(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0052195
(43) 공개일자 2022년04월27일

(51) 국제특허분류(Int. Cl.)
G01C 21/34 (2006.01)
(52) CPC특허분류
G01C 21/3438 (2013.01)
G01C 21/3446 (2013.01)
(21) 출원번호 10-2020-0136265
(22) 출원일자 2020년10월20일
심사청구일자 없음

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
박석
서울특별시 종로구 평창12길 16-6, 2동 202호 (평창동, 평창삼호빌라)
이준규
경기도 안양시 동안구 관악대로263번길 68-16, 302호(비산동, 남흥빌라)
(74) 대리인
유미특허법인

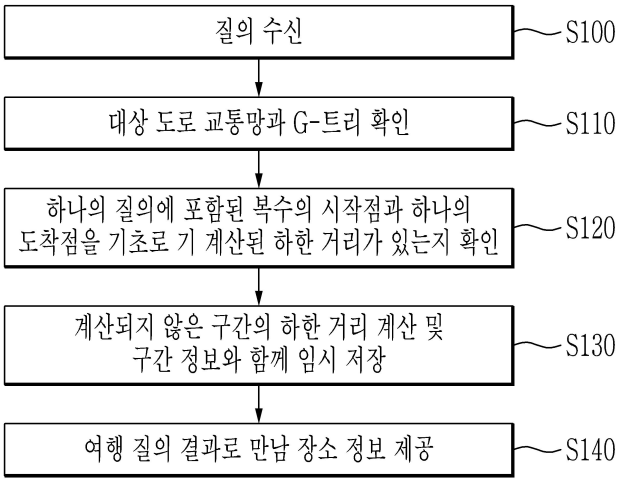
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 **도로 교통망 환경에서 G-트리 구조를 이용한 단일 그룹 콜렉티브 여행 질의 처리 장치 및 방법**

(57) 요 약

질의 처리 장치가 단일 그룹 콜렉티브 여행 질의를 처리하는 방법으로서, 단일 그룹에 속한 복수의 사용자들 각각의 위치 정보와 복수의 사용자들을 단일 그룹으로서 이동할 도착점 위치 정보를 포함하는 질의를 수신한다. 도로 교통망을 기초로 생성된 G-트리에서, 각각의 위치 정보와 도착점 위치 정보를 노드로 확인하고, 단일 그룹에 속한 복수의 사용자들 중 대상 사용자의 위치 정보와 도착점 위치 정보를 기초로 G-트리에서 복수의 구간들을 생성한 후, 복수의 구간들 중 단일 그룹 콜렉티브 여행 질의에 대응하여 미리 계산되어 저장된 하한 거리들의 구간들에 대응하는 구간이 있는지 확인한다. 대상 사용자의 구간들 중 이미 계산되어 저장된 하한 거리에 대응하는 구간이 있으면, 하한 거리가 계산된 구간을 제외한 나머지 구간에서의 하한 거리를 계산한다.

대 표 도 - 도3



이 발명을 지원한 국가연구개발사업

과제고유번호	1711110998
과제번호	2019R1A2C1088126
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	중견연구자지원사업
연구과제명	개인정보 보호를 위해 차분 프라이버시가 적용된 사용자 정보에 기반한 도로교통망에서의 최적 장소 선택
기 여 율	1/2
과제수행기관명	서강대학교 산학협력단
연구기간	2019.09.01~2021.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호	1345315071
과제번호	2017R1D1A1B03036252
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학개인기초연구지원사업
연구과제명	도로교통망 이동객체에 대한 사용자 의도 추론 기반의 프라이버시를 보호한 예측 질의처리 기법
기 여 율	1/2
과제수행기관명	서강대학교 산학협력단
연구기간	2017.06.01~2020.11.30

명 세 서

청구범위

청구항 1

적어도 하나의 프로세서에 의해 동작하는 질의 처리 장치가 단일 그룹 콜렉티브 여행 질의를 처리하는 방법으로서,

단일 그룹으로 설정된 복수의 사용자들 각각의 위치 정보들과, 상기 단일 그룹이 이동할 도착점 위치 정보를 포함하는 질의를 수신하는 단계,

상기 복수의 사용자들이 이동할 도로 교통망에 대응하는 G-트리에서, 상기 각각의 위치 정보와 도착점 위치 정보를 노드로 확인하는 단계,

상기 복수의 사용자들 중 대상 사용자의 위치 정보와 상기 도착점 위치 정보를 기초로 상기 G-트리에서 복수의 구간들을 생성하고, 상기 복수의 구간들 중 상기 질의에 대응하여 미리 계산된 하한 거리들의 구간들에 대응하는 구간이 있는지 확인하는 단계, 그리고

상기 대상 사용자의 구간들 중 이미 계산되어 저장된 하한 거리에 대응하는 구간이 있으면, 상기 하한 거리가 계산된 구간을 제외한 나머지 구간에서의 하한 거리를 계산하는 단계

를 포함하는, 단일 그룹 콜렉티브 여행 질의 처리 방법.

청구항 2

제1항에 있어서,

상기 하한 거리를 계산하는 단계는,

계산한 하한 거리들과 각 하한 거리들에 대응하는 구간 정보를 상기 단일 그룹 콜렉티브 여행 질의의 처리가 완료될 때까지 저장하는 단계

를 포함하는, 단일 그룹 콜렉티브 여행 질의 처리 방법.

청구항 3

제2항에 있어서,

상기 하한 거리를 계산하는 단계 이후에,

상기 복수의 사용자들 각각에 대해 계산한 하한 거리와 기 계산된 구간의 하한 거리를 기초로, 상기 단일 그룹 콜렉티브 여행 질의에 대한 결과인 만남 장소 정보를 제공하는 단계

를 포함하는, 단일 그룹 콜렉티브 여행 질의 처리 방법.

청구항 4

제1항에 있어서,

상기 G-트리 구조는 상기 도로 교통망을 계층적으로 쌓아 형성되며,

복수의 정점들과, 각 정점들을 연결하는 간선들을 포함하는, 단일 그룹 콜렉티브 여행 질의 처리 방법.

청구항 5

제4항에 있어서,

상기 복수의 정점들은 각각 어느 하나의 서브 그래프에 포함되며,

각 정점은 해당 정점이 포함된 서브 그래프에서 다른 정점의 서브 그래프로 이동할 때 반드시 지나야 하는 가장자리 정점 정보를 포함하는, 단일 그룹 콜렉티브 여행 질의 처리 방법.

청구항 6

단일 그룹 콜렉티브 여행 질의를 처리하는 장치로서,

복수의 사용자들 각각의 위치 정보와 상기 복수의 사용자들을 단일 그룹으로 하여 이동할 도착점 위치 정보를 포함하는 여행 질의를 수신하는 인터페이스,

도로 교통망을 기초로 생성된 G-트리와, 상기 질의에 대응하여 각 사용자들에 대해 계산된 하한 거리와 하한 거리를 포함하는 구간 정보를 저장하는 스토리지, 그리고

프로세서

를 포함하며,

상기 프로세서는,

상기 단일 그룹에 속한 복수의 사용자들 중 대상 사용자의 위치 정보와 도착점 위치 정보를 기초로 상기 G-트리에서 복수의 구간들을 생성하고, 상기 복수의 구간들 중 상기 단일 그룹 콜렉티브 여행 질의에 대응하여 미리 계산되어 저장된 하한 거리들의 구간들에 대응하는 구간이 있는지 확인하고, 상기 대상 사용자의 구간들 중 이미 계산되어 저장된 하한 거리에 대응하는 구간이 있으면, 상기 하한 거리가 계산된 구간을 제외한 나머지 구간에서의 하한 거리를 계산하는, 단일 그룹 콜렉티브 여행 질의 처리 장치.

청구항 7

제6항에 있어서,

상기 프로세서는,

계산한 하한 거리들과 각 하한 거리들에 대응하는 구간 정보를 상기 단일 그룹 콜렉티브 여행 질의의 처리가 완료될 때까지 상기 스토리지에 임시로 저장하는, 단일 그룹 콜렉티브 여행 질의 처리 장치.

청구항 8

제7항에 있어서,

상기 프로세서는,

상기 복수의 사용자들 각각에 대해 계산한 하한 거리와 기 계산된 구간의 하한 거리를 기초로, 상기 단일 그룹 콜렉티브 여행 질의에 대한 결과인 만남 장소 정보를 제공하는, 단일 그룹 콜렉티브 여행 질의 처리 장치.

청구항 9

제8항에 있어서,

상기 프로세서는,

상기 G-트리 그래프를 복수의 그래프들로 분할하고, 분할한 그래프들에서 가장자리 정점을 G-트리 노드들로 확인하며, 상기 복수의 사용자들 각각의 위치와 상기 G-트리 노드들 사이의 상기 하한 거리를 계산하는, 단일 그룹 콜렉티브 여행 질의 처리 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 도로 교통망 환경에서 G-트리 구조를 이용한 단일 그룹 콜렉티브 여행 질의 처리 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 위치 기반 서비스(Location Based Service)와 공유 경제 서비스(sharing economy)가 결합된 승차 공유 서비스는, 기존의 운송 수단을 빠르게 대체하고 있다. 승차 공유 서비스를 통해 사용자는 기존의 운송 수단과 달리 차를 소유하지 않아도 개인 차량을 이용하는 것처럼 이동을 할 수 있고, 대중 교통을 이용할 필요도 없다.

[0003] 이러한 승차 공유 서비스를 제공할 때, 사람들이 모이는 장소를 선택하기 위해서는 전체 여행 비용을 최소화하는 기법이 사용되어야 한다. 전체 여행 비용을 최소화하는 기법은 다양하며, 단일 그룹 콜렉티브 여행 질의(SGCTP: Single Group Collective Travel Planning)가 그 중 하나이다.

[0004] 단일 그룹 콜렉티브 여행 질의는 여러 명의 사용자들이 특정 지점에 모여서 하나의 운송 수단을 이용하여 도착 지점으로 이동을 할 때, 소요되는 전체 비용을 최소화 하는 지점을 찾는 것을 목표로 한다.

[0005] 콜렉티브 여행 계획 질의와 관련된 연구들이 많이 진행이 되었지만 특정 상황에서만 효과인 성능을 보인다는 문제점을 가지고 있다.

[0006] 예를 들어, 콜렉티브 여행 계획 질의는 여러 명의 사용자와 하나의 도착점, 그리고 만남 장소의 후보 집합이 주어졌을 때, 최대 k개의 만남 장소에 사용자들을 매칭하여 최소 비용을 갖는 만남 장소 집합을 구하는 질의 계획이다. 사용자는 반드시 적어도 하나의 만남 장소와 매칭 되어야 하나, 매칭이 되지 않은 경우는 도착 장소로 직행하는 경우에 해당한다.

[0007] 이러한 콜렉티브 여행 계획 질의는 NP-hard라고 증명되어 있기 때문에, 사용자와 만남 장소를 빠르게 매칭시키기 위해 근사 알고리즘을 사용한다. 하지만 근사 알고리즘은 단위 연산이 되는 최단 거리 계산을 선행 계산으로 취하기 때문에, 단일 그룹 콜렉티브 여행 계획 질의를 해결하기 위한 방법이 될 수 없다.

[0008] 그룹 여행 계획(Group Trip Planning) 질의는 사용자들의 출발점 집합, 후보 여행지 집합, 그리고 사용자들의 도착점 집합이 주어지면, 어떤 순서로 후보 여행지를 방문해야 최적의 비용이 소모되는지에 관한 문제와 그 해결법을 제시하였다. 그룹 여행 계획 질의는 여행 후보 지역(POI: Point Of Interest)을 입력 받아서 최의 해를 구하기 때문에, 그래프 상에 존재하는 모든 정점을 후보로 삼는 단일 그룹 콜렉티브 여행 질의에는 적용 불가능하다.

[0009] 또 다른 연구인 최적 도착 장소(Optimal End Stops)는 여러 명의 사용자들이 하나의 만남 지점에서 모인 후 다같이 이동 하다가 임의의 지점에서 흩어져서 사용자들이 각각의 도착 지점으로 이동했을 때, 모든 비용을 최소화 하는 문제를 해결하는 것이다. 이때, 모든 사용자들이 같은 도착점으로 이동한다면, 최적 도착 장소에서 해결하고자 하는 문제는 단일 그룹 콜렉티브 여행 질의와 동일한 문제가 된다. 하지만 최적 도착 장소 문제는 여러 명의 사용자가 각각의 도착점을 요청 하는 상황을 우선으로 고려했기 때문에, 불필요한 시간 복잡도가 발생하는 단점이 있다.

[0010] 이 외에도 최적 만남 장소(Optimal Meeting Point)는 유클리드 거리를 하한과 상한으로 사용하기 때문에, 가지치기할 때 실제 네트워크 거리와 비교 시 값의 차이가 크게 발생한다. 이에 따라 최적 값을 찾기 위해 필요한 탐색보다 더 많은 탐색이 이루어진다는 단점이 있다.

발명의 내용

해결하려는 과제

[0011] 따라서, 본 발명은 단일 그룹 콜렉티브 여행 질의를 처리하는 방법과 가지치기 기반의 동적 계획을 제공하는, 도로 교통망 환경에서 G-트리 구조를 이용한 단일 그룹 콜렉티브 여행 질의 처리 장치 및 방법을 제공한다.

과제의 해결 수단

[0012] 상기 본 발명의 기술적 과제를 달성하기 위한 본 발명의 하나의 특징인 적어도 하나의 프로세서에 의해 동작하는 질의 처리 장치가 단일 그룹 콜렉티브 여행 질의를 처리하는 방법으로서,

[0013] 단일 그룹으로 설정된 복수의 사용자들 각각의 위치 정보들과, 상기 단일 그룹이 이동할 도착점 위치 정보를 포함하는 질의를 수신하는 단계, 상기 복수의 사용자들이 이동할 도로 교통망에 대응하는 G-트리에서, 상기 각각의 위치 정보와 도착점 위치 정보를 노드로 확인하는 단계, 상기 복수의 사용자들 중 대상 사용자의 위치 정보와 상기 도착점 위치 정보를 기초로 상기 G-트리에서 복수의 구간들을 생성하고, 상기 복수의 구간들 중 상기 질의에 대응하여 미리 계산된 하한 거리들의 구간들에 대응하는 구간이 있는지 확인하는 단계, 그리고 상기 대상 사용자의 구간들 중 이미 계산되어 저장된 하한 거리에 대응하는 구간이 있으면, 상기 하한 거리가 계산된 구간을 제외한 나머지 구간에서의 하한 거리를 계산하는 단계를 포함한다.

[0014] 상기 하한 거리를 계산하는 단계는, 계산한 하한 거리들과 각 하한 거리들에 대응하는 구간 정보를 상기 단일

그룹 콜렉티브 여행 질의의 처리가 완료될 때까지 저장하는 단계를 포함할 수 있다.

[0015] 상기 하한 거리를 계산하는 단계 이후에, 상기 복수의 사용자들 각각에 대해 계산한 하한 거리와 기 계산된 구간의 하한 거리를 기초로, 상기 단일 그룹 콜렉티브 여행 질의에 대한 결과인 만남 장소 정보를 제공하는 단계를 포함할 수 있다.

[0016] 상기 G-트리 구조는 상기 도로 교통망을 계층적으로 쌓아 형성되며, 복수의 정점들과, 각 정점들을 연결하는 간선들을 포함할 수 있다.

[0017] 상기 복수의 정점들은 각각 어느 하나의 서브 그래프에 포함되며, 각 정점은 해당 정점이 포함된 서브 그래프에서 다른 정점의 서브 그래프로 이동할 때 반드시 지나야 하는 가장자리 정점 정보를 포함할 수 있다.

[0018] 상기 본 발명의 기술적 과제를 달성하기 위한 본 발명의 또 다른 특징인 단일 그룹 콜렉티브 여행 질의를 처리하는 장치로서,

[0019] 복수의 사용자들 각각의 위치 정보와 상기 복수의 사용자들을 단일 그룹으로 하여 이동할 도착점 위치 정보를 포함하는 여행 질의를 수신하는 인터페이스, 도로 교통망을 기초로 생성된 G-트리와, 상기 질의에 대응하여 각 사용자들에 대해 계산된 하한 거리와 하한 거리를 포함하는 구간 정보를 저장하는 스토리지, 그리고 프로세서를 포함하며, 상기 프로세서는, 상기 단일 그룹에 속한 복수의 사용자들 중 대상 사용자의 위치 정보와 도착점 위치 정보를 기초로 상기 G-트리에서 복수의 구간들을 생성하고, 상기 복수의 구간들 중 상기 단일 그룹 콜렉티브 여행 질의에 대응하여 미리 계산되어 저장된 하한 거리들의 구간들에 대응하는 구간이 있는지 확인하고, 상기 대상 사용자의 구간들 중 이미 계산되어 저장된 하한 거리에 대응하는 구간이 있으면, 상기 하한 거리가 계산된 구간을 제외한 나머지 구간에서의 하한 거리를 계산한다.

[0020] 상기 프로세서는, 계산한 하한 거리들과 각 하한 거리들에 대응하는 구간 정보를 상기 단일 그룹 콜렉티브 여행 질의의 처리가 완료될 때까지 상기 스토리지에 임시로 저장할 수 있다.

[0021] 상기 프로세서는, 상기 복수의 사용자들 각각에 대해 계산한 하한 거리와 기 계산된 구간의 하한 거리를 기초로, 상기 단일 그룹 콜렉티브 여행 질의에 대한 결과인 만남 장소 정보를 제공할 수 있다.

[0022] 상기 프로세서는, 상기 G-트리 그래프를 복수의 그래프들로 분할하고, 분할한 그래프들에서 가장자리 정점을 G-트리 노드들로 확인하며, 상기 복수의 사용자들 각각의 위치와 상기 G-트리 노드들 사이의 상기 하한 거리를 계산할 수 있다.

발명의 효과

[0023] 본 발명에 따르면, 최적 해를 계산하는 방법이면서도 근사 해를 구하는 비교 기법보다 더 빠른 성능을 갖질 수 있다.

[0024] 또한, 본 발명에 따르면 G-트리 인덱스 구조를 사용하는 최적 우선 탐색 방법으로 인해 발생할 수 있는 중복 계산을 방지하기 위한 동적 프로그래밍 방법을 제안하여, 빠른 탐색을 진행할 수 있다.

도면의 간단한 설명

[0025] 도 1은 단일 그룹 콜렉티브 여행 계획 질의의 예시도이다.

도 2는 본 발명의 실시예에 따른 단일 그룹 콜렉티브 여행 질의 처리 장치의 구조도이다.

도 3은 본 발명의 실시예에 따른 단일 그룹 콜렉티브 여행 질의 처리 방법에 대한 흐름도이다.

도 4는 본 발명의 실시예에 따른 G-트리 인덱스 구조의 예시도이다.

도 5는 최단 거리 계산 시 발생하는 중복의 예시도이다.

도 6은 본 발명의 실시예에 따른 동적 프로그래밍 기반의 최단 거리를 계산하는 예시도이다.

도 7은 본 발명의 실시예에 따라 G-트리 매개변수 별 인덱스 구성 시간을 나타낸 그래프이다.

도 8은 본 발명의 실시예에 따라 G-트리 매개변수 별 제1 비용 함수의 질의 시간을 나타낸 그래프이다.

도 9는 본 발명의 실시예에 따라 G-트리 매개변수 별 제2 비용 함수의 질의 시간을 나타낸 그래프이다.

도 10은 본 발명의 실시예에 따라 제1 비용 함수에서 균등 생성 방식 질의 시간을 나타낸 그래프이다.

도 11은 본 발명의 실시예에 따라 제2 비용 함수에서 균등 생성 방식 질의 시간을 나타낸 그래프이다.

도 12는 본 발명의 실시예에 따라 제1 비용 함수에서 격자 생성 방식 질의 시간을 나타낸 그래프이다.

도 13은 본 발명의 실시예에 따라 제2 비용 함수에서 결자 생성 방식 질의 시간을 나타낸 그래프이다.

도 14는 본 발명의 실시예에 따른 가지치기 효과를 나타낸 그래프이다.

도 15는 본 발명의 실시예에 따른 에러 비율을 나타낸 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0026] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0027] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0028] 이하 도면을 참조로 하여 본 발명의 실시예에 따른 도로 교통망 환경에서 G-트리 구조를 이용한 단일 그룹 콜렉티브 여행 질의 처리 장치 및 방법에 대해 설명한다.

[0029] 도 1은 단일 그룹 콜렉티브 여행 계획 질의의 예시도이다.

[0030] 도 1의 (a)는 사용자들의 이동 거리의 합을 최소화하는 이동 경로의 비용 합을 나타낸 예시도이고, 도 1의 (b)는 사용자들의 이동 경로 중 가장 오래 걸린 사용자의 비용을 최소화하는 예시도이다.

[0031] 사용자들의 위치들과 하나의 도착점이 입력으로 주어지면, 질의 처리 시스템은 사용자를 분할하지 않고 하나의 그룹으로 형성한다. 그리고 사용자들이 만나서 하나의 운송 수단을 가지고 도착점으로 이동하려고 한다고 가정한다. 이 때, 사용자로 하여금 모이는 장소(이하, 설명의 편의를 위하여 '만남 장소'라 지칭함)를 선택하는데 가장 적은 노력을 필요로 하는 지점을 선택하는 방법을, 두 가지 비용 함수로 분류하여 제공할 수 있다.

[0032] 첫 번째는 사용자들의 이동 경로의 합을 나타내는 비용 함수로, 다음 수학식 1과 같이 정의한다.

수학식 1

[0033] $$F_{sum}=\arg_{x\in G.V}\sum_{t}^{|Q|}dist(q_i,x)+dist(x,d)$$

[0034] 여기서, $dist(q_i,x)$는 임의의 사용자 q_i와 임의의 정점 x 사이의 최단 거리이며, d는 단일 도착점이며, $Q=\{q_1,q_2,\cdots,q_{|Q|}\}$는 입력으로 들어오는 사용자들의 위치를 나타내는 질의의 집합이다.

[0035] F_{sum}은 전체 여행 비용의 합을 나타내는 비용 함수로, 여행에 소요되는 이동 경비를 최소화하는 것을 최종 목표로 한다. 도 1의 (a)에 도시된 바와 같이 사용자들의 위치를 나타내는 질의 집합 Q, 모든 사용자들의 공통된 하나의 도착점 d가 주어졌을 때, 최적의 위치에서 만나서 이동을 했다고 가정하면 최소 비용은 27(=2+5+4+6+10)이다.

[0036] 또 다른 비용 함수는 사용자의 최대 이동 시간을 나타내는 최대 이동 비용으로, 다음 수학식 2와 같이 정의한다.

수학식 2

[0037] $$F_{max}=\arg_{x\in G.V}\max_{i}^{|Q|} dist(q_i,x)+dist(x,d)$$

[0038] F_{max}는 사용자의 여행 시간의 최대값을 최소화하는 것을 목표료 한다. 도 1의 (b)에 도시된 바와 같이 사용자들의 위치를 나타내는 질의 집합 Q와 도착점 d가 주어졌다고 가정하면, 최적의 위치에서 사용자들이 만나서 도착점으로 이동했다고 가정할 경우의 비용은 16(=10+max([2,5,4,6]))이다.

[0039] 이와 같이 정의한 두 가지 비용 함수인 Fsum과 Fmax에서 비용을 최소화하는 집합 장소를 찾는 단일 그룹 콜렉티브 여행 계획 질의를 효과적으로 처리하기 위해, 본 발명의 실시예에서는 가지치기에 기반한 동적 계획 방법을 이용한다.

[0040] 본 발명의 실시예에 대해 설명하기 앞서, 단일 그룹 콜렉티브 여행 계획 질의에 대해 먼저 다음과 같이 정의한다.

[0041] 단일 그룹 여행 질의는 사용자 집합 Q와 도착점 d, 도로 교통망 G=<V, E>, 비용 함수 F(여기서, F는 Fsum 또는 Fmax 중 어느 하나)가 주어졌을 때, F를 최소화하는 만남 지점 r을 반환하는 질의로, 수학식 3과 같이 정의할 수 있다.

수학식 3

[0042] $$SGCTP(G_i,Q,d,F)=\arg\min_{r}F(G,Q,d)$$

[0043] 본 발명의 실시예에 따라 가지치기 기반으로 단일 그룹 콜렉티브 여행 계획 질의를 동적으로 처리하는 장치에 대해 도 2를 참조로 설명한다.

[0044] 도 2는 본 발명의 실시예에 따른 단일 그룹 콜렉티브 여행 질의 처리 장치의 구조도이다.

[0045] 도 2에 도시된 바와 같이, 단일 그룹 콜렉티브 여행 질의 처리 장치(이하, 설명의 편의를 위하여 '질의 처리 장치'라 지칭함)(100)는 적어도 하나의 프로세서에 의해 동작하는 컴퓨팅 장치에서, 본 발명의 동작을 실행하도록 기술된 명령들(instructions)이 포함된 프로그램을 실행한다.

[0046] 질의 처리 장치(100)의 하드웨어는 적어도 하나의 프로세서(110), 메모리(120), 스토리지(130), 통신 인터페이스(140)를 포함할 수 있고, 버스를 통해 연결될 수 있다. 이외에도 입력 장치 및 출력 장치 등의 하드웨어가 포함될 수 있다. 질의 처리 장치(100)는 프로그램을 구동할 수 있는 운영 체제를 비롯한 각종 소프트웨어가 탑재될 수 있다.

[0047] 프로세서(110)는 질의 처리 장치(100)의 동작을 제어하는 장치로서, 프로그램에 포함된 명령들을 처리하는 다양한 형태의 프로세서(110)일 수 있고, 예를 들면, CPU(Central Processing Unit), MPU(Micro Processor Unit), MCU(Micro Controller Unit), GPU(Graphic Processing Unit) 등 일 수 있다.

[0048] 메모리(120)는 본 발명의 동작을 실행하도록 기술된 명령들이 프로세서(110)에 의해 처리되도록 해당 프로그램을 로드한다. 메모리(120)는 예를 들면, ROM(read only memory), RAM(random access memory) 등 일 수 있다. 스토리지(130)는 본 발명의 동작을 실행하는데 요구되는 각종 데이터, 프로그램 등을 저장한다. 통신 인터페이스(140)는 유/무선 통신 모듈일 수 있다.

[0049] 프로세서(110)는 통신 인터페이스(140)를 통해 수신한 입력 정보를 기초로, 도로 교통망을 나타내는 그래프 전체에 존재하는 모든 정점들을 만남 후보점들로 지정한다. 그리고 지정한 만남 후보점들 각각의 비용을 계산한다.

[0050] 여기서, 입력 정보는 사용자들에 대한 시작점들, 그리고 도착점을 포함한다. 본 발명의 실시예에서는 입력 정보에 시작점들과 도착점에 대한 정보를 포함하는 것을 예로 하여 설명하나, 도로 교통망과 G-트리 그래프가 포함될 수도 있다.

[0051] 하나의 커다란 도로 교통망은 여러 개의 작은 도로 교통망으로 분할되고, 분할된 작은 도로 교통망들은 다시 여러 개의 더 작은 도로 교통망으로 분할되는 과정이 반복될 수 있다. 이때, 하나의 도로 교통망에는 하나의 G-트리가 대응된다.

[0052] 프로세서(110)는 입력 정보를 기초로 스토리지(140)에 저장되어 있는 도로 교통망 그래프들 중 시작점과 도착점을 기초로 대상 도로 교통망을 확인하고, 대상 도로 교통망에 대응하는 G-트리 그래프를 확인한다.

[0053] 여기서, 프로세서(110)가 시작점과 도착점을 기초로 도로 교통망을 확인하는 방법은 다양하므로, 본 발명의 실시예에서는 어느 하나의 방법으로 한정하지 않는다. 또한, G-트리 그래프는 도로 교통망 그래프들을 계층적으로 쌓아 올린 그래프로서, 각각 정점(V)들과 간선(E)들로 이루어진 도로 교통망 그래프를 포함한다.

[0054] 프로세서(110)는 단일 도로 교통망을 재귀적으로 분할한 후 계층적으로 쌓아 올린 G-트리 그래프를 복수의 그래프들로 분할하고, 분할한 복수의 그래프들 각각의 가장자리 정점을 G-트리 노드로 확인한다. 그리고 프로세서(110)는 G-트리 노드와 사용자 위치 사이의 하한 거리를 계산한다. 프로세서(110)는 계산한 하한 거리를 기초로, 단일 그룹 콜렉티브 여행 질의를 처리한다.

[0055] 이때, 프로세서(110)는 각 사용자의 위치인 시작점별로 도착점까지의 하한 거리를 각각 계산한다. 그리고 복수의 계산된 하한 거리를 스토리지(130)에 질의 처리가 완료될 때까지 임시로 저장한다.

[0056] 질의 처리 장치(100)가 동일한 질의에 포함된 복수의 사용자들 각각의 하한 거리를 계산할 때, 중복으로 하한 거리가 계산될 수 있다. 따라서, 프로세서(110)는 하한 거리를 계산할 때, 스토리지(130)에 임시로 저장되어 있는 하한 거리들과 중복되는 부분이 있는지 확인하고, 중복되는 부분이 있으면 추가로 계산하지 않고도 임시 저장되어 있는 하한 거리를 사용한다.

[0057] 이상에서 설명한 질의 처리 장치(100)를 통해 동적으로 여행 질의를 처리하는 방법에 대해 도 3을 참조로 설명한다.

[0058] 도 3은 본 발명의 실시예에 따른 단일 그룹 콜렉티브 여행 질의 처리 방법에 대한 흐름도이다.

[0059] 도 3에 도시된 바와 같이, 질의 처리 장치(100)가 사용자들의 각각의 위치에 해당하는 시작점들과 하나의 도착점을 포함하는 질의를 수신하면(S100), 수신한 질의를 기초로 질의 처리 장치(100)는 도로 교통망과 해당 도로 교통망에 대한 G-트리를 확인한다(S110). 질의 처리 장치(100)가 질의를 기초로 복수의 도로 교통망들 중에서 대상 도로 교통망을 확인하고, 해당 대상 도로 교통망에 대한 G-트리를 확인하는 방법은 다양한 방법으로 수행될 수 있으므로, 본 발명의 실시예에서는 어느 하나의 방법으로 한정하지 않는다.

[0060] 질의 처리 장치(100)는 시작점들 각각에 대하여 G-트리 노드에서 하한 거리를 계산할 때, 해당 질의에 대응하여 임시로 저장되어 있는 하한 거리 중 중복되는 구간에 해당하는 하한 거리가 있는지 확인한다(S120). G-트리의 탐색은 루트 노드에서 아래 노드로 내려가는 구조이기 때문에, 질의 처리 장치(100)는 자식 노드, 형제 노드 또는 부모 노드에서부터 도착점까지의 하한 거리가 이미 메모리에 저장되어 있는지 확인한다.

[0061] 중복되는 구간이 있어 하한 거리가 이미 계산되어 있다면, 질의 처리 장치(100)는 계산되지 않은 구간의 하한 거리를 계산하고, 구간 정보와 함께 임시 저장한다(S130). 질의 처리 장치(100)는 전체 구간에 대한 하한 거리를 계산한 후, 여행 질의에 대한 결과인 만남 장소 정보를 제공한다(S140).

[0062] 이상에서 설명한 방법으로 질의 처리 장치(100)가 단일 그룹 콜렉티브 여행 질의를 처리하는 과정에서 사용되는 G-트리 인덱스 구조에 대해 도 4를 참조로 설명한다.

[0063] 도 4는 본 발명의 실시예에 따른 G-트리 인덱스 구조의 예시도이다.

[0064] 본 발명의 실시예에서는 단일 그룹 콜렉티브 여행 질의를 최적 우선 탐색으로 해결하기 위해 도로 교통망을 계층으로 쌓아 올린 G-트리 인덱스 구조를 사용한다. G-트리 인덱스는 도 4의 (a)와 같이 다층 그래프 분할 방법을 기반으로 한다.

[0065] 도 4의 (a)는 그래프를 분할한 예시도이고, 도 4의 (b)는 계층적으로 그래프를 분할하면서 이를 트리 형태로 쌓아 올린 G-트리 구조를 나타낸 예시도이다.

[0066] 트리의 노드 내부에 표시된 숫자는 노드가 표현하는 서브 그래프에서 다른 노드의 서브 그래프로 이동할 때 반드시지나야 하는 정점인 가장자리 정점(Border) 집합을 나타낸 것이다.

[0067] 여기서, 그래프 분할이라 함은, 정점 집합 V, 간선 집합 E, 그래프 G=<V, E>가 주어졌을 때, 다음 제1 조건을

만족하는 서브 그래프 집합 $\{g_1 = < V_1, E_1 >, g_2 = < V_2, E_2 >, \ldots, g_f = < V_f, E_f >\}$ 을 그래프 G가 분할된 분할 집합이라 정의한다.

[0068] 제1 조건:

$$(1)\ \cup_{1 \le i \le f} V_i = V,$$
$$(2)\ \text{For } i \neq j, V_i \cap V_j = \varnothing, \text{ and}$$
$$(3)\ \forall u, v \in V_i, \text{ if } (u,v) \in E, \text{ then } (u,v) \in E_i.$$

[0069]

[0070] 그리고 가장자리 정점은, 그래프 G의 서브 그래프 g_i가 주어졌을 때, 정점 가 다음의 제2 조건을 만족하면, u는 서브 그래프 g_i의 가장자리 정점이라 한다. 제2 조건을 만족하는 가장자리 집합들이 쌍은 2차원 행렬로 생성된다.

[0071] 제2 조건: $\exists (u,v) \in E,\ u \in V_i \text{ and } v \notin V_i$

[0072] 2차원 행렬은 이러한 가장자리 집합들의 쌍으로 행렬을 구성하여, 선행 계산으로 최단 거리를 계산한 거리 행렬(distance matrix)이고, G-트리는 이를 이용하여 두 정점 사이의 최단 거리를 빠르게 계산할 수 있다.

[0073] G-트리는 계층 구조로 이루어져 있기 때문에 최우선 탐색 방법을 적용할 수 있다. 종래의 R-트리 기반의 분기 한정 방법은 유클리드 거리를 도로 교통망에 적용했기 때문에, 네트워크 분포에 따라 실제 도로 교통망의 최단 거리와의 차이가 결정된다. 이에 대해 도 5를 참조로 설명한다.

[0074] 도 5는 최단 거리 계산 시 발생하는 중복의 예시도이다.

[0075] 도 5에 도시된 바와 같이, 도로 교통망에서 정의할 수 있는 유클리드 거리 하한과 실제 하한의 차이에서, 점선으로 표시된 거리는 유클리드 하한 거리이고 파선으로 표시된 거리는 실제 네트워크 거리의 하한을 나타낸다. 그리고 검은 점은 사용자 위치를 의미한다.

[0076] 유클리드 거리 하한이 네트워크 거리 하한보다 매우 작은 것을 확인할 수 있다. 따라서 유클리드 거리 하한과 상한 값을 사용하면 탐색하지 않아도 될 분기를 더 탐색해야 한다는 문제이 있다.

[0077] 따라서, 본 발명의 실시예에서는 각각의 사용자에서 G-트리 노드까지의 하 한을 측정하는 방법을 정의한다. 이때 정의한 하한은 R-트리를 이용한 방법과 비교했을 때, 도로 교통망의 네트워크 최단 거리와 아주 근접한 하한이 된다.

[0078] G-트리 노드 하한은 다음과 같이 정의한다.

[0079] 사용자 위치 qi와 G-트리 노드 쑤이 주어졌을 때, G-트리 노드와 사용자 위치 사이의 하한 거리를 lb(qi, Tn)으로 표기하고, 다음과 같이 정의한다.

[0080] 1) 사용자 위치 qi가 G-트리 노드 Tn의 서브 그래프에 포함된 정점이 아닐 때, Tn의 가장자리 정점 집합을 B(Tn)이라 한다면, $lb(q_i, T_n) = \min_{b \in B(T_n)} dist(q_i, b)$ 이다.

[0081] 2) 사용자 위치 qi가 G-트리 노드 G-트리 노드 Tn의 서브 그래프에 포함된 정점일 때, Tn의 가장자리 정점 집합을 B(Tn)이라고 한다면, $lb(q_i, T_n) = 0$ 이다.

[0082] 이와 같은 정의를 이용하여 G-트리 노드 하한을 분기 한정 방법에 용하려면, G-트리의 부모 노드의 하한이 항상 해당 노드의 자식 노드의 하한보다 작아야 한다.

[0083] 이때, G-트리에서 하향식으로 탐색할 때, G-트리 노드 하한은 단조 증가한다. 즉, G-트리 노드 하한의 두 가지 경우에 해서 모두 단조 증가한다는 것을 보이면 G-트리 노드 하한은 단조 증가한다고 할 수 있다.

[0084] 먼저, 사용자 위치 qi가 G -트리 노드 Tn의 서브 그래프에 포함되지 않을 때, 하향식 탐색으로 탐색할 Tn의 자식 노드를 Tn+1이라 하면, 각각의 노드의 하한 노드는 다음 수학식 4 및 수학식 5와 같이 구할 수 있다.

수학식 4

$$lb(q_i, T_n) = \min_{b_1 \in B(T_n)} dist(q_i, b_1)$$
$$= \min_{u_x \in B(g_x)} (dist(q_i, u_x) +$$
$$\min_{u_{x-1} \in B(g_{x-1})} (dist(u_x, u_{x-1}) + \ldots +$$
$$\min_{u_1 \in B(g_1)} (dist(u_2, u_1) + \min_{b_1 \in B(T_n)} dist(u_1, b_1)))$$

[0085]

수학식 5

$$lb(q_i, T_{n+1}) = \min_{b_2 \in B(T_{n+1})} dist(q_i, b_2)$$
$$= \min_{u_x \in B(g_x)} (dist(q_i, u_x) +$$
$$\min_{u_{x-1} \in B(g_{x-1})} (dist(u_x, u_{x-1}) + \ldots +$$
$$\min_{t_n \in B(T_n)} (dist(u_1, t_n) + \min_{b_2 \in B(T_{n+1})} dist(t_n, b_2)))$$

[0086]

[0087] 질의 처리 장치(100)가 하향식 탐색을 할 때, G-트리 노드 하한이 단조 증가하지 않는다고 가정하면 $lb(q_i, T_{n+1}) < lb(q_i, T_n)$ 된다.

[0088] 하지만, 수학식 4와 수학식 5를 이용하여 하한 거리를 각각 구하면, $\min_{t_n \in B(T_n)}(dist(u_1, t_n) + \min_{b_2 \in B(T_{n+1})} dist(t_n, b_2)) < \min_{b_1 \in B(T_n)} dist(u_1, b_1))$ 을 얻을 수 있다.

[0089] $\min_{b_2 \in B(T_{n+1})} dist(t_n, b_2) \geq 0$ 을 항상 만족하므로, $\min_{t_n \in B(T_n)}(dist(u_1, t_n) < \min_{b_1 \in B(T_n)} dist(u_1, b_1))$ 인데, 이는 $t_n = b_1$ 일 때 모순이다. 따라서, G-트리 노드 하한은 하향식 탐색을 할 때, 항상 단조 증가함을 알 수 있다.

[0090] 이와 같이, 단일 그룹 콜렉티브 여행 질의의 G-트리 노드 하한이 단조 증가하는 것을 알 수 있으므로, 본 발명의 실시예에서는 분기 한정 방법을 G-트리에 적용할 수 있다. 이때, 본 발명의 실시예에서는 빠른 탐색을 위해 최적 우선 탐색을 질의 처리에 적용하였으며, 다음 표 1과 같은 알고리즘을 이용한다.

표 1

```
        Input: a road network G, a set of users
        Q, a destination d, and a cost function
        F (F_sum or F_max)
        Output: r∈G.V∧argmin_{r∈G.V}F(G,Q,d)
1:      r←NULL
2:      PQ←NULL  // priority queue
3:      LB=getLowerBound(G,Q,d,F)
4:      PQ.push(<LB,G>)
5:      current_best=INF
6:      while PQ is not empty do
7:          <LB,node>=PQ.pop()
8:          if LB>current_best then
9:              continue
10:         else
11:             if node is a leaf node then
12:                 current_best=
        min(F(node,Q,d),current_best)
13:                 r←argmin_{r∈node}F(node,Q,d)
14:             for each c in node.children do
15:                 LB_c=getLowerBound(c,Q,d,F)
16:                 PQ.push(<LB_c,c>)
17:     return r
```

[0091]

[0092] 표 1에 언급한 알고리즘은 G-트리에 최적 우선 탐색 방법을 적용한 단일 그룹 콜렉티브 여행 질의를 구하는 방법의 의사 코드이다. 질의 처리 장치(100)는 우선순위 큐(Priority Queue)를 사용하여 최적 우선 탐색을 구하고, 하한 값이 작은 노드부터 탐색을 진행하다. 여기서, 7행 내지 9행들은 단조 증가로 인해 생기는 관찰 사항으로 가지치기를 하는 과정을 나타낸 것이다.

[0093] 가지치기 과정이라 함은, 정답이 될 수 없는 경우를 미리 확인할 수 있는 경우, 조기에 서브 트리를 탐색에서 제거하는 방법을 의미한다.

[0094] 즉, G-트리의 특정 노드에서 도착점(또는 사용자)까지의 하한 비용은 노드가 하위에 위치할수록 같거나 증가한다고 가정한다. 그리고, 현재까지 찾은 최소 비용이 10이라고 했을 때, 비교 대상이 되는 G-트리 노드에서 발생하는 하한 비용이 12라면, 이 노드의 자식 노드들은 항상 크거나 같은 하한 비용을 같기 때문에 12보다 좋아질 수 없다. 따라서 비교 대상이 되는 노드의 자식들은 검사하지 않아도 되며, 이러한 방법을 가지치기 방법이라 한다.

[0095] 여기서, 현재까지 찾은 최적의 단일 그룹 콜렉티브 여행 질의 결과를 current_best라고 가정한다. 이 경우, 현재 검색중인 G-트리 노드 Tn의 질의 하한 값이 current_bset보다 크면, 해당 노드는 가지치기 할 수 있다.

[0096] 또한, G-트리 하한을 계산할 때, 사용자의 위치와 가장자리 정점까지 거리를 계산해야 한다. 이때, SPSP 질의를 사용하여 빠르게 두 정점 사이의 최단 거리를 계산할 수 있으며, SPSP 질의는 이미 알려진 방법으로 본 발명의 실시예에서는 상세한 설명을 생략한다.

[0097] 질의 처리 장치(100)는 SPSP 연산을 이용하여 G-트리에서 최적 우선 탐색을 이용한 단일 그룹 콜렉티브 여행 질의의 시간 복잡도를 계산한다. 가지 치기 성능은 그래의 분포에 따라 결정되기 때문에, 본 발명의 실시예에서는 최악의 경우 발생할 수 있는 시간 복잡도를 계산한다. G-트리의 높이가 $H=\log_f(|V|/\tau)+1$로 계산되고, 질의 처리 장치(100)는 각 노드 마다 사용자와 가장자리 정점 사이의 최단 거리를 계산한다.

[0098] G-트리에 포함된 모든 가장자리 정점의 수는 $O(\sum_{1\le i\le H}\log_2 f\cdot\sqrt{|V|f^{i-1}})=O(|V|\cdot\log_2 f/\sqrt{\tau})$ 이고,

SPSP의 질의 처리 시간은 $O(\log_2^2 f \cdot |V|)$ 이므로, 전체 시간 복잡도는 $O(\log_2^3 f/\sqrt{\tau} \cdot |Q||V|^2)$ 으로 계산될 수 있다. 한편, 기존 기본 방법의 시간 복잡도는 다음 수학식 6으로 근사할 수 있다.

수학식 6

[0099]
$$O(|Q||E|+|Q||V|\log|V|+|Q||V|)$$
$$\approx O(|Q||V|\log|V|)$$
$$< O(\log_2^3 f/\sqrt{\tau} \cdot |Q||V|^2).$$

[0100] 따라서, 노드의 가지치기가 하나도 일어나지 않는 최악의 경우로 비교할 때, 본 발명의 실시예에 따라 G-트리에서 최적 우선 탐색을 하는 방법이 기본 방법보다 시간 복잡도가 크게 된다. 이는 G-트리에서 최적 우선 탐색을 적용한 단일 그룹 콜렉티브 여행 질의 처리 방법에서 중복되는 계산이 발생하기 때문이다.

[0101] 따라서, 사용자의 위치 gi와 G-트리 노드 Tn까지 계산되는 거리 행렬이 질의 처리 장치(100)에 저장되어 있다면, 다음 노드까지의 최단거리를 동적 프로그래밍을 사용하여 중복 계산을 방지할 수 있다. 이에 대해 도 6을 참조로 설명한다.

[0102] 도 6은 본 발명의 실시예에 따른 동적 프로그래밍 기반의 최단 거리를 계산하는 예시도이다.

[0103] 도 6에 도시된 바와 같이, 본 발명의 실시예에 따라 발생하는 탐색 종류는 총 세 가지가 있다.

[0104] 첫 번째 경우는 도 6의 (a)와 같이 사용자의 위치에서 탐색할 노드까지 도달하기 전에 먼저 최소 공통 조상(lowest common ancestor)에 도달할 때 발생하는 자식-부모 관계이다. 두 번째 경우는 도 6의 (b)에 도시한 바와 같이 첫 번째 경우에서 도달한 최소 공통 조상에서 탐색할 노드까지 도달하기 위해 발생하는 부모-자식 관계이다. 세 번째 경우는 도 6의 (c)에 나타낸 바와 같이 최소 공통 조상 노드의 형제 노드 사이에서 발생하는 형제-형제 관계이다.

[0105] 도 6의 (d)에는 동적 프로그래밍으로 인해 채워지는 계산 결과를 나타내는 테이블로, 질의 처리 장치(100)의 스토리지(140)에 임시 저장된다.

[0106] 여기서, G-트리 하한을 계산하는 알고리즘은 다음 표 2와 같다.

표 2

[0107]
```
Input: a vertex u, a G-tree node g, and
3-dimensional matrix table
Output: shortest distance between u and g
1:  LCA = lowest common ancestor of u and g.
2:  if LCA is equal to parent of g then
3:      son = a child of LCA if child is an
ancestor of g.
4:      if table[u][son] is not computed then
5:          fillToLCA(u, LCA, table[u])
6:      fillTable(s − s, son, g, table[u], u)
7:  else if table[u][parent of g] is already
computed then
8:      fillTable(p − c, parent of g, g, table[u], u)
9:  else
10:     fillToLCA(u, LCA, table[u])
11: return min(table[u][g])
```

[0108] u, g, 그리고 도 6의 (d)에 나타낸 테이블을 입력으로 받으면, 세 번째 조건인 자식-부모 관계에 해당하는지 판단한다. 자식-부모 관계 조건에 해당하면, 공통 조상 노드까지 동적 프로그래밍으로 계산된 거리 행렬을 가져온다.

[0109] 이때 거리 행렬이 계산되어 있지 않으면, 첫 번째 경우인 자식-부모 관계 연산을 반복으로 수행하여 공통 조상 노드까지 거리 행렬을 계산한다. 그 후에 세 번째 경우인 형제-형제 관계 연산을 하여, 검색할 G-트리 노드까지 최종 결과를 계산 한다.

[0110] 또한, 프로그램의 7-8행에는 두 번째 경우인 부모-자식 관계 연산을 나타내었다. 하향식 탐색 순서로 인해 반드시 부모 노드까지 결과가 계산되어 있다. 따라서 2행의 형제-형제 관계와는 달리 바로 직전 노드의 결과를 가져와서 계산하면 된다. 그리고 10행은 첫 번째 경우인 자식-부모 관계 연산에 대한 것이다.

[0111] 이와 같이, 동적 프로그래밍 기법을 이용하면 G-트리의 노드 사이의 거리를 계산하고, 그 이후에는 계산된 결과를 불러오기만 하면 된다.

[0112] 즉, 각각의 G-트리 노드는 사용자 한 명에 대해서는 반드시 한 번의 거리 행렬 계산만 필요로 한다. G-트리의 전체 노드 개수는 $O(|V|/\tau)$개가 있고, 노드간의 거리 행렬 계산 시간은 $O(\tau \log_2^2 f)$이므로 동적 프로그래밍 기법의 최종 시간 복잡도는 $O(|Q||V|\log_2^2 f)$ 이다.

[0113] 이를 기본 방법과 비교하면, $O(|Q||V|\log_2^2 f) < O(|Q||V|\log|V|)$ 이다. G-트리의 매개변수인 f는 트리의 자식 수를 나타 내고 이는 전체 그래프의 정점의 수인 |V|보다 매우 작은 값이므로, 동적 프로그래밍 기법이 적용된 최우선 탐색 기법의 최악의 경우 시간 복잡도가 기 방법의 복잡도보다 낮은 것을 확인할 수 있다.

[0114] 이와 같이, 본 발명에 따른 질의 처리 방법에 대한 실험 결과에 대해 도 7 내지 도 9를 참조로 설명한다.

[0115] 도 7은 본 발명의 실시예에 따라 G-트리 매개변수 별 인덱스 구성 시간을 나타낸 그래프이다. 도 8은 본 발명의 실시예에 따라 G-트리 매개변수 별 제1 비용 함수의 질의 시간을 나타낸 그래프이고, 도 9는 본 발명의 실시예에 따라 G-트리 매개변수 별 제2 비용 함수의 질의 시간을 나타낸 그래프이다.

[0116] 본 발명의 실시예에서는 동적 프로그래밍 기법이 적용된 최우선 탐색 방법이 실제 도로 교통망에서 얼마만큼의 성능을 보이는지 확인하기 위해, 실제 도로 교통망 데이터인 CA(California)와 SF(San Francisco)를 사용한다. 각각의 도로 교통망에 대한 정보는 표 3과 같다.

표 3

[0117]

Data	Description	# Vertices	# Edges
CA	California	21,048	21,693
SF	San Francisco	174,956	223,001

[0118] 모든 사용자의 위치, 도착점, 질의 결과는 정점 위에 있다고 가정한다. 사용자의 위치와 도착점은 두 가지 방식으로 생성한다. 첫 번째 방식은 전체 그래프에서 균등하게 위치를 생성하는 균등 생성 방식이고, 두 번째는 전체 지도를 격자 모양으로 잘라서 임의의 격자 하나에서 균등하게 위치를 생성하는 격자 균등 생성 방식이다.

[0119] 실험에서 사용한 매개변수는 표 4와 같다.

표 4

[0120]

Parameter	Value
Size of query set Q	100, 200, …, 1000
Grid size (%)	5, 10, 15, 20, 25, 30
G-tree parameter f	2, 4, 8, 16
G-tree parameter τ	32, 64, 128, 256, 512

[0121] 원본 G-트리는 SPSP 질의 성능을 극화하기 위해 매개변수 f=4, τ=64(CA), τ=128(SF)를 사용한다. 하지만 본 발명의 실시예에서는 최단 거리를 구하는 시간을 최적화 하는 것보다 최우선 탐색에서 가지치기로 얻는 효과를 극대화 하는 것이 더 좋기 때문에, G-트리 매개변수 값을 실험으로 찾을 필요가 있다.

[0122] 질의 성능을 실험으로 측정하기 이전에 매개변수에 따른 G-트리의 인덱스 구성 시간을 먼저 측정하다. 도 7은 f={2, 4, 8, 16}, τ={32, 64, 128, 256, 512}에서 인덱스 구성 시간을 측정한 실험이다.

[0123] CA 데이터셋에서는 f=4, τ=32에서 가장 좋은 인덱스 구성 시간을 보였고, SF 데이터셋에서는 f=4, τ=64에서 가장 좋은 인덱스 구성 시간을 보였다. 하지만 τ=512일 때를 제외한 나머지 경우에서도 합리적인 시간을 나타내었다.

[0124] G-트리의 매개 변수에 따른 단일 그룹 콜렉티브 여행 계획 질의 성능을 비교하기 위해 f={2, 4, 8, 16}, τ={32, 64, 128, 256, 512}와 비용함수 Fsum, Fmax에서 질의 처리 시간의 평균값을 측정하였다.

[0125] 원본 G-트리의 SPSP 질의는 f=4, τ=64(CA), τ=128(SF)에서 가장 좋은 질의 성능을 보이나, 본 발명의 실시예에 따른 단일 그룹 콜렉티브 여행 계획 질의에서는 f=16, τ=32에서 가장 좋은 실험 성능을 보였다.

[0126] 도 8과 도 9는 이에 대한 Fsum, Fmax 비용 함수 각각에 대한 실험 결과를 나타낸 것으로, G-트리 매개변수 f가 커질수록 그래프는 더 많은 서브 그래프로 쪼개지는 것을 알 수 있다.

[0127] 같은 크기의 그래프를 이전보다 더 작은 단위로 쪼개면, 하나의 서브 그래프에 포함된 정점의 수는 이전보다 작아진다. 서브 그래프에 포함된 정점의 수가 이전보다 작아진다면, 해당 서브 그래프를 표현한 G-트리 노드 하한 값은 더 정확해진다.

[0128] G-트리 매개변수 τ가 작아질수록 G-트리의 리프 노드가 갖는 정점의 최대 수는 작아진다. 리프 노드가 표현하는 서브 그래프의 크기가 작아진다면, 같은 크기의 그래프를 좀 더 쪼개서 깊게 표현했기 때문에, 가지치기 효과를 향상시킬 수 있다. 이렇게 얻은 효과가 SPSP 질의 성능을 극대화하여 얻은 효과보다 더 많은 이득을 준다는 것을 확인할 수 있다. 따라서, 단일 그룹 콜렉티브 여행 질의 성능을 가장 극대화하는 매개변수인 f=16, τ=32으로 설정하여 성능을 평가하였다.

[0129] 이에 대해 도 10 내지 도 15를 참조로 설명한다.

[0130] 도 10은 본 발명의 실시예에 따라 제1 비용 함수에서 균등 생성 방식 질의 시간을 나타낸 그래프이고, 도 11은 본 발명의 실시예에 따라 제2 비용 함수에서 균등 생성 방식 질의 시간을 나타낸 그래프이다.

[0131] 도 10 및 도 11에서 표현된 BL은 기본 방법이고, GD는 기존의 경사 하강 방법에 대한 실험이다. 그리고 DPBFS가 본 발명의 실시예에 따른 동적 프로그래밍 기법이 적용된 최적 우선 탐색 기법에 대한 실험이다. 사용자 위치 집합 Q의 크기는 100부터 1,000까지 100씩 증가시켰다.

[0132] 도 10 및 도 11에 도시된 바와 같이, 본 발명의 실시예에 따른 방법이 최적의 해를 구하는 방법임에도 불구하고 질의를 고르게 생성했을 때, 기본 방법보다 뛰어난 성능을 보임을 확인할 수 있다.

[0133] 도 12는 본 발명의 실시예에 따라 제1 비용 함수에서 격자 생성 방식 질의 시간을 나타낸 그래프이고, 도 13은 본 발명의 실시예에 따라 제2 비용 함수에서 결자 생성 방식 질의 시간을 나타낸 그래프이다.

[0134] 사용자의 위치를 전체 도로 교통망에서 선택하지 않고 격자 모양으로 쪼개서 특정 격자 내에서 균등하게 선택했을 때에도 질의 처리 성능을 측정할 수 있다. 그리드 영역은 전체 도로 교통망 크기의 5%부터 30%까지, 5%씩 증가시키면서 사용자의 위치를 선택하였다. 그리고 사용자 집합 Q의 크기를 1,000으로 고정하였다.

[0135] 이 경우, 도 12에 도시된 바와 같이 GD 방법은 경사 하강 방법에 의한 지역 최적 값을 찾는 방법이기 때문에, 그래프의 분포에 따라 수렴하는 속도가 다르다. 따라서, 격자의 크기는 증가하지만 GD 방법은 불규칙한 성능 분포를 나타내었다.

[0136] 그러나, 본 발명의 실시예에 따른 방법을 사용하면, 격자의 크기와 상관 없이 같은 크기의 질의 집합에 대하여 일정한 성능 분포를 부였으며, 비교 기법인 BB와 GD에 비해 뛰어난 성능을 보이는 것을 알 수 있다.

[0137] 그리고 도 13에 도시된 바와 같이, SF 데이터셋에서 불규칙한 성능 분포를 보였으나, 본 발명의 실시예에 따른 방법을 이용할 경우 Fmax 비용 함수에서도 비교 기법 GD 방법과 비슷한 수준의 성능을 보이며 최적 해를 찾는 것을 확인할 수 있다. 또한, 격자의 크기에 상관없이 일정한 성능 분포를 보여주는 것을 확인할 수 있다.

[0138] CA, SF 각각의 데이터 셋에서 사용자 집합 크기를 100부터 1,000까지 변화시키면서 비용 함수 Fsum과 Fmax에서 단일 그룹 콜렉티브 여행 계획 질의 처리 시, 공간 인덱스 트리 구조에서 방문하는 노드의 수를 측정한 결과에 대해 도 14 및 도 15를 참조로 설명한다.

[0139] 도 14는 본 발명의 실시예에 따른 가지치기 효과를 나타낸 그래프이다. 그리고 도 15는 본 발명의 실시예에 따른 에러 비율을 나타낸 그래프이다.

[0140] 가지치기 기법을 사용한 R-트리 기반의 분기 한정 방법(BB: Branch and Bound)과 본 발명의 실시예에 따른 DPBFS 방법의 방문 노드 수를 측정하였다. 도 14 및 도 15에 도시된 바와 같이, 집합의 크기가 증가할 때, 본 발명의 실시예에 따른 방법은 일정한 수의 방문 노드를 갖는 반면, 기존의 BB 방법은 방문하는 노드의 수가 사용자 집합의 크기에 비례하여 증가하는 것을 확인할 수 있다.

[0141] 또한, 기존 연구인 GD 방법의 경우는 최적 해가 아닐 수 있으므로, 근사 해가 얼마나 정확한지 확인하였다. 도 15에 도시된 바와 같이, GD 방법이 실제 정답과 얼마나 차이를 보이는지 실험을 한 결과이다.

[0142] 정답의 오차는 ratio = (approx-optimal)/optimal-1로 계산한다. 도 15에 나타난 그래프를 살펴보면 주어진 도로 교통망에 따라 서로 다른 정답 오차를 갖는 것을 확인할 수 있다.

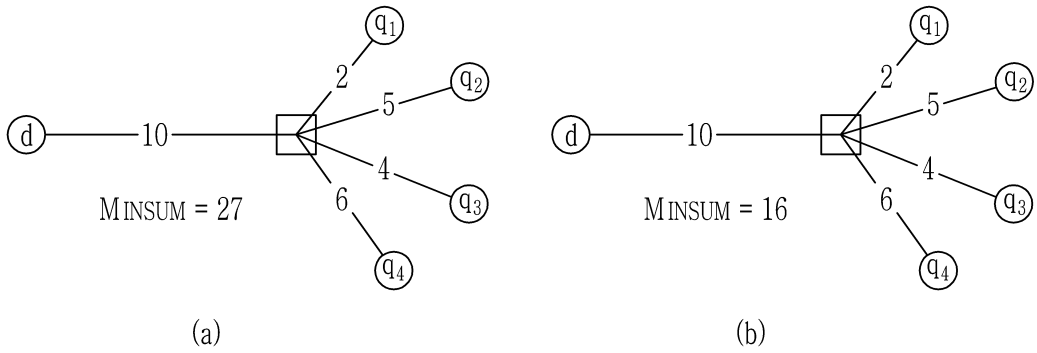
[0143] 하지만 본 발명의 실시예에 따른 기법은 도로 교통망에 종속인 정답 오차를 갖지 않고 항상 최적의 해를 계산할 수 있다. 즉, 본 발명의 실시예에 따른 방법은 사용자 집합의 크기 변화에 민감하지 않고 일정한 가지치기 성능을 보인다.

[0144] 따라서 집합의 크기가 증가할 때 비교 기법에 비해 항상 일정한 가지치기 성능을 보이며, 사용자 집합의 크기에 비례하여 선형으로 질의 수행 시간이 증가하는 것을 볼 수 있다. 또한 비교 기법인 GD와 BB에 비해 좋은 성능을 보이는 것을 확인할 수 있다.

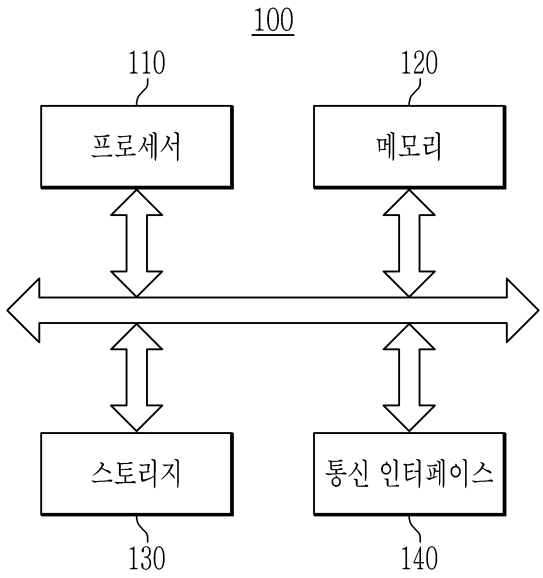
[0145] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

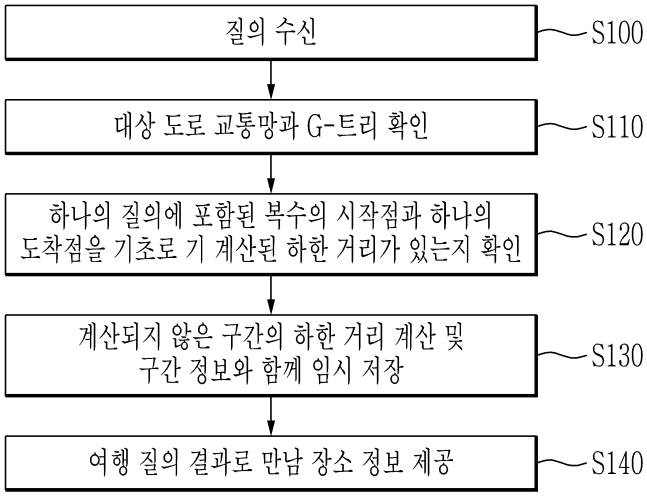
도면1



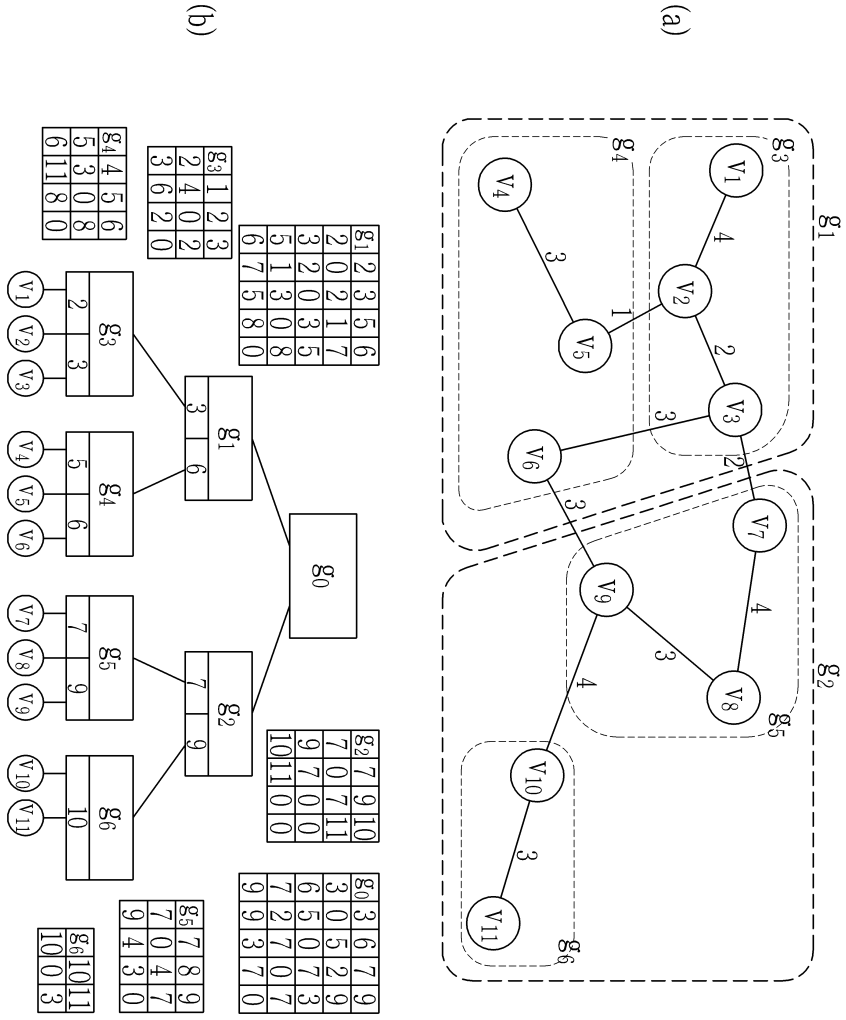
도면2


100
110
120
프로세서
메모리
스토리지
통신 인터페이스
130
140

도면3

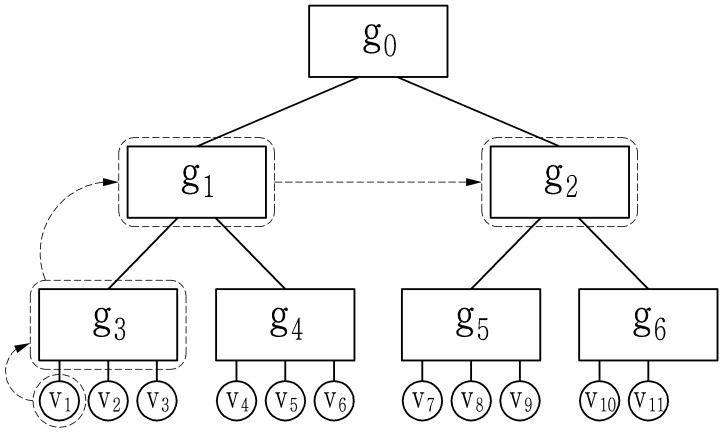

질의 수신
S100
대상 도로 교통망과 G-트리 확인
S110
하나의 질의에 포함된 복수의 시작점과 하나의 도착점을 기초로 기 계산된 하한 거리가 있는지 확인
S120
계산되지 않은 구간의 하한 거리 계산 및 구간 정보와 함께 임시 저장
S130
여행 질의 결과로 만남 장소 정보 제공
S140

도면4

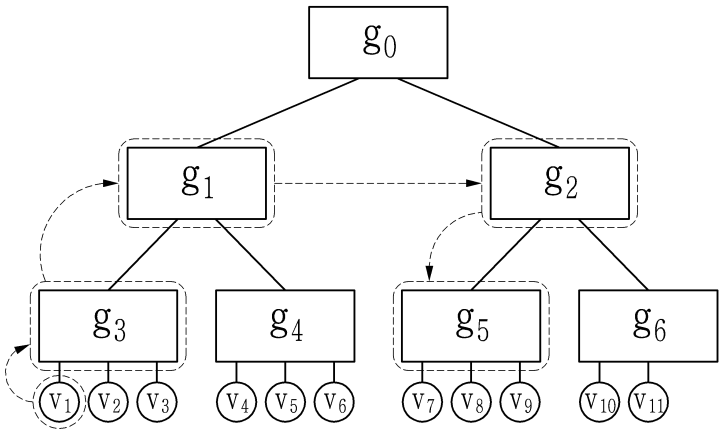


도면5

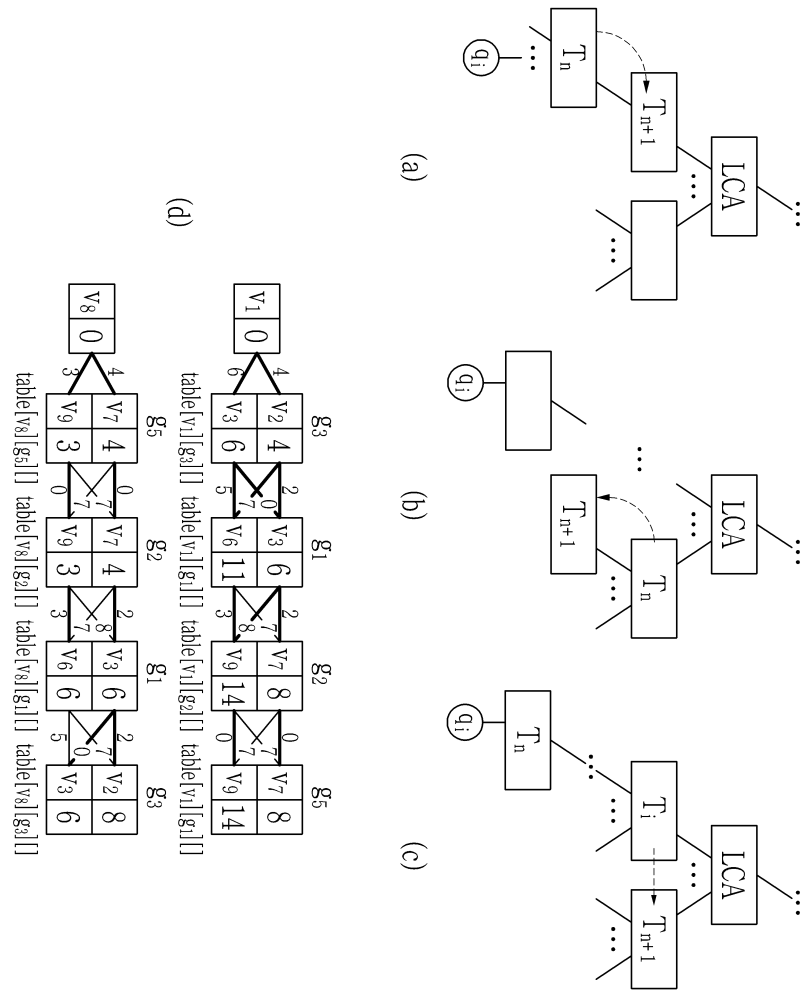
(a)


g0
g1
g2
g3
g4
g5
g6
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
v11

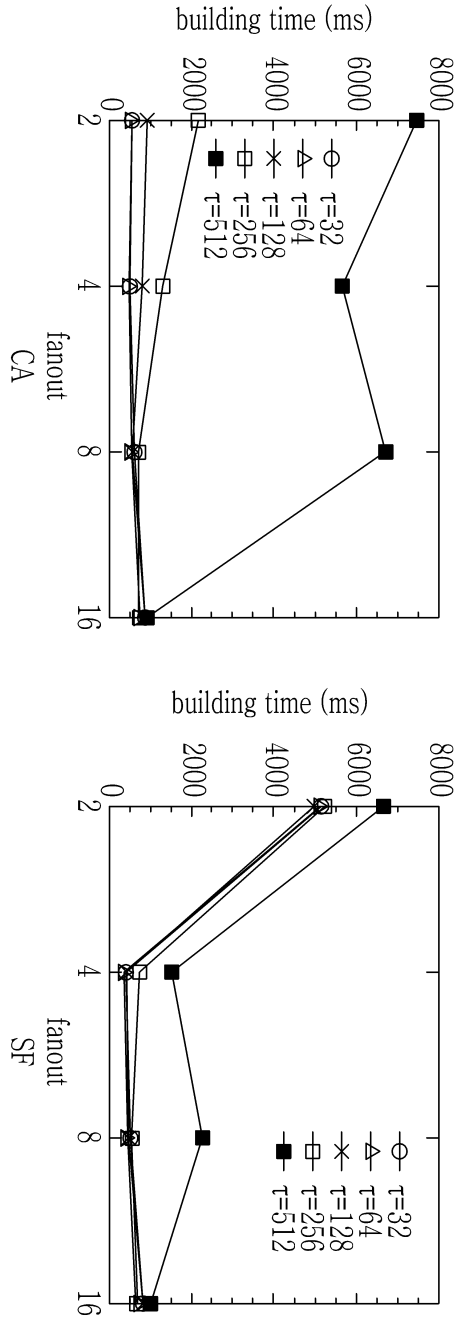
(b)


g0
g1
g2
g3
g4
g5
g6
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
v11

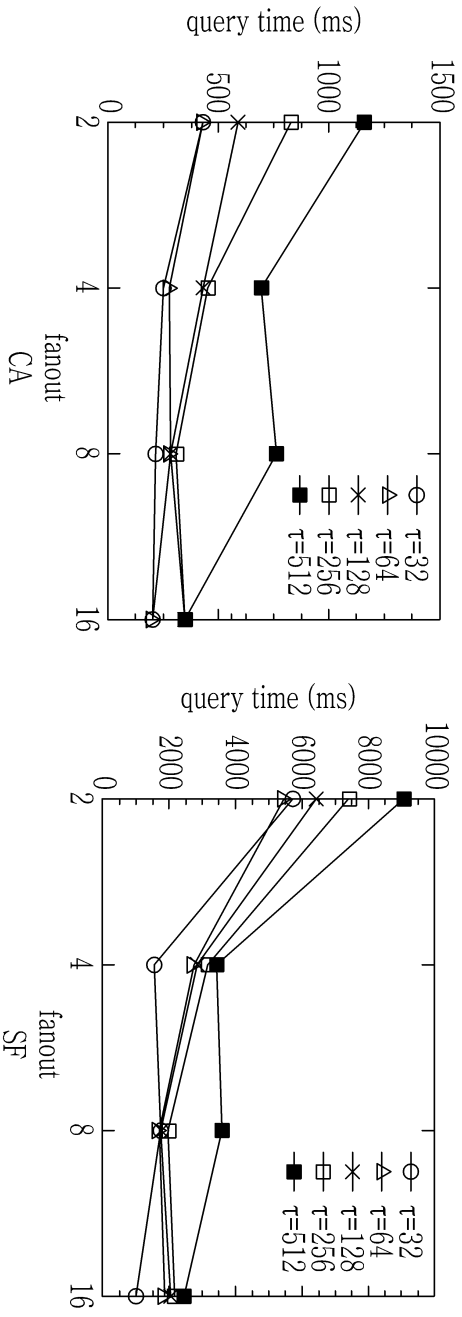
도면6


(a)
(b)
(c)
(d)
LCA
T_n
T_{n+1}
T_i
q_i
g_1
g_2
g_3
g_5
table[v_1][g_3][] table[v_1][g_1][] table[v_1][g_2][] table[v_1][g_1][]
table[v_8][g_3][] table[v_8][g_2][] table[v_8][g_1][] table[v_8][g_3][]

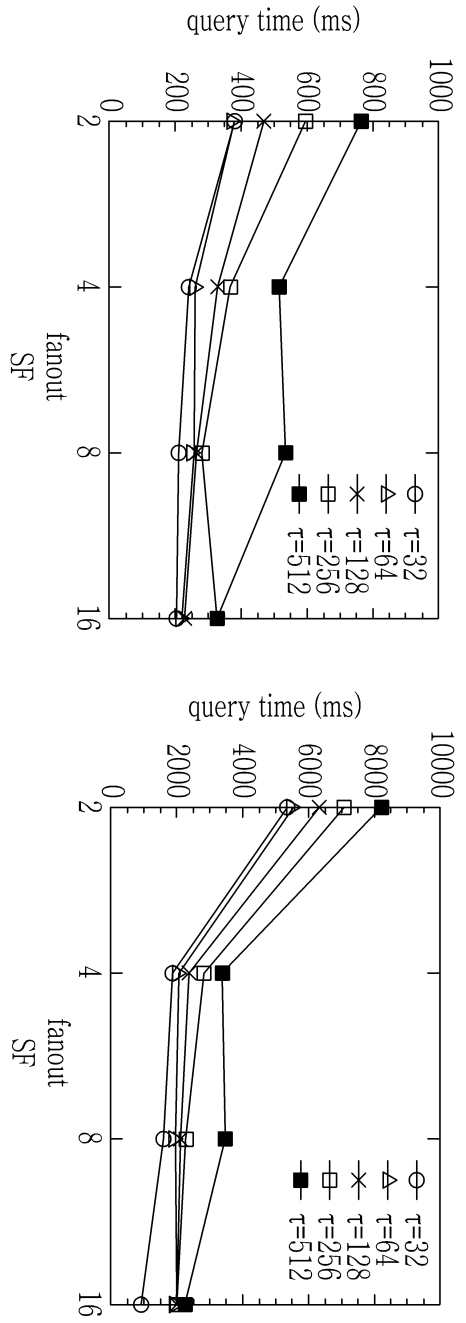
도면7


building time (ms)
0
2000
4000
6000
8000
2
4
8
16
fanout
CA
τ=32
τ=64
τ=128
τ=256
τ=512
building time (ms)
0
2000
4000
6000
8000
2
4
8
16
fanout
SF
τ=32
τ=64
τ=128
τ=256
τ=512

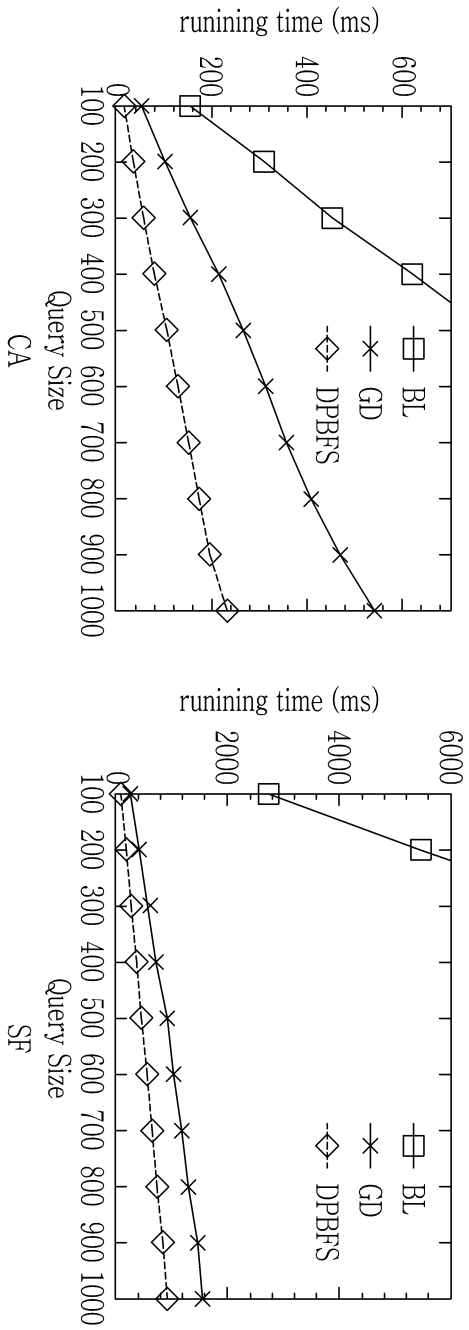
도면8


query time (ms)
0
500
1000
1500
2
4
8
16
fanout
CA
τ=32
τ=64
τ=128
τ=256
τ=512
query time (ms)
0
2000
4000
6000
8000
10000
2
4
8
16
fanout
SF
τ=32
τ=64
τ=128
τ=256
τ=512

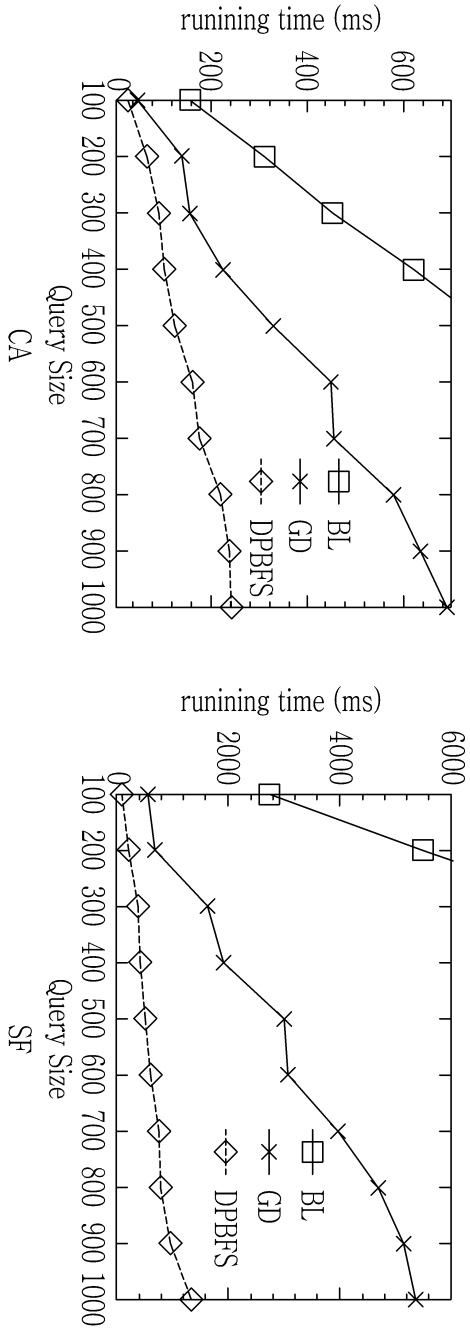
도면9


query time (ms)
0
200
400
600
800
1000
2
4
8
16
fanout
SF
τ=32
τ=64
τ=128
τ=256
τ=512
query time (ms)
0
2000
4000
6000
8000
10000
2
4
8
16
fanout
SF
τ=32
τ=64
τ=128
τ=256
τ=512

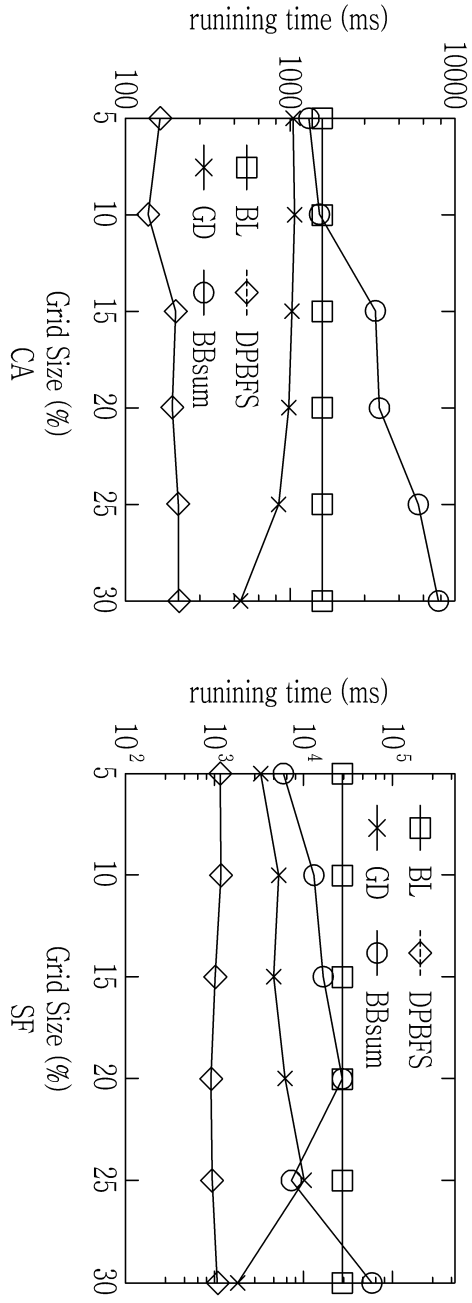
도면10


runining time (ms)
0
200
400
600
100 200 300 400 500 600 700 800 900 1000
Query Size
CA
BL
GD
DPBFS
runining time (ms)
0
2000
4000
6000
100 200 300 400 500 600 700 800 900 1000
Query Size
SF
BL
GD
DPBFS

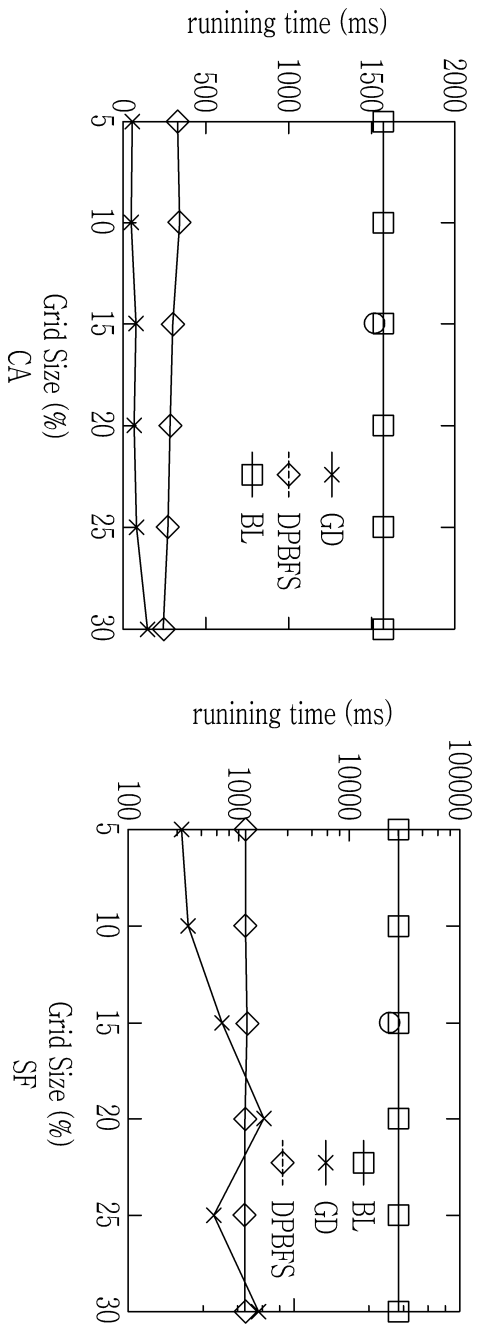
도면11


runining time (ms)
0
200
400
600
100 200 300 400 500 600 700 800 900 1000
Query Size
CA
BL
GD
DPBFS
runining time (ms)
0
2000
4000
6000
100 200 300 400 500 600 700 800 900 1000
Query Size
SF
BL
GD
DPBFS

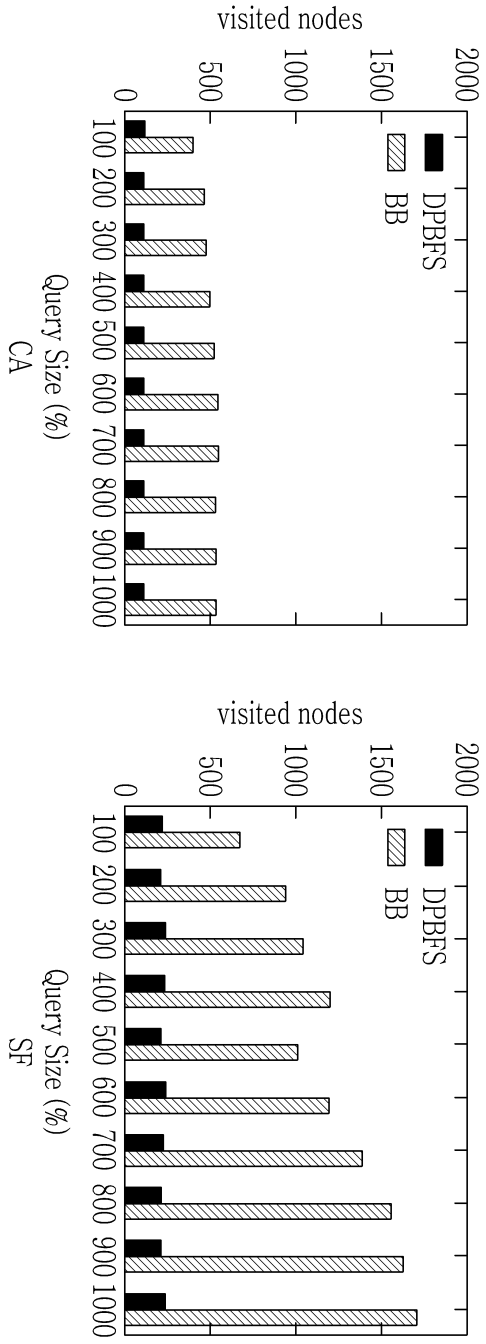
도면12


runining time (ms)
10000
1000
100
5
10
15
20
25
30
Grid Size (%)
CA
BL
DPBFS
GD
BBsum
runining time (ms)
10^5
10^4
10^3
10^2
5
10
15
20
25
30
Grid Size (%)
SF
BL
DPBFS
GD
BBsum

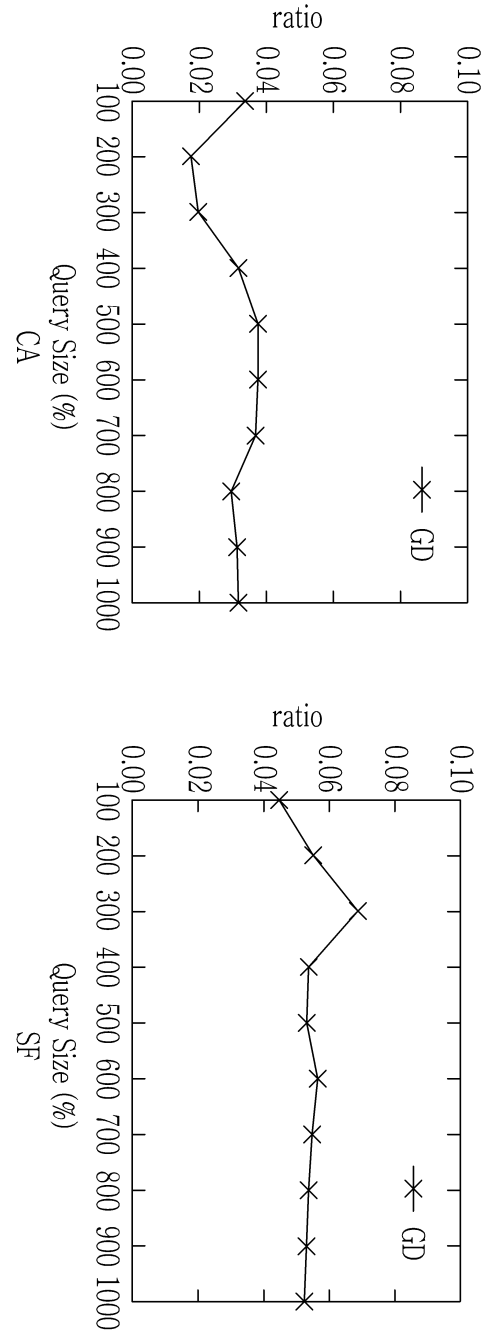
도면13


runining time (ms)
0
500
1000
1500
2000
5
10
15
20
25
30
Grid Size (%)
CA
BL
DPBFS
GD
runining time (ms)
100
1000
10000
100000
5
10
15
20
25
30
Grid Size (%)
SF
BL
GD
DPBFS

도면14


visited nodes
0
500
1000
1500
2000
DPBFS
BB
100 200 300 400 500 600 700 800 900 1000
Query Size (%)
CA
visited nodes
0
500
1000
1500
2000
DPBFS
BB
100 200 300 400 500 600 700 800 900 1000
Query Size (%)
SF

도면15


ratio
0.00
0.02
0.04
0.06
0.08
0.10
100 200 300 400 500 600 700 800 900 1000
Query Size (%)
CA
GD
ratio
0.00
0.02
0.04
0.06
0.08
0.10
100 200 300 400 500 600 700 800 900 1000
Query Size (%)
SF
GD