(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0165577
(43) 공개일자 2024년11월25일

(51) 국제특허분류(Int. Cl.)
B60W 60/00 (2020.01) *B60W 30/06* (2006.01)
G06N 3/045 (2023.01) *G06N 3/08* (2023.01)
(52) CPC특허분류
B60W 60/0025 (2020.02)
B60W 30/06 (2024.01)
(21) 출원번호 10-2023-0062873
(22) 출원일자 2023년05월16일
심사청구일자 2023년05월16일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
장경희
서울특별시 마포구 백범로 205, 101동 1110호(신공덕동, 펜트라우스)
아라티
인천광역시 미추홀구 경인남길30번길 53, 301호(용현동)
(뒷면에 계속)
(74) 대리인
양성보

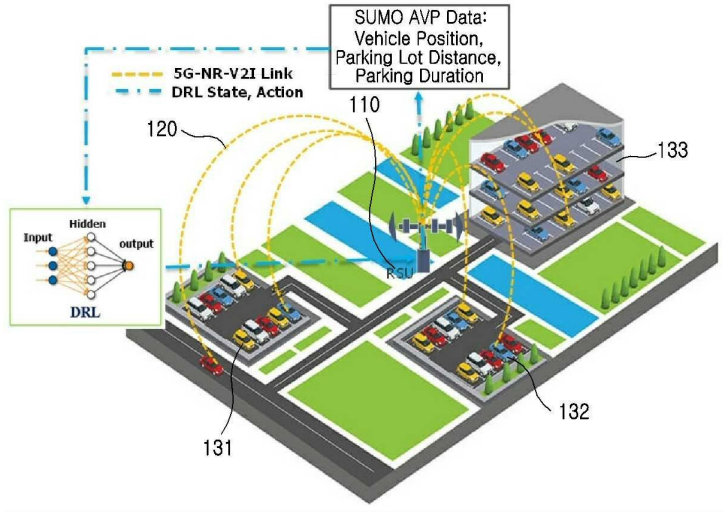
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 **V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템 및 그 동작 방법**

(57) 요 약

V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템 및 그 동작 방법이 제시된다. 본 발명에서 제안하는 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템은 V2I 통신을 이용하여 인공지능 기능이 탑재된 RSU로부터 V2I 링크 데이터를 수신하는 데이터 수집부, 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 시뮬레이션부 및 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 DDPG 에이전트를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06N 3/045 (2023.01)
G06N 3/08 (2023.01)
B60W 2556/45 (2020.02)

(72) 발명자
사지드
인천광역시 미추홀구 인하로134번길 24-35 (용현동)

박창호
중국, 충칭 난안 디스트릭트, 총웬 로드 2, 충칭 유니버시티 오브 포스츠 앤드 텔레커뮤니케이션스

이 발명을 지원한 국가연구개발사업

과제고유번호	1711175434
과제번호	2021K1A3A1A61003236
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	국제화기반조성사업
연구과제명	[Ezbaro] 5G-NR V2X 기술을 활용한 완전 자율주행 차량 교통 흐름 최적화 및 안전성 향상 연구
기 여 율	1/1
과제수행기관명	인하대학교
연구기간	2022.10.29 ~ 2023.10.28

명 세 서

청구범위

청구항 1

V2I 통신을 이용하여 인공지능 기증이 탑재된 RSU로부터 V2I 링크 데이터를 수신하는 데이터 수집부;

상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 시뮬레이션부; 및

결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 DDPG 에이전트

를 포함하는 AVP(Autonomous Valet Parking) 시스템.

청구항 2

제1항에 있어서,

상기 시뮬레이션부는,

RSU와의 V2I 통신을 통해 수집된 V2I 링크 데이터를 이용하여 차량 위치 및 거리 정보를 수집하고, 복수의 주차 구역 내 차량의 점유율, 위치, 거리 및 대기시간에 따라 주차 공간을 할당하기 위해 차량이 하차 위치에서 주차 공간으로 이동하여 해당 공간으로 이동하는 데 걸리는 시간인 평균 주차 시간, 운전자가 픽업 지점에서 차를 받기 위해 기다리는 시간인 평균 대기 시간, 주차 공간을 향해 주행하거나 출발하는 동안 교통 충돌로 인해 차량이 대기하는 시간인 평균 지연 시간을 포함하는 차량이 주차 과정에서 보내는 총 시간을 이용하는 마르코프 의사 결정 프로세스를 통해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는

AVP 시스템.

청구항 3

제1항에 있어서,

상기 DDPG 에이전트는,

현재 상태를 입력으로 사용하고 해당 작업을 생성하여 결정론적 정책을 학습하기 위한 액터(actor) 네트워크 및 상태와 작업을 사용하여 특정 상태-행동 쌍에 대한 기대 수익을 생성하여 상태 가치 함수를 학습하는 크리틱(critic) 네트워크를 통해 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는

AVP 시스템.

청구항 4

제3항에 있어서,

상기 DDPG 에이전트는,

액터 네트워크가 시뮬레이션 환경과 상호 작용하기 위한 행동을 예측하고, 탐색을 위한 노이즈를 추가하여 행동을 예측하며,

시뮬레이션 환경에 따른 보상을 생성하고 상기 예측된 행동에 따라 다음 상태로 이동한 후, 해당 현재 상태, 행동, 다음 상태 및 보상을 하나의 샘플로서 재생 버퍼에 저장하며,

상기 액터 네트워크의 미러 액터 네트워크(Mirror Actor Network)는 상기 버퍼에 저장된 각 샘플에 대해 현재 상태로부터 다음 행동을 예측하고,

상기 크리틱 네트워크의 미러 크리틱 네트워크(Mirror Critic Network)는 상기 버퍼에 저장된 각 샘플에 대해 상태-행동 쌍에 대한 가치를 예측하여 크리틱 네트워크학습에 사용함으로써 보상 값을 최대화하는 주차 구역을

할당하는

AVP 시스템.

청구항 5

데이터 수집부를 통해 V2I 통신을 이용하여 인공지능 기증이 탑재된 RSU로부터 V2I 링크 데이터를 수신하는 단계;

시뮬레이션부를 통해 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 단계; 및

DDPG 에이전트를 통해 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 단계

를 포함하는 AVP(Autonomous Valet Parking) 시스템의 동작 방법.

청구항 6

제5항에 있어서,

상기 시뮬레이션부를 통해 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 단계는,

RSU와의 V2I 통신을 통해 수집된 V2I 링크 데이터를 이용하여 차량 위치 및 거리 정보를 수집하고, 복수의 주차 구역 내 차량의 점유율, 위치, 거리 및 대기시간에 따라 주차 공간을 할당하기 위해 차량이 하차 위치에서 주차 공간으로 이동하여 해당 공간으로 이동하는 데 걸리는 시간인 평균 주차 시간, 운전자가 픽업 지점에서 차를 받기 위해 기다리는 시간인 평균 대기 시간, 주차 공간을 향해 주행하거나 출발하는 동안 교통 충돌로 인해 차량이 대기하는 시간인 평균 지연 시간을 포함하는 차량이 주차 과정에서 보내는 총 시간을 이용하는 마르코프 의사 결정 프로세스를 통해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는

AVP 시스템의 동작 방법.

청구항 7

제5항에 있어서,

상기 DDPG 에이전트를 통해 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 단계는,

현재 상태를 입력으로 사용하고 해당 작업을 생성하여 결정론적 정책을 학습하기 위한 액터(actor) 네트워크 및 상태와 작업을 사용하여 특정 상태-행동 쌍에 대한 기대 수익을 생성하여 상태 가치 함수를 학습하는 크리틱(critic) 네트워크를 통해 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는

AVP 시스템의 동작 방법.

청구항 8

제7항에 있어서,

상기 DDPG 에이전트를 통해 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 단계는,

액터 네트워크가 시뮬레이션 환경과 상호 작용하기 위한 행동을 예측하고, 탐색을 위한 노이즈를 추가하여 행동을 예측하며,

시뮬레이션 환경에 따른 보상을 생성하고 상기 예측된 행동에 따라 다음 상태로 이동한 후, 해당 현재 상태, 행동, 다음 상태 및 보상을 하나의 샘플로서 재생 버퍼에 저장하며,

상기 액터 네트워크의 미러 액터 네트워크(Mirror Actor Network)는 상기 버퍼에 저장된 각 샘플에 대해 현재 상태로부터 다음 행동을 예측하고,

상기 크리틱 네트워크의 미러 크리틱 네트워크(Mirror Critic Network)는 상기 버퍼에 저장된 각 샘플에 대해

상태-행동 쌍에 대한 가치를 예측하여 크리틱 네트워크학습에 사용함으로써 보상 값을 최대화하는 주차 구역을 할당하는

AVP 시스템의 동작 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템 및 그 동작 방법에 관한 것이다.

배 경 기 술

[0002] AVP(Autonomous Valet Parking) 시스템은 사람의 개입 없이도 차량이 지정된 장소에 스스로 주차할 수 있는 기술 솔루션이다. 이 시스템은 센서, 카메라 및 고급 알고리즘과 같은 다양한 기술을 사용하여 차량을 주차장으로 이동시키고 안전하고 효율적인 방식으로 주차한다. 종래기술에서는 결정론적인 차량 주차 작업을 해결하기 위한 휴리스틱 알고리즘을 개발하는 데 중점을 두었지만, 이러한 접근 방식은 동적 및 랜덤 주차 시나리오에서 실시간 주차 공간 할당에 적합하지 않다. 종래기술에서는 자동 주차를 위해 강화 학습(Reinforcement Learning; RL) 기반 종단 간 주차 알고리즘이 제안되어 경로 추적 오류를 피하면서 지속적인 학습과 최적의 스티어링 각도 결정이 가능하다. 이 알고리즘은 비전 및 차량 섀시 정보를 기반으로 한 주차 슬롯 추적 알고리즘을 사용한다. 이전의 연구는 결정론적인 차량 주차 작업을 해결하기 위한 휴리스틱 알고리즘을 개발하는 데 중점을 두었지만, 이러한 접근 방식은 동적 및 대규모 주차 시나리오에서 실시간 주차 공간 할당에 적합하지 않다.

[0003] AVP 시스템의 주차 공간 할당 문제에 대하여 심층 강화 학습(Deep Reinforcement Learning; DRL) 기반 접근 방식이 제안되었다. 이 방법은 간단한 위상 배치로 작은 주차장에 주차 공간을 할당하는 데 사용되었다. 자동차 생산 증가로 기존 주차 인프라 시설에 부담이 생겼고, 늘어나는 차량 수를 따라가지 못하고 있다. 이 문제를 해결하기 위해서는 멀티존 시나리오를 관리하고 증가하는 주차 공간에 대한 수요를 충족시킬 수 있는 완전 자율 주차 시스템을 위한 알고리즘을 개발하는 것이 중요하다. 머신 러닝 알고리즘과 지능형 인프라 시설 기술을 주차 구역과 도로변에 통합함으로써 이동성을 높이고 주차 검색에 걸리는 시간을 줄일 수 있어 전반적인 주차 효율을 향상시킬 수 있다. AVP 솔루션은 두 가지 유형으로 분류할 수 있다: 단거리 완전 자율 주차(Short-Range Autonomous Valet Parking; SAVP) 및 장거리 완전 자율 주차(Long-Range Autonomous Valet Parking; LAVP)이다. 사용자는 LAVP를 통해 차량을 내려주고 자율주행 차량이 기존 지도를 이용해 이용 가능한 주차장으로 이동할 수 있다. AVP 시스템에는 모션 계획, 경로 계획, 코스 계획, 기동 계획, 궤적 계획이 포함된다. 도심의 혼잡을 줄이고 여행 경험을 향상시키기 위해 학습 기반 및 DRL 기반 알고리즘을 장거리 완전 자율 주차에 사용할 수 있으며, 이는 경로 계획 및 서비스 시간 간격을 최적화할 수 있다.

[0004] 종래기술에서는 사전 훈련이 필요하지 않지만 의사 결정에 더 오래 걸리고 더 많은 전력을 소비할 수 있는 이중 계층 개미 군집 최적화(Double-Layer Ant Colony Optimization; DL-ACO) 알고리즘이 개발되었다. DQN 기반 알고리즘은 학습이 필요하지만 더 빠른 결정을 내릴 수 있어 중요한 시나리오에 적합하지만 DL-ACO 알고리즘은 동적 환경에 적합하지 않을 수 있다.

[0005] AVP 시스템은 자율 주행 차량(Autonomous Vehicle; AV)이 사용자를 특정 위치에 내려준 다음 사용 가능한 주차 지점을 찾기 위해 독립적으로 탐색할 수 있는 가능성을 제공한다. AVP 시스템은 효율적이고 편리한 주차 솔루션으로 인해 관심을 끌고 있다. DRL은 AVP 시스템을 개발하기 위한 유망한 머신 러닝 기술이다. AV는 첨단 기술을 보유하고 있으며, 첨단 센서, 카메라, AI 알고리즘을 탑재해 자율적으로 주행 조작을 수행하고 장애물을 피한다.

[0006] 종래기술에서, 지능형 스마트 주차는 운전자의 선호도와 도시 요인을 고려하여 혼합 정수 선형 프로그래밍(Mixed Integer Linear Programming; MILP)을 사용하여 운전자가 부담하는 비용을 최소화하는 I-Parker와 같은 주차 공간을 예약한다. 종래기술에서는 자기 신호 및 수신 신호 강도를 이용한 차량 검출을 위한 저비용 방법을 제안하고 있으며, 이는 정확하고 에너지 효율적이어서 배터리 구동 무선 차량 검출기에 적합하다. 이러한 기술은 심층 강화 학습에 기반한 신경망을 사용하여 빈 주차 공간을 찾아 주차할 수 있는 자율 주행 프로토타입 차량을 설계하고 구현한다. 이 접근 방식은 시뮬레이션 환경에서 심층 RL 알고리즘을 사용하여 두 개의 서로 다른 인공 신경망(Neural Networks; NN)을 훈련하는 것을 포함하며, 그런 다음 프로토타입 자동차의 컴퓨팅 플랫폼에 내장된다. 종래기술의 군중 감지 및 스마트 주차는 주차 감지기 역할을 하는 시민을 통해 실시간으로 접근 가능한 주차 정보를 수집하기 위해 지능형 주차에 사용되는 새로운 도시 서비스이다. S-Park는 인증을 위한 신뢰할

수 있는 기관(Trusted Authority; TA)과 함께 온보드 유닛(Onboard Unit; OBU) 및 RSU(Road-Side Unit)를 통해 실시간 주차장 정보를 제공하며, 이중 선형 페어링 기술을 제공한다. 종래기술의 스마트 주차 할당 및 예약 시스템은 비용 함수를 이용하여 현재 도로 상황 및 주차장 정보를 기반으로 최적의 주차 공간을 할당 및 예약한다. 주차 지리 정보 시스템, 운전자 요청 처리 센터, 주차 자원 관리 센터를 포함하며, MILP를 사용하여 사용자가 제공한 정보를 기반으로 최적의 주차 공간을 선택한다.

[0007] 또 다른 종래기술에서는 IoT 기술을 활용한 클라우드 기반 지능형 주차 시스템을 위한 알고리즘을 제안하여 주차 성공 확률을 높이고 사용자 대기 시간을 최소화했다. 또한, 무선 센서 네트워크에서 노드 배치를 위한 다목적 최적화 모델을 제시하고 재난 구조에서 스마트 주차 및 하이브리드 UAV/센서 네트워크와의 관련성을 제안한다. 또 다른 종래기술에서는 새로운 무선 이중 연결(NR-DC)과 패킷 복제를 사용하여 5G 네트워크의 성능을 향상시키기 위한 혁신적인 기술을 제시한다. 이러한 접근 방식은 차세대 셀룰러 네트워크에서 신뢰성, 처리량 및 대기 시간을 개선하도록 설계되었다.

선행기술문헌

비특허문헌

[0008] (비특허문헌 0001) J. Xie, Z. He, and Y. Zhu, "A DRL based cooperative approach for parking space allocation in an automated valet parking system," Appl. Intell., pp. 1-20, 2022.

(비특허문헌 0002) Z. Xu et al., "Experience-driven Networking: A Deep Reinforcement Learning based Approach," in IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, 2018, pp. 1871-1879.

발명의 내용

해결하려는 과제

[0009] 본 발명이 이루고자 하는 기술적 과제는 환경현황에 따라 차량에 주차공간을 할당하여 주차장의 효율성을 높이기 위해 지각, 행동 및 목표 요소를 포함하는 순차적 의사 결정 프로세스를 이용하여, 마르코프 의사 결정 프로세스(Markov Decision Process; MDP)를 사용하여 모델링하는 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템 및 그 동작 방법을 제공하는데 있다. 또한, 지능형 인프라 시설과 통합된 대규모 멀티존 동적 환경에 적합한 딥 러닝 알고리즘을 제안하여 멀티존 AVP 시나리오에 RSU(Road-Side Unit)이 존재한다고 가정하고 AI와 V2X 통신 기술의 통합으로 차량 주차 방식을 혁신하여 더 안전하고 효율적이며 편리한 프로세스를 제공하고자 한다.

과제의 해결 수단

[0010] 일 측면에 있어서, 본 발명에서 제안하는 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템은 V2I 통신을 이용하여 인공지능 기능이 탑재된 RSU로부터 V2I 링크 데이터를 수신하는 데이터 수집부, 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 시뮬레이션부 및 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 DDPG 에이전트를 포함한다.

[0011] 상기 시뮬레이션부는 RSU와의 V2I 통신을 통해 수집된 V2I 링크 데이터를 이용하여 차량 위치 및 거리 정보를 수집하고, 복수의 주차 구역 내 차량의 점유율, 위치, 거리 및 대기시간에 따라 주차 공간을 할당하기 위해 차량이 하차 위치에서 주차 공간으로 이동하여 해당 공간으로 이동하는 데 걸리는 시간인 평균 주차 시간, 운전자가 픽업 지점에서 차를 받기 위해 기다리는 시간인 평균 대기 시간, 주차 공간을 향해 주행하거나 출발하는 동안 교통 충돌로 인해 차량이 대기하는 시간인 평균 지연 시간을 포함하는 차량이 주차 과정에서 보내는 총 시간을 이용하는 마르코프 의사 결정 프로세스를 통해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션한다.

[0012] 상기 DDPG 에이전트는 현재 상태를 입력으로 사용하고 해당 작업을 생성하여 결정론적 정책을 학습하기 위한 액터(actor) 네트워크 및 상태와 작업을 사용하여 특정 상태-행동 쌍에 대한 기대 수익을 생성하여 상태 가치 함수를 학습하는 크리틱(critic) 네트워크를 통해 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습한다.

[0013] 상기 DDPG 에이전트는 액터 네트워크가 시뮬레이션 환경과 상호 작용하기 위한 행동을 예측하고, 탐색을 위한

노이즈를 추가하여 행동을 예측하며, 시뮬레이션 환경에 따른 보상을 생성하고 상기 예측된 행동에 따라 다음 상태로 이동한 후, 해당 현재 상태, 행동, 다음 상태 및 보상을 하나의 샘플로서 재생 버퍼에 저장하며, 상기 액터 네트워크의 미러 액터 네트워크(Mirror Actor Network)는 상기 버퍼에 저장된 각 샘플에 대해 현재 상태로부터 다음 행동을 예측하고, 상기 크리틱 네트워크의 미러 크리틱 네트워크(Mirror Critic Network)는 상기 버퍼에 저장된 각 샘플에 대해 상태-행동 쌍에 대한 가치를 예측하여 크리틱 네트워크학습에 사용함으로써 보상값을 최대화하는 주차 구역을 할당한다.

[0014] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템의 동작 방법은 데이터 수집부를 통해 V2I 통신을 이용하여 인공지능 기능이 탑재된 RSU로부터 V2I 링크 데이터를 수신하는 단계, 시뮬레이션부를 통해 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 단계 및 DDPG 에이전트를 통해 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 단계를 포함한다.

발명의 효과

[0015] 본 발명의 실시예들에 따르면 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템 및 그 동작 방법을 통해 환경 현황에 따라 차량에 주차공간을 할당하여 주차장의 효율성을 높이기 위해 지각, 행동 및 목표 요소를 포함하는 순차적 의사 결정 프로세스를 이용하여, 마르코프 의사 결정 프로세스(Markov Decision Process; MDP)를 사용하여 모델링할 수 있다. 또한, 지능형 인프라 시설과 통합된 대규모 멀티존 동적 환경에 적합한 딥 러닝 알고리즘을 제안하여 멀티존 AVP 시나리오에 RSU(Road-Side Unit)이 존재한다고 가정하고 AI와 V2X 통신 기술의 통합으로 차량 주차 방식을 혁신하여 프로세스를 더 안전하고 효율적이며 편리하게 할 수 있다.

도면의 간단한 설명

[0016] 도 1은 본 발명의 일 실시예에 따른 5G-NR-V2I 통신을 활용한 DRL 기반 멀티존 AVP 시스템 모델을 나타내는 도면이다.

도 2는 본 발명의 일 실시예에 따른 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템의 구성을 나타내는 도면이다.

도 3은 본 발명의 일 실시예에 따른 다중 영역 AVP 운영 관리를 위한 DDPG 알고리즘의 아키텍처를 나타내는 도면이다.

도 4는 본 발명의 일 실시예에 따른 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템의 동작 방법을 설명하기 위한 흐름도이다.

도 5는 본 발명의 일 실시예에 따른 DDPG 알고리즘을 나타내는 도면이다.

도 6은 본 발명의 일 실시예에 따른 DRL 방법의 평균 보상 및 대기 시간을 나타내는 도면이다.

도 7은 본 발명의 일 실시예에 따른 DDPG 알고리즘의 학습 과정에서 모든 주차 구역의 평균 대기 시간 및 보상을 나타내는 도면이다.

도 8은 본 발명의 일 실시예에 따른 V2I를 적용한 경우와 적용하지 않은 경우의 세 가지 성능을 각 주차구역별 점유율과 거리에 따라 비교한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0017] AVP(Autonomous Valet Parking)는 사람의 개입 없이 차량을 직접 주차할 수 있는 기술이다. 첨단 감지 및 통신 시스템을 사용하여 적절한 주차 공간을 찾고 차량을 안전하고 효율적으로 주차한다. 다양한 인공지능(Artificial Intelligence; AI) 기반의 방법들이 교통 혼잡을 줄이고, 안전을 개선하며, 운전자들의 편의와 편안함을 향상시키는 등 AVP의 장점들을 보여주고 있는 가운데, 실제 환경에서 멀티존 주차 구역을 효과적으로 처리할 수 있는 AVP 시스템을 개발하고 평가하는 문제는 아직 해결되지 않았다.

[0018] 본 발명에서는, 예를 들어 반경 1km 내에 위치한 세 개의 주차 구역에 대한 AVP 시스템을 제시하고 기존 도구와 심층 결정론적 정책 그레디언트(Deep Deterministic Policy Gradient; DDPG) 알고리즘을 조합하여 문제를 해결한다. DDPG 알고리즘은 AVP 시스템을 제어하여 차량을 자율적으로 탐색하고 주차할 수 있도록 주차 공간을 효율

적으로 할당한다. 제안하는 시스템은 차량과 시스템 간의 정보 교환을 위한 5G-NR V2I(Vehicle-to-Infrastructure) 통신의 활용을 가정한다. 또한 통신 대기 시간이 시스템 성능에 미치는 영향도 연구한다. 시뮬레이션 결과, 제안하는 시스템은 세 개의 주차 구역에 차량을 효율적이고 안전하게 주차하여 기존의 심층 강화 학습 방법에 비해 대기 시간을 7% 단축하는 것으로 나타났다. 제안하는 시스템은 종래기술보다 눈에 띄는 발전을 나타내며 미래를 위한 스마트 도시의 비전을 발전시키는 데 도움이 된다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0020] 도 1은 본 발명의 일 실시예에 따른 5G-NR-V2I 통신을 활용한 DRL 기반 멀티존 AVP 시스템 모델을 나타내는 도면이다.

[0021] 주차 관리를 위해 제안된 솔루션은 분석 모델의 복잡성과 시뮬레이션 확장의 어려움으로 인해 소규모로만 평가되었다. 그러나 대기 시간을 줄이고 도시 성장을 지속 가능하게 최적화하기 위해 스마트 인프라 시설과 통합할 수 있는 AI 기반 멀티존 AVP 시스템에 대한 필요성이 증가하고 있다. 이 문제를 해결하기 위해 V2I 통신에서 통신 지연의 영향을 고려하면서 멀티존 주차 최적화를 가능하게 하는 DDPG 기반 알고리즘을 제안한다.

[0022] 제안된 시스템은 네트워크 구축과 관련된 문제를 해결하기 위해 도시 지역에 대규모 배치를 목적으로 한다. PyPML(Python Parking Monitoring Library)과 이동성 시뮬레이션 프레임워크는 스마트 도시 및 스마트 이동성 문제에 대한 투자를 통해 교통 혼잡 및 지속 가능한 성장 문제를 해결하기 위해 도입되었다.

[0023] 본 발명의 실시예에 따르면, 멀티존 AVP 시스템에 대한 실시간 영향을 관찰하기 위해 SUMO 시뮬레이터를 통해 PyMPL과 함께 모델이 없는 DDPG 체계를 적용한다. 이 접근 방식은 기존 DQN(Deep Q-Network) 알고리즘보다 안정적인 것으로 확인되었다.

[0024] 도 1은 V2I 통신을 가정한 멀티존 AVP 애플리케이션 시나리오를 보여준다. 지능형 멀티존 AVP 애플리케이션 시스템의 경우 RSU(110)에 AI 기능이 탑재되어야 한다. 본 발명에서는 AVP 시스템이 V2I 통신(120)을 이용하여 RSU로부터 데이터를 수신하고 있다고 가정한다. RSU는 SUMO 시뮬레이션 내에서 모델링되어야 한다. V2I 링크 데이터는 AVP 시나리오에서 각 주차 구역(131, 132, 133)의 위치, 거리 및 점유 상태에 대한 정보를 제공할 수 있다. 이 시나리오에서 멀티존 AVP 시스템은 도로변에 설치된 RSU(110)와 통신하여 사용 가능한 주차 공간, 점유 상태 및 기타 관련 데이터에 대한 실시간 정보를 AVP 시스템에 제공한다. 도 1은 각각 700m, 850m, 950m에서 서로 떨어져 위치한 세 개의 AVP 구역(131, 132, 133)을 보여준다.

[0025] 주차 할당 시스템은 차량에 적절한 주차 공간을 할당하기 위해 세 주차 구역의 점유 상태 및 거리를 고려한다. 본 발명에서는 AVP 시나리오를 만들기 위해 SUMO 시뮬레이터를 사용했다. SUMO는 지능형 교통 시스템을 테스트하고 평가하기 위해 교통 상황을 시뮬레이션하는 오픈 소스 소프트웨어이다. 교차로, 원형 교차로, 고속도로 및 기타 복잡한 교통 상황을 모델링한다.

[0026] 이러한 AVP 시스템에서 DDPG 에이전트는 DDPG 알고리즘을 사용하여 멀티존 AVP 시스템이 시간 경과에 따라 주차 행동을 최적화하도록 학습할 수 있다. AVP 시스템은 과거의 경험을 바탕으로 학습하여 성능을 점진적으로 개선함으로써 보다 효율적이고 효과적인 주차 시스템을 구축할 수 있다. 본 발명에서는 TraCI 인터페이스를 사용하여 SUMO의 AVP 시나리오를 모델링할 수 있는 PyPML이라는 기존 주차 시뮬레이션 소프트웨어를 활용하여 지능형 AVP 시스템을 모델링하기 위해 DDPG 알고리즘을 설계했다. 멀티존 AVP 시스템은 AVP 프로세스의 대기 시간 단축을 목표로 설계되었다.

[0027] 본 발명의 실시예에 따르면, DDPG 기반 학습 알고리즘을 설계하고 시뮬레이션한다. 상태 공간은 이용 가능한 주차 공간의 수, 각 구역의 거리 및 점유, 각 구역의 평균 대기 시간, 차량의 현재 위치와 주차 구역 사이의 거리와 같은 SUMO 환경에서 DDPG 에이전트에게 제공된 정보를 기반으로 정의할 수 있다. 작업 공간은 주차 구역 조건(예를 들어, 거리, 점유율)을 기준으로 정의할 수 있으며, 운전자는 주차 기간과 검색을 먼저 제공해야 한다. AVP 시스템은 차량이 주차 인프라 시설과 통신하여 주차 요청을 생성하고 주차 정보를 수신하는 시스템을 의미하는 V2I 통신을 활용한다. 이 시스템에서 차량은 인프라 시설로 주차 요청을 전송하여 원하는 주차 위치, 기간, 주차 거리, 주차장별 점유율 등의 정보를 얻는다. 그런 다음 인프라 시설이 요청을 평가하고 사용 가능한 주차 공간을 할당한다. 그러면 주차 정보가 다시 차량으로 전송되고, 차량은 할당된 주차 공간으로 이동하여 주차할 수 있다. V2I 통신은 차량과 주차 인프라 시설이 실시간으로 정보를 교환하여 원활하고 효율적인 주차 환경을 보장하기 때문에 멀티존 AVP 시스템의 필수 요소이다.

[0029] 도 2는 본 발명의 일 실시예에 따른 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템의 구성을 나타내는 도면이다.

[0030] 본 실시예에 따른 멀티존 AVP 시스템(200)은 프로세서(210), 버스(220), 네트워크 인터페이스(230), 메모리(240) 및 데이터베이스(250)를 포함할 수 있다. 메모리(240)는 운영체제(241) 및 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 루틴(242)을 포함할 수 있다. 프로세서(210)는 데이터 수집부(211), 시뮬레이션부(212) 및 DDPG 에이전트(213)를 포함할 수 있다. 다른 실시예들에서 멀티존 AVP 시스템(200)은 도 2의 구성요소들보다 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 멀티존 AVP 시스템(200)은 디스플레이나 트랜시버(transceiver)와 같은 다른 구성요소들을 포함할 수도 있다.

[0031] 메모리(240)는 컴퓨터에서 판독 가능한 기록 매체로서, RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 또한, 메모리(240)에는 운영체제(241)와 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 루틴(242)을 위한 프로그램 코드가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 드라이브 메커니즘(drive mechanism, 미도시)을 이용하여 메모리(240)와는 별도의 컴퓨터에서 판독 가능한 기록 매체로부터 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록 매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록 매체(미도시)를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록 매체가 아닌 네트워크 인터페이스(230)를 통해 메모리(240)에 로딩될 수도 있다.

[0032] 버스(220)는 멀티존 AVP 시스템(200)의 구성요소들간의 통신 및 데이터 전송을 가능하게 할 수 있다. 버스(220)는 고속 시리얼 버스(high-speed serial bus), 병렬 버스(parallel bus), SAN(Storage Area Network) 및/또는 다른 적절한 통신 기술을 이용하여 구성될 수 있다.

[0033] 네트워크 인터페이스(230)는 멀티존 AVP 시스템(200)을 컴퓨터 네트워크에 연결하기 위한 컴퓨터 하드웨어 구성요소일 수 있다. 네트워크 인터페이스(230)는 멀티존 AVP 시스템(200)을 무선 또는 유선 커넥션을 통해 컴퓨터 네트워크에 연결시킬 수 있다.

[0034] 데이터베이스(250)는 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP를 위해 필요한 모든 정보를 저장 및 유지하는 역할을 할 수 있다. 도 2에서는 멀티존 AVP 시스템(200)의 내부에 데이터베이스(250)를 구축하여 포함하는 것으로 도시하고 있으나, 이에 한정되는 것은 아니며 시스템 구현 방식이나 환경 등에 따라 생략될 수 있고 혹은 전체 또는 일부의 데이터베이스가 별개의 다른 시스템 상에 구축된 외부 데이터베이스로서 존재하는 것 또한 가능하다.

[0035] 프로세서(210)는 기본적인 산술, 로직 및 멀티존 AVP 시스템(200)의 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(240) 또는 네트워크 인터페이스(230)에 의해, 그리고 버스(220)를 통해 프로세서(210)로 제공될 수 있다. 프로세서(210)는 데이터 수집부(211), 시뮬레이션부(212) 및 DDPG 에이전트(213)를 위한 프로그램 코드를 실행하도록 구성될 수 있다. 이러한 프로그램 코드는 메모리(240)와 같은 기록 장치에 저장될 수 있다.

[0036] 데이터 수집부(211), 시뮬레이션부(212) 및 DDPG 에이전트(213)는 도 4의 단계들(410~430)을 수행하기 위해 구성될 수 있다.

[0037] 멀티존 AVP 시스템(200)은 데이터 수집부(211), 시뮬레이션부(212) 및 DDPG 에이전트(213)를 포함할 수 있다.

[0038] 데이터 수집부(211)는 V2I 통신을 이용하여 인공지능 기능이 탑재된 RSU로부터 V2I 링크 데이터를 수신한다.

[0039] 시뮬레이션부(212)는 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션한다.

[0040] 시뮬레이션부(212)는 RSU와의 V2I 통신을 통해 수집된 V2I 링크 데이터를 이용하여 차량 위치 및 거리 정보를 수집하고, 복수의 주차 구역 내 차량의 점유율, 위치, 거리 및 대기시간에 따라 주차 공간을 할당하기 위해 차량이 하차 위치에서 주차 공간으로 이동하여 해당 공간으로 이동하는 데 걸리는 시간인 평균 주차 시간, 운전자가 픽업 지점에서 차를 받기 위해 기다리는 시간인 평균 대기 시간, 주차 공간을 향해 주행하거나 출발하는 동안 교통 충돌로 인해 차량이 대기하는 시간인 평균 지연 시간을 포함하는 차량이 주차 과정에서 보내는 총 시간

을 이용하는 마르코프 의사 결정 프로세스를 통해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션할 수 있다.

[0041] DDPG 에이전트(213)는 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습한다.

[0042] DDPG 에이전트(213)는 현재 상태를 입력으로 사용하고 해당 작업을 생성하여 결정론적 정책을 학습하기 위한 액터(actor) 네트워크 및 상태와 작업을 사용하여 특정 상태-행동 쌍에 대한 기대 수익을 생성하여 상태 가치 함수를 학습하는 크리틱(critic) 네트워크를 통해 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습한다.

[0043] DDPG 에이전트(213)는 액터 네트워크가 시뮬레이션 환경과 상호 작용하기 위한 행동을 예측하고, 탐색을 위한 노이즈를 추가하여 행동을 예측한다. 이후, 시뮬레이션 환경에 따른 보상을 생성하고 상기 예측된 행동에 따라 다음 상태로 이동한 후, 해당 현재 상태, 행동, 다음 상태 및 보상을 하나의 샘플로서 재생 버퍼에 저장한다.

[0044] 상기 액터 네트워크의 미러 액터 네트워크(Mirror Actor Network)는 상기 버퍼에 저장된 각 샘플에 대해 현재 상태로부터 다음 행동을 예측하고, 상기 크리틱 네트워크의 미러 크리틱 네트워크(Mirror Critic Network)는 상기 버퍼에 저장된 각 샘플에 대해 상태-행동 쌍에 대한 가치를 예측하여 크리틱 네트워크학습에 사용함으로써 보상 값을 최대화하는 주차 구역을 할당할 수 있다.

[0046] 제안하는 AVP 시스템을 평가하기 위해, 저차원 관찰을 사용하여 모든 작업에 대한 경쟁 정책을 학습할 수 있는 모델이 없는 접근 방식 DDPG를 선택하는데, 이는 작업 공간이 연속적인 연속 제어 작업에 일반적으로 사용되는 강화 학습 알고리즘이다. 이는 심층 신경망을 사용하여 정책과 가치 함수를 나타내는 모델이 없는 off-policy 알고리즘이다. 그것은 간단한 Actor-Critic 아키텍처와 학습 알고리즘만을 필요로 한다. DPG 알고리즘은 Q-러닝에 필요한 최적화 단계를 필요로 하지 않는 결정론적 정책을 사용하기 때문에 연속 작업 공간에 적합하다. 따라서 제한되지 않은 큰 함수 근사값과 사소한 작업 공간에 대해 계산적으로 효율적이고 실용적이다.

[0047] Actor-Critic 알고리즘의 확장인 DDPG는 actor가 결정론적 정책을 학습하고, critic가 상태 가치 함수를 학습하는 것을 포함한다. 다른 Actor-Critic 방법과 비교하여 DDPG는 재생 버퍼를 통합하여 경험 튜플을 저장한 다음 Actor-Critic 네트워크를 학습하는 데 사용된다. 본 발명의 실시예에 따른 알고리즘은 Actor-Critic 네트워크의 두 가지 주요 구성 요소로 구성된다. actor 네트워크는 현재 상태를 입력으로 사용하고 해당 작업을 생성하는 반면, critic 네트워크는 상태와 작업을 활용하여 특정 상태-행동 쌍에 대한 기대 수익을 생성한다. 두 네트워크 모두 시간적 차이(Temporal Difference; TD) 오류를 기반으로 동일한 손실 함수를 사용하여 학습된다. TD 오차는 현재 상태-행동 쌍의 예측 가치와 보상 및 다음 상태에서 얻은 실제 가치 사이의 차이이다. 손실 함수는 예측 가치와 실제 가치 사이의 평균 제곱 오차이다. DDPG는 Actor-Critic 네트워크 모두에 대해 미러 네트워크를 사용하여 TD 오류의 분산을 감소시킨다. 미러 네트워크는 소프트 업데이트 규칙을 사용하여 가중치가 천천히 업데이트되는 원래 네트워크의 복사본이다. 이는 학습 프로세스를 안정화하고 네트워크가 진동하는 것을 방지하는 데 도움이 된다. DDPG는 환경으로부터 피드백을 받는 데 있어 부분적인 관찰 가능성과 지연 시간을 관리할 수 있기 때문에 통신 대기 시간으로 구성된 멀티존 AVP 시스템에 적합한 알고리즘이다. 완전 자율 주차 시스템은 몇 가지 요인으로 인해 통신 지연이 발생할 수 있다. 이러한 대기 시간으로 인해 자율 주행 차량이 환경으로부터 지연된 피드백을 수신하여 차량을 정확하게 제어하기 어려울 수 있다. DDPG는 통신 대기 시간과 부분적인 관찰 가능성에 강력한 정책을 학습하여 이 문제를 해결할 수 있다. 제안하는 알고리즘은 재생 버퍼를 사용하여 과거의 경험을 저장하므로 지연된 피드백과 부분적인 관찰을 통해 학습할 수 있다.

[0048] 통신 지연이 있는 완전 자율 주차 시스템에서 DDPG를 사용하는 방법의 예시를 설명한다. 자율 주행 차량이 이용 가능한 주차 공간을 찾기 위해 주차장을 탐색하고 있다고 가정한다. 차량의 센서가 환경을 감지하여 중앙 컨트롤러로 정보를 전송하여 처리한다. 그러나 통신 지연으로 인해 중앙 컨트롤러는 몇 초의 지연과 함께 정보를 수신한다. 차량은 DDPG를 사용하여 이러한 대기 시간과 부분적인 관찰 가능성을 관리하는 정책을 학습할 수 있다. 이 알고리즘은 재생 버퍼에 저장된 과거 경험을 통해 학습할 수 있으며, 이를 통해 통신 대기 시간 및 부분 관찰에 적응할 수 있다. 예를 들어, 이 알고리즘은 중앙 컨트롤러로부터 피드백을 아직 받지 못하더라도 잠재적인 주차 지점을 감지할 때 차량 속도를 낮추는 방법을 배울 수 있다. 이렇게 하면 통신 대기 시간이 있더라도 차량이 안전하게 작동할 수 있다. 전반적으로 DDPG는 완전 자율 주차 시스템에서 통신 대기 시간 및 부분 관찰 가능성 문제를 관리할 수 있는 강력한 알고리즘으로, 향후 자율 차량 기술에 적합한 접근 방식이다.

[0049] 현재 AVP 시스템은 단일 주차장으로 제한되며, 주차장이 꽉 찬 경우에 대한 솔루션이 없다. 종래기술에서는 차

량 위치 및 거리 정보를 수집하는 방법을 구체적으로 설명하지 않았으며, 단일 주차장 시나리오에서 주차 할당을 최적화하기 위해 DRL만 사용했다. 다시 말해, 대기 시간과 통신 대기 시간이 주차장이 여러 개인 AVP 시스템에 미치는 영향은 철저히 조사되지 않았다. 대부분의 AVP 시스템은 그리디(greedy) 접근 방식을 기반으로 설계되며 단일 주차장 면적만 고려한다. DRL의 상태 공간에서 대기 시간이 고려되었지만, 대기 시간과 통신 대기 시간이 주차장이 여러 개인 AVP 시스템에 미치는 영향은 아직 연구되지 않았다. 이러한 측면을 검토하여 현재 AVP 시스템의 한계를 이해하고 여러 주차장이 포함된 실제 시나리오에서 주차 할당을 위한 보다 효율적이고 효과적인 솔루션을 개발하는 것이 중요하다.

[0050] 기존 AVP 시스템의 한계를 해결하기 위해 본 발명의 실시예에서는 서로 다른 거리에 위치한 세 개의 개별 주차 구역을 만드는 것을 포함한다. 이 AVP 시스템은 차량에 주차 공간을 할당할 때 모든 주차 구역의 점유율과 거리를 고려한다. 단일 RSU와의 V2I 통신을 통해 SUMO 생성 데이터를 이용하여 차량 위치 및 거리 정보를 수집하고 있으며, DDL 알고리즘, 특히 DDPG 알고리즘은 3개 주차구역 내 차량의 점유, 위치, 거리 및 대기시간을 고려하여 주차 공간을 효율적으로 할당한다. AVP 시스템에서 평균 주차 시간(Average Parking Time; APT)은 차량이 하차 위치에서 주차 공간으로 이동하여 해당 공간으로 이동하는 데 걸리는 시간이다. 평균 대기 시간(Average Waiting Time; AWT)은 운전자가 픽업 지점에서 차를 받기 위해 기다리는 시간이고, 평균 지연 시간(Average Delay Time; ADT)은 주차 공간을 향해 주행하거나 출발하는 동안 교통 충돌로 인해 차량이 대기하는 시간이다. 제안하는 AVP 시스템은 차량이 주차 과정에서 보내는 총 시간을 고려하며, 평균 대기 시간은 DDPG 알고리즘의 성능을 평가하기 위한 성능 척도로 사용된다. 본 발명에서는 주차 공간 할당 문제를 마르코프 의사 결정 과정으로 설계했다. 에이전트의 작업에는 수요에 따라 주차 공간을 할당하는 작업이 포함되며, 환경 상태에는 주차 공간 가용성, 차량 위치 및 상태 및 기타 관련 정보가 포함된다. 정책 $\pi(a_t|s_t)$은 주차 공간을 효율적으로 할당하기 위해 즉시 보상과 미래 보상을 포함하여 예상 할인 보상을 최대화하는 것을 목표로 한다. 목표는 n대의 차량의 지속적인 도착, 주차 및 출발과 관련된 일시적인 작업에 대한 주차 공간 할당으로 인해 발생하는 주차 혼잡을 해결하는 것이다. 본 발명의 실시예에 따른 할인된 예상 보상은 다음과 같다:

[0051] $$R_t = \sum_{\tau}^{N+t-1} \gamma^{\tau-t} r_t \quad (1)$$

[0052] N은 에피소드 기간에 해당하는 반면 $Q^{\pi}(s, a)$는 주어진 정책 π과 관련된 예상 수익률을 나타내는 상태-행동 가치 함수를 나타낸다.

[0053] $$Q^{\pi}(s,a) = E[R_t|s_t = s, a_t = a, \pi] \quad (2)$$

[0054] MDP의 기본 목표는 다음과 같이 상태-행동 가치 함수를 최대화하는 최적의 정책 π^*를 식별하는 것이다:

[0055] $$Q^{\pi^*}(s, a) = max_{\pi} Q^{\pi}(s,a) \quad (3)$$

[0056] 도 3을 참조하여, 제안된 멀티존 AVP 운영 관리 아키텍처에 대한 상태 공간(S), 작업 공간(A) 및 보상 함수(R)를 설명한다.

[0058] 도 3은 본 발명의 일 실시예에 따른 다중 영역 AVP 운영 관리를 위한 DDPG 알고리즘의 아키텍처를 나타내는 도면이다.

[0059] 본 발명의 일 실시예에 따른 다중 영역 AVP 운영 관리를 위한 DDPG 알고리즘에 있어서, actor 네트워크(310)를 학습시키기 위해, 정책 함수의 그레디언트는 파라미터 π_{μ}에 대해 계산된다. 그런 다음 이 그레디언트를 사용하여 정책을 개선하는 방식으로 actor 네트워크 파라미터를 조정한다. 그러나 작업 공간이 충분하고 셀 수 없는 연속 작업 공간에서는 정확한 그레디언트를 계산하는 것이 불가능하다. 이를 해결하기 위해 결정론적 정책 그레디언트(Deterministic Policy Gradient; DPG) 알고리즘을 기반으로 한 근사 기법을 채택한다. DPG 알고리즘은 critic 네트워크(320)의 그레디언트를 활용하여 정책 네트워크의 그레디언트를 추정하고, 식(4)로 공식화된다:

[0060] $$\nabla_{\theta^{\mu}} J \approx E[\nabla_a Q_{\theta}(s, \mu(s))|_{s=s_i} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s=s_i} \quad (4)$$

[0061] 여기서, $\mu(s|\theta^{\mu})$는 주어진 상태 s의 행동을 출력하는 actor-네트워크이고, $Q_{\theta}(s, \mu(s))$은 특정 상태-행동

조합과 연결된 예상 미래 보상을 추정하는 데 사용된다. 이 근사치는 현재 상태 s_i에서 평가된 파라미터에 대한 actor 네트워크의 그레디언트를 곱하여 행동에 대한 critic 네트워크의 그레디언트를 사용한다. critic 네트워크는 상태와 행동을 가치에 매핑하는 행동-가치 함수 $Q^{\pi}(s, a)$를 근사화하는 데 사용된다. DDPG에서, critic 네트워크는 재생 버퍼(360)의 각 상태-행동 쌍(s, a)에 대한 추정된 상태-행동 가치 $Q(s,a)$와 예상되는 상태-행동 가치 y_i 사이의 평균 제곱 오차(Mean Squared Error; MSE)를 최소화하도록 학습되어 있으며, 식(5)의 손실 함수로 공식화 된다:

[0062] $$Loss = 1/N \sum_i (y_i - Q(s,a|\theta^Q))^2 \qquad (5)$$

[0063] 여기서 N은 재생 버퍼(360)에 있는 샘플의 수이고, θ^Q는 critic 네트워크의 파라미터이다. 기대 상태-행동 가치 y_i는 예상 미래 보상을 즉시 보상 r_i의 합으로 표현하는 벨만 방정식을 사용하여 계산되며 할인된 예상 미래 보상은 다음 상태 s_{i+1}와 대상 actor-네트워크에 의해 계산된 다음 행동 a'에 주어졌다:

[0064] $$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \qquad (6)$$

[0065] $Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$은 목표 행동-가치 함수이며, γ은 동결 파라미터 $\theta^{Q'}$와 $\theta^{\mu'}$를 갖는 critic 네트워크의 복사본인 할인 계수이다. 대상 actor-네트워크 μ'은 다음 상태 s_{i+1}를 기준으로 다음 행동 a'를 계산하는 데 사용된다.

[0066] 본 발명의 실시예에 따른 재생 버퍼(360)는 상태, 행동, 보상, 다음 상태 및 에피소드 완료 여부로 구성된 과거 경험을 튜플로 저장하는 데이터 세트이다. 제안하는 DDPG 알고리즘은 신경망 훈련 목적으로 배치를 무작위로 샘플링할 수 있는 버퍼에 경험을 저장하여 훈련 프로세스의 안정성을 높이고 샘플 상관 관계를 완화한다. 이 접근 방식은 DQN 알고리즘에서 사용되는 경험 재생 기술과 유사하며, 개별 작업 공간이 있는 작업에 일반적으로 사용된다. 두 알고리즘 모두에서 재생 버퍼(360)는 이전 경험을 저장하는 데 사용되며, 신경망을 훈련시키기 위해 랜덤 배치로 샘플링할 수 있다. 이는 연속적인 샘플 간의 상관 관계를 끊는 데 도움이 되며, 결과적으로 더 안정적이고 효율적인 학습 과정을 제공한다. 이중 네트워크 구조의 대상 네트워크는 일반적으로 두 개의 별도 신경망, 즉 대상 actor-네트워크 $\mu'(s|\theta^{\mu'})$과 대상 critic 네트워크 $Q'(s,a|\theta^{Q'})$로 표시된다. actor 네트워크는 현재 상태를 기반으로 환경(330)의 작업을 선택하는 반면, critic 네트워크는 주어진 상태-행동 쌍에 대한 Q-값을 추정한다. 대상 네트워크의 가중치를 업데이트하기 위해 사용되는 특정 업데이트 공식은 다음과 같다:

[0067] $$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{Q'} \qquad (7)$$

[0068] 도 3을 참조하면, 본 발명의 실시예에 따른 DDPG라는 Actor-Critic 네트워크(310, 320)를 통해 DRL을 활용하는 AVP 시스템의 관리 아키텍처를 도시한다.

[0069] 본 발명의 실시예에서는 차량이 세 가지 주차 구역 중 하나에서 빈 주차 공간을 확보하기 위해 대기하는 시간을 기준으로 보상을 제공하는 것을 예시로서 설명한다. 이 정보는 V2I 통신에서 얻은 것으로, V2I 통신은 SUMO 환경에 존재하는 것으로 가정된다. 현재 상태 S_t 는, SUMO 환경(330)을 관찰한 다음 actor-네트워크(310)가 환경(330)과 상호 작용하기 위한 행동 a_t를 예측한다. actor 네트워크는 탐색을 위한 노이즈를 추가하여 행동 a_t를 예측한다(340).

[0070] $$a_t = \mu(S) + \varphi \qquad (8)$$

[0071] 여기서 φ는 노이즈이며, OU(Ornstein-Uhlenbeck) 노이즈 프로세스는 구현에 사용된다. 게다가, 노이즈의 진폭

은 학습 초기의 탐색을 보장하기 위해 풀 스케일에서 0으로 오목해지고 있다.

[0072] 환경(330)은 보상 r_t를 생성(350)하고 행동 공간에 따라 다음 상태 S_{t+1}로 이동한다. 이후, 샘플 (S_t, a_t, S_{t+1}, r_t)이 재생 버퍼(360)에 저장된다. 재생 버퍼에서 샘플 배치인 N_{batch}가 랜덤으로 선택된다. 각 샘플은 (S_t, a_t, S_{t+1}, r_t)로 구성된다. 각 샘플에 대해 미러 액터 네트워크(Mirror Actor Network)는 상태 S_t로부터 다음 행동 a_{t+1}을 예측하는 데 사용된다. 각 샘플에 대해 미러 크리틱 네트워크(Mirror Critic Network)는 상태-행동 쌍 S_{t+1}, a_{t+1}, 즉 $Q'(S_{t+1}, a_{t+1})$에 대한 Q 가치를 예측하는 데 사용된다. 그런 다음 상태-행동 쌍 S_t, a_t에 대한 Q-가치를 계산할 수 있으며, y_t로 표시된다:

[0073] $$y_t = r_t + \gamma Q'(S_{t+1}, a_{t+1}) \qquad (9)$$

[0074] 그런 다음 (S_t, a_t, y_t)를 나중에 critic 네트워크(320)의 학습에 사용할 수 있다. RL 알고리즘은 상태 정보를 사용하여 주차 공간을 할당하고 주차장의 주차 활용을 최적화하는 결정을 내린다. 본 발명의 실시예에 따른 상태 공간 표현은 특정 주차장에 명시되어 있으며 세 개의 주차 구역에 걸친 주차 공간 할당 문제를 해결하기 위해 활용될 수 있다. 이 문제를 해결하기 위해, 여러 주차 구역에 대한 세부 사항을 포함하도록 표현을 확장했다. 이를 통해 각 개별 주차 구역과 관련된 거리, 점유 상태 및 시간 관련 정보를 고려할 수 있다. 이와 같이 수정된 상태 공간 표현은 다음과 같다:

[0075] $$S_{t_i} = [\, O_{k_i}, T_{k_i}, t_i \,] \qquad (10)$$

[0076] 점유 상태 변수 O_{k_i}는 세 개의 주차 구역 모두의 점유 상태를 연결한 것이다. O_k를 주차 구역 k에 대한 점유 상태 벡터라 하자. 여기서 k = 1, 2, 3은 주차 구역 ID를 나타낸다.

[0077] $O_{k_i} = O_{1_i}, O_{2_i}, O_{3_i}$는 주차 구역 k_i에 대한 점유 상태 벡터이다.

[0078] $T_{k_i} = T_{1_i}, T_{2_i}, T_{3_i}$는 주차 구역의 실제 조건(예를 들어, 사용 가능한 공간의 수, 사용 가능한 공간의 위치 등)을 나타내는 시간 t_i에서의 주차 시스템의 실제 상태이다.

[0079] t_i는 관측치와 상태가 기록되는 시간 단계를 나타낸다.

[0080] 제안하는 시스템에서, 0과 1 사이에서 t_i를 정규화할 때, 정규화 없이, t_i의 값이 훨씬 더 큰 값인 O_{t_i}와 T_i로 증가하고 있음을 시뮬레이션을 통해 확인했다. 이는 불안정과 단계 변화를 가져오며, 이러한 정규화를 적용한 후 Actor-Critic 네트워크 수렴의 높은 변동을 초래할 것이며, 제안하는 시스템은 더 안정된 성능을 가질 수 있다. 차량 Vi에 주차 공간을 할당할 때, 선택된 공간은 현재 상태 S_{t_i}에 의해 결정되는 행동 a_{t_i}로 표시된다. 따라서 행동 공간 A의 크기는 n으로 표시되는 사용 가능한 주차 공간의 수와 일치해야 한다.

[0081] $$a_{t_i} \in A = \{0, 1, 2, \ldots K_{K_i}, \ldots, n_{k_i}\} \qquad (11)$$

[0082] 행동 a_{t_i}는 0에서 n_{k_i}사이의 값을 취할 수 있다. 여기서 n_{k_i}는 세 개의 주차 구역을 합친 총 주차 가능 공간 수 $(n_{1_i}, n_{2_i}, n_{3_i})$이고 K_{K_i}는 세 개의 주차 구역 각각에 있는 차량에 주차 공간을 할당하는 것을 나타낸다. 특정 구역에 주차 공간이 할당되지 않은 경우 해당 지수가 0.0으로 설정되면 차량에 주차 공간이 할당되지 않았음을 나타낸다. 주차 공간 할당 문제에서 MDP는 보상 함수를 활용하여 주차 행동의 효과를 평가하고 에이전트가 최적의 할당 정책을 준수하는지 여부를 결정한다. 주요 목표는 모든 차량의 평균 대기 시간을 최소화하는 것이

다. r_t로 표현되는 시간 t_i에서의 보상은 그 때의 상태 S_{t_i}와 행동 a_{t_i}에 의해 결정된다.

$$min\frac{1}{N_{K_i}} \sum_{i=1}^{N_{K_i}} w_i \qquad (12)$$

[0083]

[0084] 여기서, N_{K_i}는 세 개의 주차 구역으로부터의 총 주차 미션 횟수이고, 각각 k = 1, 2, 3이다. w_i는 차량 Vi 소유자의 대기 시간이다.

[0085] 본 발명의 실시예에 따르면, 주차 구역 내의 교통 혼잡을 평가하기 위해 몇 가지 지표를 사용할 수 있다. 여기에는 차량이 구역을 출입하는 데 걸리는 시간, 운전자가 차량을 회수할 때까지 기다려야 하는 시간, 차량이 운송 중에 정지해 있는 시간 등이 포함된다. 세 개의 주차 구역이 있는 AVP 시스템에서, 보상을 정의하기 위해 N_{K_i} 주차 작업의 평균 성능을 계산한다. 이는 문제를 각 주차 구역에 걸쳐 N_{K_i}주차 공간을 할당하는 것과 관련된 주기적인 임무로 구성하는 것을 수반한다. 기간화를 구현함으로써 지연된 피드백의 특성을 유지하면서 적시에 할당 정책을 평가할 수 있다. 이러한 요인을 고려하고 활용하여 후속 보상 함수를 정의한다. 주차 할당 시스템을 평가하는 데 사용되는 기본 측정 기준은 N_{K_i} 차량이 주차 구역을 출입하는 데 소요되는 평균 시간이다. 시간 소모는 고속도로 입구에서 주차 공간까지, 그리고 주차 공간에서 고속도로 출구까지 운전하는 데 걸린 총 시간으로 구성된다. 이를 고려하여 다음과 같이 보상 함수 R_t를 고안한다.

$$R_t = r_{allocated} + [r_{episode}]_{N_{K_i}}$$

$$r_{allocated} = \begin{cases} \frac{d_{max} - D_{K_i}}{d_{max} - d_{min}} \end{cases} \qquad (13)$$

[0086]

[0087] 여기서 D_{K_i}는 K_i = 1, 2, 3에서의 경로 거리이며, 여기서 D_{K_i}는 차량 Vi에 할당된 각 주차 공간의 세 주차 구역 $(d_{1_i}, d_{2_i}, d_{3_i})$로부터의 경로 거리이다. d_{max} 및 d_{min}은 경로 및 $[r_{episode}]_{N_{K_i}}$를 따라 이동한 거리의 가장 높은 값과 가장 낮은 값을 나타낸다. DRL 기반 할당 방법이 n대의 차량에 대한 주차 공간 할당을 정기적으로 완료하면 결과 값은 0 이하가 된다. 보상은 제안된 방법대로 점유율이 낮은 짧은 거리를 이동하도록 차량을 안내해야 한다. 안전하고 유효한 주차를 보장하기 위해 보상 시스템은 이미 점유된 주차 공간을 점유할 수 있는 실행 불가능한 행동을 방지해야 한다. 그러나 거리만을 기준으로 주차 공간을 할당하여 단일 단계 보상을 R_s로 공식화했다.

$$R_s = \begin{cases} 0 - \alpha & a_{t_i} \in Emptyset_{t_i} \\ f_p - \alpha & a_{t_i} \notin Emptyset_{t_i} \end{cases}$$

$$f_p = a_n * A_{pu} * |a_{t_i} - Emptyset_{t_i}| \qquad (14)$$

[0088]

[0089] 주차 할당 정책에서 단일 단계 보상 R_s는 여러 요인의 함수로 정의된다. 특히, 요인 α는 잘못된 주차 할당에 적용되는 페널티를 결정한다. 세 개의 주차 구역 중 어느 곳에도 주차 공간이 없는 경우 α는 0으로 설정된다. 그렇지 않으면 α가 0.001로 설정된다. 또 다른 중요한 요인은 패널티의 크기를 제어하는 음의 스케일링 계수인 a_n이다. a_n값이 클수록 잘못된 할당에 대한 패널티가 커지며, 값이 작을수록 패널티가 가벼워진다. 이용 가능한 주차 공간의 수, A_{pu}또한 페널티의 강도를 결정하는 역할을 한다. 주차 공간이 많은 경우 정책에 따라 차량을 올바른 주차 구역에 할당하기가 쉬워지므로 패널티 요소가 더 높아진다. 반대로 이용 가능한 주차공간이 적을 때는 정책적으로 적합한 주차공간을 찾는 것이 더 까다로워 페널티 요소가 낮아진다. 마지막으로, 할당된 주차 공간과 원하는 주차 공간 사이의 거리도 패널티 요소에 영향을 미친다. 거리가 클수록 페널티 요소가 높아

원하는 주차 공간에 가까운 주차 공간을 배정하도록 정책을 유도한다. 이것은 결국 주차장을 더 효율적으로 사용하게 된다. 이러한 요인들이 상호 작용하여 잘못된 주차 할당에 적용되는 적절한 페널티를 결정한다. 파라미터를 적절히 조정함으로써 주차 할당 정책을 효율성과 효과에 최적화할 수 있다.

[0090] 본 발명의 실시예에 따른 행동 공간은 자율 주행 차량이 선택할 수 있는 사용 가능한 주차 공간의 모든 조합의 집합이다. 각 주차 공간은 이진 변수로 표시된다. 여기서 1은 공간이 선택되었음을 나타내고 0은 공간이 선택되지 않았음을 나타낸다. 주차 구역 1의 주차 공간에 대한 행동 공간은 A_1로 표시된다:

[0091] $$A_1 = \{ a_{11}, a_{12}, a_{13}, a_{14}.....a_{1n} \} \qquad (15)$$

[0092] 여기서 i번째 주차 공간이 선택되면 $a_{1i} = 1$이고, 선택되지 않은 경우 $a_{1i} = 0$이다. 마찬가지로 주차 구역 2와 3의 주차 공간에 대한 행동 공간을 A_2와 A_3으로 정의할 수 있다.

$$A_2 = \{ a_{21}, a_{22}, a_{23}, a_{24}.....a_{2n} \} \qquad (16)$$

[0093] $$A_3 = \{ a_{31}, a_{32}, a_{33}, a_{34}.....a_{3n} \} \qquad (17)$$

[0094] 각 주차 공간에 대한 점유율은 주차 공간이 위치한 주차 구역의 점유율을 사용하여 계산할 수 있다. 점유율 = 1 - (점유율 공간 / 총 공간)이다. 여기서 점유율 공간은 현재 점유하고 있는 주차 공간의 수이고, 총 공간은 구역 내의 총 주차 공간의 수이다. 예를 들어, 주차 공간이 주차 구역 1에 있고 총 100개의 주차 공간 중 사용 가능한 주차 공간이 10개인 경우 점유율은 1 - (10 / 100) = 0.9가 된다. 행동 공간의 각 주차 공간에 대해 각 주차 구역 i에 대한 거리 계수를 계산한다. 보상 정보에서 D_{K_i}는 K_i = 1, 2, 3에서 경로 거리이며, 이는 세 주차 구역으로부터의 경로 거리이다.

[0095] $$D_{K_i} = \rho * d_{k_i}^{h_e} + (1 - \rho)\, d_{k_i}^{h_x} \qquad (18)$$

[0096] 빈 주차 공간에 차량을 주차하기 위한 대기 시간을 고려하기 위해, ρ=0.1이다. 여기서 d_{k_i}는 고속도로와 차량 V_i에 할당된 주차공간인 각 주차 구역 k_i 사이와의 거리를 나타내며, h_e와 h_x는 진입 및 출구로부터 고속도로 거리를 나타낸다. 앞서 설명된 것처럼 상태 공간은 주차 지속 시간 분포를 기반으로 정의된다. T_{k_i}는 시간 t_i에서의 주차 시스템의 실제 상태이며, 이는 주차장의 실제 상태를 나타낸다. t_i는 각 차량 V_i의 추정 주차 기간 T_d 값으로 0과 1에서 정규화된다. 입력 데이터는 차량의 유입 및 유출에 기초한 SUMO를 사용하여 생성되므로 각 차량 T_i^{fac}의 주차 지속 시간이 계산될 수 있다. 예상 주차 시간 T_d는 30분 단위만 지원하며, 이 정규화된 값은 $Emptyset_{t_i}$의 계산에 영향을 미친다. 이를 해결하기 위해 T_i^{fac}을 다시 계산한다.

$$T_d = [\frac{T_i^{fac}}{1800}].1800 \qquad (19)$$

[0097] $$T_i^{fac} = (1 - \beta) T_i^{P_{t_i}^D} + \beta * W_T \qquad (20)$$

[0098] 여기서 W_T는 각 주차 구역으로부터 주차공간과 연관된 대기시간으로, D_{K_i}를 기준으로 각 주차 구역에 대해 계산되며, 이는 고속도로 입구와 출구까지의 거리이다. β의 기본값은 0.1이며, 조건에 따라 조정할 수 있다. 그런 다음 시간 t_i에서 $S_{t_i} = [O_{k_i}, T_{k_i}, t_i]$에 대해 모든 주차 지속 시간을 정렬한 다음, 거리 D_{K_i}를 만족하

도록 작업 공간 $A = \{0, 1, 2, \ldots K_{K_i}, \ldots, n_{k_i}\}$ 을 설정한다.

[0099] 본 발명의 실시예에 따른 통신 대기 시간을 고려하면, 보상은 각 주차 구역에서 빈 주차 공간 정보를 얻기 위한 대기 시간으로 관찰될 수 있다. 경험 중심 접근 방식은 DDPG를 능가하고 전체 유틸리티를 향상시키는 동시에 종단 간 지연 시간을 줄이는 통신 네트워크의 리소스를 할당한다. 보상 함수를 설계하기 위해 대기 시간에 추가된 거리, 점유 및 통신 대기 시간 요소를 고려해야 하며 식(21)에 공식화되어 있다:

[0100] $$R_s = -((1 - w) * dist_i + w * o_i) - d_i \qquad (21)$$

[0101] 여기서, $dist_i$ 는 차량 i와 RSU 사이의 거리이고, o_i 는 주차 구역 i의 점유이며, 0과 100 사이의 백분율로 측정되며, d_i 는 주차장 i에 대한 추정 통신 지연 시간이고, 초 단위로 측정되며, w는 보상 함수에서 점유 요소의 가중치를 결정하는 상수이다. 이전과 마찬가지로 보상 함수의 처음 두 항은 거리와 점유 요인을 계산한다. 세 번째 항은 보상에서 선택한 주차 구역에 대한 예상 통신 대기 시간을 뺀다. 에이전트가 대기 시간을 최소화하고 대기 시간이 길수록 대기 시간이 증가하기 때문에 통신 대기 시간이 차감된다. 거리, 탑승자 및 통신 지연 요인을 결합하여, 에이전트는 탑승자 및 통신 지연이 가장 낮은 가장 가까운 주차 구역을 선택하는 방법을 학습할 것이다. 이전과 같이 w = 0.5를 설정하고 다음과 같이 통신 대기 시간을 추정한다. 주차 구역 1의 경우 10초, 주차 구역 2의 경우 15초, 주차 구역 3의 경우 20초이다. 주차 구역 1을 선택하면 보상은 -((1 - 0.5) * 700 + 0.5 * 10) - 10 = -355가 된다. 주차 구역 2를 선택하면 보상은 -((1 - 0.5) * 850 + 0.5 * 50) - 15 = -490이다. 주차 구역 3을 선택하면 보상은 -((1 - 0.5) * 900 + 0.5 * 30) - 20 = -480이다. 이 예에서 에이전트는 주차 구역 1이 가장 낮은 결합 거리, 승객 및 통신 지연 시간 요소를 가지고 있기 때문에 주차 구역 1을 선택하는 방법을 재학습하여 최고의 보상을 받게 된다. 그러나 가중치 w의 선택과 통신 대기 시간의 추정은 AVP 시스템의 특정 요구 사항에 따라 다르며 필요에 따라 조정할 수 있다. 전송 대기 시간은 식(22)을 사용하여 계산되며 보상 함수의 대기 시간에 추가된다. 따라서 대기 시간과 전송 지연을 모두 고려하여 결과 보상 값을 최대화하기 위한 작업 공간을 선택해야 한다. 차량은 보상 가치가 가장 높은 주차 공간을 선택해야 하며, 이는 업데이트된 보상 함수를 사용하여 계산할 수 있다. 각 주차 구역의 신호 전송 지연 시간은 식(22)을 사용하여 계산한다:

[0102] $$d_i = T_{hop} * ceil\left(\frac{d}{100}\right) \qquad (22)$$

[0103] 여기서 d는 차량과 RSU 사이의 절대 거리이며, 각 홉의 길이가 100m임을 고려할 때 T_{hop}= 0.005초이다. HARQ(Hybrid Automatic Repeat Request) 재전송의 경우, 재전송 횟수는 차량과 RSU 사이의 거리에 따라 달라질 수 있다. 특히, 거리가 100m 이하인 경우, HARQ는 패킷을 한 번만 재전송하도록 구성될 수 있다. 거리가 100m에서 150m 사이인 경우, HARQ는 패킷을 두 번까지 재전송하도록 구성될 수 있다. 따라서, 전송 대기 시간은 HARQ 재전송 횟수에 간접적으로 영향을 미칠 수 있다. 지연 시간이 너무 크면 차량이 패킷을 올바르게 수신하지 못할 수 있다. 이 경우 전송 오류가 발생한다. 이 오류는 HARQ 메커니즘을 트리거하여 재전송을 시작한다. 이러한 방식으로, RSU와 차량 간의 신뢰할 수 있는 통신을 보장하기 위해서는 전송 대기 시간 및 HARQ 재전송이 필수적이다. 대기 시간은 초기 전송의 신뢰성에 영향을 미치는 반면, HARQ 재전송은 전송 오류를 수정하고 전반적인 통신 신뢰성을 향상시키는 메커니즘을 제공한다. AVP 시스템의 경우, 전송 대기 시간은 다음과 같은 명령에 소요되는 시간이다:

[0104] 주차 요청: 차량이 주차 인프라 시설에 주차 프로세스를 시작하도록 요청한다.

[0105] 주차 확인: 주차 인프라 시설은 주차 공간의 가용성을 확인하고 위치 및 기타 필요한 정보를 차량으로 전송한다.

[0106] 주차 지원: 주차 인프라 시설은 다양한 센서, 카메라 및 기타 기술을 사용하여 차량을 주차 공간으로 안내한다.

[0107] 주차 완료: 차량이 지정된 공간에 주차되면 주차 인프라 시설이 차량에 완료 신호를 보낸다.

[0108] 검색 요청: 차량이 주차 인프라 시설에 검색을 시작하도록 요청한다.

[0109] 검색 지원: 주차 인프라는 다양한 센서, 카메라 및 기타 기술을 사용하여 차량을 출구로 안내한다.

[0110] 그러나 무선 채널을 통한 전송은 노이즈, 간섭 및 감쇠로 인해 오류가 발생하기 쉽다. 이러한 오류로 인해 패킷이 잘못 수신되어 전송 오류가 발생할 수 있다. 신뢰할 수 있는 통신을 보장하기 위해, 5G-NR-V2X는 수신기가 손실되거나 잘못된 패킷의 재전송을 요청할 수 있는 HARQ를 사용한다. 필요한 HARQ 재전송 횟수는 무선 링크의 품질, 패킷 크기 및 송신기와 수신기 사이의 거리에 따라 달라진다. 앞서 설명한 바와 같이 주차 시스템의 경우, 차량과 RSU 사이의 거리에 따라 HARQ 재전송 횟수가 달라진다. 전송 대기 시간이 너무 크면 신호가 RSU에서 차량으로 더 오래 이동한다. 이는 전송 오류의 가능성을 증가시켜 HARQ 메커니즘이 재전송을 시작하도록 트리거할 수 있다. 이제 거리 때문에 RSU가 차량으로 전송한 패킷이 전송 중에 손실되거나 손상되었다고 가정한다. 이 경우, RSU는 HARQ 메커니즘을 통해 재전송을 요청한다. HARQ 재전송 횟수가 2로 설정된 경우, RSU는 패킷을 두 번 더 재전송한다. 즉, 이 패킷의 전송 대기 시간은 원래 지연의 3배인 0.03초가 된다. 따라서 차량 대기 시간이 예상보다 길어지므로 주차 시스템의 성능에 부정적인 영향을 미칠 수 있다.

[0112] 도 4는 본 발명의 일 실시예에 따른 V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템의 동작 방법을 설명하기 위한 흐름도이다.

[0113] V2I 통신을 이용한 딥러닝 기반 멀티존 AVP 시스템의 동작 방법은 데이터 수집부를 통해 V2I 통신을 이용하여 인공지능 기증이 탑재된 RSU로부터 V2I 링크 데이터를 수신하는 단계(410), 시뮬레이션부를 통해 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 단계(420) 및 DDPG 에이전트를 통해 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습하는 단계(430)를 포함할 수 있다.

[0114] 단계(410)에서, 데이터 수집부를 통해 V2I 통신을 이용하여 인공지능 기증이 탑재된 RSU로부터 V2I 링크 데이터를 수신한다.

[0115] 단계(420)에서, 시뮬레이션부를 통해 상기 수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션한다.

[0116] 시뮬레이션부는 RSU와의 V2I 통신을 통해 수집된 V2I 링크 데이터를 이용하여 차량 위치 및 거리 정보를 수집하고, 복수의 주차 구역 내 차량의 점유율, 위치, 거리 및 대기시간에 따라 주차 공간을 할당하기 위해 차량이 하차 위치에서 주차 공간으로 이동하여 해당 공간으로 이동하는 데 걸리는 시간인 평균 주차 시간, 운전자가 픽업 지점에서 차를 받기 위해 기다리는 시간인 평균 대기 시간, 주차 공간을 향해 주행하거나 출발하는 동안 교통 충돌로 인해 차량이 대기하는 시간인 평균 지연 시간을 포함하는 차량이 주차 과정에서 보내는 총 시간을 이용하는 마르코프 의사 결정 프로세스를 통해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션할 수 있다.

[0117] 단계(430)에서, DDPG 에이전트를 통해 결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습한다.

[0118] DDPG 에이전트는 현재 상태를 입력으로 사용하고 해당 작업을 생성하여 결정론적 정책을 학습하기 위한 액터(actor) 네트워크 및 상태와 작업을 사용하여 특정 상태-행동 쌍에 대한 기대 수익을 생성하여 상태 가치 함수를 학습하는 크리틱(critic) 네트워크를 통해 주차 구역을 할당하기 위한 주차 행동을 최적화하도록 학습한다.

[0119] DDPG 에이전트는 액터 네트워크가 시뮬레이션 환경과 상호 작용하기 위한 행동을 예측하고, 탐색을 위한 노이즈를 추가하여 행동을 예측한다. 이후, 시뮬레이션 환경에 따른 보상을 생성하고 상기 예측된 행동에 따라 다음 상태로 이동한 후, 해당 현재 상태, 행동, 다음 상태 및 보상을 하나의 샘플로서 재생 버퍼에 저장한다.

[0120] 상기 액터 네트워크의 미러 액터 네트워크(Mirror Actor Network)는 상기 버퍼에 저장된 각 샘플에 대해 현재 상태로부터 다음 행동을 예측하고, 상기 크리틱 네트워크의 미러 크리틱 네트워크(Mirror Critic Network)는 상기 버퍼에 저장된 각 샘플에 대해 상태-행동 쌍에 대한 가치를 예측하여 크리틱 네트워크학습에 사용함으로써 보상 값을 최대화하는 주차 구역을 할당할 수 있다.

[0122] 도 5는 본 발명의 일 실시예에 따른 DDPG 알고리즘을 나타내는 도면이다.

[0123] 본 발명의 실시예에 따른 강화 학습에서 개별 행동 공간에서 행동을 선택하려면 종종 각 행동을 평가해야 하며, 이는 행동 수가 많을 때 계산 비용이 많이 든다.

[0124] 본 발명의 실시예에 따른 DDPG 학습 알고리즘은 도 5에 제시된다. 이로 인해 행동 공간 크기 $|A|$에 따라 선형적으로 증가하는 실행 복잡성이 증가한다. 결과적으로 정책 개선을 위해 가치 함수에 의존하는 전통적인 가치 기반 방법은 행동의 수가 증가함에 따라 구현이 어려워진다. 이 문제를 해결하기 위해 행동 임베딩은 행동을 선택하는 일반적인 방법 중 하나이다. 즉, 값 함수에 따라 가장 높은 값을 가진 행동을 선택하는 정책을 사용하는 것이다. 이 정책은 그리디 정책으로 알려져 있으며 다음과 같이 기록될 수 있다:

[0125] $$\pi_Q(s) = \mathrm{argmax}_{a \in A} Q_{(s,a)} \qquad (23)$$

[0126] 강화 학습에서 가치 기반 접근법을 사용할 때 가치 함수는 정책결정에 직접적인 영향을 미친다. 정책의 결정은 예상되는 장기 수익의 추정치를 제공하는 가치 함수에 의해 알려진다. 따라서 정책은 이 추정치를 기준으로 작업을 선택한다. 목표 정책의 탐색을 촉진하기 위해 에이전트는 OU(Ornstein-Uhlenbeck) 프로세스에서 생성된 노이즈를 사용하는 대신 가우스 노이즈를 평균이 0인 상태로 통합하고 분산을 점진적으로 줄인다. OU 프로세스는 식(24)에 설명된 바와 같이, 시간적으로 상관관계가 있는 탐색을 생성하고 탐색 효율을 향상시키는 데 사용된다:

[0127] $$\widetilde{\mu(s)} = \mu(s) + \aleph(\mu(s)) \qquad (24)$$

[0128] 여기서 $\aleph$는 행동 $\mu(s)$와 관련된 부가적인 노이즈이다

[0129] $$\aleph(\mu) = \theta(\beta - \mu) + \sigma\varkappa \qquad (25)$$

[0130] 여기서 $\varkappa$는 0-평균 단위-분산 가우스 랜덤 변수이며, 실험에서 β는 0.5로 상대적으로 (-1보다는) 1에 더 가깝다, 이는 출구 근처의 주차 구역에 매핑되어 출구 근처의 주차 구역이 할당될 때 탐색을 축소하고 출구에서 멀리 떨어진 주차 구역이 할당될 때 탐색을 확장한다.

[0132] 도 6은 본 발명의 일 실시예에 따른 DRL 방법의 평균 보상 및 대기 시간을 나타내는 도면이다.

[0133] 도 6(a)는 학습 진행 중 DRL 방법의 평균 보상을 나타내는 그래프이고, 도 6(b)는 학습 진행 중 DRL 방법의 대기 시간을 나타내는 그래프이다.

[0134] 본 발명의 실시예에 따른 성능 평가하기 위해 DDPG, Greedy 및 랜덤 방식의 세 가지 방법 및 알고리즘을 고려한다. DDPG와 Greedy는 모두 DRL을 기반으로 V2I 통신을 활용하는 반면, 랜덤 방식은 V2I 통신을 고려하지 않는 non-DRL 방식이다. DDPG 할당 방식은 차량의 현재 위치에서 가장 가까운 주차 구역의 공간을 식별하여 주차 공간을 할당한다. Greedy 할당 시스템은 주차 공간을 효율적으로 할당하거나 거리를 고려하지 않고 사용 가능한 주차 공간을 찾아 차량에 할당한다. DDPG 방식이 평균 대기 시간을 초 단위로 줄이는 것을 보여 주며, 119에서 122 사이이다. 반면에, 그리디 방식의 평균 대기 시간은 125에서 130 사이입니다.

[0136] 도 7은 본 발명의 일 실시예에 따른 DDPG 알고리즘의 학습 과정에서 모든 주차 구역의 평균 대기 시간 및 보상을 나타내는 도면이다.

[0137] 본 발명의 실시예에 따라 시간당 2,000대의 차량이 다중 구역 주차 공간을 탐색하면서 평균 대기 시간을 계산하는 시뮬레이션을 수행한다. 평균 대기 시간 및 보상에 대한 DDPG 학습 결과, 주차 과정에서 3개의 AVP 구역에 걸쳐 대기 시간을 크게 줄이는 DRL 기반 할당 방법의 효과를 확인할 수 있다. 특히, 차량이 최단 주차 거리를 찾아야 하는 경우에 효과적이다.

[0139] 도 8은 본 발명의 일 실시예에 따른 V2I를 적용한 경우와 적용하지 않은 경우의 세 가지 성능을 각 주차구역별 점유율과 거리에 따라 비교한 도면이다.

[0140] 본 발명의 실시예에 따른 접근 방식의 효율성을 평가하기 위해 3개의 주차 구역에 대한 평균 대기 시간을 개별

적으로 비교하였다. 그 결과 본 발명에서 제안한 V2I를 이용한 DRL 방식은, DRL방식 및 V2I 통신을 적용하지 않은 랜덤 방식에 비해 평균 대기 시간을 크게 단축할 수 있음을 확인할 수 있다. V2I가 없는 Non-DRL(다시 말해, 랜덤 방식)과 V2I가 있는 DRL 방법(다시 말해, DDPG 및 Greedy) 사이의 평균 대기 시간이 134.36초에서 125.85초로 감소하는 것을 확인할 수 있다.

[0141] 도 8을 참조하면, 구역 1(Z1) = 출발점(고속도로)에서 거리 700m, 초기 점유율은 10%이고, 구역 2(Z2) = 시작점(고속도로)에서 거리 850m, 초기 점유율은 30%이며, 구역 3(Z3) = 출발점(고속도로)에서 거리 950m, 초기 점유율은 50%이다.

[0143] 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, 예를 들어, 프로세서, 콘트롤러, 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 애플리케이션을 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 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0144] 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, 가상 장치(virtual equipment), 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.

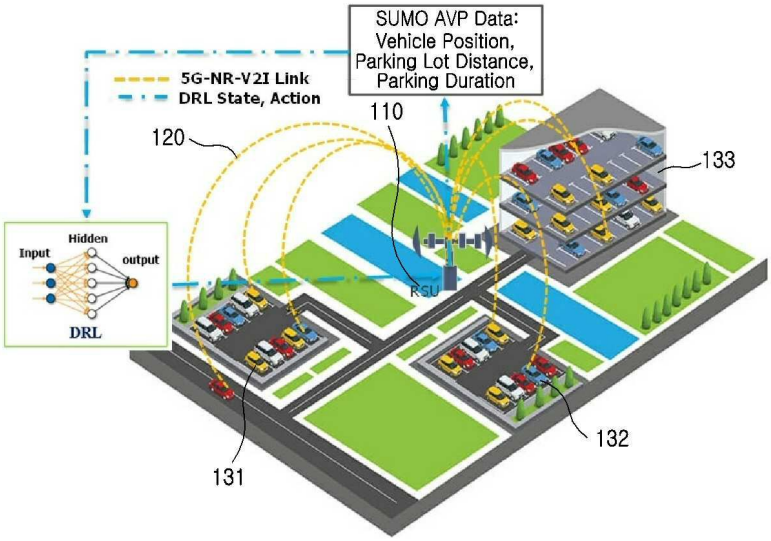
[0145] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0146] 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.

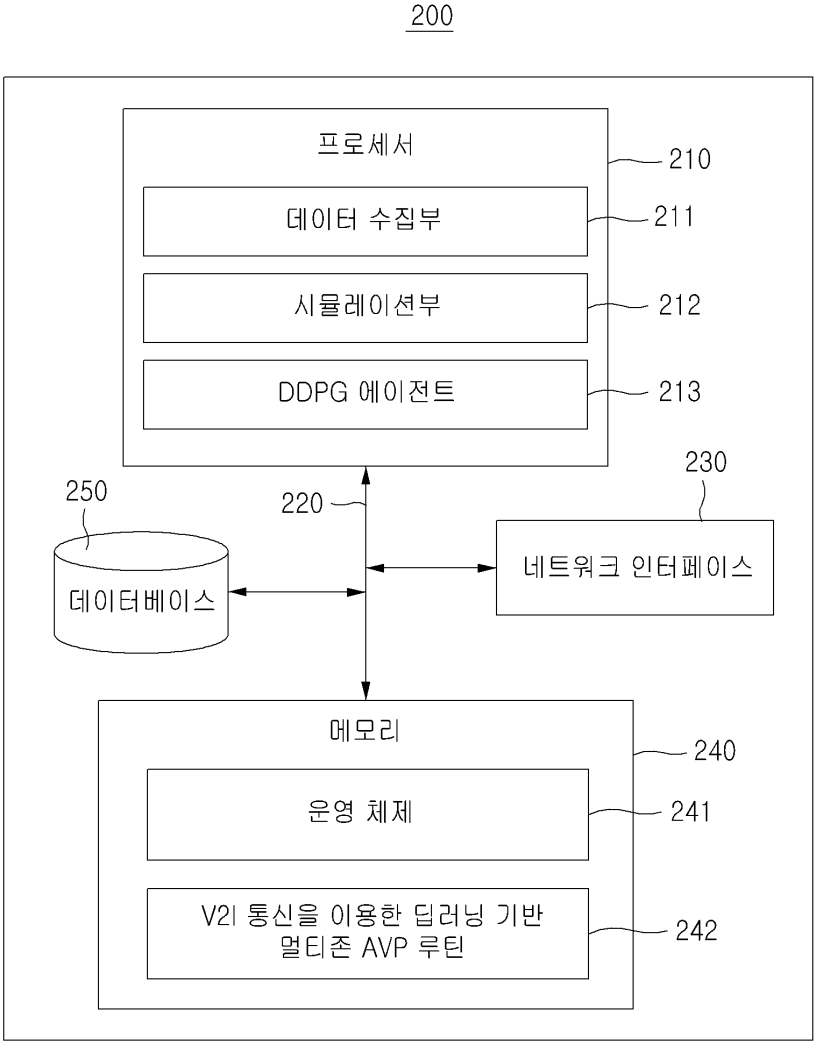
[0147] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

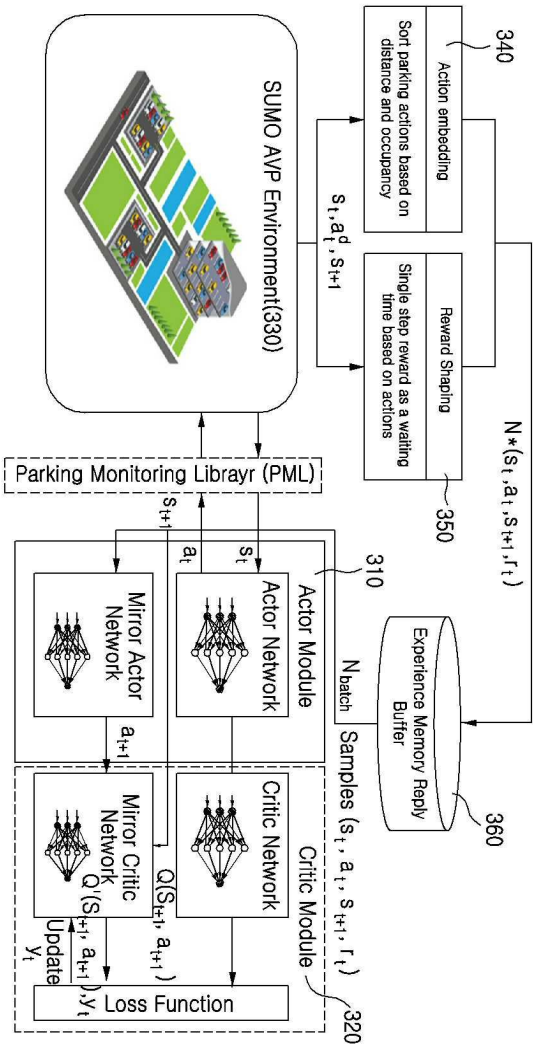
도면1


SUMO AVP Data:
Vehicle Position,
Parking Lot Distance,
Parking Duration
5G-NR-V2I Link
DRL State, Action
110
120
133
Hidden
Input
output
DRL
RSU
131
132

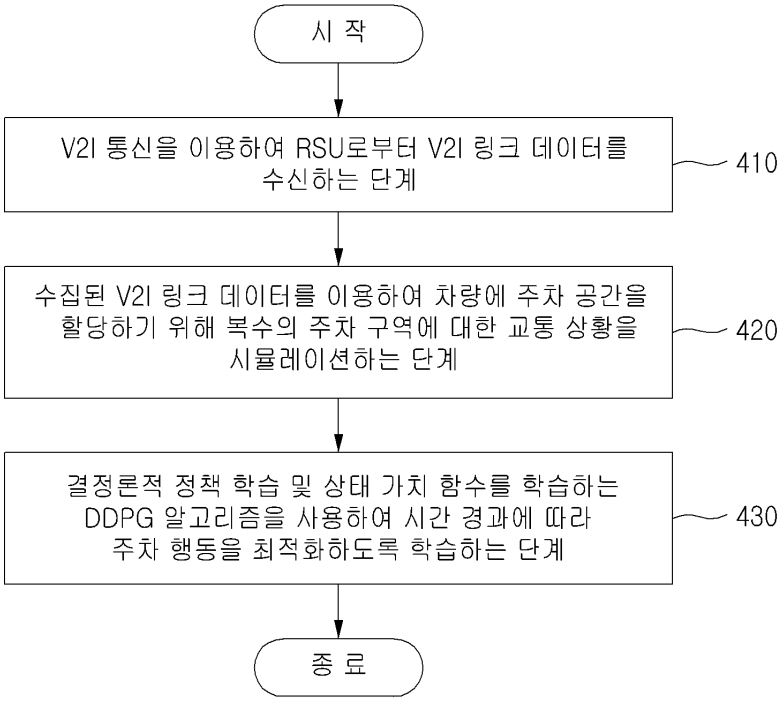
도면2


200
프로세서
210
데이터 수집부
211
시뮬레이션부
212
DDPG 에이전트
213
230
250
220
네트워크 인터페이스
데이터베이스
메모리
240
운영 체제
241
V2I 통신을 이용한 딥러닝 기반
멀티존 AVP 루틴
242

도면3


SUMO AVP Environment(330)
Parking Monitoring Librayr (PML)
Action embedding
Sort parking actions based on distance and occupancy
Reward Shaping
Single step reward as a waiting time based on actions
340
350
Actor Module
Mirror Actor Network
Actor Network
310
Critic Module
Mirror Critic Network
Critic Network
320
Loss Function
Update
Experience Memory Reply Buffer
360
Samples

도면4


시 작
V2I 통신을 이용하여 RSU로부터 V2I 링크 데이터를 수신하는 단계
410
수집된 V2I 링크 데이터를 이용하여 차량에 주차 공간을 할당하기 위해 복수의 주차 구역에 대한 교통 상황을 시뮬레이션하는 단계
420
결정론적 정책 학습 및 상태 가치 함수를 학습하는 DDPG 알고리즘을 사용하여 시간 경과에 따라 주차 행동을 최적화하도록 학습하는 단계
430
종 료

도면5

Algorithm 1 The training algorithm based on DDPG

Randomly initialize critic network Q (s, a|θ^{Q}) and actor network μ (s |θ^{Q}) with weight θ^{μ}

Initialize target network Q' and μ' with weights a_1, b_1

Initialize reply buffer R with max size S_M

for t = 1 to N do (N is the total number of parking missions)

Observe new state s_t

Make a prediction of the continuous action from the Actor Network

Calculate continuous action $\hat{a}_t = \mu(s_t|\ \theta_t^{\mu})$ from the Actor Network with addictive OU noise

Map from $\hat{a}_t$ to a discrete action a_t (after considering the occupancy status of the parking zone and its neighboring zones), and reroute the vehicle according to the parking zone corresponding to the discrete action a_t

Map from a_t back to a continuous action $\hat{a}_t$ (which is exact action being applied)

Store the transition (s_{t-1}, $\widetilde{a_{t-1}}$, R_{t-1}, s_t) in the replay buffer //(s_{t-1}, $\widetilde{a_{t-1}}$, R_{t-1}, s_{t-1} were already stored in a previous for-loop-iteration at t-1

Sample a minibatch of N_s transitions from the replay buffer followed by a shuffling

Update the Critic Network by minimizing the loss function

Update current policy network weight (i.e., Actor Network) by applying the policy gradient

Update the weights both target Actor-/Critic Networks

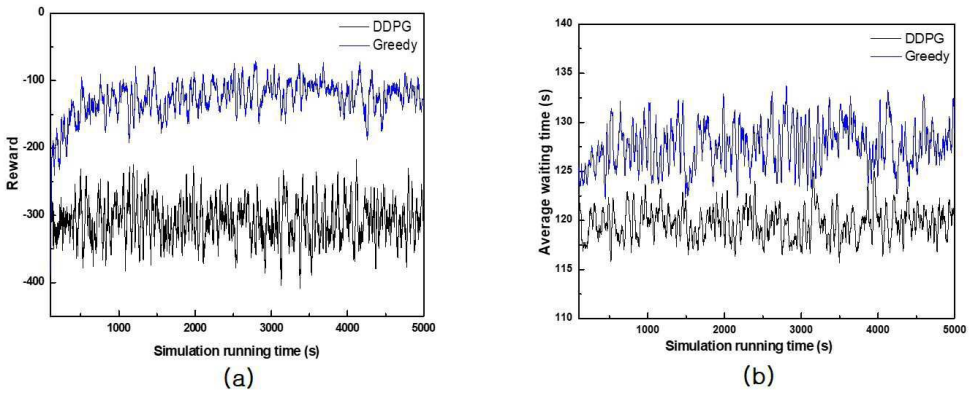
Calculate the reward R_t

Store s_t, $\hat{a}_t$, R_t temporarily until the next vehicle has arrived and a new state is observed.

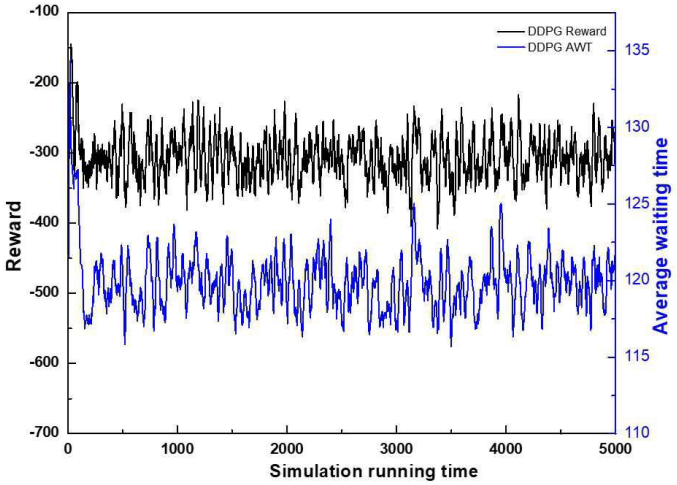
end for

end for

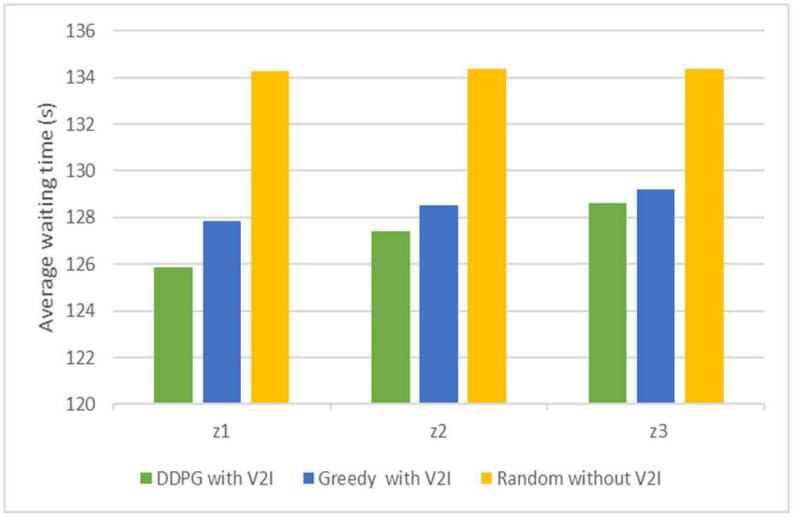
도면6


DDPG
Greedy
Reward
Simulation running time (s)
(a)
DDPG
Greedy
Average waiting time (s)
Simulation running time (s)
(b)

도면7


DDPG Reward
DDPG AWT
Reward
Average waiting time
Simulation running time

도면8


Average waiting time (s)
z1
z2
z3
DDPG with V2I
Greedy with V2I
Random without V2I