



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2021년05월17일  
(11) 등록번호 10-2252678  
(24) 등록일자 2021년05월11일

- (51) 국제특허분류(Int. Cl.)  
G06Q 50/30 (2012.01) G06F 16/28 (2019.01)  
G06Q 10/04 (2012.01)
- (52) CPC특허분류  
G06Q 50/30 (2013.01)  
G06F 16/285 (2019.01)
- (21) 출원번호 10-2019-0172708
- (22) 출원일자 2019년12월23일  
심사청구일자 2019년12월23일
- (56) 선행기술조사문헌  
공개특허공보 제10-2015-0061324호(2015.06.04.)\*  
김경옥, 자전거 불균형을 고려한 공유자전거 시스템 운영을 위한 공유자전거 이용 특성에 대한 연구, 2018 서울연구논문공모전, 서울연구원, 2018.10.12.\*  
\*는 심사관에 의하여 인용된 문헌
- (73) 특허권자  
세종대학교산학협력단  
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
- (72) 발명자  
아불가셈  
서울특별시 광진구 능동로 209(군자동, 세종대학교)  
최수미  
서울특별시 광진구 능동로 209(군자동, 세종대학교)
- (74) 대리인  
양성보

전체 청구항 수 : 총 7 항

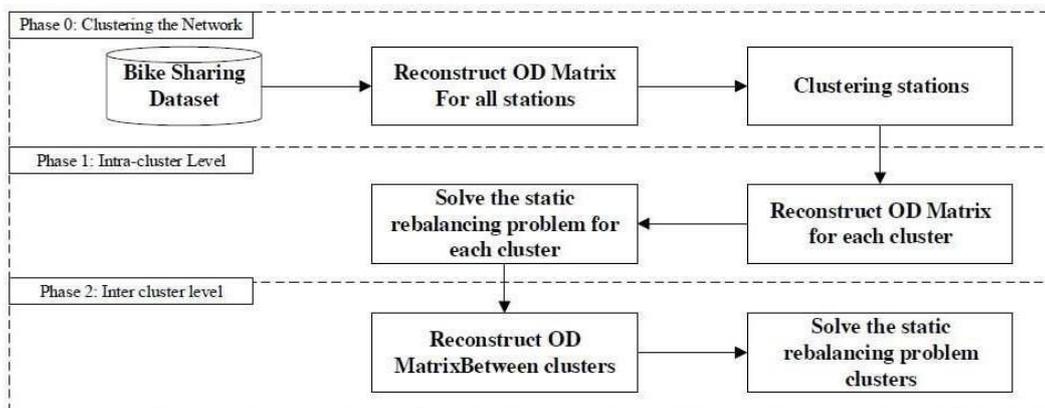
심사관 : 박지은

(54) 발명의 명칭 정적 재조정 자전거 공유 문제를 위한 공간 클러스터 기반 모델

(57) 요약

정적 재조정 자전거 공유 문제를 위한 공간 클러스터 기반 모델이 개시된다. 자전거 공유 재조정 방법은, 자전거 공유 네트워크에 포함된 스테이션 간 여행 패턴에 기초하여 상기 스테이션을 클러스터링하는 단계; 및 클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 단계를 포함한다.

대표도



(52) CPC특허분류  
**G06Q 10/043** (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711093218
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신기술인력양성(R&D)
연구과제명	모바일 플랫폼 기반 엔터테인먼트 VR 기술 연구
기 여 율	1/1
과제수행기관명	세종대학교 산학협력단
연구기간	2019.01.01 ~ 2019.12.31

공지예외적용 : 있음

---

## 명세서

### 청구범위

#### 청구항 1

컴퓨터 시스템에서 실행되는 자전거 공유 재조정 방법에 있어서,

상기 컴퓨터 시스템은 메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고,

상기 자전거 공유 재조정 방법은,

상기 적어도 하나의 프로세서에 의해, 자전거 공유 네트워크에 포함된 스테이션 간 여행 패턴에 기초하여 상기 스테이션을 클러스터링하는 단계; 및

상기 적어도 하나의 프로세서에 의해, 클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 단계

를 포함하고,

상기 재조정을 수행하는 단계는,

단일 개체 유전 알고리즘을 사용하여 클러스터 내부 레벨(Intra-cluster Level) 및 클러스터 간 레벨(Inter-cluster level)에서 양 또는 음의 균형에 따라 재조정 투어 경로를 최적화하는 단계

를 포함하는 자전거 공유 재조정 방법.

#### 청구항 2

제1항에 있어서,

상기 클러스터링하는 단계는,

상기 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지(origin-destination) 매트릭스를 재구성하는 단계; 및

상기 목적지 매트릭스를 이용하여 상기 스테이션 간 유사성에 따라 상기 스테이션을 클러스터링하는 단계

를 포함하는 자전거 공유 재조정 방법.

#### 청구항 3

제1항에 있어서,

상기 클러스터링하는 단계는,

상기 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지 매트릭스를 재구성하는 단계; 및

상기 목적지 매트릭스를 이용하여 상기 스테이션 간 운행 횟수에 따라 상기 스테이션을 클러스터링하는 단계

를 포함하는 자전거 공유 재조정 방법.

#### 청구항 4

제2항에 있어서,

상기 재조정을 수행하는 단계는,

각 클러스터에 대한 목적지 매트릭스를 재구성하여 클러스터 내부 레벨(Intra-cluster Level)에서 각 클러스터 내의 스테이션을 재조정하는 단계

를 포함하는 자전거 공유 재조정 방법.

**청구항 5**

제4항에 있어서,

상기 재조정을 수행하는 단계는,

클러스터들 사이의 목적지 매트릭스를 재구성하여 클러스터 간 레벨(Inter cluster level)에서 클러스터를 다른 클러스터와 연결하는 보상을 통해 재조정하는 단계

를 더 포함하는 자전거 공유 재조정 방법.

**청구항 6**

삭제

**청구항 7**

컴퓨터 시스템에서 실행되는 자전거 공유 재조정 방법에 있어서,

상기 컴퓨터 시스템은 메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고,

상기 자전거 공유 재조정 방법은,

상기 적어도 하나의 프로세서에 의해, 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지 매트릭스를 재구성하는 단계;

상기 적어도 하나의 프로세서에 의해, 상기 목적지 매트릭스를 이용하여 상기 스테이션 간 유사성에 따라 상기 스테이션을 클러스터링하는 단계; 및

상기 적어도 하나의 프로세서에 의해, 클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 단계

를 포함하고,

상기 재조정을 수행하는 단계는,

각 클러스터에 대한 목적지 매트릭스를 재구성하여 클러스터 내부 레벨에서 각 클러스터 내의 스테이션을 재조정하는 단계; 및

클러스터들 사이의 목적지 매트릭스를 재구성하여 클러스터 간 레벨에서 클러스터를 다른 클러스터와 연결하는 보상을 통해 재조정하는 단계

를 포함하는 자전거 공유 재조정 방법.

**청구항 8**

컴퓨터 시스템에 있어서,

메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를

를 포함하고,

상기 적어도 하나의 프로세서는,

자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지 매트릭스를 재구성하는 과정;

상기 목적지 매트릭스를 이용하여 상기 스테이션 간 유사성에 따라 상기 스테이션을 클러스터링하는 과정; 및

클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 과정

을 처리하고,

상기 재조정을 수행하는 과정은,

단일 개체 유전 알고리즘을 사용하여 클러스터 내부 레벨(Intra-cluster Level) 및 클러스터 간 레벨(Inter-cluster level)에서 양 또는 음의 균형에 따라 재조정 투어 경로를 최적화하는 과정을 포함하는 컴퓨터 시스템.

**발명의 설명**

**기술 분야**

[0001] 아래의 설명은 자전거 공유 기술에 관한 것이다.

**배경 기술**

[0002] 근래에 자전거를 이용하는 사용자가 증가하면서 자전거의 공유에 대한 관심이 높아지고 있다. 정부나 여러 지자체에서는 여러 사용자가 자전거를 공유하여 사용하도록 공유자전거를 대여해주는 공공 서비스를 확대 시행하고 있다.

[0003] 예컨대, 한국공개특허공보 제10-2014-0016595호에는 자전거 대여 및 반납시 자전거 거치대로부터 자전거 대여정보를 입력받아 자전거를 반납하는 사용자 정보, 자전거 반납시간정보를 포함하는 자전거 반납정보를 생성하여 통신부를 통해 자전거 거치대로 전송하는 기술이 개시된다.

[0004] 자전거 공유 시스템의 서비스 품질은 관계자에게 중요한 운영 상의 문제다. 서비스 품질을 일정 레벨로 유지하기 위해서는 네트워크의 불균형을 보상하기 위한 재조정이 필요하다. 정적 재조정 문제는 네트워크의 모든 스테이션의 균형을 미리 정해진 레벨로 유지하는 것이다. 재조정의 과정은 보통 차량에 의해 수행된다. 이 차량은 차고에서 출발하여 스테이션의 빈 도크(dock)로 자전거를 배달하거나 과부하된 스테이션에서 자전거를 픽업한다. 이 과정을 네트워크를 통해 실행하는 데 드는 비용은 매우 높을 수 있으며, 이는 자전거 공유 시스템 관리에 있어 중요한 문제가 된다. 따라서 최소한의 비용으로 네트워크의 균형을 유지하기 위한 재조정 전략을 찾는 것이 운영자에게 이상적인 경우가 될 수 있다.

[0005] 자전거 공유 시스템은 첫/마지막 마일 문제에 대한 효과적이고 환경 친화적인 해결책 중 하나이다. 자전거 공유 시스템은 특히 자동차 여행, 대기 오염, 연료 소비 및 주차 공간의 필요성을 감소시킨다. 이러한 시스템은 많은 도시 지역에서 공공 교통 시스템의 주요 보완 모드 중 하나가 되고 있다. 스테이션에 대한 접근성, 자전거의 가용성 및 빈 도크 및 요금과 같은 다른 요인들이 이러한 시스템의 효과에 영향을 미치며, 그 중 하루 동안의 자전거 가용성은 이러한 서비스의 설계, 운영 및 관리에 있어 가장 어려운 문제 중 하나이다.

[0006] 자전거 공유 시스템은 편도 여행에 많이 사용되며, 그러한 경향은 시간과 공간의 부적절한 자전거 분배로 이어진다. 결과적으로, 시스템에 충족되지 않은 수요와 그에 따른 사용자 불만이 발생한다. 수요 불만족(즉, 사용량 및 수익 극대화)을 최소화하기 위해서는 스테이션에 자전거의 재분배가 필요하다. 재조정이라고도 하는 이 재분배는 차량(대개 트럭)을 채용하여 과부하된 스테이션에서 잉여 자전거를 수집하고 자전거가 부족한 스테이션에 납품함으로써 구현된다. 각 스테이션의 자전거와 빈 도크의 수가 사전 정의된 목표값으로 주기적으로 복원되도록 하기 위해서는 관계자의 재조정 운영의 효율성이 중요하다.

[0007] 네트워크에서 자전거의 균형 재조정은 두 가지 다른 방식으로 수행된다. 즉, 사용자가 매일 여행하거나 특정 여행에 대해 보너스를 받을 수 있는 사용자 기반 접근 방식과 정적이든 동적이든 간에 자전거가 시스템 운영자에 의해 재조정되는 운영자 기반 접근방식이다. 정적 재조정은 시스템의 수요가 미미한 밤중에 이루어진다. 동적 재조정은 자전거 공유 시스템의 실시간 사용을 고려하여 시스템이 활성 상태인 낮에 수행된다. 운영자 기반 자전거 재조정 문제(BRP)는 픽업(Pickup), 배송(Delivery), 및 외판원(traveling Salesman)을 포함한다. 재조정의 목적은 충족되지 않은 수요를 최소화하고, 총 여행 시간과 비용을 최소화하며, 차량의 투어 길이를 최소화하고, 투어 수를 최소화하는 것이다.

[0008] 정적 재조정은 대조적으로 동적 재조정은 대개 시스템의 크기와 범위가 상대적으로 큰 대부분의 경우에 활용되지 않을 정도로 복잡하고 비용이 상당히 많이 든다. 자전거 공유 시스템에서는 가급적 아침 수요(아침 피크)에 대한 자전거의 균형을 재조정하기 위해 운영자에 의해 정적 재조정이 수행된다. 다른 시간대에 사용자는 시스템에서 동적 자전거 재조정을 처리한다.

[0009] 궁극적으로, 사용자의 행동과 여행 패턴은 시스템에 대한 자전거의 분배에 상당한 영향을 미치며, 야간 운영자에 의해 재조정되어야 하는 자전거의 수와 방법을 결정한다. 따라서, 착수된 여행의 원래 도착지(OD) 매트릭스

는 정적 재조정에서 고려될 필요가 있는 주요 사실 중 하나이다. 어느 스테이션이 재조정이 필요한지 구체적으로 밝히는 것이 중요하다. 이로써 스테이션 간 추가적인 투어 재조정을 없앨 수 있을 것이다. 즉, 사용자들이 재조정된 비율과 자전거 공유 서비스 직원이 얼마나 많은 것을 수행해야 하는지를 보여준다.

**발명의 내용**

**해결하려는 과제**

[0010] 본 발명은 자전거 공유 시스템의 정적 재조정 문제를 해결하기 위한 상향식 공간 클러스터 기반 모델을 제안한다. 첫째, 네트워크에서 자전거 공유 여행의 공간적, 시간적 패턴을 조사한다. 둘째, 스테이션 간 이동에 기초한 유사성 측정은 계층적 집적 클러스터링 방법을 사용하여 상관 관계 스테이션 그룹을 발견하기 위해 정의된다. 셋째, 네트워크의 균형을 하루의 시작으로 유지하기 위한 목적으로 두 단계의 재조정을 클러스터 내와 클러스터 간으로 가정한다. 클러스터 내 레벨은 각 클러스터 내의 자전거 분배의 균형을 유지하며, 클러스터 간 균형을 유지하기 위해 클러스터 간 레벨이 서로 다른 클러스터를 연결한다. 마지막으로, 단일 개체 유전 알고리즘을 사용하여 클러스터 내 및 클러스터 간 레벨에서 양 또는 음의 균형에 따라 재조정 투어가 최적화된다. 재조정 문제는 최적화 문제로 모델링되어 있어 투어 길이를 최소화하는 것을 목표로 한다.

**과제의 해결 수단**

[0011] 컴퓨터 시스템에서 실행되는 자전거 공유 재조정 방법에 있어서, 상기 컴퓨터 시스템은 메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 상기 자전거 공유 재조정 방법은, 상기 적어도 하나의 프로세서에 의해, 자전거 공유 네트워크에 포함된 스테이션 간 여행 패턴에 기초하여 상기 스테이션을 클러스터링하는 단계; 및 상기 적어도 하나의 프로세서에 의해, 클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 단계를 포함하는 자전거 공유 재조정 방법을 제공한다.

[0012] 일 측면에 따르면, 상기 클러스터링하는 단계는, 상기 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지(origin-destination) 매트릭스를 재구성하는 단계; 및 상기 목적지 매트릭스를 이용하여 상기 스테이션 간 유사성에 따라 상기 스테이션을 클러스터링하는 단계를 포함할 수 있다.

[0013] 다른 측면에 따르면, 상기 클러스터링하는 단계는, 상기 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지 매트릭스를 재구성하는 단계; 및 상기 목적지 매트릭스를 이용하여 상기 스테이션 간 운행 횟수에 따라 상기 스테이션을 클러스터링하는 단계를 포함할 수 있다.

[0014] 또 다른 측면에 따르면, 상기 재조정을 수행하는 단계는, 각 클러스터에 대한 목적지 매트릭스를 재구성하여 클러스터 내부 레벨(Intra-cluster Level)에서 각 클러스터 내의 스테이션을 재조정하는 단계를 포함할 수 있다.

[0015] 또 다른 측면에 따르면, 상기 재조정을 수행하는 단계는, 클러스터들 사이의 목적지 매트릭스를 재구성하여 클러스터 간 레벨(Inter cluster level)에서 클러스터를 다른 클러스터와 연결하는 보상을 통해 재조정하는 단계를 더 포함할 수 있다.

[0016] 또 다른 측면에 따르면, 상기 재조정을 수행하는 단계는, 단일 개체 유전 알고리즘을 사용하여 클러스터 내부 레벨(Intra-cluster Level) 및 클러스터 간 레벨(Inter cluster level)에서 양 또는 음의 균형에 따라 재조정 투어 경로를 최적화하는 단계를 포함할 수 있다.

[0017] 컴퓨터 시스템에서 실행되는 자전거 공유 재조정 방법에 있어서, 상기 컴퓨터 시스템은 메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 상기 자전거 공유 재조정 방법은, 상기 적어도 하나의 프로세서에 의해, 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기 스테이션 간 여행에서 파생된 목적지 매트릭스를 재구성하는 단계; 상기 적어도 하나의 프로세서에 의해, 상기 목적지 매트릭스를 이용하여 상기 스테이션 간 유사성에 따라 상기 스테이션을 클러스터링하는 단계; 및 상기 적어도 하나의 프로세서에 의해, 클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 단계를 포함하고, 상기 재조정을 수행하는 단계는, 각 클러스터에 대한 목적지 매트릭스를 재구성하여 클러스터 내부 레벨에서 각 클러스터 내의 스테이션을 재조정하는 단계; 및 클러스터들 사이의 목적지 매트릭스를 재구성하여 클러스터 간 레벨에서 클러스터를 다른 클러스터와 연결하는 보상을 통해 재조정하는 단계를 포함하는 자전거 공유 재조정 방법을 제공한다.

[0018] 컴퓨터 시스템에 있어서, 메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, 자전거 공유 네트워크에 포함된 모든 스테이션에 대해 상기

스테이션 간 여행에서 파생된 목적지 매트릭스를 재구성하는 과정; 상기 목적지 매트릭스를 이용하여 상기 스테이션 간 유사성에 따라 상기 스테이션을 클러스터링하는 과정; 및 클러스터를 이용하여 상기 스테이션의 자전거 재분배를 위한 재조정을 수행하는 과정을 처리하는 컴퓨터 시스템을 제공한다.

**도면의 간단한 설명**

- [0019] 도 1은 본 발명의 일 실시예에 있어서 자전거 공유를 위한 재조정 실행 절차의 예를 도시한 것이다.
- 도 2는 본 발명의 일 실시예에 있어서 클러스터를 기반으로 자전거 공유 시스템의 정적 재조정 과정을 도시한 흐름도이다.
- 도 3은 본 발명의 일 실시예에 있어서 자전거 공유 데이터 셋과 출발지(OD) 매트릭스 예시를 도시한 것이다.
- 도 4는 본 발명의 일 실시예에 있어서 유전 알고리즘 주요 과정을 도시한 흐름도이다.
- 도 5는 본 발명의 일 실시예에 있어서 유전 알고리즘 시각화 예시를 도시한 것이다.
- 도 6 내지 도 9는 본 발명의 일 실시예에 있어서 정적 재조정 문제에서 유전 알고리즘을 구현하기 위한 의사코드 예시를 도시한 것이다.
- 도 10은 본 발명의 일 실시예에 있어서 컴퓨터 시스템의 내부 구성의 일례를 설명하기 위한 블록도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0020] 이하, 본 발명의 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0021] 본 발명의 실시예들은 자전거 공유 기술에 관한 것이다.
- [0022] 본 명세서에서 구체적으로 개시되는 것들을 포함하는 실시예들은 자전거 공유 시스템의 정적 재조정 문제를 해결하기 위한 상향식 공간 클러스터 기반 모델을 제공한다.
- [0023] 자전거 재조정 문제는 자전거 공유 시스템 작동의 주요 문제 중 하나이다. 재조정의 목적은 각 스테이션에서 적재/하차하는 자전거의 수를 명시하고 다양한 제약을 받는 최적의 트럭 경로를 찾는 것이다. 정적 및 동적 재조에 대한 대부분의 연구는 차량 여행 시간 및 비용 최소화, 투어 길이, 차량 배출량, 사용자 불만족 및 충족되지 않은 요구 최소화를 포함한 목표를 채택한다.
- [0024] Static Rebalancing Problem
- [0025] 정적 재조정 문제는 대부분 수학적 접근방식으로 최적화 문제를 해결하는 것을 다루는데, 여기서 간단히 설명된다. 재조정 문제에 대한 연구도 클러스터링 단계가 있는 연구와 없는 연구로 나눌 수 있다.
- [0026] 일부 연구에서는 총 여행 거리를 최소화하는 항로를 찾기 위해 수학-경험적 방법을 제안한다. 그들의 알고리즘은 분기 절단법 절차에 의해 풀리는 문제의 완화 솔루션에 근거한다. 한편, 다른 연구에서는 여행 비용 최소화를 목표로 4개의 혼합 정수 선형 프로그래밍을 제시한다. 그들의 계산 시간은 효율적으로 최대 50개의 꼭지점까지 네트워크를 해결하고, 기하 급수적으로 더 큰 네트워크로 확장한다. 다른 목표로, 또 다른 연구에서는 각 스테이션의 불만족 수요를 예측하고 추정된 조치에 대한 알고리즘을 제안함으로써 자전거 공유 시스템의 서비스 레벨을 최적화한다. 상대적으로 작은 네트워크에서 시험한 그들의 절차는 매우 빠른 경험적 알고리즘임을 증명한다. 일부 연구에서는 재조정을 위한 정확한 알고리즘을 제시한다. 일례로, 견잡을 수 없는 자전거 공유(도크가 없는) 시스템에서 정적 완전 재조정 문제를 해결하기 위한 혼합 정수 선형 프로그램을 제공한다. 다른 예로는, 다양한 유형의 자전거의 총 재조정 비용을 최소화하기 위해 혼합 정수 선형 프로그래밍을 제시한다. 복합 유전 알고리즘을 제안하고 있는데, 이 알고리즘은 복합 유전 검색이 최적의 경로와 스테이션에서 자전거의 적재/하차 횟수를 명시하기 위한 욕심 많은 경험적 방법을 결정한다. 상기한 연구들은 소규모 네트워크에 효과적이다. 또 다른 연구에서는 하이브리드 반복 로컬 검색(ILS) 알고리즘을 제안한다. 상기한 연구들은 효율적인 방법에서 실세계의 크기 문제를 해결할 수 있는 더 나은 수학적 접근방법을 제안함에 초점을 맞추고 있다.
- [0027] 다른 연구들은 재조정 문제를 단순화하기 위해 클러스터링 방법을 사용한다. 일례로는 다항식 크기 클러스터링 문제에 대한 재고 재조정 및 차량 경로(클러스터-첫 번째 경로-두 번째 경험적 방법)를 제안하고 서비스 레벨 타당성과 대략적인 경로 비용을 동시에 고려한다. 다른 예로는 3단계 경험적 방법과 하이브리드를 사용하여 서로 다른 시간 간격과 이질적인 차량의 다중 차고 경로 문제를 조사한다. 이들의 접근방식은 경험적 클러스터링 알고리즘에 의해 소형 혼합 정수 선형 프로그래밍(MILP)의 하위 문제로 나눈 다음 차고 이기종 차량 경로 문제

에 대한 최적 경로(최소 여행 비용)를 찾는 것을 목표로 한다. 유사한 접근방식의 다른 연구에서는 지리적으로 스테이션을 클러스터한 후에 차량 경로를 결정한다. 또 다른 연구에서는 투어 길이를 최소화할 목적으로 혼합 정수 비선형 프로그래밍을 해결하기 위한 적응형 용량 제약 K-센터 클러스터링 알고리즘을 제안한다. 결과적으로 클러스터링 알고리즘을 사용하여 재조정 문제의 질을 높이면 문제의 크기를 성공적으로 처리하고 더 작은 문제로 분해할 수 있다.

[0028] Dynamic Rebalancing Problem

[0029] 동적 재조정의 주된 목적은 각 시간 간격에 유지되어야 하는 최적의 재고 레벨을 결정하는 것이다. 동적 재조정에서 발생하는 과제는 전용적이고 효율적인 방법론을 필요로 한다. 세 가지 핵심 요소(사용자 불만족 추정, 자전거 재조정 및 차량 경로)를 완전히 통합하는 복잡성을 고려하여 동적 재조정의 몇 가지 부분적 또는 순차적 솔루션 방법론을 제안한다.

[0030] 일부 연구에서는 순차적인 프레임워크를 제안한다. 첫째, 수요 예측 모델과 재고 레벨 모델을 해결한 다음, 재분배 수요를 창출하고 마지막으로 차량 경로를 계산한다. 다른 연구에서는 차량 경로 및 충족되지 않은 수요의 최적화 외에도, 환경 측면을 동적 재조정 문제의 목표로 삼아 CO2 배출 및 관련 비용을 최소화한다. 자전거에 대한 다양한 수요를 처리하기 위해 동적 재조정이 몇 가지 정적 재조정의 문제로 분해되는 롤링 수평선 접근법을 사용한다. 강화된 인공 벌 군집 알고리즘과 경로 절단 경험적 방법이 각 단계에서 경로 설계를 최적화하기 위해 공동으로 사용되며, 적재 및 하차 경험적 방법은 각 단계에서 경로를 따라 적재 하차하는 하위 문제를 해결하기 위해 사용된다. 그들은 단계 길이가 짧을수록 더 좋은 해결책이 된다는 것을 보여준다; 추가적으로 더 긴 단계 길이에서 계산 시간, 총 미충족 수요, 그리고 총 연료와 CO2 배출 비용이 더 크다. 이것은 문제를 하위 문제로 나누는 것을 강화한다. 문제가 작을수록 더 좋고 빠른 솔루션이다.

[0031] 또 다른 연구에서는 동적 재조정의 세 가지 핵심 요소(자전거 재조정, 차량 경로 및 사용자 불만족)를 완전히 통합하기 위한 비선형 시간 공간 네트워크 플로우 베이스 공식을 제안한다. 차량 도착 시간을 기준으로 예측 기간을 두 개의 하위 기간으로 분해하여 사용자의 불만을 추정한다. 한편, 또 다른 연구에서는 균형 잡힌 수요를 강요하고 따라서 재조정의 필요성을 완전히 절감하는 동적 가격 결정 메커니즘을 제시한다. 그러한 접근방식은 분명히 시스템 사용자에게 제공되는 서비스 레벨을 줄이는 데 비용이 들지만 그럼에도 불구하고 재조정이 너무 비싼 자동차 공유 시스템의 균형을 맞추는 유일한 실행 가능한 대안으로 보인다. 또 다른 연구에서는 사용자 불만족 추정치가 단순화되고 결과적으로 추정 정확성이 희생되는 시간공간 네트워크에 기초한 방법론을 제안한다.

[0032] 사용자 만족도는 건잡을 수 없는 자전거 공유 시스템에 중요해지고 동적 재조정은 문제를 더 복잡하게 만든다. 복잡성을 줄이기 위해 일부 연구에서는 일정한 시간 간격에서 동적 클러스터링 방법을 제안한다. 도시 내 서로 다른 구역의 공간-임시 상관관계 패턴 후 높은 레벨의 사용자 만족도를 달성하고 재조정의 운영 비용을 가능한 낮게 유지하기 위해 의사결정 지원 시스템을 개발한다. 따라서 동적 재조정 문제의 연구는 공간적 및 시간적 차원의 문제 모두를 해결하는 다양한 경험적 최적화 방법에 초점을 맞추고 있다.

[0033] 결론적으로, 자전거 공유 재조정 문제는 최근 몇 년 동안 많은 연구자들의 관심을 끌고 있다. 대다수의 연구는 사용자의 행동과 재조정 문제에서의 그들의 역할을 고려하지 않고 최적화 문제 자체를 해결하는 데 초점을 맞추고 있다. 또한, 대부분의 연구는 비교적 작은 네트워크에 대한 실험을 수행하며 문헌은 대부분 벡터 그래프로 네트워크를 단순화하고 조사한다. 대규모 최적화 문제의 복잡성을 줄이기 위해 클러스터링을 사용하는 연구는 거의 없다. 사용자의 행동은 수요와 공급 사이의 내재적 상호작용의 결과물이다. 거리 및 지리적 지형에 따른 자전거(공급)의 가용성은 사용자의 행동과 결과적으로 자전거와 도크의 수요에 영향을 미친다. 스테이션 간 이동을 고려하여 네트워크를 클러스터링하면 네트워크의 분석이 더 간단하고 효율적이며 불필요한 투어를 없앨 수 있다.

[0034] 본 발명에서는 자전거 공유 시스템의 정적 재조정 문제를 해결하기 위해 상향식 클러스터 기반 모델을 제안한다. 첫째, 네트워크에서 자전거 공유 여행의 공간적, 시간적 패턴을 조사한다. 둘째, 스테이션 간 이동에 기초한 유사성 측정은 계층적 집적 클러스터링 방법을 사용하여 상관 관계 스테이션 그룹을 발견하기 위해 정의된다. 셋째, 네트워크의 균형을 하루의 시작으로 유지하기 위한 목적으로 재조정을 위한 두 가지 레벨을 클러스터 내와 클러스터 간으로 가정한다. 클러스터 내 레벨은 각 클러스터 내의 자전거 분배의 균형을 유지하며, 클러스터 간 균형을 유지하기 위해 클러스터 간 레벨이 서로 다른 클러스터를 연결한다. 마지막으로, 단일 개체 유전 알고리즘을 사용하여 클러스터 내 및 클러스터 간 레벨에서 양 또는 음의 균형에 따라 재조정 투어가 최적화된다. 재조정 문제는 최적화 문제로 모델링되어 있어 투어 길이를 최소화하는 것을 목표로 한다.

- [0035] 본 발명은 다음을 목표로 상향식 클러스터 기반 모델을 제안한다.
- [0036] (1) 네트워크에서 사용자의 행동 및 사용자 기반 재조정 부분 결정
- [0037] (2) 재조정이 필요한 스테이션 클러스터링
- [0038] (3) 재조정 투어 길이(비용) 최소화
- [0039] 스테이션을 클러스터링한 후 여행을 두 개의 다른 범주, 즉 클러스터 내와 클러스터 간으로 나뉜다. 클러스터 내 여행은 클러스터 내에서 발생하는 것이며, 클러스터 간 여행은 두 클러스터 간에 발생하는 것을 의미한다. 이러한 여행은 각 클러스터 내부와 외부 자전거의 분배를 분산시키고 다시 균형을 맞출 필요가 있다. 이러한 재조정은 부분적으로 사용자 활동을 통해 이루어지며, 일부는 재조정의 차량에 의해 이루어진다. 따라서, 사용자 기반 재조정의 비율을 결정하여 운영자가 재조정된 자전거의 양을 명시할 필요가 있다. 여행과 마찬가지로, 재조정은 클러스터 간과 클러스터 내를 통해 수행되어야 한다. 특정 용량의 단일 차량과의 정적 재조정 작업이 가정된다. 이 차량은 한 번의 투어로 여러 번 스테이션을 방문할 수 있다. 또한, 차량은 일부 자전거와 함께 차고를 출발하여 최종 단계에서 동일한 수의 자전거로 차고로 되돌아간다. 이 자전거의 수는 클러스터 간 재조정 최적화에 따라 결정된다. 재조정 실행 절차는 도 1에 요약되어 있다.
- [0040] 재조정의 목표는 각 스테이션의 균형을 미리 정해진 레벨(초기 균형)으로 유지하는 것이다. 초기 상태는 자전거 공유 거래의 데이터 집합으로 계산된다. 각 클러스터의 스테이션을 재조정하기 위해 정적 재조정 문제가 해결되어야 한다. 각 클러스터의 재조정은 다음 단계로 다른 클러스터와 클러스터를 연결함으로써 보상하는 전체 클러스터의 양 또는 음의 균형으로 이어질 것이다. 쉬운 유전적 최적화 알고리즘은 정적 재조정 문제를 다른 연구에서의 라우팅 문제를 해결하는 데 긍정적인 실행을 가지고 있는 재조정을 해결하는 데 사용된다.
- [0041] 먼저, 명확성을 위해 이하에서 사용되는 수학적 기호를 설명한다.
- [0042] (1) 인덱스 표기
- [0043]  $i$ : 스테이션 ID
- [0044]  $j$ : 차량 ID
- [0045] (2) 문제 매개변수 표기
- [0046]  $I$ : 네트워크에 있는 모든 스테이션의 ID 집합
- [0047]  $IBI_i$ : 스테이션  $i$ 에 진입하는 자전거의 수
- [0048]  $OBI_i$ : 스테이션  $i$ 에서 나가는 자전거의 수
- [0049]  $B_i$ : 스테이션  $i$ 의 균형(스테이션  $i$ 에 마지막까지 남아 있는 자전거의 수)
- [0050] ( $B_i = IBI_i - OBI_i$ )
- [0051]  $Cap_j$ : 차량의 용량
- [0052] (3) 의사결정 변수 표기
- [0053]  $T$ : 차량 투어 ( $T = S_1, \Delta B_1, S_2, \Delta B_2, \dots, S_m, \Delta B_m$ )
- [0054]  $\Delta B_m$ :  $m$ 번째 스테이션으로 배달되거나 혹은  $m$ 번째 스테이션에서 픽업되는 자전거 수
- [0055]  $S_m$ :  $m$ 번째 스테이션 ID
- [0056] 자전거 공유 네트워크는 스테이션의 숫자로 구성되어 있으며, 스테이션에는 차지하거나 비어있을 수 있는 도크가 포함되어 있다. 하루 동안 이용자에 의해 여러 대의 자전거가 스테이션에 진입하거나 빠져 나오므로, 하루의 끝에는 스테이션의 균형이 깨진다. 각 스테이션에 대한 자전거의 입출고 간의 차이는 0이 되어야 하는 스테이션의 균형으로 간주된다. 일부 스테이션에서 사용자는 하루의 끝에 스테이션의 균형을 0으로 유지한다(사용자 기반 재조정). 그러나 일부 다른 스테이션은 차량에 의해 0으로 재조정될 필요가 있다. 관계자는 보통 네트워크 균형을 이루기 위하여 내/외 스테이션에서 배달/픽업하려고 특정 용량의 차량을 사용한다. 투어를 시작할 때 차

량의 자전거 수는 0개 또는 차량 내부에 있는 자전거의 양수일 수 있다.

[0057] 스테이션 재조정을 위해 차량을 운행하는 비용은 관계자에게 어려운 문제여서 가능한 한 최소한으로 할 필요가 있다. 재조정의 운영 비용은 많은 요인에 따라 달라지는데, 그 중 차량 수와 주행 거리라는 두 가지 요인에 의해 비용이 크게 영향을 받는다. 따라서, 의사결정 변수는 네트워크를 재조정하는 차량의 투어이다. 모든 투어에는 스테이션 ID와 특정 스테이션에서 적재/하차(배송/픽업)해야 하는 자전거 수가 포함된다. 정적 재조정 문제에 대한 목적함수와 제약조건은 다음 수학적식 1 내지 수학적식 6에 제시되어 있다. 여기서,  $D$ 는 네트워크에서 두 스테이션 사이의 네트워크 거리를 나타내고,  $m$ 은 투어의 스테이션 수이고,  $abs$ 는 절대값을 나타낸다.

[0058] 목적함수는 모든 스테이션의 균형을 재조정하기 위해 차량이 횡단해야 하는 총 네트워크 거리를 최소화하는 것이다.

[0059] [수학적식 1]

$$Min \sum_{i=1}^{m-1} D(S_i, S_{i+1})$$

[0060] 수학적식 1은 투어의 모든 후속 스테이션 사이의 거리를 합산한 총 네트워크 거리를 나타낸다.

[0062] 다음 다섯 가지 제약조건이 있다.

[0063] [수학적식 2]

$$\Delta B_m \leq Cap$$

[0065] [수학적식 3]

$$Cap_j \geq 0$$

[0067] [수학적식 4]

$$abs(\Delta B_m) \leq abs(B_m)$$

[0069] [수학적식 5]

$$\Delta B_m \times B_m < 0$$

[0071] [수학적식 6]

$$\Delta B_m + B_m = 0 : i \in I$$

[0073] 수학적식 2는 스테이션 간 이동 자전거 수를 나타내며, 차량 용량과 적거나 같아야 한다. 수학적식 3은 차량의 용량이 음수가 될 수 없음을 보장한다. 수학적식 4는 모든 스테이션에서 배달/픽업된 자전거의 수가 해당 스테이션의 균형보다 클 수 없음을 나타낸다. 수학적식 5는 차량이 항상 스테이션의 균형을 재조정하는 데 도움이 되도록 보장하며, 예를 들어, 스테이션의 균형이 음일 경우, 차량은 스테이션에 자전거만 전달할 수 있다(되도록 음이 만들어지지 않는다). 수학적식 6은 네트워크 상의 모든 스테이션의 균형은 투어가 끝날 때 0이어야 한다는 것을 보증한다.

[0074] 이하에서는 정적 재조정 문제를 해결하기 위해 제안된 접근방식을 설명한다.

[0075] 도 2는 재조정 문제를 해결하기 위해 제안된 접근방식을 도시한 것이다.

[0076] 도 2를 참조하면, 본 발명에 따른 접근방식은 자전거 공유 거래의 데이터 집합에서 네트워크 내의 스테이션 사이의 OD(origin-destination) 매트릭스를 도출하는 것으로 시작한다. 그 다음, 스테이션은 그 사이의 운행 횟수에 따라 클러스터 된다. 다음으로, 이전 단계에서 발견된 각 클러스터에 대해 하나의 OD 매트릭스가 재구성된다. 각 클러스터의 스테이션(클러스터 내 레벨)을 재조정하기 위해 각 클러스터에 대해 정적 재조정 문제가 해결된다. 각 클러스터를 재조정하면 전체 클러스터에 양 또는 음의 균형이 발생하며, 다음 단계(클러스터 내부 레벨)에서 해당 클러스터를 다른 클러스터와 연결하여 보상된다. OD 매트릭스는 클러스터 간 여행에 의해

계산되며, 이는 실제로 재조정의 첫 번째 레벨(클러스터 내 레벨)에서 자전거 외에 클러스터 간 사용자들의 자전거 거래 수에 초점을 맞춘다.

[0077] 자전거 공유 데이터 집합에는 사용자의 여행이 포함된다. 모든 여행은 독특한 ID를 가지고 있다. 모든 여행은 두 역 사이에서 출발지와 목적지로 일어난다. 또한 데이터 집합에 여행 시작 및 종료 시간이 포함된다. 데이터 집합은 일반적으로 데이터 누락이나 불합리한 여행 길이 또는 지속 시간의 경우 명확화 프로세스가 필요하다는 점에 유의해야 한다. 네트워크의 모든 스테이션에는 OD 매트릭스를 재구성하는 데 도움이 되는 고유한 ID가 있다. OD 매트릭스의 모든 셀은 여행 방향을 고려한 두 스테이션 사이의 여행 수를 보여준다. 도 3은 네트워크 상의 5개 스테이션에 대한 데이터 집합과 파생된 OD 매트릭스의 예를 제시한다. 이 예는 네트워크상의 5개 스테이션 사이의 17번의 여행을 보여준다. 그리고 이러한 스테이션 간 여행에서 파생된 OD 매트릭스는 도 3의 아래 도면에 제시된다.

[0078] 네트워크의 모든 스테이션에 대해 생성된 OD 매트릭스는 스테이션 클러스터링에 사용된다. 여기서의 유사성 측정은 스테이션 간 운행 횟수이다. 이 논문은 클러스터 내 총 분산을 최소화하는 워드(Ward) 방법을 사용하여 집적 계층적 클러스터링 알고리즘을 활용한다. Single, Average, Complete, Ward 등 다양한 방법으로 집적 계층적 클러스터링 알고리즘을 구현할 수 있으며, 일례로 Ward 방법을 선택한다. 클러스터 수를 결정할 필요가 없고 서로 다른 유사성 척도로 유연하기 때문에 선택된다. 각 개체가 자체 클러스터를 가지고 있는 바닥에서 시작하여 상단에서 모든 개체가 하나의 클러스터를 형성할 때까지 이들을 병합한다. 계층적 집적형 클러스터링의 결과는 각 단계에서 개체가 어떻게 병합되는지를 보여주는 계통도이다. 계통도와 Silhouette 정보의 모양에 따라 계통도는 적절한 레벨에서 자를 수 있다. Silhouette 정보는 데이터 클러스터 내의 일관성을 해석하고 검증하는 방법을 의미한다. Ward의 최소 분산 방법에서 초기 클러스터 거리는 수학적 7과 같이 지점 사이의 제곱 유클리드(Euclidean) 거리로 정의되며, 이 거리  $d_{qp}$ 가 두 스테이션 q와 p 사이의 제곱 유클리드 거리를 나타낸다. 스테이션을 클러스터링하면 대부분의 상호 작용(여행)을 함께 가지는 스테이션 그룹을 발견하는 데 도움이 된다. 클러스터링의 분명한 이점은 문제를 계산적으로 해결하기 쉬운 몇 가지 하위 문제로 나누는 것이다.

[0079] [수학식 7]

[0080] 
$$d_{qp} = d(X_q, X_p) = \| X_q - X_p \|^2$$

[0081] 스테이션 클러스터를 발견한 후, 네트워크의 각 단일 클러스터에 대해 정적 재조정 문제가 해결된다. 클러스터 내 재조정 문제는 클러스터 내 여행으로 인해 발생하는 불균형에 초점을 맞춘다. 클러스터 내 재조정은 각 클러스터 내의 스테이션을 재조정하는 것을 목표로 한다. 그러나 서로 다른 클러스터 간의 여행은 피할 수 없으며 다음 레벨(클러스터 간) 재조정에서 다루어지는 클러스터의 전체 양 또는 음의 균형으로 간주해야 한다. 정적 재조정 문제는 각 클러스터에 대해 해결된다. 문제는 문제의 크기와 복잡성에 따라 다양한 결정론적 또는 경험적 알고리즘으로 해결할 수 있는 단일 개체 최적화 문제이다. 일례로, 최적화 알고리즘은 경로 문제를 해결하는 데 긍정적인 성과를 가지고 있는 유전 최적화 알고리즘을 이용한다.

[0082] 클러스터 내 재조정의 다음 단계는 클러스터 간 이동과 이전 재조정의 결과에 초점을 맞춘 클러스터 간 재조정이다. 클러스터 내 레벨과 마찬가지로 클러스터 간에 발생한 모든 여행 사이에 OD 매트릭스가 생성되며, 여행의 원점과 목적지는 동일한 클러스터에서 생성되지 않는다. 또한 클러스터 간 레벨에서 정적 재조정 문제는 이전 단계와 유사한 유전 알고리즘으로 해결된다.

[0083] 유전 알고리즘은 진화적 경험적 최적화 방법으로서, 생물학적 용어로 개체 발생의 염색체를 닮은 솔루션 집합을 생산한다. 모든 염색체는 유전자라고 불리는 독립된 단위로 구성되어 있는데, 이것은 사실 솔루션의 성분이다. 유전 최적화 알고리즘은 문제의 제약조건에 근거하여 솔루션의 초기 집단이 생성되는 초기화 단계에서 시작한다. 또한 모집단 크기, 교차 및 돌연변이 발생률은 초기화 레벨에서 설정해야 한다. 유전 알고리즘 메인 루프의 1차 단계는 도 4와 같다.

[0084] 도 5는 가정한 길이 값을 가진 두 개의 가능한 투어를 나타내는 두 개의 염색체에 대한 간단한 시각화를 보여준다. 각 염색체는 분리된 셀로 구성되어 있으며, 홀수 셀은 스테이션의 ID를 나타내며, 셀도 재조정을 위한 자전거의 수를 나타낸다. 첫 번째의 두 염색체는 크로스오버(cross-over)를 위해 선택되며, B 스테이션에서 부착된다. 또한, 크로스오버에서 생성된 아이는 투어 길이가 다른 새로운 염색체로 변형된다. 마지막으로, 이 네 염색체(투어 길이 값에 관한)의 순위가 부모 중 한 명(염색체 2)은 순위의 바닥에 있음을 보여 준다 제시되어 있다.

[0085] 도 6 내지 도 9에 도시된 알고리즘 1 내지 알고리즘 4는 정적 재조정 문제에서 유전 알고리즘을 구현하기 위한

의사 코드를 제시한다.

- [0086] 도 10은 본 발명의 실시시에 따른 컴퓨터 시스템의 예를 도시한 블록도이다. 예를 들어, 본 발명의 실시예들에 따른 자전거 공유 시스템은 도 10을 통해 도시된 컴퓨터 시스템(1000)에 의해 구현될 수 있다.
- [0087] 도 10에 도시된 바와 같이 컴퓨터 시스템(1000)은 본 발명의 실시예들에 따른 정적 재조정 방법을 실행하기 위한 구성요소로서, 메모리(1010), 프로세서(1020), 통신 인터페이스(1030), 그리고 입출력 인터페이스(1040)를 포함할 수 있다.
- [0088] 메모리(1010)는 컴퓨터에서 판독 가능한 기록매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 여기서 ROM과 디스크 드라이브와 같은 비소멸성 대용량 기록장치는 메모리(1010)와는 구분되는 별도의 영구 저장 장치로서 컴퓨터 시스템(1000)에 포함될 수도 있다. 또한, 메모리(1010)에는 운영체제와 적어도 하나의 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 메모리(1010)와는 별도의 컴퓨터에서 판독 가능한 기록매체로부터 메모리(1010)로 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록매체를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록매체가 아닌 통신 인터페이스(1030)를 통해 메모리(1010)에 로딩될 수도 있다. 예를 들어, 소프트웨어 구성요소들은 네트워크(1060)를 통해 수신되는 파일들에 의해 설치되는 컴퓨터 프로그램에 기반하여 컴퓨터 시스템(1000)의 메모리(1010)에 로딩될 수 있다.
- [0089] 프로세서(1020)는 기본적인 산술, 로직 및 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(1010) 또는 통신 인터페이스(1030)에 의해 프로세서(1020)로 제공될 수 있다. 예를 들어 프로세서(1020)는 메모리(1010)와 같은 기록 장치에 저장된 프로그램 코드에 따라 수신되는 명령을 실행하도록 구성될 수 있다.
- [0090] 통신 인터페이스(1030)는 네트워크(1060)를 통해 컴퓨터 시스템(1000)이 다른 장치와 서로 통신하기 위한 기능을 제공할 수 있다. 일례로, 컴퓨터 시스템(1000)의 프로세서(1020)가 메모리(1010)와 같은 기록 장치에 저장된 프로그램 코드에 따라 생성한 요청이나 명령, 데이터, 파일 등이 통신 인터페이스(1030)의 제어에 따라 네트워크(1060)를 통해 다른 장치들로 전달될 수 있다. 역으로, 다른 장치로부터의 신호나 명령, 데이터, 파일 등이 네트워크(1060)를 거쳐 컴퓨터 시스템(1000)의 통신 인터페이스(1030)를 통해 컴퓨터 시스템(1000)으로 수신될 수 있다. 통신 인터페이스(1030)를 통해 수신된 신호나 명령, 데이터 등은 프로세서(1020)나 메모리(1010)로 전달될 수 있고, 파일 등은 컴퓨터 시스템(1000)이 더 포함할 수 있는 저장 매체(상술한 영구 저장 장치)로 저장될 수 있다.
- [0091] 통신 방식은 제한되지 않으며, 네트워크(1060)가 포함할 수 있는 통신망(일례로, 이동통신망, 유선 인터넷, 무선 인터넷, 방송망)을 활용하는 통신 방식뿐만 아니라 기기들간의 근거리 유선/무선 통신 역시 포함될 수 있다. 예를 들어, 네트워크(1060)는, PAN(personal area network), LAN(local area network), CAN(campus area network), MAN(metropolitan area network), WAN(wide area network), BBN(broadband network), 인터넷 등의 네트워크 중 하나 이상의 임의의 네트워크를 포함할 수 있다. 또한, 네트워크(1060)는 버스 네트워크, 스타 네트워크, 링 네트워크, 메쉬 네트워크, 스타-버스 네트워크, 트리 또는 계층적(hierarchical) 네트워크 등을 포함하는 네트워크 토폴로지 중 임의의 하나 이상을 포함할 수 있으나, 이에 제한되지 않는다.
- [0092] 입출력 인터페이스(1040)는 입출력 장치(1050)와의 인터페이스를 위한 수단일 수 있다. 예를 들어, 입력 장치는 마이크, 키보드, 카메라 또는 마우스 등의 장치를, 그리고 출력 장치는 디스플레이, 스피커와 같은 장치를 포함할 수 있다. 다른 예로 입출력 인터페이스(1040)는 터치스크린과 같이 입력과 출력을 위한 기능이 하나로 통합된 장치와의 인터페이스를 위한 수단일 수도 있다. 입출력 장치(1050)는 컴퓨터 시스템(1000)과 하나의 장치로 구성될 수도 있다.
- [0093] 이러한 도 10의 실시예는, 컴퓨터 시스템(1000)의 일례일 뿐이고, 컴퓨터 시스템(1000)은 도 10에 도시되지 않은 추가의 컴포넌트를 더 구비하거나, 2개 이상의 컴포넌트를 결합시키는 구성 또는 배치를 가질 수 있다. 컴퓨터 시스템(1000)에 포함 가능한 컴포넌트들은 하나 이상의 신호 처리 또는 어플리케이션에 특화된 집적 회로를 포함하는 하드웨어, 소프트웨어, 또는 하드웨어 및 소프트웨어 양자의 조합으로 구현될 수 있다.
- [0094] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 프로세서, 컨트롤

러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

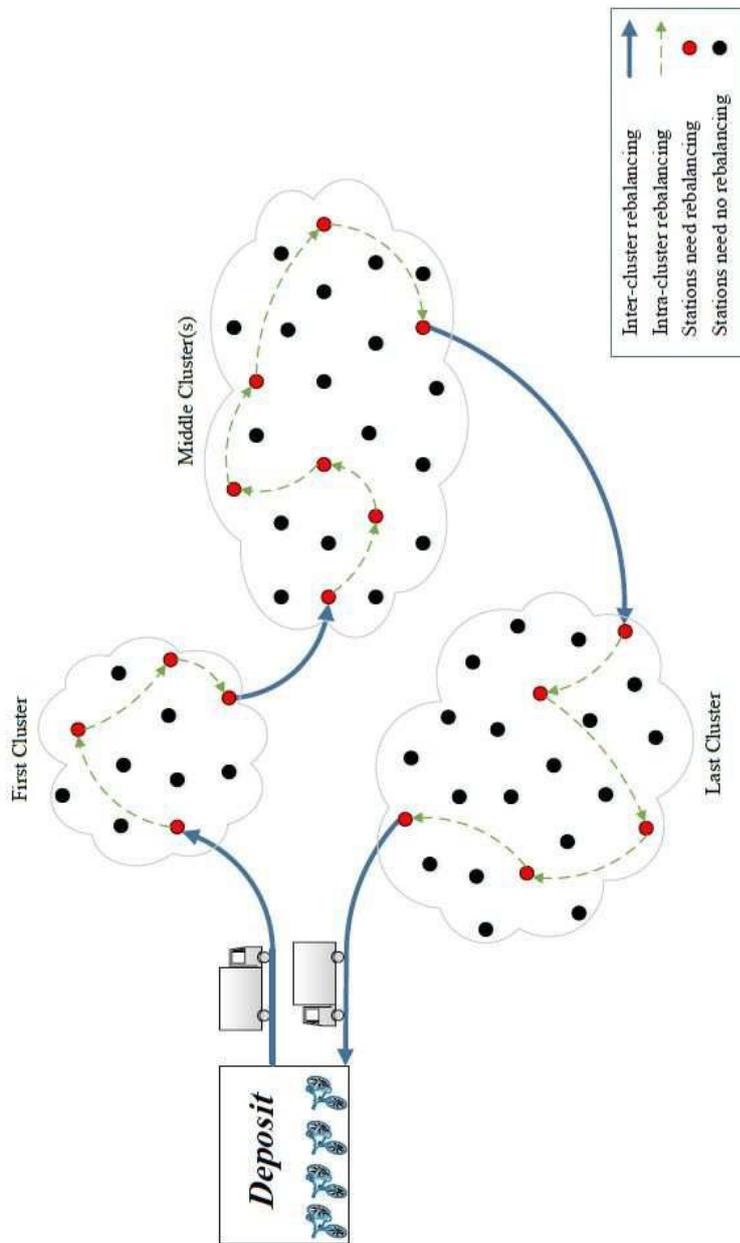
[0095] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0096] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 이때, 매체는 컴퓨터로 실행 가능한 프로그램을 계속 저장하거나, 실행 또는 다운로드를 위해 임시 저장하는 것일 수도 있다. 또한, 매체는 단일 또는 수 개의 하드웨어가 결합된 형태의 다양한 기록수단 또는 저장수단일 수 있는데, 어떤 컴퓨터 시스템에 직접 접속되는 매체에 한정되지 않고, 네트워크 상에 분산 존재하는 것일 수도 있다. 매체의 예시로는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM 및 DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical medium), 및 ROM, RAM, 플래시 메모리 등을 포함하여 프로그램 명령어가 저장되도록 구성된 것이 있을 수 있다. 또한, 다른 매체의 예시로, 어플리케이션을 유통하는 앱 스토어나 기타 다양한 소프트웨어를 공급 내지 유통하는 사이트, 서버 등에서 관리하는 기록매체 내지 저장매체도 들 수 있다.

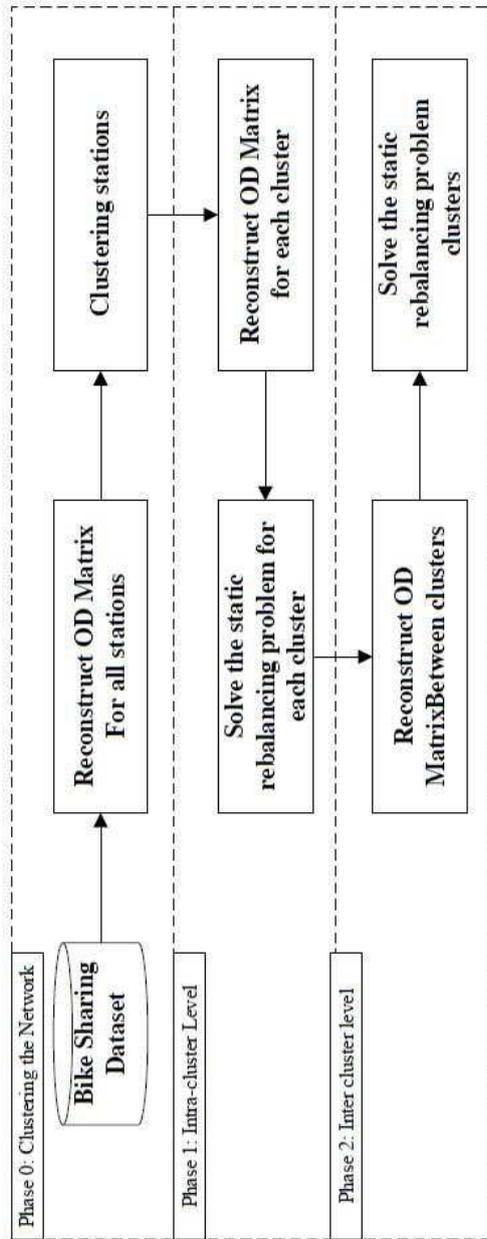
[0097] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0098] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면  
도면1



도면2



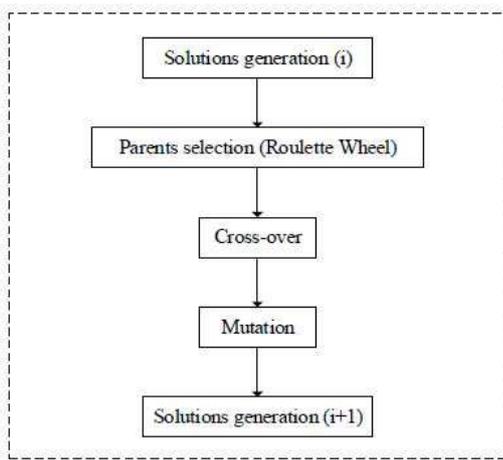
도면3

trip_id	start_time	end_time	tripduration	from_station_id	to_station_id
13550373	4/4/2017 23:26	4/4/2017 23:40	867	9	9
13549223	4/4/2017 19:11	4/4/2017 19:31	1182	4	4
13549015	4/4/2017 18:52	4/4/2017 19:14	1298	3	3
13548817	4/4/2017 18:34	4/4/2017 18:52	1071	6	3
13548299	4/4/2017 17:59	4/4/2017 18:24	1535	4	6
13548272	4/4/2017 17:58	4/4/2017 18:24	1599	4	6
13546231	4/4/2017 16:20	4/4/2017 17:18	3483	4	4
13546228	4/4/2017 16:20	4/4/2017 17:18	3499	4	4
13546225	4/4/2017 16:20	4/4/2017 17:18	3503	4	4
13545825	4/4/2017 15:46	4/4/2017 15:57	695	6	3
13545570	4/4/2017 15:17	4/4/2017 15:34	999	4	4
13545519	4/4/2017 15:10	4/4/2017 15:17	375	4	3
13545098	4/4/2017 14:08	4/4/2017 14:23	894	4	7
13544607	4/4/2017 12:40	4/4/2017 13:04	1435	3	3
13544602	4/4/2017 12:40	4/4/2017 13:04	1468	3	3
13543314	4/4/2017 9:13	4/4/2017 9:37	1437	6	4
13543312	4/4/2017 9:13	4/4/2017 9:37	1448	6	4

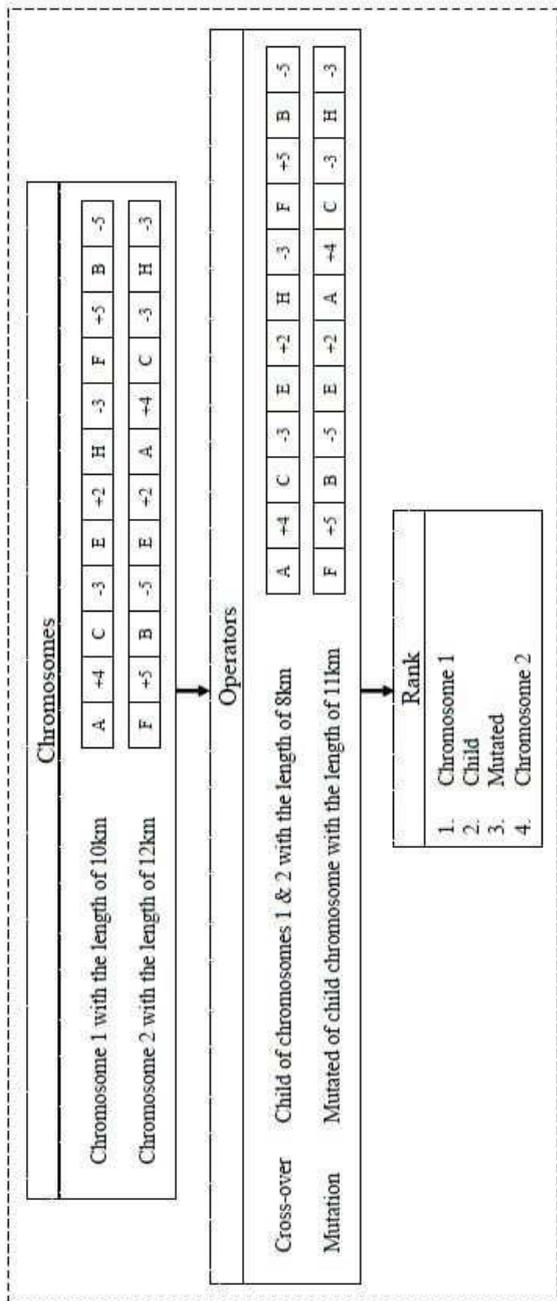


OD	3	4	6	7	9
3	0	0	0	0	0
4	1	5	2	1	0
6	2	2	0	0	0
7	0	0	0	0	0
9	0	0	0	0	1

도면4



도면5



## 도면6

---

**Algorithm 1:** Genetic algorithm framework for the static rebalancing problem

---

**Data:** OD matrix

**Result:** tours

```

1  $p \leftarrow$  population size;
2  $gen \leftarrow$  maximum number of generations;
3  $cr \leftarrow$  cross over rate;
4  $mu \leftarrow$  mutation rate ;
5  $P \leftarrow$  Initialize-Population ();
6 mutated  $\leftarrow$  Mutation ( $mut$ );
7 for  $i = 1 : gen$  do
8   rank the solutions;
9   parents  $\leftarrow$  select ( $cr \times p$ ) solutions using Roulette Wheel method;
10  children  $\leftarrow$  Cross-over (parents);
11  add children to the solutions;
12   $mut \leftarrow$  select ( $mu \times p$ ) solutions;
13  mutated  $\leftarrow$  Mutation ( $mut$ );
14  add mutated to the solutions;
15  rank the solutions;
16  eliminate ( $(cr + mu) \times p$ ) of the worst solutions (highest cost);
```

---

## 도면7

---

**Algorithm 2:** Initialize-Population function

---

```
1 tour ←  $\emptyset$ ;  
2 while tour does not meet the constraints do  
3   randomly choose stations;  
4   randomly choose numbers of bikes;  
5   add stations and number of bikes to the tour;  
6 return tour;
```

---

## 도면8

**Algorithm 3:** Cross-over function

```
1 while child tour does not meet the constraints do  
2   randomly cut parent tours;  
3   child ← attach cut parents;  
4 return child;
```

## 도면9

---

**Algorithm 4:** Mutation function

---

```
1 while mutated tour does not meet the constraints do
2   randomly choose number of stations in the selected tours;
3   mutated ← replace the chosen stations with random stations;
4 return mutated;
```

---

도면10

