

FEC 환경에서 효율적 자원 배치를 위한 엣지 디바이스의 최적 이동패턴 추출

이연식^{1*} · 남광우¹ · 장민석¹

Extracting optimal moving patterns of edge devices for efficient resource placement in an FEC environment

YonSik Lee^{1*} · KwangWoo Nam¹ · MinSeok Jang¹

^{1*}Professor, School of Computer Info. & Comm., Kunsan National University, Kunsan, 54150 Korea

요약

동적으로 변하는 시간 가변적 네트워크 환경에서 엣지 디바이스의 최적 이동패턴은 FEC 환경에서 응용 서비스 사용자에 근접한 에지 클라우드 서버에 컴퓨팅 리소스를 분배하거나 새로운 에지 서버(기지국)를 배치하는데 적용함으로써, 클라우드 컴퓨팅의 단점인 지연시간 문제 완화를 위한 효율적 계산 오프로딩이 가능한 환경 구축에 활용이 가능하다. 본 논문은 임의의 시간제약 및 이동규칙 등이 적용되는 시공간 환경에서 응용 서비스를 요구하는 다수의 엣지 디바이스(이동객체)들의 이동경로를 빈발도 기반으로 분석하여 최적 이동패턴을 추출하는 알고리즘을 제안한다. 제안한 OPE_freq 알고리즘을 A* 및 Dijkstra 알고리즘들과 비교 실험을 통하여, 제안 알고리즘이 상대적으로 빠른 연산시간과 적은 메모리를 사용하고 보다 정확한 최적경로를 추출함을 알 수 있다. 또한 A* 알고리즘과의 비교 결과를 통하여 가중치를 빈발도와 동시에 적용함으로써 경로 추출의 정확도를 향상시킬 수 있음을 도출하였다.

ABSTRACT

In a dynamically changing time-varying network environment, the optimal moving pattern of edge devices can be applied to distributing computing resources to edge cloud servers or deploying new edge servers in the FEC(Fog/Edge Computing) environment. In addition, this can be used to build an environment capable of efficient computation offloading to alleviate latency problems, which are disadvantages of cloud computing. This paper proposes an algorithm to extract the optimal moving pattern by analyzing the moving path of multiple edge devices requiring application services in an arbitrary spatio-temporal environment based on frequency. A comparative experiment with A* and Dijkstra algorithms shows that the proposed algorithm uses a relatively fast execution time and less memory, and extracts a more accurate optimal path. Furthermore, it was deduced from the comparison result with the A* algorithm that applying weights (preference, congestion, etc.) simultaneously with frequency can increase path extraction accuracy.

키워드 : 포그/엣지 컴퓨팅, 이동객체, 최적 이동패턴, 계산 오프로딩, 자원 분배 및 배치

Keywords : Fog/Edge computing, Moving object, Optimal moving pattern, Computation offloading, Resource placement

Received 2 November 2021, Revised 11 November 2021, Accepted 18 November 2021

* Corresponding Author YonSik Lee (E-mail: yslee@kunsan.ac.kr, Tel:+82-63-469-4553)

Professor, School of Computer Info. & Comm., Kunsan National University, Kunsan, 54150 Korea

Open Access <http://doi.org/10.6109/jkiice.2022.26.1.162>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. Introduction

최근 무선측위 및 모바일 컴퓨팅 기술 기반의 이동기기 사용의 증가에 따라 사용자의 동적 이동성을 적용한 응용들이 다양하게 개발되고 있다. 그러나 부득이하게 중앙 집중식 클라우드로 오프로딩이 요구되는 응용들은 상당한 네트워크 부하나 지연시간을 초래한다[1]. ‘AI City Safety & Security Summit 2021’에서 ‘미래의 스마트 산업은 에지(Edge)에 달려있다’는 견해가 발표되는 현 시점에서, 사용자 중심의 차세대 컴퓨팅 서비스를 위한 사용자 기기의 정보 분석 및 성능 향상과 효율적 트래픽 처리, 특히 서비스 지연시간 문제 해결을 위하여 클라우드 서비스를 새로운 FEC(Fog/Edge Computing) 패러다임에서 제안되는 모바일 네트워크의 가장자리(edge cloud)로 이동시켜 사용자 장비와의 근접성을 확보하는 방법이 요구된다[2,3]. 또한 서비스를 요구하는 단말기기(이동객체)들은 시공간 속성이 지속적으로 변화하므로, 이들의 동적 위치 변화의 패턴을 예측하여 FEC 환경에서 컴퓨팅 리소스를 효율적으로 분배 및 배치하는 방법이 요구된다[4,5].

이동객체 관련 데이터 모델링, 인덱싱, 불확실성 처리, 궤적 추적 및 예측방법 등이 다양하게 연구되었지만, 이동객체의 시공간적 이동에 따른 의미 패턴 추출과 이를 이용한 컴퓨팅 및 자원 분배에 대한 연구는 미흡한 실정이다[2,4,6]. 이동객체의 이동패턴 추출은 빈번한 데이터베이스 검색 및 다량의 데이터 탐사로 인한 시간 지연과 메모리 소모가 발생하며, 시공간적 특성을 고려하기 위한 비용 문제 등이 발생한다. 따라서 특정 영역 내에서 시공간 제약을 적용한 효율적이고 능동적인 빈발 이동경로 추출 및 경로 예측방법이 요구된다.

본 논문은 FEC 환경에서 응용 서비스의 사용자 근접성을 지원하는 컴퓨팅 리소스의 최적할당 및 지연시간 감소를 위한 이동에이전트[7,8,9] 기반 오프로딩 기술 개발[2,3,6]의 선행연구로써, 응용 서비스를 요구하는 다수의 이동객체의 이력데이터로부터 시공간적 제약조건들을 만족하는 빈발도 기반의 최적 이동경로를 추출하는 알고리즘을 제안한다. 또한 제안방법의 성능평가를 통하여 계산 오프로딩을 위한 컴퓨팅 리소스들의 효율적 분배 및 배치방법에의 적용성을 제시한다.

II. Related works

객체의 시공간 패턴 추출방법은 진화규칙을 적용하여 시공간 속성 변화에 따른 반복 또는 공통적 패턴을 추출하는 것으로, 위치변화패턴, 군집패턴 및 동작패턴 등의 추출을 위한 다양한 연구들이 진행되었다[10,11,12]. 이동객체의 이동패턴 추출방법은 주로 Apriori 계열 알고리즘을 사용하였으나, 이는 수준별 추출방법 구분 적용으로 데이터베이스 반복 접근에 따른 수행성능 저하를 초래하며, 이를 해결하기 위한 최소지지도 적용 방법은 수행시간은 감소시키지만 비교 연산 및 소요 메모리량의 증가를 유도한다[10,14]. 데이터 일반화 및 이동패턴 트리 적용 알고리즘은 소요 메모리량과 수행시간을 감소시키지만, 데이터양이 많을 경우 과다한 이동패턴 트리 생성 및 유지비용이 발생하는 단점이 있다[13].

네트워크 내에서 객체의 이동성 예측을 위하여 순차 패턴 추출, 특정 지지도 산출 및 조인 연산된 k-시퀀스 집합 등의 개념을 적용하였으나[10,11], 이동객체 수와 최소지지도에 따른 지지도 산출 및 후보패턴 집합 생성을 위한 데이터베이스 스캔 요구로 인하여 수행시간이 지연되는 문제점을 가진다. Apriori-MSP 알고리즘의 유효 소요시간 및 중복시퀀스를 적용한 지지도 산출방법은 패턴 추출의 정확성이 저하되는 문제점이 있다[15]. 따라서 시공간 제약을 만족하면서 데이터베이스 접근 최소화, 소요 시간 및 메모리량을 최적화시킬 수 있는 이동객체의 최적 이동패턴 추출방법이 요구된다.

임의의 네트워크에서 두 노드 사이의 주어진 가중치가 최소인 최적경로 탐색방법들은 Dijkstra, Moore, Shier, Floyd-Marshall 알고리즘 등을 사용하는데 이들은 정확하고 계산이 빠른 장점이 있지만, 탐색 범위의 크기에 따라 탐색시간과 공간비용이 증가하는 문제점이 있다[14,15]. 반면에 A* 알고리즘은 최적경로 탐색과정에서 잔여 경로비용의 최소 추정치를 통한 경험적 가중치를 각 노드에 적용하여 탐색함으로써, 연산 시간과 접근노드 수를 감소시키는 장점이 있다. 그러나 A* 알고리즘은 가중치에 따라 탐색 범위가 증가할 수 있으며, 이를 해결하기 위한 양방향성 Dijkstra 및 A* 알고리즘이 제안되었으나 경우에 따라 탐색 계산량이 오히려 증가하는 문제점을 지닌다[16]. 따라서 효율적 계산 오프로딩을 위한 컴퓨팅 리소스의 최적할당 및 지연시간 감소를 위하여, 시간 가변적 네트워크를 고려하지 않은 기

존의 정적 알고리즘의 한계를 극복하는 동적으로 변하는 환경(시간 제약, 이동규칙 등)에서의 최적경로 탐색 방법이 필수적으로 요구된다.

III. Definitions of moving object and optimal moving pattern

3.1. Moving object

무선 센서네트워크의 엣지 디바이스에서 요구되는 응용(AoE : App. on Edge)은 시간의 흐름에 따라 이동하며 요구 자원과 환경이 지속적으로 변하는 특성을 가지는 시공간 이동객체이다. 이동객체 표현을 위한 데이터 모델은 점들의 집합을 사용하여 연속적 또는 이산적 모델을 사용할 수 있으며, 이들의 위치 속성은 x-y 2차원 공간에서 연속적으로 변화한다. 본 논문에서는 이동객체의 위치가 연속적으로 변하는 이산적 시점에서 획득됨을 가정하고, 센서네트워크 상의 AoE를 이동객체로 정의한다. AoE의 위치 데이터는 객체 식별자(oid)와 이산적으로 획득된 연속적 이동위치 데이터(객체 위치(l), 시간(vt))로 구성하여 (oid, l, vt)로 표현한다. 객체 위치 l은 x-y 좌표 값이고, vt는 유효시간을 나타낸다. 이동객체 AoE는 $AoE = \langle oid, s, t, a \rangle$ 와 같이 정의한다. 여기서, 공간적 속성 s는 2차원 공간이며 공간 차원 도메인 내에 있고, 공간상의 위치점이나 영역을 표시하는 값이다. 시간적 속성 t는 유효시간이며 시간 차원 도메인 내에 있고, 타임라인 상의 한 시점이나 기간을 표시하는 값이다. a는 속도나 상태를 나타내는 일반 속성 값이다. AoE의 이력데이터 집합은 AoE의 인스턴스 $AoE_i = \langle oid_i, li, vt_i, ai \rangle, 0 \leq i \leq n$ 들의 집합이며, 동일 oid의 이력들은 서로 다른 시점에서 획득된 데이터이므로 서로 다르다.

3.2. Optimal moving pattern

최적 이동패턴 추출을 위한 빈발 이동패턴은 시공간 속성에 따른 AoE의 위치변화 패턴으로부터 위치 기반 서비스에 응용이 가능한 유의미한 패턴이며, 요구되는 지식들은 모두 AoE의 공간 영역 변화에 대한 연관성 및 시간 제약조건을 적용한 것이다.

본 논문에서는 AoE의 순차적으로 변하는 이동패턴 추출을 위하여 AoE 데이터베이스(AoE-DB), 사용자 지

정 최소지지도(min_sup) 및 인접 공간영역 간의 최대 시간간격(max_gap)을 적용하여 이들을 만족하는 이동시퀀스를 빈발 이동시퀀스로 정의한다. 최소지지도는 임의의 이동시퀀스를 빈발 시퀀스로 결정하기 위한 지지도의 하한 값이다. 최대 시간간격은 시퀀스 내에서 인접영역으로 이동한 시간 (ti - ti-1) 이상의 크기인 시간(ti - ti-1 ≤ max_gap)이며, 시퀀스 생성 시 시간 구분에 따라 이동시퀀스를 여러 개로 분리하여 생성한다.

본 논문에서 적용하는 공간제약은 응용의 요구에 따른 특별한 위치나 영역들에 존재하는 AoE의 이동데이터들을 대상으로 한다. 또한, AoE의 이력데이터로부터 특정 지점들 사이에서 시공간 한정자 및 특정제약을 만족하는 가장 빈발한 이동패턴을 최적(지연시간 및 네트워크 트래픽 부하 최소화) 이동패턴으로 정의한다. 빈발도 기반 최적 이동패턴은 AoE-DB와 사용자 지정 특정 지점들 및 시공간 제약이 주어졌을 때, 특정지점(출발지, 목적지)들 사이를 최대지름으로 하는 영역 내의 경로들을 대상으로 주어진 조건들을 만족하는 이동시퀀스 집합들 중에서 순차적으로 추출한다.

IV. Optimal moving pattern extraction

이동패턴 추출은 AoE의 이동이력데이터로부터 이동시퀀스를 생성 및 일반화하여 트랜잭션 데이터베이스를 생성하는 전처리과정과 최적 이동패턴 추출과정으로 구분한다.

4.1. Moving sequence extraction

AoE의 이동패턴 추출을 위해서는 AoE의 이동이력 데이터에 시간간격 제약조건을 적용하여 이동시퀀스로 생성하고, 데이터 일반화를 통해 이들을 일반화된 시퀀스로 변환하여 사용한다. 다음 그림 1은 AoE가 이동 가능한 시공간속성 네트워크상에서의 이동경로에 따른 물리적 요소를 나타내며, 공간의 최상위 레벨은 $R = \{R1, R2, R3\}$ 이며, $R1 = \{A1, A2, \dots, A1\}$, $R2 = \{B1, B2, \dots, Bm\}$, $R3 = \{C1, C2, \dots, Cn\}$ 이고, 각 A, B, C는 최하위 레벨인 각 링크를 연결하는 노드들의 집합으로 AoE의 위치 샘플링에 사용한다. AoE가 통과하는 순서에 의한 엣지 클라우드의 집합을 $EC = \{EC_1, EC_2, \dots, EC_n\}$ 으로 정의한다. 모든 요소 $EC_i = \{t_i, t_{i+1}\}$ 에서 t_i 와

t_{i+1} 은 AoE의 i 번째 엣지 클라우드 커버리지 영역의 진입과 이탈 시간을 각각 나타낸다. AoE의 이동에 따라 응용 서비스 요청은 언제든지 발생할 수 있으며, 이는 AoE와 일부 엣지 클라우드 서버의 공동 실행으로 처리될 수 있다. AoE의 이동속도에 따라 하나의 커버리지에서 머무는 시간이 짧아서 완료를 위하여 두 개 이상의 엣지 서버가 필요할 수 있으며(그림 1의 N14), 이러한 엣지 서버는 서로 통신이 가능해야 한다. 서비스 요청의 시작 시간 t_α 와 서비스 작업이 완료되는 시간 t_ϵ 을 정의하고, AoE가 서비스 작업을 처리하기 위해 동일 엣지 클라우드에서 보내는 시간과 작업을 완료할 때까지의 시간 간격들을 일반화 과정을 통하여 지점(노드) 단위로 구분한다. 그림 1에서 t_α 에서 서비스 요청이 발생하여 t_ϵ 에서 작업이 완료되는 경우, 7개의 시간 간격들 $((t_1 - t_\alpha), (t_2 - t_1), (t_3 - t_2), (t_4 - t_3), (t_5 - t_4), (t_6 - t_5), (t_\epsilon - t_6))$ 로 구성된 이동시퀀스가 발생한다.

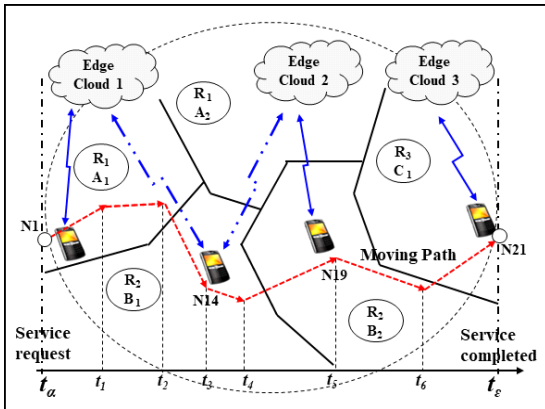


Fig. 1 Physical elements according to the moving path of an AoE on spatio-temporal property network

2차원 공간상에서 샘플링된 AoE의 이동이력은 시공간 속성을 갖는 이동시퀀스로 표현하며, k 개의 이동이력으로 구성되는 이동시퀀스 seq 는 $seq = \{(l_1, vt_1), (l_2, vt_2), \dots, (l_k, vt_k)\}$ 이다. 각 AoE별로 유효시간을 기준으로 추출된 이동시퀀스는 위치 변화에 대한 시간 간격이 적용되지 않아 패턴을 추출하기 위한 트랜잭션으로 사용하기에는 부적절하므로, 인접 위치로 이동한 시간 간격이 주어진 최대 시간 제약인 max_gap 을 초과하면 별도의 이동시퀀스로 분리하여 추출한다.

최적 이동패턴 추출은 공간상의 특정 지점들 간의 상

세 수준 이동경로가 요구되므로, 시공간 속성에서 최하위 수준으로 일반화된 각 oid 의 SEQ_ID 에 해당하는 노드들의 연결리스트들을 이동시퀀스 집합으로 사용하며, 빈발 2-node 시퀀스 집합으로부터 생성된 후보 시퀀스 집합들에 지지도를 적용하여 빈발 시퀀스를 순차적으로 생성한다(그림 3 참조). 제안 방법은 후보 시퀀스 생성과정에서 시퀀스들의 항목 차수가 점차 감소하므로 연산시간을 감소시킬 수 있으며, 시공간 속성으로 일반화된 데이터 집합을 사용함으로써 추출된 패턴의 지식화를 통하여 FEC 컴퓨팅 환경에서 요구되는 컴퓨팅 리소스 분배 및 배치에 효율적 적용이 가능하다.

4.2. Frequency-based optimal moving pattern extraction algorithm

빈발도 기반 최적 이동패턴 추출 알고리즘의 2-node 시퀀스 생성모듈은 최초 출발지(S)로부터 시작하는 2-node 시퀀스 $\langle nodeS \ nodeN \rangle$ 들을 구한 후 최대지지도를 갖는 2-node 시퀀스를 추출한다. 추출된 2-node 시퀀스의 연결노드인 $nodeN$ 을 기준으로 주어진 공간 영역 내에서 목적지(D)를 포함하는 모든 부분 이동시퀀스들의 집합을 구한 후, S를 대신한 $nodeN$ 으로부터 이전 과정을 반복 수행한다. 이 과정은 $nodeN$ 이 D이면 종료하고, S로부터 순차적으로 연결된 $nodeN$ 들의 리스트를 최적 이동패턴으로 결정 반환한다.

다음 그림 2는 빈발도 기반 최적 이동패턴 추출 알고리즘 OPE_freq이다. 먼저 AoE-DB로부터 공간 한정자 C_s 와 공간 제약 조건을 만족하는 이력데이터 집합을 추출한 후 SEQ_Extract 함수를 통해 max_gap 을 적용한 이동시퀀스 데이터를 생성한다. 생성된 이동시퀀스 데이터로부터 시공간 속성으로 일반화된 이동시퀀스 집합을 생성하여 최적 이동패턴 추출 모듈을 통해 최대 빈발패턴인 최적 이동패턴을 추출한다.

```

Input : AoE-DB,  $C_s$  (Spatial Constraint),  $S$  (Start node),
currN (Current node),  $D$  (Destination node),
max_gap (Temporal Constraint), min_sup (Threshold
of Minimum Support factor), G-Mseq (Generalized
Moving Sequence)
Output : Optimal Moving Pattern (OMP_list)
Procedure OPE-freq (AoE-DB,  $C_s$ ,  $S$ ,  $D$ , max_gap, min_sup)
Begin
PrevA = null;
// Set of historical data satisfied  $C_s$ 
    
```

```

Drt = limited Data Set (AoE-DB, Cs, S, D);
// Extraction of set of moving sequences from Drt
Dseq = SEQ_Extract (Drt, max_gap);
// Data Generalization
// Extraction of optimal moving pattern
Begin
OMP_list = null;
nextN = S;
while (nextN ≠ D) {
    add nextN to OMP_list;
    Begin
    subD = Sequence containing currN and
    D in G-Mseq;
    for each moving sequence S ∈ subD
        c = subset(currN, S);
        if c ∉ Cs then add c to Cs;
        c.count++;
    End for
    c_max_sup = 2-node sequence with
    maximum support factor from Cs;
    nextN = new node unconnected with
    currN in c_max_sup;
    Return nextN;
    End
}
add D to OMP_list;
End
Return OMP_list;
End
    
```

Fig. 2 OPE_freq algorithm

다음 그림 3은 6개 AoE의 이동이력데이터로부터 구해진 이동시퀀스 집합에 대하여 OPE_freq 알고리즘을 적용하여 최적 이동패턴을 추출하는 과정을 보인다.

OPE_freq 알고리즘을 이용한 최적 이동패턴 추출 시스템은 빈발 이동패턴 추출을 위한 조건과 질의의 입력을 위한 사용자 인터페이스 모듈, 데이터 일반화를 위한 데이터 전처리 및 요구에 적합한 지식 추출을 위한 패턴 추출 모듈, 추출된 패턴들의 저장, 관리 및 가공을 통한 서비스를 위한 규칙 모듈, 공간 정보 및 AoE-DB와 지식 베이스로 구성된 데이터 및 정보 모듈 등으로 구성한다.

본 논문에서 제안한 최적 이동패턴 추출방법은 사용자의 의도에 따른 시공간제약을 만족하는 빈발패턴을 추출하며, 이동에 따른 시간, 거리, 비용 등의 기준 대신 최대 빈발경로를 추출함으로써 AoE들이 특정 목표지점으로 이동하는 최적경로를 추출하는 특징을 갖는다. 이러한 특징을 FEC환경에서 AoE들의 최적 이동경로에 근접한 에지 클라우드 서버에 컴퓨팅 리소스를 분배하

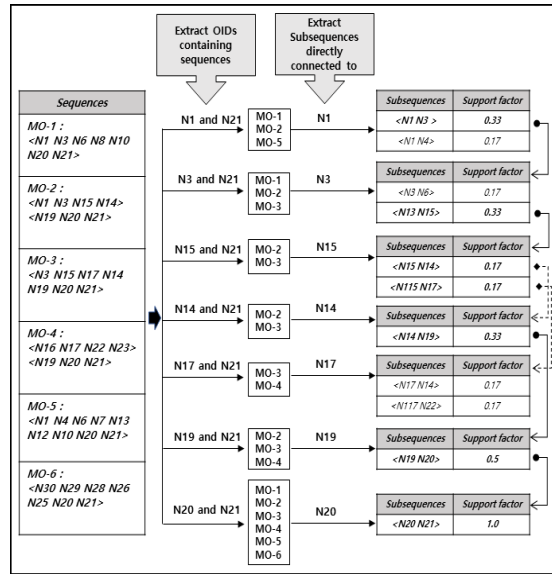


Fig. 3 Optimal moving pattern extraction process based on frequency

거나 새로운 에지 서버(기지국)를 배치하는데 적용함으로써, 클라우드 컴퓨팅의 단점인 지연시간 문제 완화를 위한 효율적 계산 오프로딩이 가능한 환경 구축에 활용이 가능하다.

V. Experiments and results

본 장에서는 OPE_freq 알고리즘의 성능을 A* 및 Dijkstra 알고리즘을 대상으로 최적 이동패턴 추출의 효율성과 정확성 비교실험 결과를 통하여 평가한다.

5.1. Experimental environment

OPE_freq 알고리즘을 적용한 최적 이동패턴 추출 시스템은 Windows x64 OS에서 Eclipse IDE와 JDK 8, Oracle 10g 버전 데이터베이스를 기반으로 구현하였고, 실험은 Oracle 데이터베이스 시스템을 포함하여 Intel core i7-9700 CPU 3GHz, 8 GByte main memory PC들을 사용하였다. AoE의 이력데이터인 실험 데이터는 서울시 도로 네트워크상에서 x-y 좌표로 측정된 택시 운행 기록을 사용하고, 임의의 노드(교차, 분기 및 목적지)들을 설정하여 AoE의 위치데이터를 생성하였다. 실험 데이터는 'AoE ID, 노드의 x-y 좌표 값, 유효시간'으로 표

시하고, AoE 1,000개에 대한 10시간의 위치데이터를 수집하였고 그 크기는 15.7MB이다. 최적 이동경로 추출 성능을 평가하기 위하여 Dijkstra 알고리즘과 A* 알고리즘을 비교대상으로 설정하고, 동일한 조건에서 출발지로부터 목적지를 임의로 변경하여 각각 150, 300, 450 번의 경로탐색을 위한 연산시간, 접근 노드 수 및 경로 추출의 정확도를 기준으로 평균값을 산출하여 비교 평가한다. A* 알고리즘은 경험적 가중치에 따라 성능 차이가 크게 발생하므로 두 개의 구간범위(A*-a: 5~10Km, A*-b: 15~20Km)로 구분하여 실험한다.

5.2. Performance comparison and evaluation

5.2.1. Comparison of execution time

주어진 출발지와 목적지 사이의 최적 이동경로 탐색을 동일 시공간 범위에서 연산횟수를 구분하여 실험한 평균 연산시간 결과는 다음 그림 4와 같다.

제안 알고리즘인 OPE_freq 알고리즘은 경험적 가중치를 적용한 A* 알고리즘에 비하여 연산시간이 무시할 정도로 조금 길지만, 동심원 내에 포함된 모든 노드들을 대상으로 탐색을 수행하는 Dijkstra 알고리즘과는 큰 차이가 있다. 구간범위의 크기에 따라 비교한 두 A* 알고리즘의 수행시간 차이는 A* 알고리즘의 경험적 가중치가 가지는 직진성에 의한 차이로써, 구간범위가 넓어질수록 A* 알고리즘의 수행시간은 길어진다.

OPE_freq 알고리즘은 최대 빈발도 산출을 위한 시간을 포함하지만, 수행횟수와와의 관계에서 거의 정비례하듯이 증가하는 Dijkstra 알고리즘을 제외한 다른 알고리즘



Fig. 4 Comparison of execution time

들은 연산시간의 성능이 거의 차이가 없는 특징이 있다.

5.2.2. Comparison of number of nodes accessed

최적경로 추출 시 노드에 접근하는 횟수는 연결 노드를 최적경로로 선택할지의 여부 결정으로 인한 수행시간이 소요되므로 알고리즘 성능과 밀접한 관계가 있다.

빈발도를 적용한 OPE_freq 알고리즘은 빈발도가 작은 접근 대상 노드들의 수가 매 단계에서 줄어들기 때문에, 그림 5와 같이 다른 방법에 비하여 가장 적은 수의 노드 접근 성능을 가진다. A* 알고리즘은 구간범위가 넓어질수록 알고리즘의 직진성으로 인하여 접근 노드의 수가 비효율적으로 증가하므로 경험적 가중치의 크기가 성능 개선에 중요한 변수로 작용함을 알 수 있다.

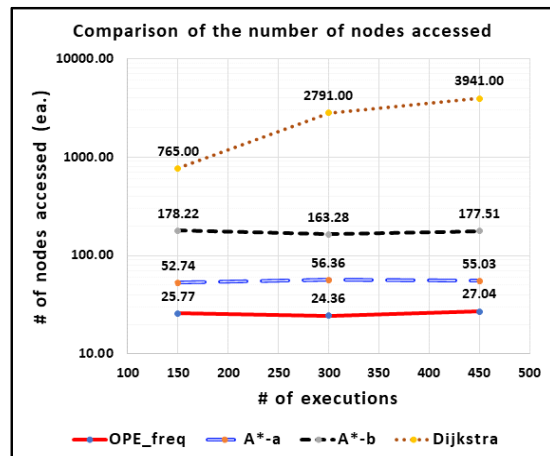


Fig. 5 Comparison of the number of nodes accessed

5.2.3. Comparison of accuracy

추출된 최적경로의 정확도 평가를 위하여 모든 대상 노드들을 탐색하는 Dijkstra 알고리즘의 정확도를 100%로 설정한다. 실험 결과인 Fig. 6에서 보듯이 작은 구간 범위에 적용된 A* 알고리즘이 경험적 가중치에 의한 직진성에 의해 탐색 범위가 작으므로 정확도가 낮아짐을 알 수 있고, 경험적 가중치를 넓은 영역에 적용하면 정확도는 향상되지만 연산시간이 증가하게 된다. OPE_freq 알고리즘은 짧은 연산시간 대비 최적경로 추출의 정확도는 상대적으로 우수함을 알 수 있다.

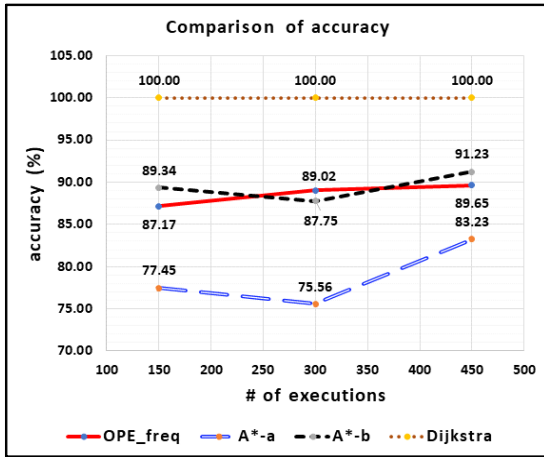


Fig. 6 Comparison of accuracy

VI. Conclusions

본 논문은 FEC 환경에서 응용 서비스의 사용자 근접성을 지원하는 컴퓨팅 리소스의 최적 할당 및 지연시간 감소를 위한 이동에이전트 기반 오프로딩의 효율성을 제고하기 위한 선행연구로써, AoE들의 방대한 시공간 이동이력데이터 집합으로부터 복잡한 시공간 제약을 적용한 최적 이동패턴 추출 시 수행시간 및 소요 메모리 양을 효과적으로 감소시키는 OPE_freq 알고리즘을 제안하였다. OPE_freq 알고리즘은 빈발도 기반의 이동시퀀스를 이용하므로, 데이터베이스 접근 횟수를 줄여 수행시간을 감소시킬 뿐만 아니라 시공간 속성 일반화를 통하여 소요 메모리를 줄일 수 있다.

실험을 통하여 OPE_freq 알고리즘의 성능을 연산 처리시간, 접근노드 수 및 최적경로 추출의 정확도 등 기준으로 Dijkstra 및 A* 알고리즘과 비교하여 평가하였다. 성능평가 결과에서 연산 처리시간과 접근노드 수 기준 OPE_freq 알고리즘은 상대적으로 안정적인 처리시간 성능을 보인 반면에 A* 알고리즘은 구간범위가 커질수록 처리시간이 가파르게 증가하였고, Dijkstra 알고리즘은 구간범위 내의 모든 노드들을 탐색하므로 최악의 처리시간이 소요되었다. 최적경로 추출의 정확도에 대한 실험 결과 완벽한 정확도를 갖는 Dijkstra 알고리즘의 성능을 기준으로 OPE_freq 알고리즘은 A* 알고리즘들보다 우수한 성능을 보였다. 추가적으로 경험적 가중치

를 적용한 A* 알고리즘과의 비교 실험 결과를 통하여 이동경로 상의 사용자 가중치(선호도, 혼잡도 등)를 빈발도와 동시에 적용함으로써 최적경로 추출의 정확도를 향상시킬 수 있음을 도출하였다.

이러한 임의의 시공간 환경에서 능동적인 이동패턴 추출방법은 AoE의 이동경로나 시공간적 동적 특성을 적용한 응용 서비스에 적용이 가능하므로, FEC 환경에서 컴퓨팅 및 스토리지 리소스의 사용자 근접 배치 및 이동에이전트 기반의 효율적 계산 오프로딩 방법 개발에 적용이 가능하다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1F1A1047768) and a grant (21RITD-C161698-01) from Regional Innovation Technology Development Program funded by Ministry of Land, Infrastructure and Transport of Korean government

REFERENCES

- [1] P. Ren, X. Qiao, Y. Huang, L. Liu, S. Dustdar, and J. Chen, "Edge-Assisted Distributed DNN Collaborative Computing Approach for Mobile Web Augmented Reality in 5G Networks," *IEEE Networks*, vol. 34, no. 2, pp. 254-261, Mar. 2020.
- [2] M. Chen, B. Liang, and M. Dong, "Multi-user multi-task offloading and resource allocation in mobile cloud systems," *IEEE Transaction on Wireless Communication*, vol. 17, no. 10, pp. 6790-6805, Aug. 2018.
- [3] H. Alameddine, S. Sharafeddine, S. Sebbah, and S. Ayoubi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE Journal of Selected Areas Communication*, vol. 37, no. 3, pp. 668-682, Jan. 2019.
- [4] A. Nadembega, A. Hafid, and R. Brisebois, "Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE," in *Proceedings of the 2016 IEEE International Conference on Communications*, pp. 22-27, 2016.

- [5] S. Jošilo and G. Dán, "Selfish Decentralized Computation Offloading for Mobile Cloud Computing in Dense Wireless Networks," *IEEE Transactions on Mobile Computing*, vol. 18, pp. 207-220, Apr. 2018.
- [6] Z. Chen, J. Hu, X. Chen, X. Zheng, and G. Min, "Computation Offloading and Task Scheduling for DNN-Based Applications in Cloud-Edge Computing," *IEEE Access*, vol. 8, pp. 115537-115547, Jun. 2020.
- [7] Y. Lee, "Lightweight and Migration Optimization Algorithms for Reliability Assurance of Migration of the Mobile Agent," *Journal of The Korea Society of Computer and Information*, vol. 25, no. 5, pp. 91-98, May. 2020.
- [8] S. P. Ardakani, J. Padget, and M. De Vos, "A Mobile Agent Routing Protocol for Data Aggregation in Wireless Sensor Networks," *International Journal of Wireless Information Networks*, vol. 24, no. 1, pp. 27-41, Dec. 2017.
- [9] S. Feng, C. Wu, Y. Zhang, and G. Olivia, "WSN Deployment and Localization Using a Mobile Agent," *Wireless Personal Communications*, vol. 97, no. 4, pp. 4921-4931, Nov. 2017.
- [10] T. Shi, W. Han, and N. Tao, "Mining Aggregation Moving Pattern of Moving Object From Spatio-temporal Trajectories," *Minimicro Systems*, vol. 40, no. 5, pp. 1099-1106, 2019.
- [11] K. Bok, C. Lee, and J. Yoo, "Recommending similar users using moving patterns in mobile social networks," *Computers & Electrical Engineering*, vol. 77, pp. 47-60, Jul. 2019.
- [12] B. Qian, Y. Wang, R. Hong, M. Wang, and L. Shao, "Diversifying Inference Path Selection: Moving-Mobile-Network for Landmark Recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 4894-4904, May. 2021.
- [13] T. Thianniwet, S. Phosaard, and W. Pattara, "Classification of Road Traffic Congestion Levels from Vehicle's Moving Patterns: A Comparison Between Artificial Neural Network and Decision Tree Algorithm," *Electronic Engineering and Computing Technology*, vol. 60, pp. 261-272, Feb. 2010.
- [14] Y. Ye, "Research and Application of Apriori Algorithm for Mining Association Rules," *Advanced Materials Research*, vol. 1079-1080, no. 2, pp. 737-742, Dec. 2015.
- [15] N. Benhamouda, H. Drias, and C. Hirèche, "Meta-Apriori: A New Algorithm for Frequent Pattern Detection," *Asian Conference on Intelligent Information and Database Systems*, vol. 2016, pp. 277-285, 2016.
- [16] Q. Ji and S. Zhang, "Research on sensor network optimization based on improved Apriori algorithm," *EURASIP Journal on Wireless Communications and Networking*, Nov. 2018.



이연식(YonSik Lee)

전남대학교 전자계산학과 이학사
 전남대학교 대학원 전자계산학과 이학석사
 전북대학교 대학원 전자공학과 전산응용공학
 전공 공학박사
 군산대학교 컴퓨터정보통신공학부 정교수
 ※관심분야: 에이전트 시스템, S/W 미들웨어



남광우(KwangWoo Nam)

충북대학교 컴퓨터과학과 이학사
 충북대학교 대학원 전자계산학과 이학석사
 충북대학교 대학원 전자계산학과 이학박사
 2001년~2004년 한국전자통신연구원 선임연구원
 군산대학교 컴퓨터정보통신공학부 정교수
 ※관심분야: 모바일, 컨버전스, GIS,
 빅데이터 마이닝



장민석(MinSeok Jang)

연세대학교 전자공학과 공학사
 연세대학교 대학원 전자공학과 공학석사
 연세대학교 대학원 전자공학과 공학박사
 군산대학교 컴퓨터정보통신공학부 정교수
 ※관심분야: IoT 시스템, 드론 응용, 딥러닝