

**UNIQ-SALT: 신호최적화 지원을 위한
시뮬레이션 기능 보완**

개요

1. 시뮬레이션 주기적 통계 산출값 수정/업데이트
2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석
3. 통계값 산출 방식 변경
4. 신호데이터 적용 오류 해결

1. 시뮬레이션 주기적 통계 산출값 수정/업데이트

- 1) 통계 산출값 간 산출 조건/기준 통일
 - (이슈) 시뮬레이션 결과 출력 파일 (링크 레벨) 내, 주기적 통계 산출값 간 산출 조건/기준이 통일되지 않음
 - (예) VehPassed (통과 차량 수) = 0 인 경우에도, SumTravelLength (통과 차량 기준 통행 거리 합)/ SumTravelTime (통과 차량 기준 통행 시간 합) 값 이 존재하는 경우 다수
 - (원인) 시뮬레이터 자체가 셀 기준으로 시뮬레이션 되며, 셀 기준 통계값을 주(main) 산출되고, 링크 기준으로는 셀 기준 통계값을 이용해서 계산하게 됨. 이 과정에서 셀 기준으로는 통과 차량이나, 링크 기준으로는 통과 차량이 되지 않는 경우를 현재는 별도 체크하지 않고 있어 해당 이슈 발생.
 - (해결) 링크 기준으로 통과 정보를 트래킹할 수 있도록 기능 추가
- 2) 통계 산출값 계산 오류 해결
 - (이슈) getAverageWaitingQLength 값 계산 오류
 - (원인) Periodic Output 값 변경 시, AverageWaitingQLength / AverageWaitingTime 값을 AverageVehicleWaitingQLength / AverageVehicleWaitingTime 값으로 변경 출력하도록 수정하면서, 원래 값 (AverageWaitingQLength / AverageWaitingTime) 계산하는 함수 호출 부분이 누락됨.
 - (해결) 현재 해당 이슈 해결한 버전 git 업데이터터 완료
 - 이전에 말씀드렸듯이, AverageWaitingQLength 값의 경우, 코드 상으로는 통계 정보 수집 구간 중에 발생한 셀 내 차량의 Waiting Time의 합을 수집 기간으로 나눈 값으로 주어짐.
 - 즉, 이는 수집 기간 동안에 Waiting Q에 머무는 시간에 대한 기대 평균값을 나타낸 것이라고 생각되는데, 명칭으로 그 의미에서 바로 파악할 수 없는 값으로 판단되어, 이번에 출력 파일 계산 관련 부분을 수정/업데이트하면서 AverageVehicleWaitingQLength로 변경하였음. (--> 혹시 이전으로의 변경을 원하면 원복 가능함.)
 - 참고: 현재 AverageVehicleWaitingQLength는 통계 정보 수집 구간 중에 해당 링크 내 셀의 WaitingQ 길이의 합으로 주어짐.

3) API 함수 별 반환 통계값 의미 명확화

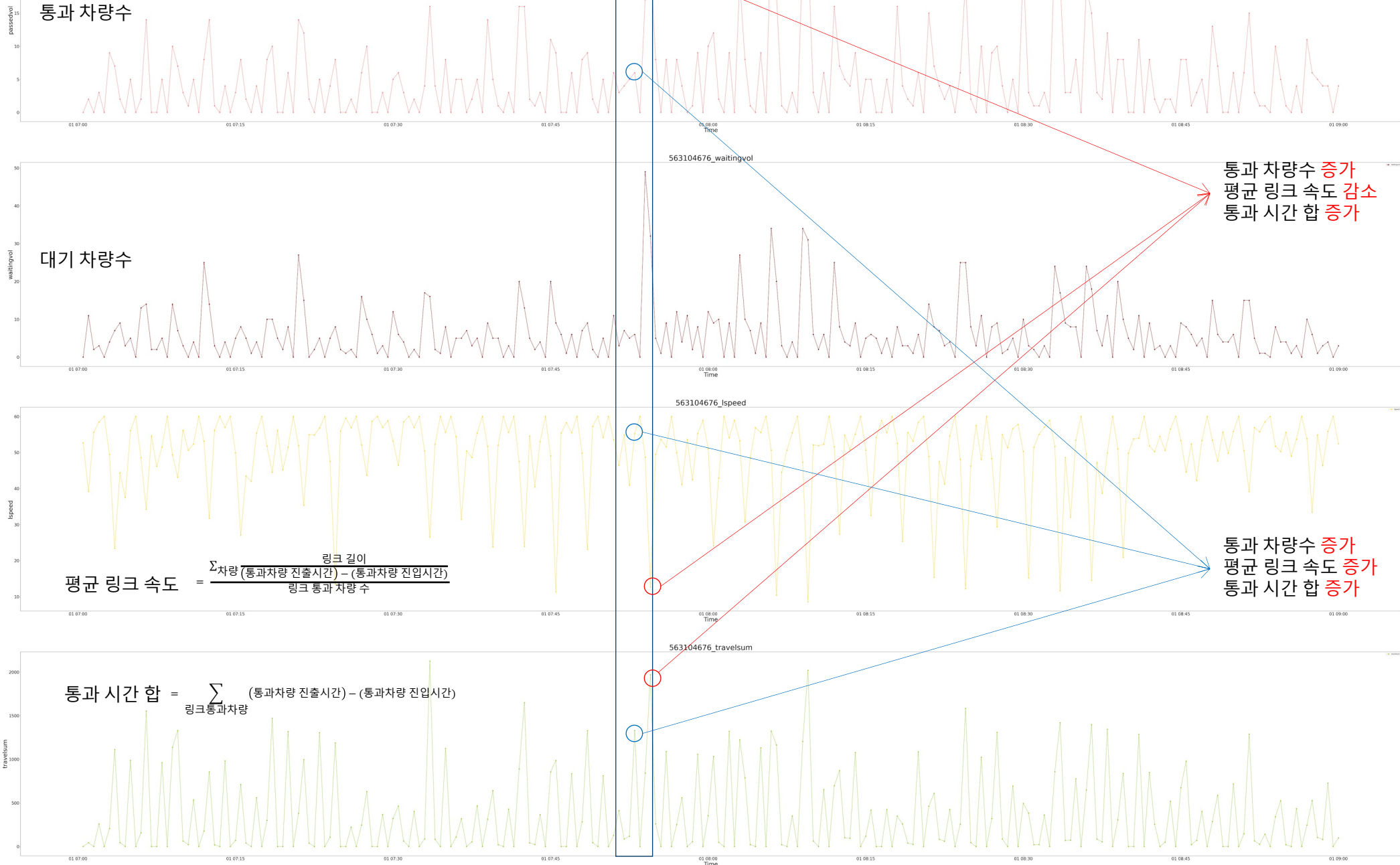
함수	이전 버전	최신 버전
getAverageWaitingQLength	통계 정보 수집 구간 중에 발생한 셀 내 차량의 Waiting Time의 합을 수집 기간으로 나눈 값 ??도로와 평균 대가류 길이	통계 정보 수집 구간 중에 발생한 셀 내 차량의 Waiting Time의 합을 수집 기간으로 나눈 값(링크의 경우, 링크 내 모든 셀에서 계산된 값에 대한 평균)
getAverageVehicleWaitingQLength	?? 최신 신호 변경 이후 도로의 평균차량 대기 큐 길이	최신 신호 변경 이후 해당 셀의 WaitingQ 길이의 합(링크의 경우, 링크 내 모든 셀에서의 값에 대한 총합)
getAverageWaitingTime	?? 도로의 평균대기 시간	통계 정보 수집 구간 중에 셀 내 차량에 대한 평균대기 시간 (= 차량별 대기시간 합 / 차량수)(링크의 경우, 링크 내 모든 셀에서 계산된 값에 대한 평균)
getAverageVehicleWaitingTime	?? 최신 신호 변경 이후 도로의 평균차량 대기 시간	최신 신호 변경 이후 해당 셀 내 차량에 대한 평균대기 시간 (= 차량별 대기시간 합 / 차량수)(링크의 경우, 링크 내 모든 셀에서 계산된 값에 대한 평균)

- 통계 정보 수집 구간 --> 시뮬레이션 시나리오 파일 내 output 산출 주기값에 의해서 정해진 시간 구간(예: 300/600/1800 등)
- 최신 신호 변경 시간 --> 특정 시간 기준을 의미, 즉, 신호 변경 시간이 아니더라도 특정 시간 기준으로 현재 시간까지의 구간 동안 계산 가능함. 단, 셀 내 통과 차량 정보는 현재 통계 정보 수집 구간 기준으로 관리되고 있어, 현재 통계 정보 수집 기간 이전에 통과 완료된 정보는 삭제되어 참고 할 수 없음.
- 평균 대기 시간 계산의 경우, 통계 정보 수집 구간에 의한 정보 소실이 있어, 현재 이를 보완하여, 특정 시간 셀 혹은 링크 내 차량들의 평균 대기 시간 및 대기 시간 합 정보를 제공하는 API가 있음.

함수	의미
getCurrentAverageWaitingTimeBasedVehicle	특정 시간 (예: 현재 타임스텝) 에 셀 혹은 링크 내 차량들의 평균 대기 시간
getCurrentWaitingTimeSumBaseVehicle	특정 시간 (예: 현재 타임스텝) 에 셀 혹은 링크 내 차량들의 대기 시간 합

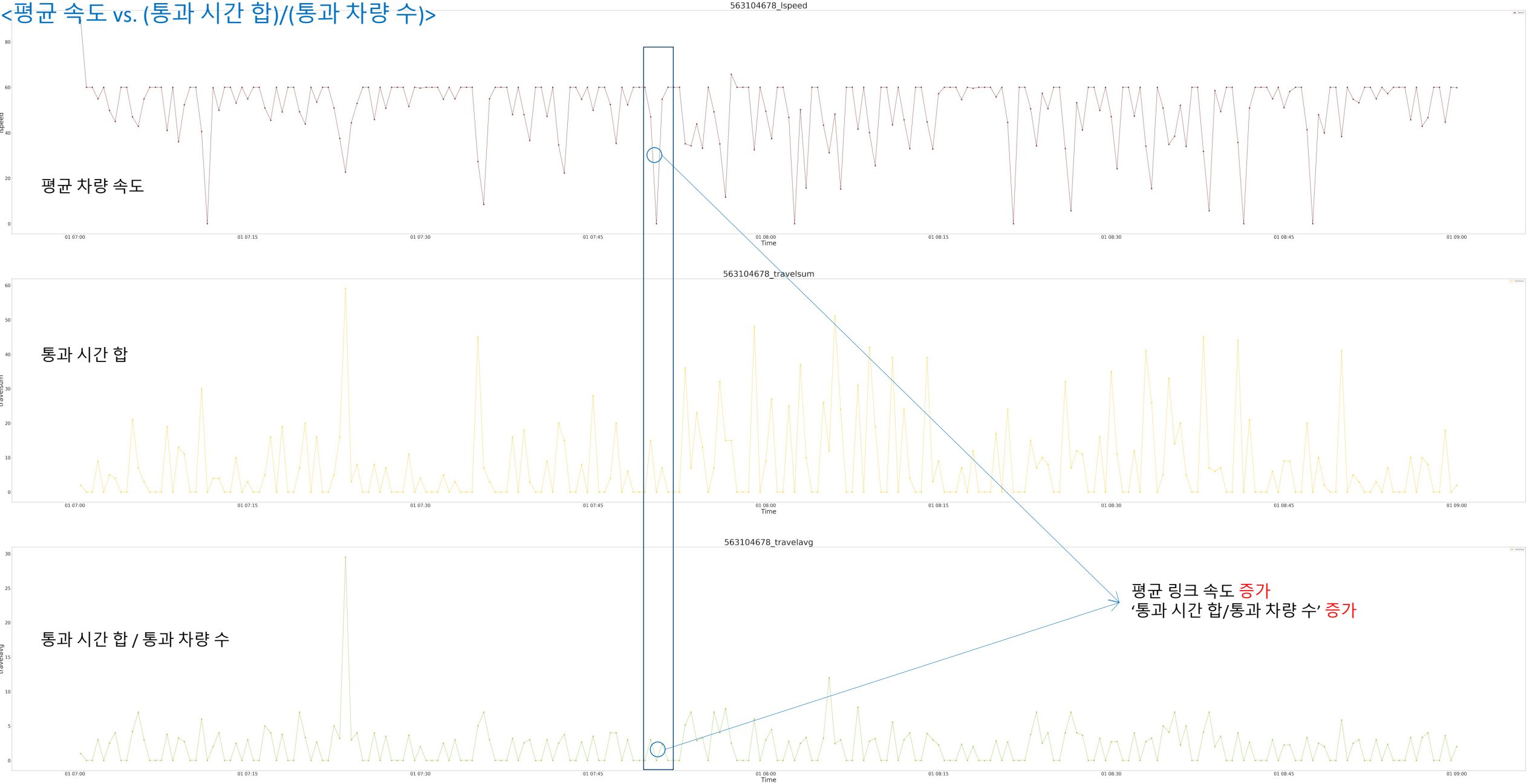
2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<평균 속도 vs. 통과 시간 합 vs. 통과 차량 수>



2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<평균 속도 vs. (통과 시간 합)/(통과 차량 수)>



2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<분석 결과>

- 평균 속도와 통과 시간 합의 경우,
✓ 산출 공식에 의해서, 통과 차량 값의 변동에 의한 영향으로 그 관계가 비례 혹은 반비례로 고정적이지 않음.
- 평균 속도와 통과 시간 합 / 통과 차량 수 의 경우,
✓ 여전히 그 관계가 비례 혹은 반비례로 고정적이지 않음.
- 원인 요소 (복합적)
 - 1) 링크 속도 산출의 기본 단위인 셀 속도 산출 시, 통과 차량에 대한 속도 산출만이 가능한데, 셀 내 대기 차량이 많은 경우가 그 변수로 작용하는 것으로 생각됨.
 - 셀 내 대기차량이 많다가 다음 스텝에서 다음 셀로의 이동이 발생할 경우, 해당 차량의 속도값이 반영되면서 속도가 떨어지는 현상이 있으며, 이 과정에서 정확한 기준값을 알 수는 없지만, 해당 현상으로 통과 시간 값이 영향을 받는 것으로 판단됨.
 - 2) 또한, 현재 링크 속도 산출은 각 셀 별로 산출된 속도들의 취합하여 링크 평균 속도로 계산하나, 링크 통과 시간의 합은 셀 단위가 아닌 직접 링크 단위로 그 값을 산출하게 되어 있어 평균 속도 값 계산 시 반영된 차량이 통과 시간 계산 시에는 반영되지 않는 경우가 있음.

2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<원인 추정 - (1)>

- 교차로 진입 링크들(예: 4거리의 경우, 4개 링크)에 대해서 평균 속도 및 통과 시간 합에 대한 평균값 이용
✓ 예시: 도안5단지네거리 (아래 표 참조)
 - 각 링크별로 비교 시, 평균 속도와 통과 시간 합(혹은 통과 시간 합/통과 차량 수)은 개선률에 이상 없음.
 - 링크들의 평균값 비교 시, 개선률 이슈 발생

		평균속도			통과 시간 합			통과 시간 합 / 통과 차량 수		
		FT	RL	개선률(%)	FT	RL	개선률(%)	FT	RL	개선률(%)
도안5단지네거리	563104785	34.85366667	32.96942857	-5.40614023	441.4571429	474.5142857	-7.488188467	55.04627665	67.14757592	-21.98386524
	-563104366	4.169666667	3.808238095	-8.668044722	454.9571429	552.7047619	-21.4850169	58.41298139	85.58360362	-46.51469859
	563104782	34.95566667	38.26652381	9.471589183	440.6190476	350.0142857	20.56306063	42.72670048	36.97776782	13.45512899
	563104734	14.78242857	14.56752381	-1.453785221	51.7	60.87142857	-17.7397071	24.84166667	31.60261905	-27.21617866

		평균속도			통과 시간 합			통과 시간 합 / 통과 차량 수		
		FT	RL	개선률(%)	FT	RL	개선률(%)	FT	RL	개선률(%)
도안5단지네거리		22.19035714	22.40292857	0.957945053	347.1833333	359.5261905	-3.555141033	45.25690629	55.3278916	-22.25292476

평균 속도 값의 변화 정도가, 통과 시간 값의 변화 정도에 비해서 절대값이 작을 확률이 높음
→ 각 링크별 변화도에 따라, 전체 평균값으로 계산했을 때, 상쇄효과가 발생할 수 있음.

2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<원인 추정 - (2)>

- 평균 속도 산출 시,
 - 1) 셀 기반 산출 속도의 평균값으로 제공
 - 링크 통과하지 않았으나, 셀 통과한 차량에 의해 속도값이 반영될 수 있음
 - 2) 통과 차량이 없는 경우, 대기 차량 유무에 따라 속도 산출값 편차가 클 수 밖에 없는 현상 존재
 - 통과 차량이 없는 경우, 셀 내 대기 차량이 있을 경우, 속도는 0, 셀 내 대기 차량이 없는 경우, Freeway 속도 즉 제한속도 (예: 50km/h)
 - 3) 링크 내 차량 포화도에 따라 산출 속도의 편차/오차가 커질 확률이 있음
 - 링크 내 차량 포화도가 커서, 정체(대기) 중인 차량이 많을 수록 속도값 산출에 영향을 많이 주게 됨.

속도 산출 시, 속도값의 상쇄(?) 현상 발생 가능성
→ 속도 vs. 통과 시간 관계에 어떤 영향을 미치는지??

속도 개선률과 통과 시간 개선률이 다른 링크 여전히 존재
→ 평균 속도 산출 시 다른 요소/환경들이 달라 효과 상쇄가 발생하는지에 대한 확인이 필요할듯.....

		평균속도			통과 시간 합			통과 시간 합 / 통과 차량 수		
		FT	RL	개선률(%)	FT	RL	개선률(%)	FT	RL	개선률(%)
도안네거리	553810834	31.90095238	28.42242857	-10.90413781	165.752381	178.2952381	-7.567225925	22.8522572	23.76143396	-3.978498751
	553810859	44.30047619	44.184	-0.262923112	35.28571429	24.85238095	29.56815115	17.62190476	11.81420635	32.95726819
	-553810865	48.99495238	50.36638095	2.799122164	66.20952381	63.6952381	3.797468354	30.55916667	28.70142857	6.079151685
	553810845	16.04904762	14.69580952	-8.431890336	127.7428571	105.0047619	17.79989562	12.92867146	10.4197784	19.40565247

		평균속도			통과 시간 합			통과 시간 합 / 통과 차량 수		
		FT	RL	개선률(%)	FT	RL	개선률(%)	FT	RL	개선률(%)
도안네거리		35.31135714	34.41715476	-2.532336487	98.74761905	92.96190476	5.859092443	20.99050002	18.67421182	11.03493578

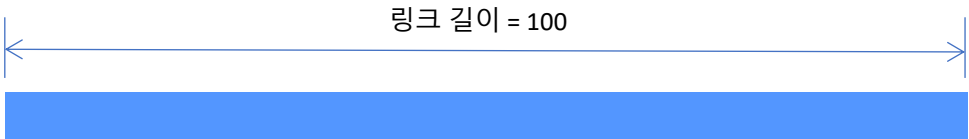


2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<개선률 이슈에 대한 분석 결과>

- 평균 속도 및 통과 시간 합 산출 방식 상 발생할 수 있는 현상 --> 이슈 아님 (아래 예시 참고)

<예시: 평균속도 vs. 통과시간합 관계에 대한 수식 기반 비교>



[차량 별 링크 통과 시간 테이블]

sim: 시뮬레이션
v: 차량

	v1	v2	v3	v4	v5	v6	v7	v8	v9
sim1	100	30	40	100	190	50	50	40	30
sim2	90	40	30	100	100	45	60	50	30

차량통과시간합(sim1) = 630
차량통과시간합(sim2) = 543

평균 속도(sim1) = 2.02
평균 속도(sim2) = 2.01

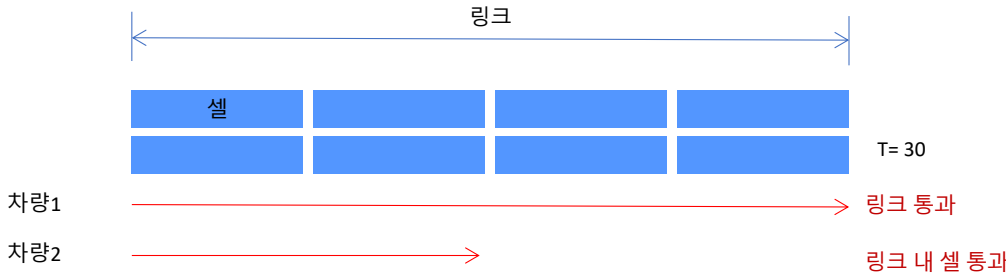
※ 평균 속도 = (차량별 속도) 합 / 차량 수

sim1 vs. sim2 차량통과시간합: 630 -> 543 (감소)
 평균속도: 2.02 -> 2.01 (감소)

2. 시뮬레이션 결과 (평균속도 vs. 통과시간) 관계 분석

<시뮬레이션 통계값 산출 시, 산출 기준 시간 구간 (예: 30초, 150초, 300초 등) 에 따른 오차 발생 영향을 최소화 이슈에 대한 분석 결과>

- ❖ 30초 산출 시, 링크 내 시뮬레이션 통계값의 경우, 링크 길이가 길 수록 30초 내 통과 못하는 차량 비율이 커지므로 통계값 산출 시 오차가 발생함.

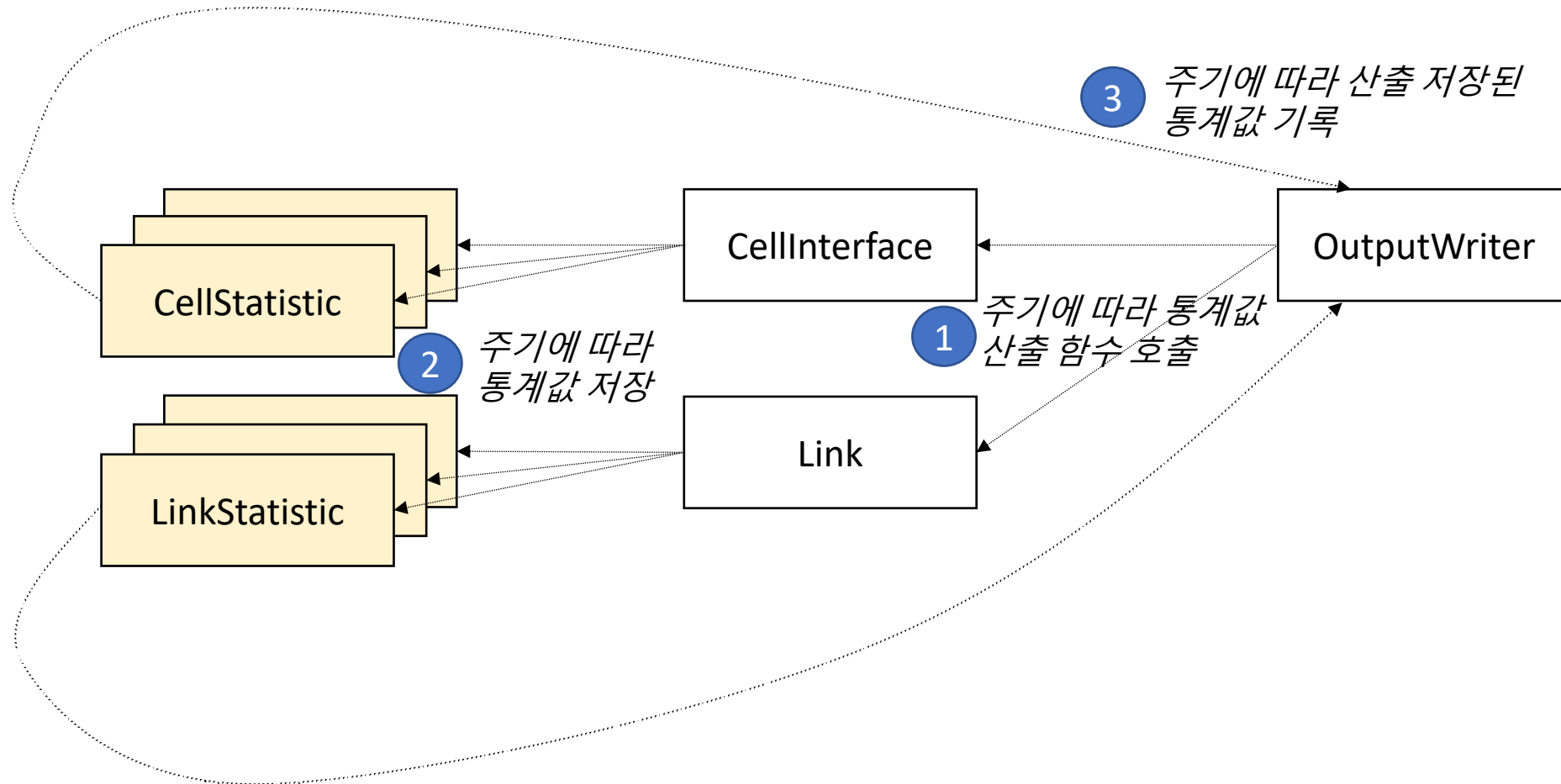


- [링크 기준]
- 통과 차량 수, 통과 시간 합 → 차량 1만 적용
 - 평균 속도, 정체/대기 시간 → 차량 1, 차량 2 적용

- ❖ 해당 오차가 최소화 되기 위해서는 적당히 긴 통계값 산출 시간 구간을 설정할 필요가 있음.
 - (이슈) 현재 시뮬레이터는 통계값 산출 시간 구간 하나만 적용 가능
 - ✓ 최적화 모델은 최적화 모델 적용 시 최적화에 필요한 정보 수집 시간 구간이 존재하는데, 현재 시뮬레이터 기능 상 해당 시간 구간을 통계값 산출 시간 구간과 동일하거나 그 이상으로 유지하고 있음.
 - (고민 중) 서로 다른 산출 기간을 만족하도록 시뮬레이터 기능 보완이 필요할까?? --> 검토 필요함.
 - ✓ 서로 다른 산출 기간을 만족하기 위해서, 기본적으로 셀별 차량 통과 정보 (셀별로 map으로 관리)를 기존과 같이 통계값 산출 기간 기반으로 관리하지 않고, 계속 유지할 필요가 있음.
 - 2시간 단위 교통 수요와 도안 범위 도로망 대상 시뮬레이션일 경우,
 - 셀별 차량 통과 정보 맵을 모두 유지한 상태로 시뮬레이션 돌릴 때, 메모리 사용량은 530~540MB 정도까지 늘어남
 - 통계구간별 메모리 사용량: 30초 --> 280M, 180초 --> 290M, 300초 --> 310MB
 - ✓ (방법1) 1) 셀별 차량 통과 정보를 메모리 사용량을 고려하여 현재 스텝 기준 특정 기간까지 유지하도록 시뮬레이터 변경하고, 2) 동적 인터페이스를 통해서 원하는 기간의 통계값을 계산하여 반환하는 기능 지원
 - 이 경우, 동적 인터페이스를 통해 시뮬레이션 되고 있는 링크/셀에 대한 raw 데이터(예: 셀별 차량 통과 정보)를 기반으로 계산하여 반환하는 함수를 호출하는 경우, 전체 시뮬레이션 수행 시간이 느려지는 원인이 될 수 있음.
 - ✓ (방법2) 1) 셀별 차량 통과 정보를 메모리 사용량을 고려하여 현재 스텝 기준 특정 기간까지 유지하도록 시뮬레이터 변경하고, 2) 시뮬레이션 입력 파라미터로 다른 통계값 구간 기준 정보를 입력받고, 시뮬레이션 수행하면서 해당 기간을 반영하여 계산/저장해두는 기능을 별도 추가
 - 예: SUMO의 simulated detector 와 유사한 기능 (정해진 링크/셀에 loop 같은 probe 포인트를 지정하고, 해당 probe 포인트에서의 정보를 별도 산출하는 기능 지원.)
 - (최종 시뮬레이션 결과 비교 시 고려사항) 시뮬레이션 시, 현재 교통수요 (2시간) 의 경우, 시뮬레이션 시작(시뮬레이션 시작에는 모든 링크 내 차량이 없는 상태에서 일정 수준의 통행차량이 채우기 위한 과정이 존재) 과 시뮬레이션 종료 (종료 시간에 대부분의 차량이 운행 종료하고 있어 링크 내 차량이 대부분 이탈하는 현상 존재)부근에서 시뮬레이션 오차 발생함.
 - ✓ 시뮬레이션 결과에 대한 유효한 비교를 위해서, 시뮬레이션 시간 중 시작 후 30분과 종료 전 30분 정도는 제외하고 나머지 시간에 대해서 통계값 산출하는 것이 타당함.

3. 통계값 산출 방식 변경

- ❖ CellStatistic 과 LinkStatistic 객체를 정의하고, 해당 통계주기가 다를 시, 각 주기별로 수집/통계/산출되는 값에 대한 객체를 생성하여 활용하는 방식으로 변경
- ❖ (성능이슈발생) 기존 CellInterface 혹은 Link 객체 내 멤버변수로 산출되던 값을 객체 생성을 통해 각각 관리하게 됨에 따라 속도 이슈가 발생
 - 도안 지역 도로망 + 2시간 수요 기준 --> 기존 수행속도: 40초, 변경 수행속도: 280초



3. 신호데이터 적용 오류 해결

- ❖ 이슈: 시뮬레이션 시작 시, Offset에 의한 이니셜 신호 ScheduleID 및 신호 PhaseIndex 값에 오류 발생
 - 문제점: 25200 (오전 7시) 혹은 61200 (오후 5시) 시작하는 시뮬레이션 수행 시, 아래와 같은 2가지 오류 현상 발생

```
(1)
[Jump currentStep from 0 to 25200]
[25200] TL: cluster_563103437_563103890_563103913_563103914 - CurrentScheduleID: 2 - CurrentPhaseIndex: 2
[25201] TL: cluster_563103437_563103890_563103913_563103914 - BeforeScheduleID: 0 - CurrentScheduleID: 2 - ScheduleTime: 0
[25201] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 0 - CurrentPhaseIndex: 3 - PhaseTime: 0
[25231] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 3 - CurrentPhaseIndex: 4 - PhaseTime: 30
[25234] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 4 - CurrentPhaseIndex: 5 - PhaseTime: 3
[25260] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 5 - CurrentPhaseIndex: 6 - PhaseTime: 26
```

<-- Offset 적용에 의해서, InitialPhaseIndex=2, InitialNextPhaseTime=28 로, 'BeforePhaseIndex: 2 - CurrentPhaseIndex: 3 - PhaseTime:28' 이 되어야 함.

```
<schedule offset="146" id="2">
  <phase state="grrrGgrrrrrrrr" duration="26" minDur="23" maxDur="42"/>
  <phase state="yrrryrrrrrrrrr" duration="3"/>
  <phase state="rGGGrrrrrGGGrrrr" duration="33" minDur="25" maxDur="60"/>
  <phase state="rGGGrrrrrGGGrrrr" duration="30" minDur="22" maxDur="57"/>
  <phase state="ryyyrrrrrrrrrr" duration="3"/>
  <phase state="rrrrrrrgrrrGgrr" duration="26" minDur="23" maxDur="42"/>
  <phase state="rrrrrrrgrrrygrr" duration="3"/>
  <phase state="rrrrrrGGgrrrrgrr" duration="23" minDur="20" maxDur="37"/>
  <phase state="rrrrrryyrrryrr" duration="3"/>
  <phase state="grrrrgrrrrrrrGG" duration="27" minDur="20" maxDur="37"/>
  <phase state="grrrrgrrrrrrryy" duration="3"/>
</schedule>
```

```
(2)
[Jump currentStep from 0 to 61200]
[61200] TL: cluster_563103437_563103890_563103913_563103914 - CurrentScheduleID: 1 - CurrentPhaseIndex: 0
[61201] TL: cluster_563103437_563103890_563103913_563103914 - BeforeScheduleID: 0 - CurrentScheduleID: 2 - ScheduleTime: 0
[61201] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 0 - CurrentPhaseIndex: 2 - PhaseTime: 0
[61229] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 2 - CurrentPhaseIndex: 3 - PhaseTime: 28
[61259] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 3 - CurrentPhaseIndex: 4 - PhaseTime: 30
[61262] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 4 - CurrentPhaseIndex: 5 - PhaseTime: 3
[61288] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 5 - CurrentPhaseIndex: 6 - PhaseTime: 26
```

<-- 해당 시간에는 신호 ScheduleID=5 가 적용되어야 함.

```
<TODPlan offset="157" defaultPlan="1">
  <plan offset="146" schedule="2" startTime="25200"/>
  <plan offset="144" schedule="3" startTime="34200"/>
  <plan offset="157" schedule="5" startTime="61200"/>
  <plan offset="157" schedule="6" startTime="75600"/>
```

3. 신호데이터 적용 오류 해결

❖ 원인 분석 / 해결

- class TrafficSignal / class TrafficSignalSchedule 에서 이니셜 신호 상태 설정하는 함수 분석 진행 --> void TrafficSignal::initTrafficSignalStatus(std::string defaultScheduleID) 함수에서 시뮬레이션 시작 시간을 '0'으로 하드 코딩되어 있음을 발견함.
- 시뮬레이션 시작 시간을 반영하여, 시뮬레이션 시작 시점 시, 신호 ScheduleID / 신호 PhaseIndex 설정하도록 코드 변경 (git: <https://github.com/etri-city-traffic-brain-jr/traffic-simulator.git> 반영 완료)
- 변경 코드 적용 시, 정상 동작 확인 완료

```
[Simulation Start]
```

```
[Jump currentStep from 0 to 61200]
```

```
[61200] TL: cluster_563103437_563103890_563103913_563103914 - CurrentScheduleID: 5 - CurrentPhaseIndex: 0
```

```
[61201] TL: cluster_563103437_563103890_563103913_563103914 - BeforeScheduleID: 0 - CurrentScheduleID: 5 - ScheduleTime: 0
```

```
[61204] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 0 - CurrentPhaseIndex: 1 - PhaseTime: 3
```

```
[61207] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 1 - CurrentPhaseIndex: 2 - PhaseTime: 3
```

```
[61242] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 2 - CurrentPhaseIndex: 3 - PhaseTime: 35
```

```
[61274] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 3 - CurrentPhaseIndex: 4 - PhaseTime: 32
```

```
[61277] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 4 - CurrentPhaseIndex: 5 - PhaseTime: 3
```

```
[61303] TL: cluster_563103437_563103890_563103913_563103914 - BeforePhaseIndex: 5 - CurrentPhaseIndex: 6 - PhaseTime: 26
```

```
<schedule offset="157" id="5">
```

```
  <phase state="grrrGgrrrrrrrr" duration="26" minDur="23" maxDur="42"/>    <-- Offset 적용 시, InitPhaseIndex=0, InitNextPhaseTime=3
```

```
  <phase state="yrrryyrrrrrrrr" duration="3"/>
```

```
  <phase state="rGGGrrrrrGGGrrrr" duration="35" minDur="25" maxDur="60"/>
```

```
  <phase state="rGGGrrrrrGGGrrrr" duration="32" minDur="22" maxDur="57"/>
```

```
  <phase state="ryyyrrrrryyyrrrr" duration="3"/>
```

```
  <phase state="rrrrrrrrrrGgrr" duration="26" minDur="23" maxDur="42"/>
```

```
  <phase state="rrrrrrrrrrygyrr" duration="3"/>
```

```
  <phase state="rrrrrGGgrrrrrgr" duration="23" minDur="20" maxDur="37"/>
```

```
  <phase state="rrrrryyyrrrryrr" duration="3"/>
```

```
  <phase state="grrrrgrrrrrrrrGG" duration="23" minDur="20" maxDur="37"/>
```

```
  <phase state="grrrrgrrrrrrryy" duration="3"/>
```

```
</schedule>
```