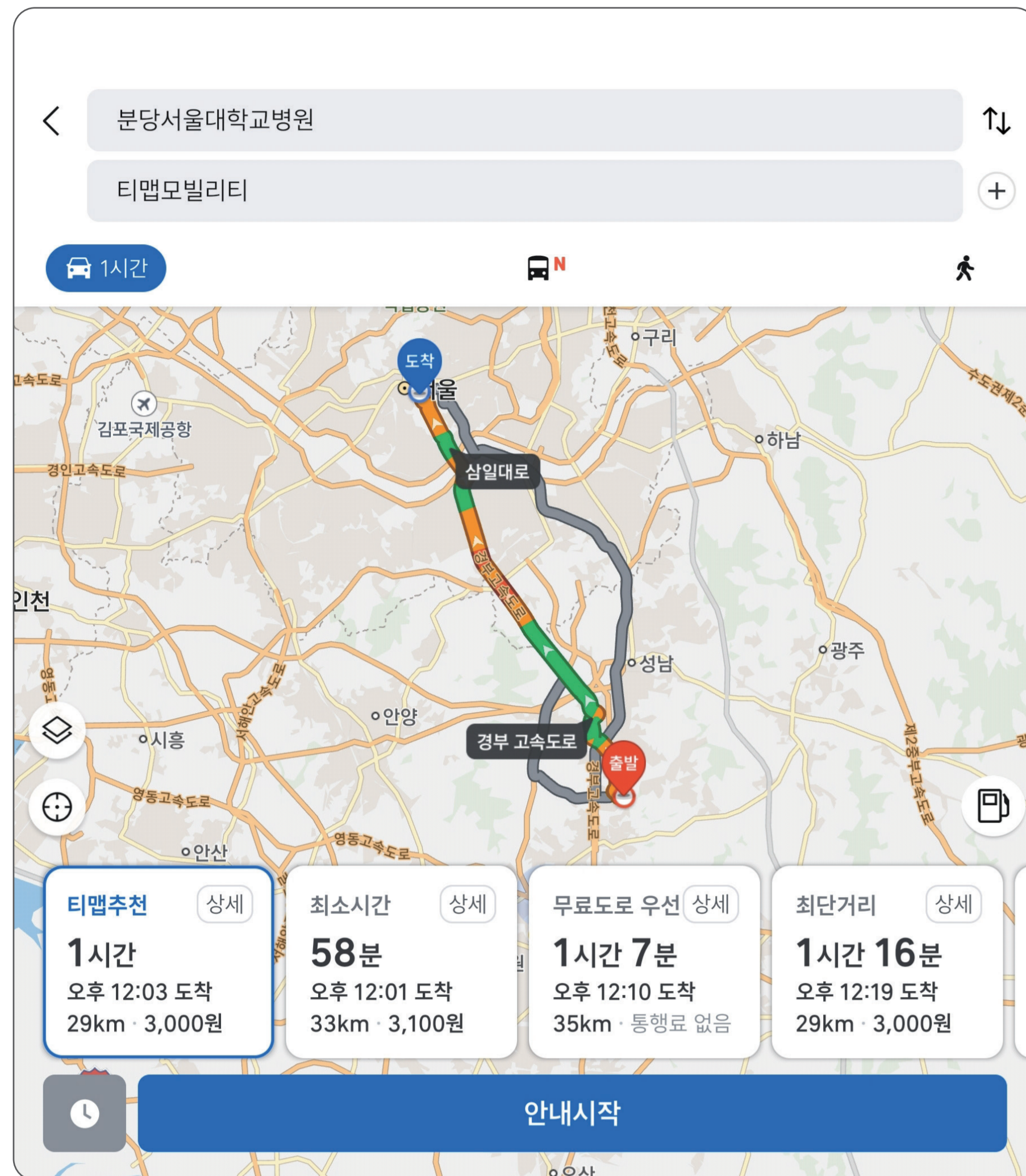


티맵 경로탐색엔진과 CCH 알고리즘 소개

티맵 경로탐색 엔진 소개



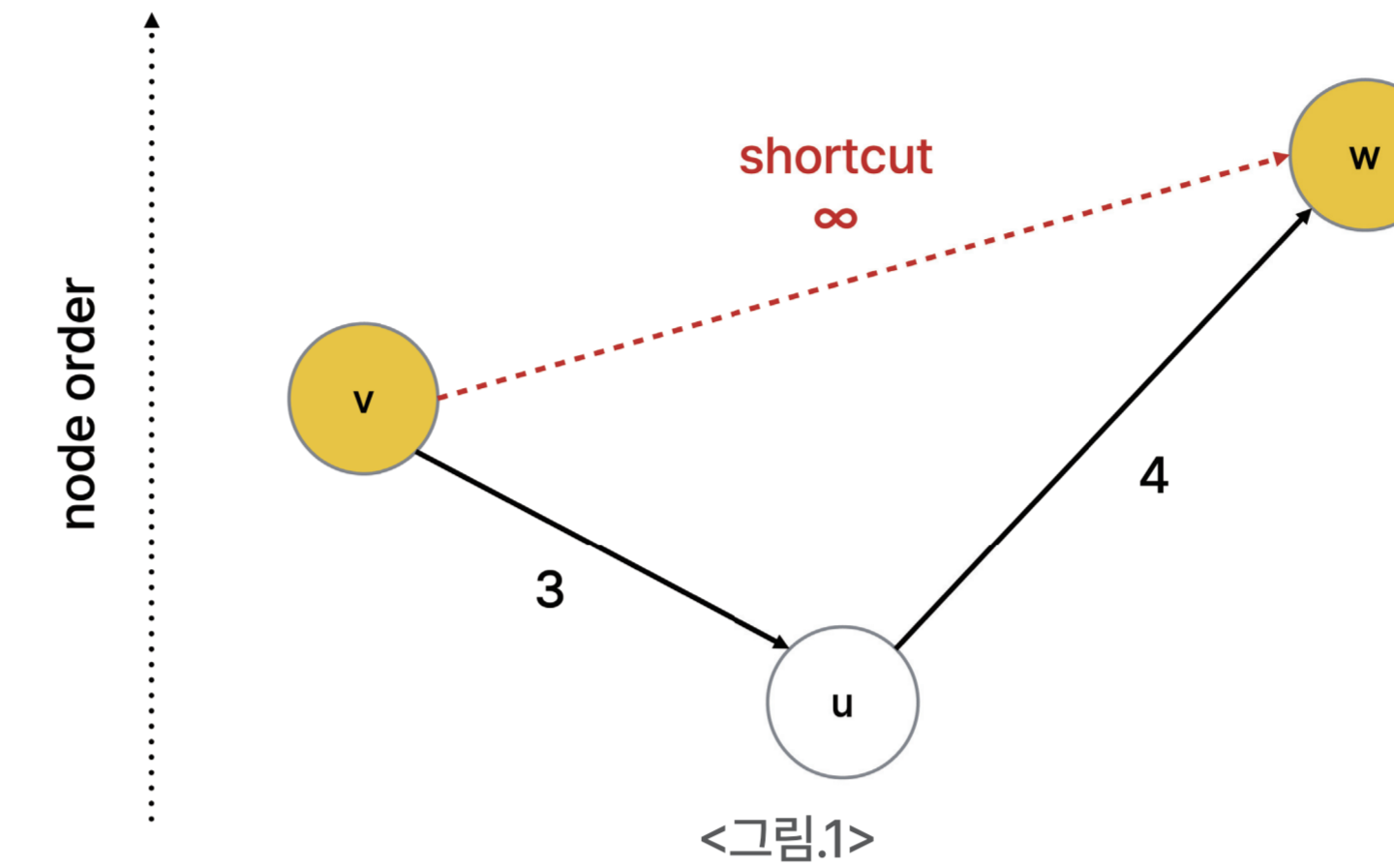
티맵에서는 경로탐색을 수행할 때, 각 옵션(cost metric) 별로 여러 경로탐색 결과를 보여줍니다.

기존에는 전통적인 경로탐색 알고리즘인 A* 알고리즘을 기반으로 개발이 되었습니다. A* 는 모든 node를 방문하면서 탐색하는 방식이기 때문에 graph의 크기가 커질수록 성능은 느려지는 경향이 있습니다.

오랜 기간동안 티맵 서비스를 제공하면서 많은 도로들이 개통되고, 도로들이 상세화 됨으로 인해 network의 크기는 지속적으로 커지고 있었습니다. 스마트폰의 보급 등으로 인해 티맵 서비스를 사용하는 유저수도 폭발적으로 늘어남으로 인해 점점 서버에 부하가 걸리기 시작하였습니다.

이를 해결하기 위해 저희는 최신 논문을 기반으로한 CCH(Customizable Contraction Hierarchies) 및 CATCHUp(Customizable Approximated Time-dependent Contraction Hierarchies through Unpacking) 알고리즘 등을 도입하여 성능을 개선하였습니다.

CCH - Step.1 Preprocessing (metric independent)



기본 아이디어는 CH(Contraction Hierarchies) 알고리즘과 유사합니다.

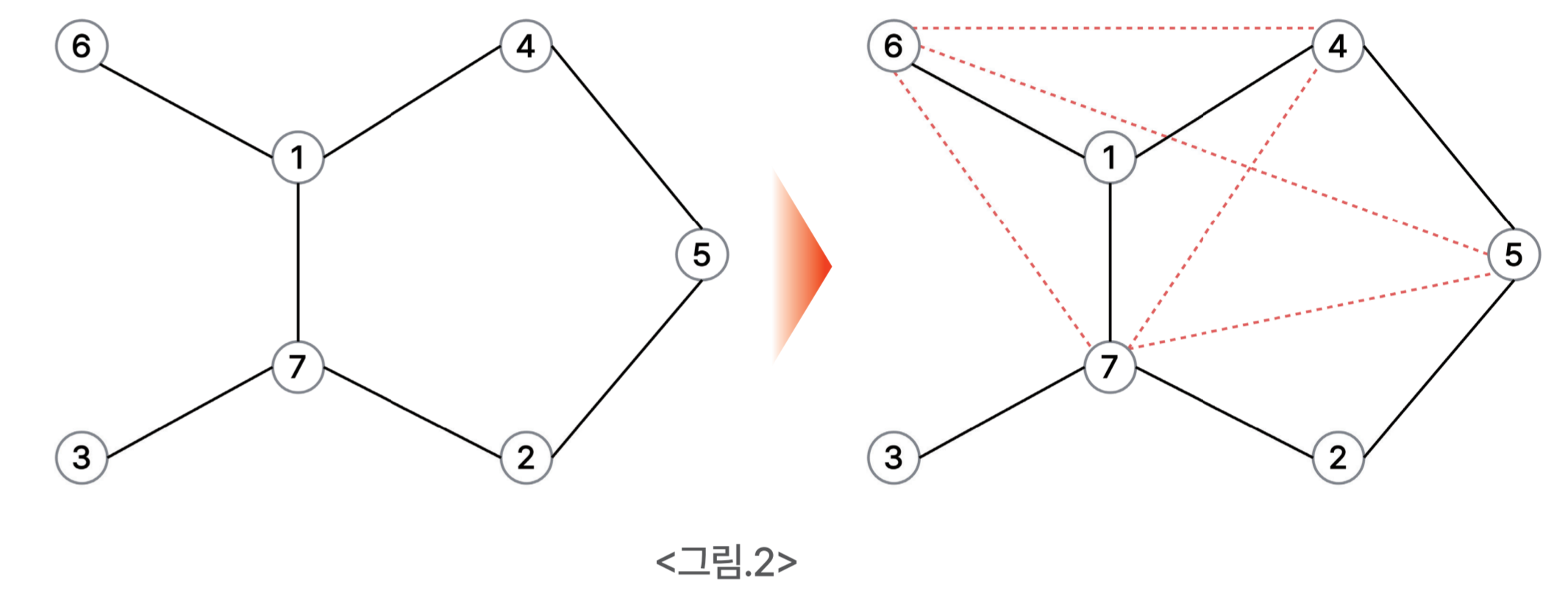
노드를 중요도(rank) 순서로 정렬한 후, <그림.1>과 같이 rank가 $v > u, u < w$ 인 조건을 만족하는 경우 $v \rightarrow w$ 에 대한 shortcut을 생성하고, $v \rightarrow w$ shortcut의 middle node인 u 를 매핑합니다. 그리고 shortcut의 cost는 무한대로 설정합니다.

이와 같이 $v \rightarrow u, u \rightarrow w$ 에 대한 shortcut $v \rightarrow w$ 를 만드는 작업을 contraction 작업이라 합니다.

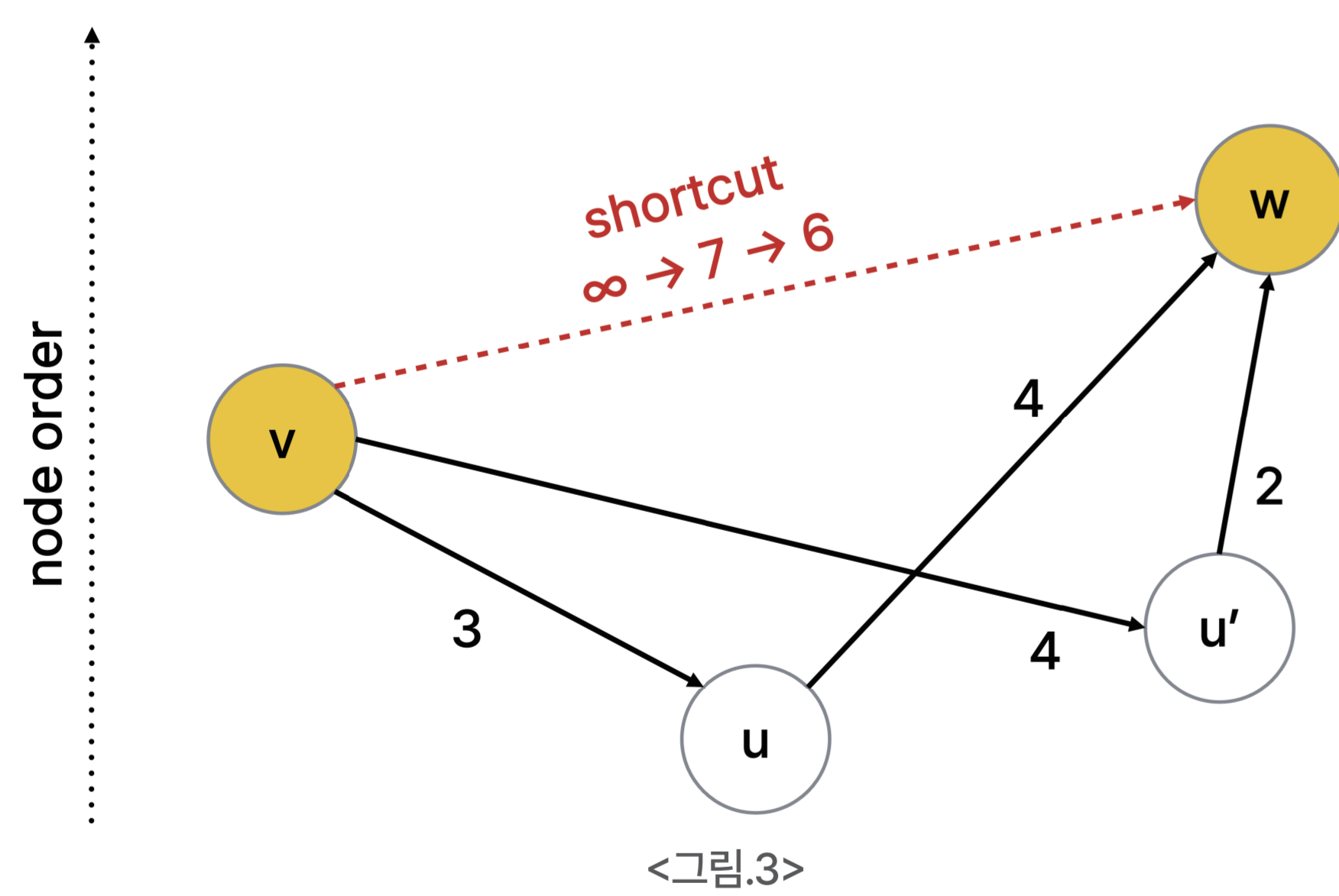
해당 작업은 Graph Topology가 변경될 때마다, 즉 맵데이터가 변경될 때마다 수행합니다. 티맵은 보통 1~2주에 한번씩 수행됩니다.

서비스에 영향이 없기 때문에 오랜 시간이 걸리더라도, 최대한 모든 연산을 preprocessing 단계에서 미리 수행해 놓는 것이 효율적입니다.

이와 같은 작업이 수행되면 <그림.2>와 같은 Graph가 생성되는데, 이를 Chordal Graph라고 합니다.



CCH - Step.2 Customization (metric dependent)



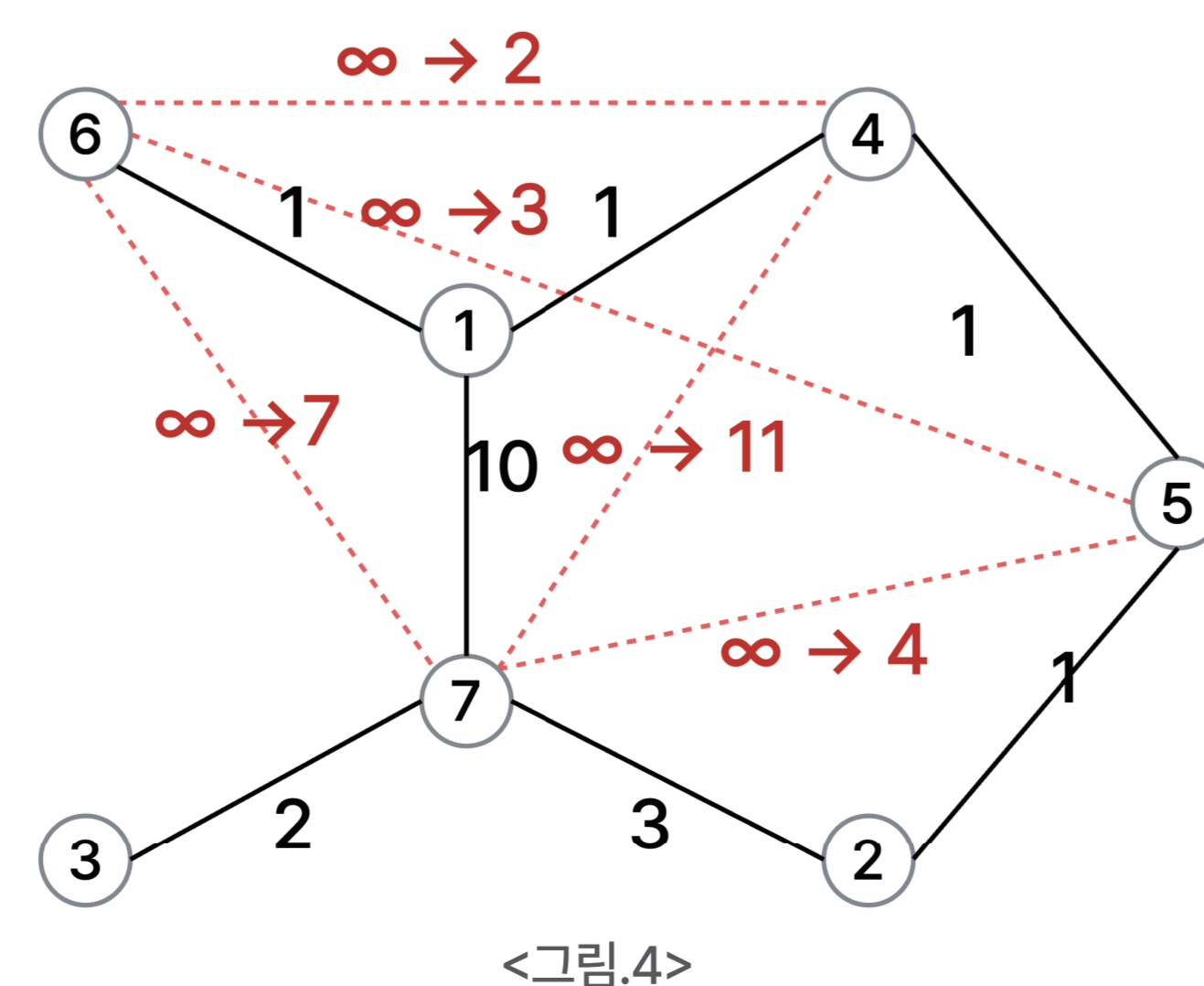
하나의 shortcut $v \rightarrow w$ 에는 여러개의 middle node (u, u')가 있을 수 있습니다.

각 metric 별로 original graph의 cost를 업데이트한 후, shortcut cost를 업데이트합니다.

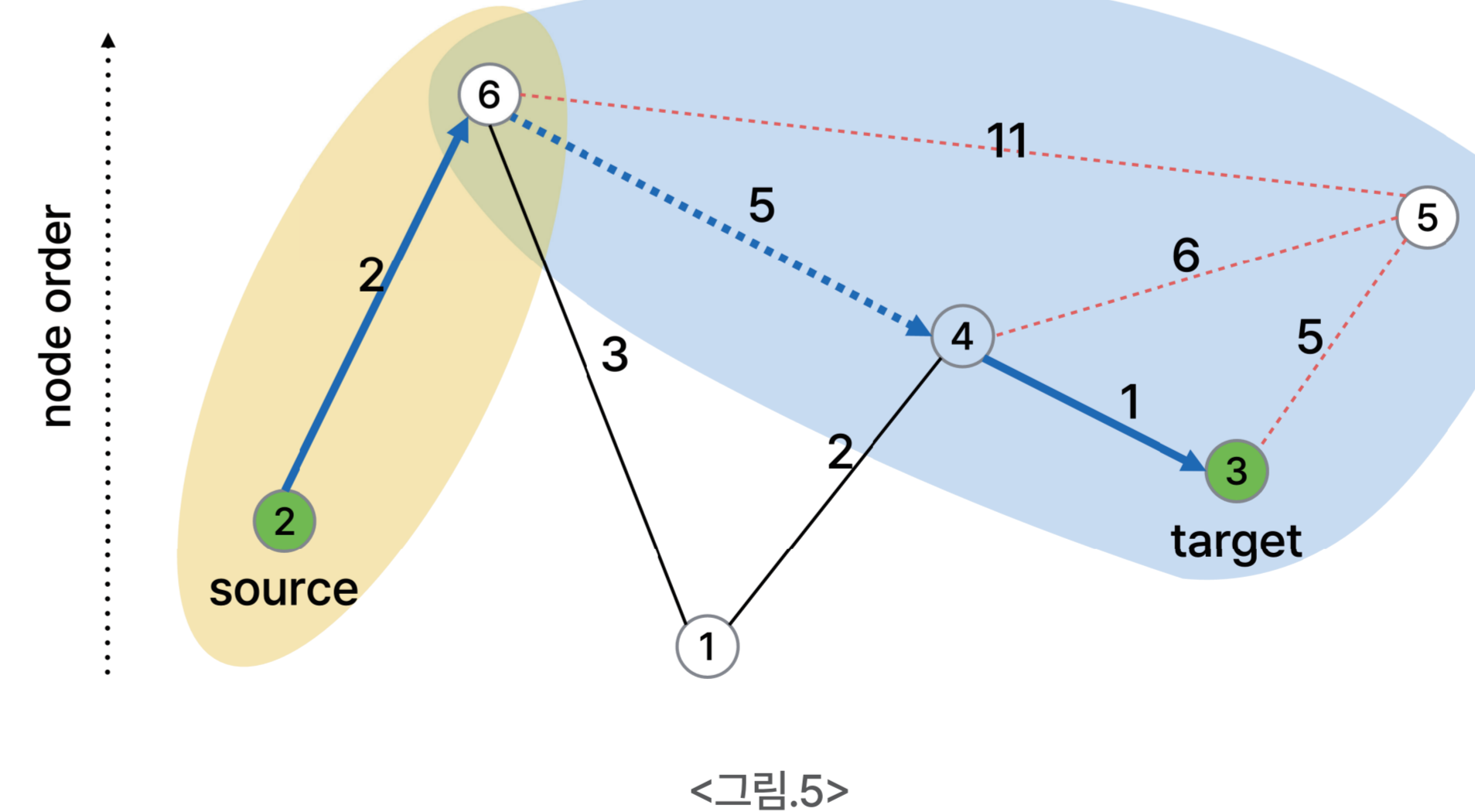
<그림.3>은 $v \rightarrow w$ shortcut에 대해 업데이트 되는 모습입니다. $v \rightarrow u \rightarrow w$ 를 통하여 $\infty \rightarrow 7$ 으로 업데이트 되고, $v \rightarrow u' \rightarrow w$ 를 통하여 $7 \rightarrow 6$ 으로 업데이트 됩니다.

도로 정보는 실시간 교통정보, 유고정보 등으로 인해 지속적으로 변화합니다. 그로인해 각 cost metric 정보도 주기적으로 업데이트 되고, 경로는 변화됩니다.

Customization 작업은 <그림.4>와 같이 node의 중요도(rank) 기준으로 정렬하고 low rank \rightarrow high rank 순서대로 순회하며 인접한 shortcut에 대해 cost metric을 업데이트 합니다



CCH - Step.3 Path Query



Shortcut이 포함된 Graph를 2개의 Graph로 나눕니다.

1. Upward Graph ($u \rightarrow v \mid u < v$): node의 중요도(rank)가 $u < v$ 인 arc만을 대상으로 함.
2. Downward Graph ($u \rightarrow v \mid u > v$): node의 중요도(rank)가 $u > v$ 인 arc만을 대상으로 함.

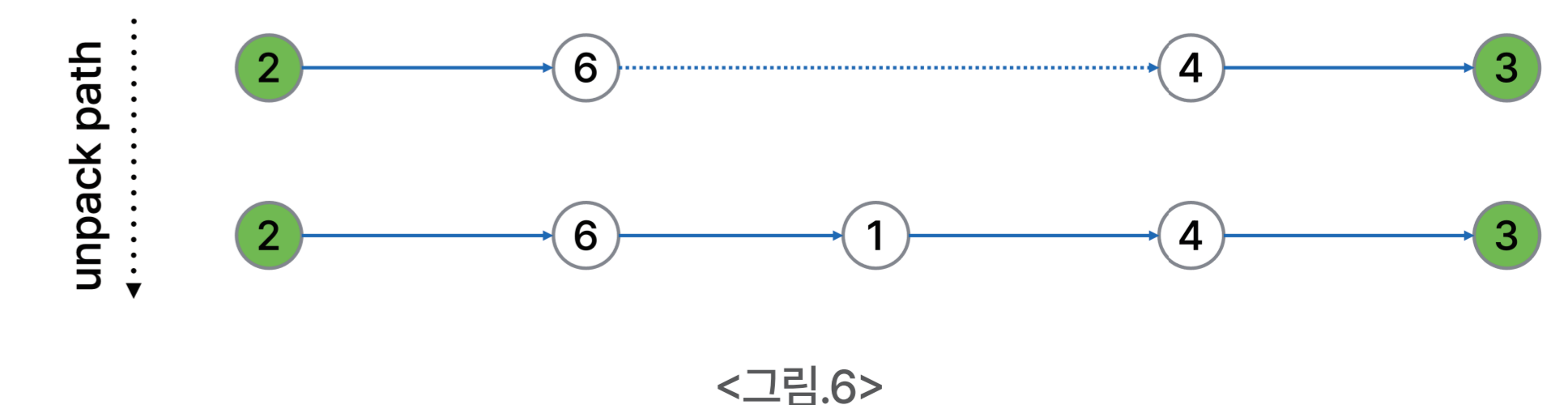
Query 알고리즘은 잘 알려진 bidirectional dijkstra를 이용합니다. (elimination tree도 이용 가능)

<그림.5>를 살펴보면 upward graph의 source node에서 forward search를 수행하고, backward graph의 target node에서 backward search를 수행합니다.

탐색 후 meeting node인 6을 통해 shortest-path를 얻을 수 있습니다.

Query를 통해 shortest-path $2 \rightarrow 6 \rightarrow 5 \rightarrow 4 \rightarrow 3$ 을 얻을 수 있습니다.

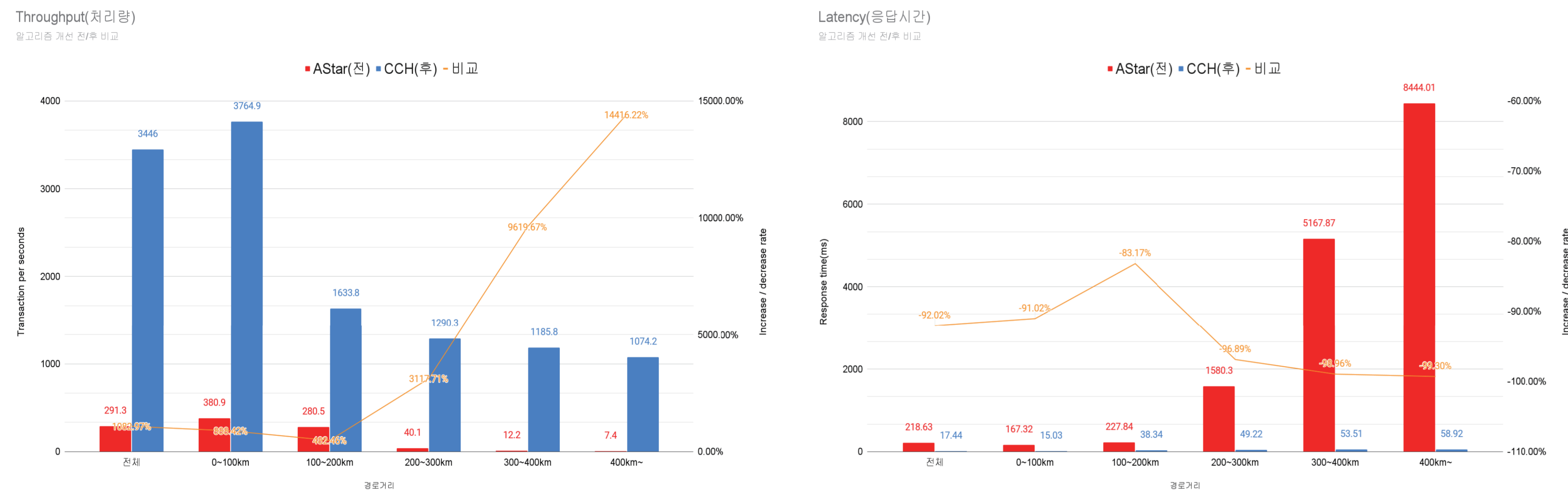
이는 shortcut을 포함하고 있기 때문에 <그림.6>과 같이 path unpacking 과정을 수행하여 최종 결과를 얻습니다.



성능개선과 CCH 알고리즘확장

알고리즘 변경 전/후 성능비교

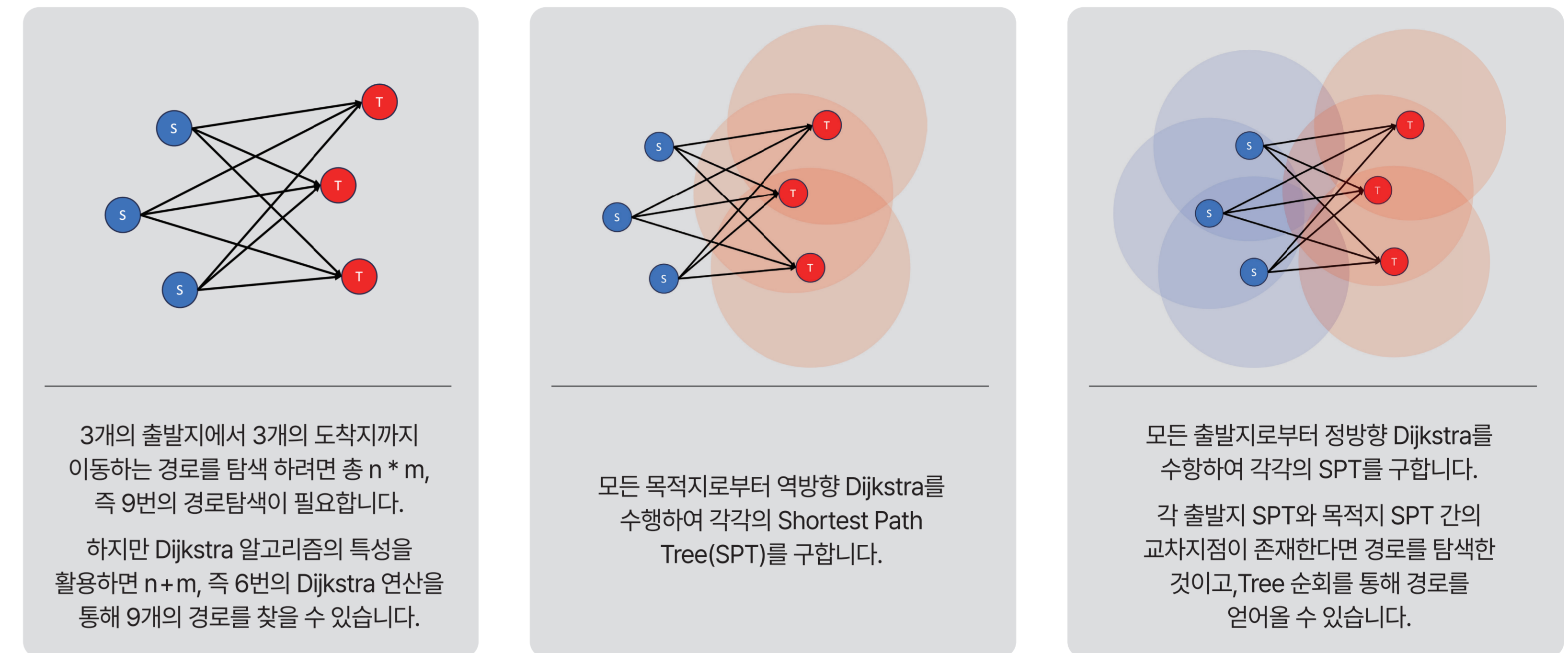
CCH 알고리즘 적용 후 AStar 대비 성능이 얼마나 개선 되었는지 측정하기 위해 동일 장비에서 테스트를 수행하였습니다. 테스트 데이터셋은 평소 일주일 동안의 실제 사용자 경로탐색 요청 정보를 랜덤 샘플링 했습니다.



성능 측정 결과 처리량, 응답시간 모두 큰 폭으로 개선되었습니다. 특히 장거리 경로의 경우 100배 이상 차이가 벌어지기도 합니다. 사용자는 장거리 경로일 수록 기존보다 쾌적한 사용 경험을 할 수 있으며 서버 운영 비용은 대폭 절감할 수 있습니다.

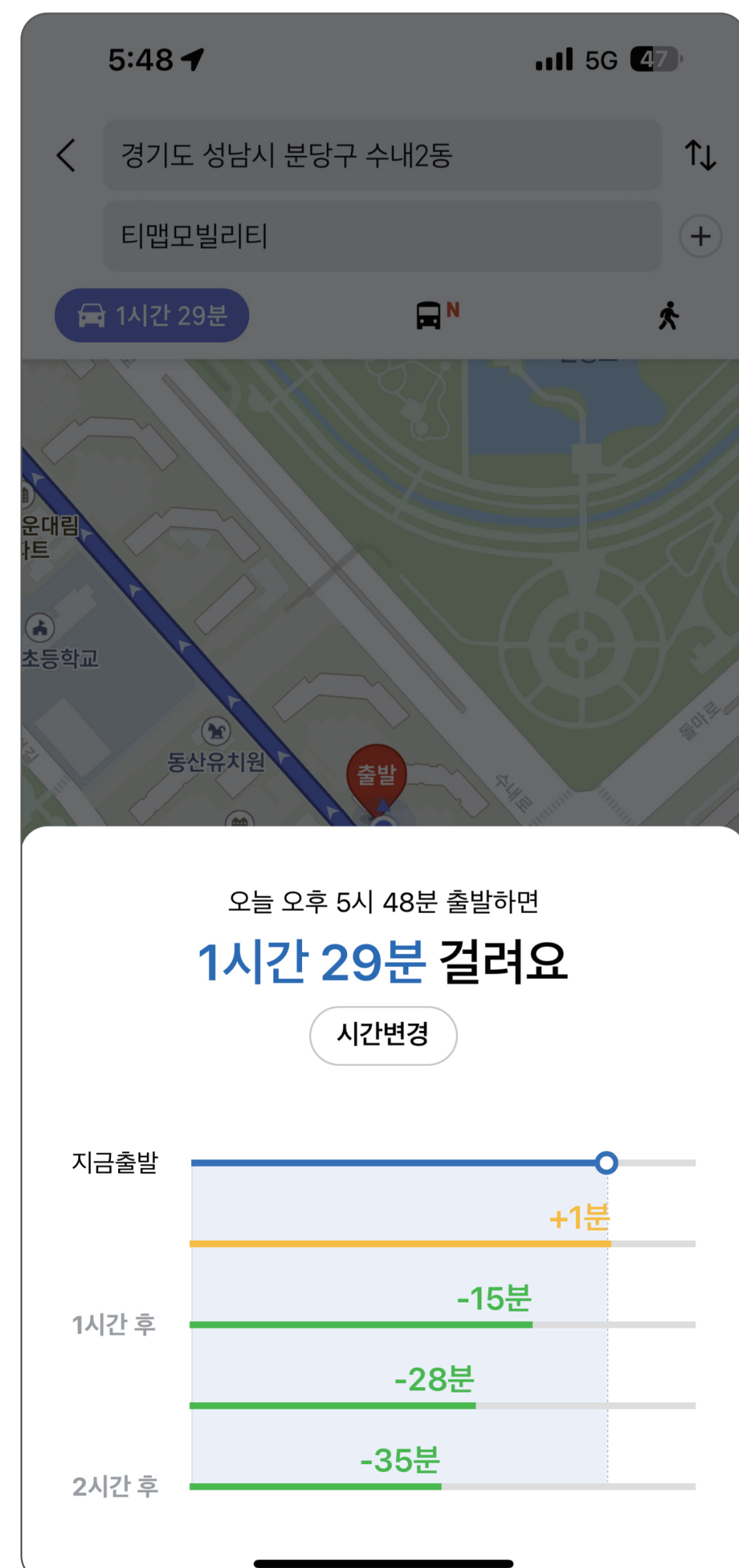
Matrix 경로탐색

CCH 알고리즘은 단순히 단일 출발지 → 도착지 경로탐색 뿐만 아니라 n개의 출발지 → m개의 도착지 경로탐색으로 확장 가능합니다. 기본적인 데이터 전처리 과정(Preprocessing, Customization)은 동일합니다.



이는 n, m의 수가 증가할수록 더욱 큰 성능 차이를 보여줍니다. 이와 같은 빠른 성능의 Matrix 경로탐색은 티맵의 배송최적화, 대리운전, 화물단가계산, 실제 경로에 기반한 주변 정보 생성 등 여러 곳에 활용되고 있습니다.

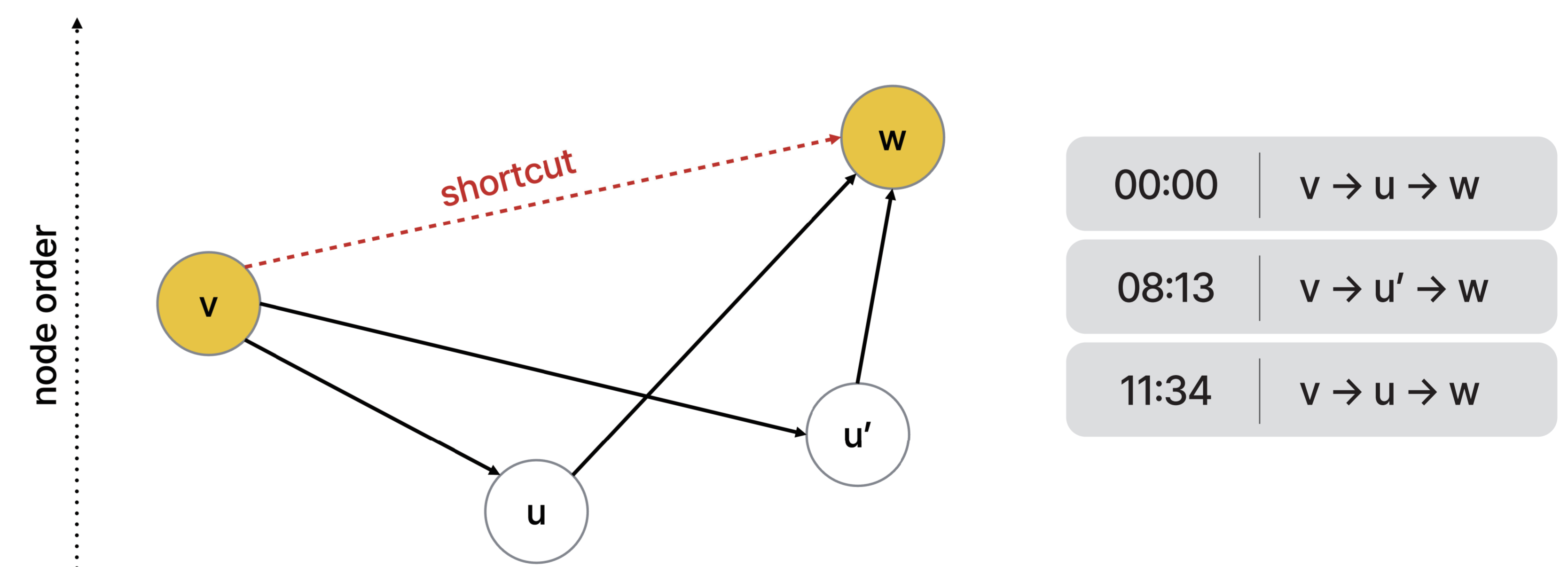
Time-Dependent 경로탐색



출발지 → 도착지 경로탐색 시 여러 정보를 비용으로 환산하여 최적의 경로를 찾게 됩니다. 보다 실 세계에 가까운 결과를 얻기 위해 시간 의존적(Time-Dependent)인 조건이 추가됩니다. Time-Dependent 경로탐색이란 출발지로부터 도로를 탐색(확장)하면서 각 도로의 실제 진입 시각을 계산하여 예상 통과 시점의 교통정보, 유고정보 등을 활용하는 경로탐색을 말합니다.

CATCHUp(Customizable Approximated Time-dependent Contraction Hierarchies through Unpacking) 알고리즘은 CCH 에서 시간 의존적인 요소를 해결하도록 확장합니다. CCH 와 Preprocessing 단계는 동일하며 Time-Dependent Customization 을 수행하는데, 시간에 따라 변화하는 Witness Path와 각 Shortcut의 최대/최소 비용을 생성합니다.

Time-Dependent 경로탐색



경로탐색 단계에서는 EliminationTree 를 순회하여 더 작은 SubGraph 를 생성하고, SubGraph 에서 Time-Dependent Witness Path 를 활용하여 정방향 Time-Dependent Dijkstra 를 수행하여 경로를 찾을 수 있습니다.

최적화로 Shortcut 의 Min/Max 비용을 이용한 EliminationTree Interval Query 로 더욱 작은 SubGraph Corridor 를 생성합니다. 또한 탐색중에 Shortcut 중복 Unpacking 을 방지하기 위해 Lazy Unpacking 을 적용하고, Corridor AStar 를 통해 빠른 Time-Dependent 경로탐색이 가능해 집니다.

현재 TMAP 에서 '다른 시간 출발' 이라는 이름으로 서비스 중이고, TMAP Open API 로 외부에도 서비스 제공중입니다.