

논문 2025-3-2 <http://dx.doi.org/10.29056/jsav.2025.09.02>

# Tottering 현상을 방지하는 개선된 Random Walk 그래프 탐색 기법에 대한 연구

윤성철\*, 김수현\*\*†, 이임영\*\*

## A Study on the Advanced Random Walk Graph Search Technique to Prevent Tottering Phenomena

SeongCheol Yoon\*, Su-Hyun Kim\*\*†, Im-Yeong Lee\*\*

### 요 약

그래프 편집 거리는 데이터 정보 손실이 적으며, 유연성이 높아 그래프 구조 유사도 검사 시 널리 사용된다. 그러나 그래프 편집 거리는 NP-Hard 문제에 속하므로 그래프의 노드 혹은 엣지가 많은 복잡한 그래프에 대해 효율성이 크게 저하된다. 이와 같은 문제를 해결하기 위해 다양한 휴리스틱 알고리즘을 연구했고, 그 중 Random Walk 기법은 구현이 쉽고, 높은 유연성으로 인해 널리 사용되고 있다. 그러나 Random Walk 기법은 중복 방문 현상인 Tottering 현상이 발생해 효율성이 저하된다. 이를 해결하고자 다양한 연구가 수행되었으나 해당 방식들 또한 Tottering 현상이 최소 1회 이상 발생하는 한계점이 존재한다. 본 논문에선 이러한 문제를 해결하기 위해 Tottering 현상이 발생하기 전 Tottering 검사를 수행하는 개선된 k-Walk 그래프 탐색 기법을 제안한다. 본 제안방식은 Tottering 현상이 발생하기 이전에 Tottering 현상을 탐지함으로써 Tottering 현상 발생을 원천적으로 차단할 수 있다. 이를 그래프 구조 기반 소스코드 유사도 검사에 활용 시 보다 효율적인 유사도 검사가 가능하다.

### Abstract

The graph edit distance algorithm is widely used for structural similarity detection due to its flexibility and minimal information loss, but its computational inefficiency on large, complex graphs stemming from its NP-Hard nature has led to the development of various heuristic, including the Random Walk approach, which, despite its advantages, suffers from the Tottering phenomenon that degrades efficiency; although prior methods attempt to address this, they fail to eliminate Tottering entirely, prompting this paper's proposal of a novel Random Walk technique that proactively detects and prevents Tottering before it occurs. This enables more efficient similarity checks when applied to graph structure based source code similarity detection.

**한글키워드 :** 그래프, 그래프 편집 거리, 휴리스틱 알고리즘, 무작위 탐색, 효율성

**keywords :** Graph, Graph Edit Distance, Heuristic Algorithm, Random Walk, Efficiency

\* 순천향대학교 SW융합학과

\*\* 순천향대학교 컴퓨터소프트웨어공학과

† 교신저자: 김수현(email: kimsh@sch.ac.kr)

접수일자: 2025.07.22. 심사완료: 2025.08.28.

게재확정: 2025.09.20.

### 1. 서 론

현대 사회에선 소스코드 연결 구조, 단백질 구

조, 네트워크, 지식 그래프 등 복잡한 관계를 표현하는데 그래프를 활용하고 있다[1, 2]. 이와 같은 환경에서 그래프로 표현된 데이터에 대한 분석을 위해, 두 그래프의 구조를 파악한 뒤 구조적 유사도를 식별하는 것은 매우 중요한 일이다. 두 그래프에 대한 구조적 유사도를 파악하기 위해 다양한 방법이 사용될 수 있으나, 그 중 그래프 편집 거리 알고리즘은 두 그래프의 구조적 유사도를 측정하는 가장 대표적인 방법이다.

그래프 편집 거리 알고리즘이란 하나의 그래프를 다른 그래프로 변환하기 위해 필요한 편집 연산의 최소 비용을 구하는 알고리즘이다[3, 4]. 그러나 그래프 편집 거리 알고리즘의 경우 NP-Hard 문제에 속한다[3, 5, 6]. 따라서 그래프의 노드 수, 엣지 수가 증가할수록 효율성이 크게 저하되는 문제가 발생한다. 이를 해결하기 위해 정확한 값을 찾는 것이 아닌, 허용할 수 있는 수준의 오차를 가진 근사값을 다항 시간 내에 찾아내는 휴리스틱 알고리즘에 대한 연구가 활발히 수행되고 있다[6, 7-10].

그래프 편집 거리에 대한 휴리스틱 알고리즘으로는 BP(Bipartite Graph Matching) 방식, LSAPE(Linear Sum Assignment with Error Correction) 방식, Subgraph 방식, Random Walk 방식 등 다양한 알고리즘이 존재한다. 그 중 Random Walk 방식은 구현이 쉽고, 유연성이 높아 다양한 환경에 적용할 수 있다. Random Walk 방식은 다음 탐색할 노드를 무작위로 선정하여 탐색을 진행한다[10, 11]. 그러나, 일반적인 Random Walk 기법들은 중복 방문을 고려하지 않아 Tottering 현상이 발생한다[12, 13].

이를 해결하기 위해 다양한 연구가 수행되었으나, 기존 연구들은 Tottering 현상이 발생하는 것을 완벽히 방지하지 못한다. 이로 인해 그래프 탐색의 효율성, 더 나아가 그래프 탐색을 기반으로 수행되는 그래프의 구조적 유사도 측정에도

정확도, 효율성 감소 등의 문제를 야기한다. 본 논문에서는 이러한 문제를 해결하기 위해 그래프 탐색 시 Tottering 현상을 사전적으로 탐지하여 Tottering 현상 발생을 방지하는 개선된 k-Walk 탐색 기법을 제안한다. 본 논문에 대한 우리의 기여는 아래와 같다.

- **Tottering 현상을 차단한 개선된 제안방식 제안:** 기존 연구들에서 감소시켰던 Tottering 현상 발생을 원천적으로 차단하는 개선된 Random Walk 탐색 기법을 제안한다. 기존 Gaüzère et al.[14]과 Yoon et al.[15]에선 각각 탐색 거리  $k$ , 큐를 활용한 Tottering 현상 탐지를 제공했으나, 해당 기법들은  $k$ 번의 탐색 동안 다수의 Tottering 현상이 발생하거나, Tottering 현상 탐지를 위해선 1회의 Tottering 현상이 발생하는 한계점이 존재한다.
- **그래프 탐색량 감소:** 기존 Yoon et al.[15]의 경우 Tottering 현상을 탐지하기 위해 1번의 Tottering 현상이 발생하는 문제점이 존재한다. 이로 인해 탐색량이 증가하는 문제가 발생한다. 본 제안방식은 Tottering 현상을 사전에 탐지하여 차단하므로, 기존 Yoon et al.[15]의 방식보다 낮은 탐색량으로 그래프를 탐색할 수 있다.
- **그래프 탐색 시간 감소:** 기존 Yoon et al.[15]의 경우 추가적인 탐색량으로 인해 탐색 시간 또한 증가하는 문제가 발생한다. 본 제안방식은 불필요한 탐색이 발생하지 않아 탐색 시간 또한 감소시킬 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에선 Random Walk의 관련 연구를 소개한다. 3장에선 제안방식의 요구사항을 분석하고, 4장에서 본 제안방식에 대해 설명한다. 5장에선 제안방식과 기존 Yoon et al.[15]의 방식을 비교분석하고, 6장에서 결론으로 마무리 짓는다.

## 2. 관련 연구

본 장에서는 Random Walk와 관련된 기존 연구와 Tottering 현상에 대해 설명한다. Random Walk와 관련된 기존 연구들로서는 Gaüzère et al.[16]과 Yoon et al.[17]이 존재한다. 이에 대한 자세한 설명은 이하 절에서 설명한다.

### 2.1 Gaüzère et al.

Gaüzère et al.[14]은 기존 이웃 노드 기반의 매칭 기법의 한계점을 개선하기 위해 k-Walk 개념을 접목하여 기존 기법의 한계점을 개선했다. k-Walk란 시작 노드를 기준으로 k번의 탐색을 수행하는 그래프 탐색 기법이다. 즉, k=3일 경우 시작 노드로부터 총 3번의 탐색을 수행하는 것이다. Gaüzère et al.[16]은 이와 같은 k-Walk를 통해 생성된 탐색 시퀀스를 기반으로 더욱 정확한 노드 매칭을 제공한다는 장점이 존재한다.

그러나 Gaüzère et al.[14]에서 사용한 k-Walk 방식은 Tottering 현상을 방지할 수 없다는 한계점이 존재한다. k-Walk 방식은 노드를 탐색 후, 해당 노드와 인접한 이웃 노드 중 하나를 임의로 선택해 탐색을 진행한다. 즉, 현재 탐색 노드 기준으로 직전 탐색 노드를 바로 탐색해 Tottering 현상이 발생할 수 있다는 한계점이 존재한다.

### 2.2 Yoon et al.

Yoon et al.[15]은 Gaüzère et al.[14]에서 발생한 Tottering 현상을 감소시키기 위해, 그래프 탐색 시 큐를 활용하여 중복 방문 검사를 수행하는 탐색 기법을 제안했다.

해당 방식은 그래프 탐색 시 탐색 노드값을 큐에 저장 후, 이후 노드 탐색 시 큐에 저장된 첫 번째 원소값과 현재 탐색 노드값을 비교하여 Tottering 현상을 탐지한다.

그러나 Yoon et al.[15]의 방식은 Tottering 현

상 탐지를 위해선 Tottering 현상이 발생한 탐색을 수행해야 한다는 한계점이 존재한다. 즉, Tottering 현상 탐지를 하기 위해선 반드시 1회 Tottering 현상이 발생해야 한다는 의미이다.

따라서 기존 해당 방식은 탐색량이 증가하고, 이로 인해 탐색 시간 또한 증가하여 이후 그래프 유사도 검사 시 효율성이 감소하는 문제가 발생하게 된다.

### 2.3 Tottering 현상

Tottering 현상이란 그래프 탐색 시 두 노드만을 번갈아가며 탐색하는 현상을 의미한다[16]. 예를 들어 노드 1, 2, 3으로 구성된 그래프 탐색 시 1, 2, 1 혹은 2, 3, 2와 같이 두 노드만을 번갈아가며 탐색하는 현상을 Tottering 현상이라고 한다. 이에 대한 그림은 아래 그림 1과 같다.

이러한 Tottering 현상이 발생할 경우 그래프 구조 파악에는 불필요한 탐색이 발생하여 이후 그래프 유사도 검사 시 정확도 감소, 탐색량 증가, 탐색 시간 증가 등 다양한 악영향을 발생시킬 수 있다[13, 17]. 이러한 문제를 방지하기 위해 그래프 탐색 시 Tottering 현상은 반드시 방지해야 하는 현상이다.

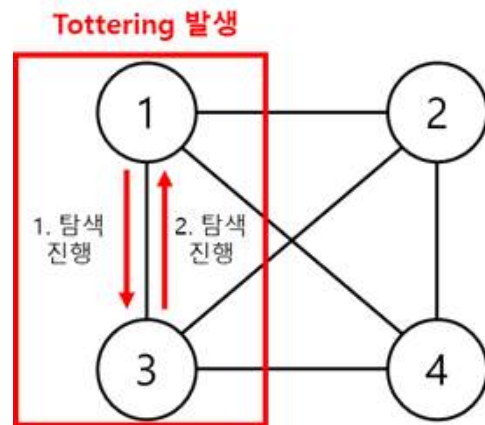


그림 1. Tottering 현상 발생 예시  
Fig. 1. Example of tottering phenomenon occurrence

### 3. 요구사항 분석

본 장에서는 본 제안방식이 만족해야 하는 요구사항에 대해 설명한다. 본 제안방식에서 만족해야 하는 요구사항은 Tottering 현상 방지, 탐색량 감소, 탐색 시간 감소가 존재한다. 이에 대한 자세한 설명은 다음과 같다.

- **Tottering 현상 방지:** Tottering 현상은 두 노드를 반복적으로 오가는 탐색 패턴으로, 그래프 구조 분석에 불필요한 영향을 주고 탐색의 효율성을 저하시킨다. 따라서 Tottering 현상을 원천적으로 방지해야 한다. 이를 위해 제안방식은 탐색 과정에서 이웃 노드를 탐색 후 Tottering 현상을 탐지하는 것이 아닌, 이웃 노드를 선정한 후, 즉 이웃 노드 탐색 전에 Tottering 현상을 탐지해야 한다. 이를 위해 큐에 탐색 노드를 저장하고, 이웃 노드 선정 시 이전 탐색 노드와 비교하여 Tottering 현상을 효과적으로 방지해야 한다.
- **탐색량 감소:** Tottering 현상을 차단하는 것은 곧 탐색 과정에서 불필요하게 생성되는 반복 경로를 제거한다는 의미이며, 이는 전체 탐색량의 감소로 이어진다. 제안방식은 Tottering이 발생한 무의미한 탐색을 사전에 탐지하여 탐색량을 최소화해야 한다. 특히, 기존 방식처럼 Tottering 경로를 생성한 후 탐지하여 제거하는 방식은 탐색량을 증가시키므로, 제안 방식은 탐색 전에 불필요한 경로를 차단함으로써 탐색량을 최소화해야 한다.
- **탐색 시간 감소:** Tottering 현상을 사전에 차단하고, 탐색량을 줄이게 되면 전체 탐색 시간 또한 감소하게 된다. 궁극적으로 제안 방식은 기존 Yoon et al.[17] 방식 대비 탐색 횟수와 탐색 시간 모두에서 우수한 성

능을 보여야 하며, 실험을 통해 탐색 시간 및 탐색 경로 수에 대한 수치적 성능 지표를 통해 이를 검증할 수 있어야 한다.

### 4. 제안방식

본 장에서는 본 논문에서 제안하는 Tottering 현상을 방지하는 새로운 k-Walk 탐색 기법을 설명한다. 본 제안방식은 큐를 활용해 Tottering 현상을 탐지하는 것은 기존 방식과 동일하나, 기존 방식과의 차별점으로, Tottering 현상 발생 전에 Tottering 현상 탐지를 수행하여 효과적으로 Tottering 현상 발생을 방지할 수 있다.

본 제안방식에 대한 알고리즘은 아래 그림 2, 그림 3과 같다. 그림 2는 본 제안방식의 전체적인 동작 과정을 나타낸 알고리즘이다. 이에 대한 동작 과정은 다음과 같다.

- Step 1. 그래프  $G$ 에 대한 정보와 탐색 거리  $k$ 를 입력한다. 여기서  $V$ 는 그래프의 노드 집합,  $E$ 는 그래프의 엣지 집합을 의미한다.
- Step 2. 탐색 경로 집합  $W$ 를 초기화 한다.
- Step 3. 그래프에 존재하는 모든 노드  $v$ 에 대해 그래프 탐색을 진행한다.
- Step 4. 시작 노드  $v$ 를 설정하고, 사전 그래프 탐색을 수행한다.
- Step 5. 모든 노드를 시작 노드로 선택하는 탐색이 종료되면, 탐색한 모든 경로가 저장된 집합  $W$ 를 반환한다.

---

#### Algorithm 1 Pre-emptive Tottering Detection k-Walk Algorithm $k$

**Input:** Graph  $G = (V, E)$ , an integer walk length  $k \geq 0$

**Input:** A list  $W$  of all non-tottering walks of length  $k$

```

1:  $W \leftarrow \emptyset$ 
2: for all vertex  $v \in V$  do
3:    $P_{init} \leftarrow [v]$ 
4:   FINDNONTOTTERINGWALKS( $G, k, v, P_{init}, \text{null}, W$ )
5: end for
6: return  $W$ 

```

---

그림 2. 제안방식 알고리즘  
Fig. 2. Proposed scheme algorithm

```

Algorithm 2 FindNonTotteringWalks
1: if  $k_{rem} = 0$  then
2:   Add  $P$  to the list  $W$ 
3:   return
4: end if
5: for all neighbor  $v_n$  of the current vertex  $v$  in  $G$  do
6:   if  $v_n \neq v_{prev}$  then
7:      $P_{new} \leftarrow P \oplus [v_n]$ 
8:     FINDNONTOTTERINGWALKS( $G, k_{rem} - 1, v_n, P_{new}, v, W$ )
9:   end if
10: end for
    
```

그림 3. 사전 Tottering 탐지 알고리즘  
Fig. 3. Pre-emptive tottering detection algorithm

이와 같은 과정을 통해 본 제안방식은 그래프  $G$ 에 존재하는 모든 노드  $v$ 에 대한 탐색을 수행할 수 있다.

아래 그림 3은 본 제안방식의 핵심 알고리즘인 사전 Tottering 탐지 알고리즘이다. 이에 대한 자세한 동작과정은 다음과 같다.

- Step 1. 만약  $k_{rem} = 0$ 일 경우, 모든 탐색이 종료된 것이므로 현재까지 생성된 탐색을  $W$ 에 저장하고 현재 탐색을 종료한다.
- Step 2.  $k_{rem} > 0$ 일 경우, 추가적인 탐색이 진행되어야 하므로 현재 노드  $v$ 의 임의의 이웃 노드  $v_n$ 을 선택한다.
- Step 3. 만약 선택된 이웃 노드  $v_n$ 이 직전 노드  $v_{prev}$ 와 동일할 경우, 이는 Tottering 현상이 발생한 것이므로 해당 탐색을 종료한다.
- Step 4. 만약  $v_n \neq v_{prev}$ 일 경우, 정상적인 탐색이므로 선정된 이웃 노드  $v_n$ 를  $P$ 에 저장한다.
- Step 5.  $v_n$ 와  $k_{rem} - 1, v$ 등을 입력해 이웃 노드 탐색을 계속해서 수행한다.

이와 같은 과정을 통해 Tottering 현상을 사전에 탐지하며 유의미한 탐색만을 수행할 수 있다.

## 5. 제안방식 분석

본 장에서는 기존 Gaüzère et al.[16], Yoon et al.[17]의 방식과 본 제안방식에 대한 비교 분석을 진행한다. 본 제안방식은 사전 Tottering 탐지를 통해 기존 Yoon et al.의 방식의 한계점을 개선한 제안방식이다.

본 비교분석을 위한 실험 환경은 다음과 같다.

- CPU: Intel(R) Core(TM) i5-13400F
- GPU: NVIDIA GeForce RTX 3050
- RAM: 32.0GB
- OS: Windows 10 Pro

본 실험에 사용된 그래프는 그림 4와 같다. 실험에 사용된 그래프는 각각 노드가 5 ~ 9개 존재하는 완전 그래프와 노드가 10개로 이루어진 연결 그래프이다. 또한, 실험 시 설정한  $k$ 값은 노드의 수 - 1이며, 각 엣지들은 무방향 엣지이다. 이에 대한 실험 결과는 다음과 같다.

### 5.1 탐색량 비교 분석

본 절에서는 기존 방식들과 본 제안방식의 탐색량에 대해 비교한다. 탐색량은 경로 탐색을 위한 재귀 함수 호출 횟수를 의미한다.

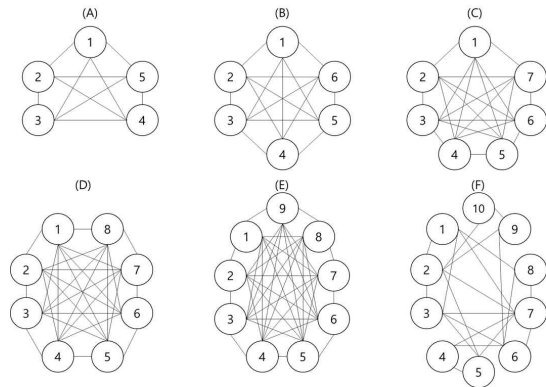


그림 4. 실험에 사용된 그래프 예시  
Fig. 4. Example of graphs used in the experiment

표 1. 탐색량 비교분석표

Table 1. Exploration effort comparison table

그래프	Gaüzère et al.	Yoon et al.	제안방식
(A)	1,705	1,065	805
(B)	23,436	12,786	8,530
(C)	391,909	196,861	164,059
(D)	7,686,400	3,657,816	3,135,280
(E)	172,565,649	79,060,113	69,177,609
(F)	12,135,582	2,026,276	1,582,138

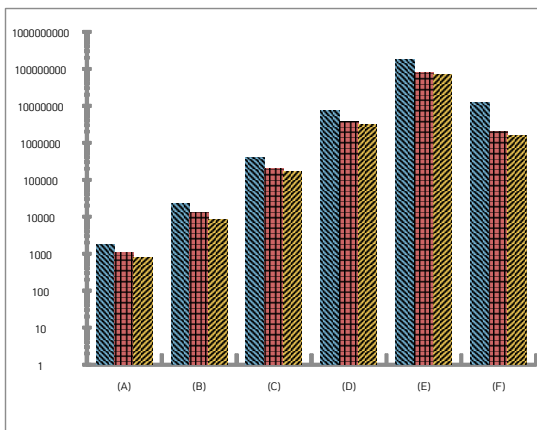


그림 5. 탐색량 비교 그래프

Fig. 5. Exploration effort comparison graph

본 제안방식과 기존 방식들에 대한 탐색량 비교분석표는 표 1, 이를 그래프로 나타낸 그림은 그림 5와 같다.

실험 결과 본 제안방식이 모든 그래프에 대해 탐색을 진행할 때 기존 방식들보다 낮은 탐색량을 보였다. 이는 본 제안방식이 Tottering 현상 탐지를 탐색 후가 아닌, 탐색 전에 시행하므로 Tottering 현상이 발생한 불필요한 탐색을 진행하지 않기 때문이다.

특히 그래프의 노드와 엣지 수가 많아질수록 본 제안방식은 기존 방식에 비해 더욱 효과적으로 탐색량을 감소시킬 수 있다.

## 5.2 탐색 시간 비교 분석

본 절에서는 기존 방식들과 본 제안방식의 탐색 시간에 대해 비교한다. 탐색 시간은 알고리즘을 10회 반복하여 걸린 시간의 평균값을 측정하였다. 또한, 탐색 시간은 소수점 네 번째 자리에서 반올림하여 표시하였다.

본 제안방식과 기존 방식들에 대한 탐색 시간 비교분석표는 아래 표 2, 이를 그래프로 나타낸 그림은 그림 6과 같다.

표 2. 탐색 시간 비교분석표 (단위: 초)

Table 2 Exploration time comparison table (sec)

그래프	Gaüzère et al.	Yoon et al.	제안방식
(A)	0.002 (±0.0008)	0.001 (±0.0001)	0.001 (±0.0002)
(B)	0.019 (±0.001)	0.015 (±0.001)	0.011 (±0.003)
(C)	0.370 (±0.014)	0.227 (±0.004)	0.181 (±0.004)
(D)	8.316 (±0.180)	4.249 (±0.119)	3.592 (±0.043)
(E)	182.844 (±7.803)	91.189 (±1.498)	78.373 (±0.679)
(F)	12.678 (±0.248)	2.369 (±0.028)	1.847 (±0.026)

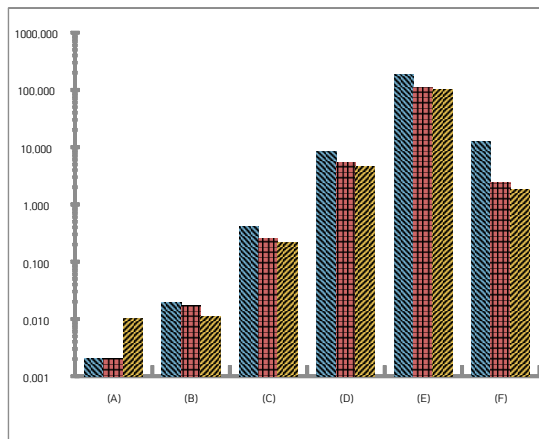


그림 6. 탐색 시간 비교 그래프

Fig. 6. Exploration time comparison graph

본 제안방식은 그래프가 복잡하지 않은 그래프 (A)에 대해서는 큰 실행 시간 차이를 보이지 않지만, 노드와 엣지가 많은 복잡한 그래프에 대해서는 큰 차이를 보인다.

특히, 그래프가 복잡해짐에 따라 본 제안방식은 기존 방식들보다 보다 빠른 시간 내에 그래프 탐색을 수행할 수 있다.

## 6. 결론

그래프 편집 거리 알고리즘의 NP-Hard 문제로 인해 다양한 휴리스틱 알고리즘이 연구되었다. 다양한 기법 중 Random Walk 기법은 구현이 쉽고 높은 유연성으로 인해 다양한 환경에서 활용되어 왔다.

그러나 Gaüzère et al.[16]와 같은 기존 Random Walk 기법은 Tottering 현상으로 인해 탐색 효율성이 크게 저하되는 한계점이 존재한다. 이를 해결하기 위해 Yoon et al.[17]의 기법은 Tottering 현상 발생을 감소시키는 탐색 기법을 제안했지만 Tottering 현상을 완벽히 방지하지 못하는 한계점이 존재했다.

본 논문에선 이를 해결하고자 Tottering 현상을 완벽히 방지하는 개선된 k-Walk 기법을 제안했다. 본 제안방식은 기존 Yoon et al.[17]의 기법과 달리 탐색 전 Tottering 현상을 탐지하여 Tottering 현상 발생 전 이를 탐지할 수 있다.

이를 통해 본 제안방식은 기존 Yoon et al.[17]의 기법보다 효과적으로 그래프 탐색을 수행할 수 있다. 추후 이를 활용해 대규모 소스코드 유사도 검사 환경, 단백질, DNA 등 분야에서 그래프 구조적 유사도 검사 시, 보다 효율적인 유사도 검사를 수행할 수 있을 것으로 사료된다.

본 연구는 그래프 구조 기반 소스코드 유사도 검사 전 그래프 탐색에 활용되는 탐색 기법에 대

한 연구로써 향후 본 제안방식을 적용하여 실제 그래프 구조 기반 유사도 검사를 수행하여 효율성을 개선을 증명하는 연구를 수행하고, 대규모 그래프 환경에서 더욱 효율적인 그래프 탐색을 위해 효율성을 개선하는 연구가 필요할 것으로 사료된다.

문화체육관광부 및 한국콘텐츠진흥원의 2025년도 SW저작권 생태계 조성 기술개발 사업으로 수행되었음 (과제명 : 클라우드 서비스 활용 구축 형태별 대규모 소프트웨어 라이선스 검증 기술개발, 과제번호 : RS-2023-00224818, 기여율: 100%)

## 참고 문헌

- [1] R. Sousa, S. Silva, C. Pesquita, "Explainable Representations for Relation Prediction in Knowledge Graphs", ArXiv, abs/2306.12687, 2023, DOI: 10.24963/kr.2023/62.
- [2] V. Romanov, V. Ivanov, G. Succi, "Approaches for Representing Software as Graphs for Machine Learning Applications", 2020 International Computer Symposium (ICS), pp.529-534, 2020, DOI: 10.1109/ICS51289.2020.00109.
- [3] J. Wallnig, L. Brun, B. Gaüzère, S. Bougleux, F. Yger, D. Blumenthal, "A Differentiable Approximation of the Graph Edit Distance", Lecture Notes in Computer Science, pp.1-10, 2024, DOI: 10.1007/978-3-031-80507-3\_1.
- [4] N. Boria, D. Blumenthal, S. Bougleux, L. Brun, "Improved Local Search for Graph Edit Distance", Pattern Recognition Letters, 128, pp.29-36, 2019, DOI: 10.1016/j.patrec.2019.10.028.

- [5] G. Zhang, S. Lin, X. Wang, T. Huang, X. Hu, L. Zou, “Experimental Comparison of Graph Edit Distance Computation Methods”, 2023 24th IEEE International Conference on Mobile Data Management (MDM), pp.303-308, 2023, DOI: 10.1109/MDM58254.2023.00056.
- [6] D. Blumenthal, N. Boria, J. Gamper, S. Bougleux, L. Brun, “Comparing Heuristics for Graph Edit Distance Computation”, The VLDB Journal, 29(2), pp.419-458, 2020, DOI: 10.1007/s00778-019-00544-1.
- [7] M. Darwiche, R. Raveaux, D. Conte, V. T'kindt, “A Local Branching Heuristic for the Graph Edit Distance Problem”, Lecture Notes in Computer Science, pp.194-202, 2017, DOI: 10.1007/978-3-319-75193-1\_24.
- [8] L. Jia, V. Tognetti, L. Joubert, B. Gaüzère, P. Honeine, “A Study on the Stability of Graph Edit Distance Heuristics”, Electronics, 11(20), pp.3312, 2022, DOI: 10.3390/electronics11203312.
- [9] F. Bause, C. Permann, N. Kriege, “Approximating the Graph Edit Distance with Compact Neighborhood Representations”, ArXiv, abs/2312.04123, pp.300-318, 2023, DOI: 10.48550/arXiv.2312.04123.
- [10] V. Millán, V. Cholvi, L. López, A. Fernández, “Resource Location Based on Partial Random Walks in Networks with Resource Dynamics”, Proceedings of the ACM Symposium on Applied Computing, pp.26-31, 2012, DOI: 10.1145/2414815.2414821.
- [11] A. Mian, R. Beraldi, R. Baldoni, “On the Coverage Process of Random Walk in Wireless Ad Hoc and Sensor Networks”, The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS), pp.146-155, 2010, DOI: 10.1109/MASS.2010.5663954.
- [12] B. Luo, W. Kropatsch, J. Jiang, C. Zhong, “Combining Bipartite Matching and Exact Neighborhood Substructure Distance”, Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science, 2019.
- [13] V. Carletti, B. Gaüzère, L. Brun, M. Vento, “Approximate Graph Edit Distance Computation Combining Bipartite Matching and Exact Neighborhood Substructure Distance”, Lecture Notes in Computer Science, pp.188-197, 2015, DOI: 10.1007/978-3-319-18224-7\_19.
- [14] B. Gaüzère, S. Bougleux, K. Riesen, L. Brun, “Approximate Graph Edit Distance Guided by Bipartite Matching of Bags of Walks”, Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), pp.73-82, Springer, 2014.
- [15] S. Yoon, D. Seo, S. H. Kim, I. Y. Lee, “A Study on k-walk Generation Algorithm to Prevent the Tottering in Graph Edit Distance Heuristic Algorithms”, Journal of Combinatorial Optimization, 49(1), pp.9, 2025, DOI: 10.1007/s10878-024-01236-5.
- [16] F. Aziz, R. Wilson, E. Hancock, “Backtrackless Walks on a Graph”, IEEE Transactions on Neural Networks and Learning Systems, 24(6), pp.977-989, 2013, DOI: 10.1109/TNNLS.2013.2248093.
- [17] L. Huang, A. Graven, D. Bindel, “Density of States Graph Kernels”, ArXiv, abs/2010.11341, 2020, DOI: 10.1137/1.9781611976700.33.

저 자 소 개



윤성철(SeongCheol Yoon)

2023.2 순천향대학교 컴퓨터소프트웨어공  
학과 졸업  
2025.2 순천향대학교 소프트웨어융합학과  
석사  
2025.3-현재: 순천향대학교 소프트웨어융합  
학과 박사과정  
<주관심분야> 정보보호, 암호학, 휴리스틱  
알고리즘



김 수 현(Su-Hyun Kim)

2010.2 순천향대학교 정보기술공학부 졸업  
2012.2 순천향대학교 컴퓨터학과 석사  
2016.2 순천향대학교 컴퓨터학과 박사  
2016.3-2018.9 순천향대학교  
IoT보안연구센터 연구교수  
2018.10-2023.2 정보통신산업진흥원 책임  
2023.3-현재 : 순천향대학교  
컴퓨터소프트웨어공학과 교수  
<주관심분야> 암호 프로토콜, IoT 보안,  
클라우드 컴퓨팅 보안



이 임 영(Im-Yeong Lee)

1981.2 홍익대학교 전자공학과 졸업  
1986.2 오사카대학 통신공학전공 석사  
1989.2 오사카대학 통신공학전공 박사  
1989-1994 한국전자통신연구원 선임연구원  
1994-현재 순천향대학교  
컴퓨터소프트웨어공학과 교수  
<주관심분야> 암호이론, 암호 프로토콜,  
컴퓨터보안