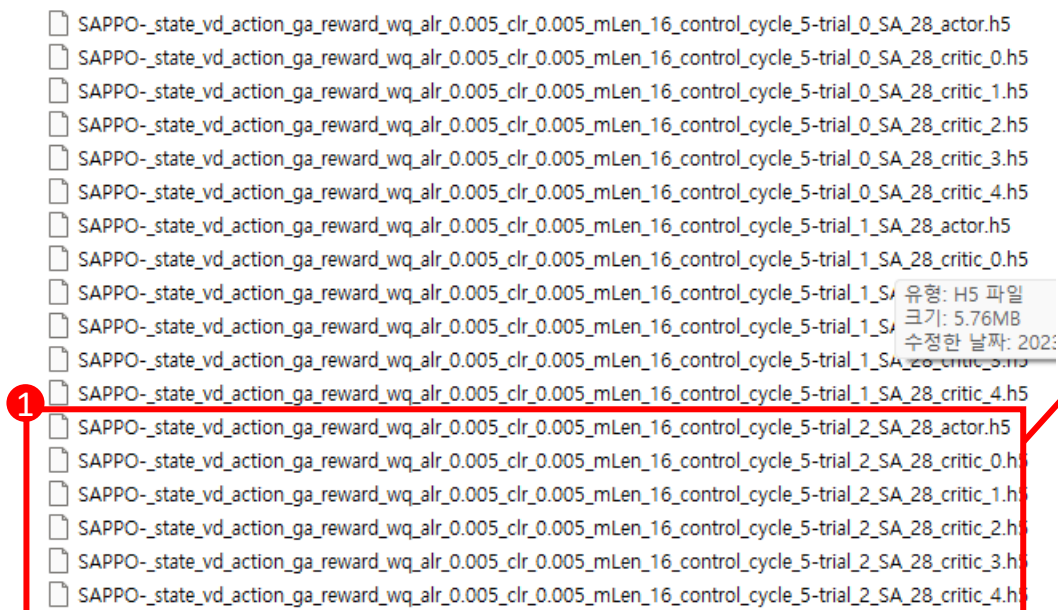
```
ubuntu@ubu1804-test:~/vis/trained_model/final/action_ga$ ls SA28
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_512_control_cycle_5-trial_300_SA_28_actor.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_512_control_cycle_5-trial_300_SA_28_critic_0.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_512_control_cycle_5-trial_300_SA_28_critic_1.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_512_control_cycle_5-trial_300_SA_28_critic_2.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_512_control_cycle_5-trial_300_SA_28_critic_3.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_512_control_cycle_5-trial_300_SA_28_critic_4.h5
```

```
ubuntu@ubu1804-test:~/vis/trained_model/final/action_ga$ ls ../../../../uniq-simv2/opt/OPTI_202309_00180/model/sappo
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_0_SA_28_actor.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_0_SA_28_critic_0.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_0_SA_28_critic_1.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_0_SA_28_critic_2.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_0_SA_28_critic_3.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_0_SA_28_critic_4.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_100_SA_28_actor.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_100_SA_28_critic_0.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_100_SA_28_critic_1.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_100_SA_28_critic_2.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_100_SA_28_critic_3.h5
SAPPO-state_vd_action_ga_reward_wq_alr_0.0001_clr_0.0001_mLen_16_control_cycle_5-trial_100_SA_28_critic_4.h5
```

데모 고려사항

- --mem-len 512 옵션 사용시 에포크별 상당한 시간이 걸림
- 데모를 위해서는 미리 학습된 모델 파일들이 준비되어 있어야 함
- 미리 학습된 데이터를 가시화 서버에서 사용하려면?
 - 모델 파일과 trained 디렉토리에 해당 모델의 csv 파일을 업데이트 필요
 - 필요에 따라서 train_epoch_tl_reward.txt, train_epoch_total_reward.txt 파일 수정

```
epoch,reward,40ep_reward  
0,-4.639986672585271,-4.639986672585271  
1,-4.5633333344740979,-4.601660008663125  
2,-4.7167733311852444,-4.640031116170499
```



/opt_xxx/trained

/opt_xxx/model/sappo

신호 최적화 지역 추가

- /uniq-simv2/data/scenario/ 디렉토리에 시나리오 준비
- /uniq-simv2/data/scenario/scenario.json 파일 수정
- 수요파일 선택 기능
 - 현재 지원 X
 - 규칙 필요
 - 현재는 아래 파일을 사용하도록 하드코딩 되어 있음
 - doan → doan_20220601.rou.xml,
 - yuseong → yuseong_20220601.rou.xml

신호최적화 학습

1. simulate 모드로 먼저 실행
2. train 모드로 실행

```
docker run --rm --name opt_xxx
-v c:\\home\\ubuntu\\uniq-simv2\\data:/uniq/data
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\input:/uniq/optimizer/input
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\output:/uniq/optimizer/output
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\model:/uniq/optimizer/model
-v c:\\home\\ubuntu\\uniq-simv2\\opt:/uniq/optimizer/opt
images4uniq/optimizer:v3.0.20230814
python ./run_modutech.py --mode simulate --map doan
```

```
docker run --rm --name opt_xxx
-v c:\\home\\ubuntu\\uniq-simv2\\data:/uniq/data
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\input:/uniq/optimizer/input
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\output:/uniq/optimizer/output
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\model:/uniq/optimizer/model
-v c:\\home\\ubuntu\\uniq-simv2\\opt:/uniq/optimizer/opt
images4uniq/optimizer:v3.0.20230814
python ./run_modutech.py --mode train --map doan
--start-time 25200 --end-time 32340 --epoch 2
--target-TL "SA 101"
--model-save-period 1
--result-comp False
--reward-func pn
--method sappo
--action gt
```

주의: --result-comp 옵션이 있어야 비교 파일 생성

시뮬레이터

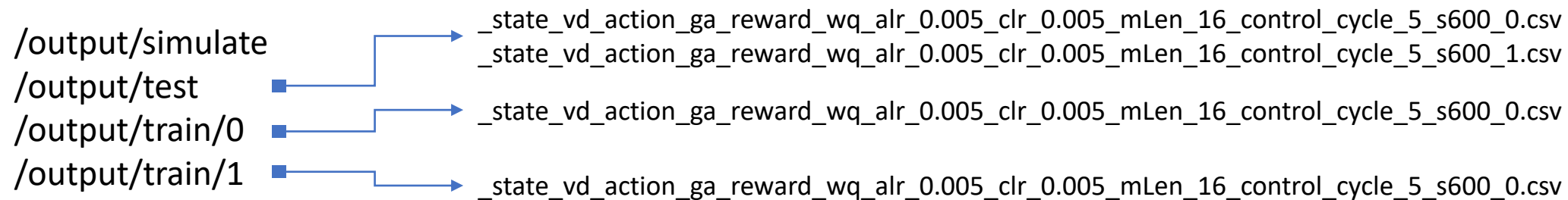
```
"scenario": {
  "id": "SIMU_202307_00448",
  "host": "192.168.1.223",
  "port": 1337,
  "interval": 10,
  "time": {
    "begin": 25200,
    "end": 32280
  },
  "input": {
    "fileType": "SALT",
    "node": "node.xml",
    "link": "edge.xml",
    "connection": "connection.xml",
    "trafficLightSystem": "tss.xml",
    "multiarea": "multiarea/edge.xml",
    "route": "/uniq/simulator/salt/volume/data/route_sim/doan.rou.xml"
  },
  "parameter": {
    "minCellLength": 30,
    "vehLength": 5
  },
  "output": {
    "fileDir": "/uniq/simulator/salt/volume/sim/SIMU_202307_00448/output",
    "period": 300,
    "level": "cell",
    "save": 1
  }
}
```

신호최적화

```
"scenario": {
  "id": "S-927708787942-1",
  "host": "192.168.1.223",
  "port": 1337,
  "interval": 10,
  "time": {
    "begin": 25200,
    "end": 32340
  },
  "input": {
    "fileType": "SALT",
    "node": "../data/scenario/yuseong/yuseong.node.xml",
    "link": "../data/scenario/yuseong/yuseong.edge.xml",
    "connection": "../data/scenario/yuseong/yuseong.connection.xml",
    "trafficLightSystem": "../data/scenario/yuseong/yuseong.tss.xml",
    "route": "../data/route/yuseong/yuseong_20220601.rou.xml"
  },
  "parameter": {
    "minCellLength": 30,
    "vehLength": 5
  },
  "output": {
    "fileDir": "output/simulate/",
    "period": 30,
    "level": "link",
    "save": 1
  }
}
```

**** SALT 시뮬레이션과 신호최적화의 입력 경로가 다름에 주의**

결과파일의 분석



신호최적화 테스트 결과파일 분석

/output/test/_state_vd_action_ga_reward_wq_alr_0.005_clr_0.005_mLen_16_control_cycle_5_s600_0.csv

ft_VehPassed_sum_0hop : 기존 통행 차량 수

ft_SumTravelTime_sum_0hop : 기존 통행 차량들의 통행 시간 합

$ft_averageTravelTime = ft_SumTravelTime_sum_0hop / ft_VehPassed_sum_0hop$

$rl_averageTravelTime = rl_SumTravelTime_sum_0hop / rl_VehPassed_sum_0hop$

$improvedRate (\%) = (ft_averageTravelTime - rl_averageTravelTime) / ft_averageTravelTime * 100$

신호 최적화 컨테이너 실행

```
docker run --rm --name opt_xxx
-v c:\\home\\ubuntu\\uniq-simv2\\data:/uniq/data
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\input:/uniq/optimizer/input
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\output:/uniq/optimizer/output
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\model:/uniq/optimizer/model
-v c:\\home\\ubuntu\\uniq-simv2\\opt:/uniq/optimizer/opt images4uniq/optimizer:v3.0.20230814
python ./run_modutech.py --mode simulate --map doan
```

```
docker run --rm --name opt_xxx -v c:\\home\\ubuntu\\uniq-simv2\\data:/uniq/data
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\input:/uniq/optimizer/input
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\output:/uniq/optimizer/output
-v c:\\home\\ubuntu\\uniq-simv2\\opt\\opt_xxx\\model:/uniq/optimizer/model
-v c:\\home\\ubuntu\\uniq-simv2\\opt:/uniq/optimizer/opt images4uniq/optimizer:v3.0.20230814
python ./run_modutech.py --mode train --map doan --start-time 25200 --end-time 32340 --epoch 2 --target-TL "SA 101" --model-save-
period 1 --reward-func pn --method sappo --action gt
```