

그래프 동형 문제를 위한 최신 알고리즘 비교

Comparison of State-of-the-Art Algorithms for Graph Isomorphism

김민형, 이민제, 송시우, 박근수

서울대학교

June 20, 2023

그래프 동형 문제

그래프 G, H 에 대하여

- $V(G)$ 에서 $V(H)$ 로 가는 전단사 함수 f 가 존재
- $u \in V(G)$ 의 레이블과 $f(u)$ 의 레이블이 같음
- (u, v) 가 $E(G)$ 에 속하는 것과 $(f(u), f(v))$ 가 $E(H)$ 에 속하는 것이 필요충분조건

일 때 G 와 H 는 동형

그래프 동형 문제

- 안정된 색칠 (stable coloring)
 - 색이 같은 임의의 두 정점에 대해 인접한 정점들의 색 구성이 동일
 - 색 개선 (color refinement)을 통해 $O((E + V)\log V)$ 시간에 계산 가능
 - 안정된 색칠이 같음은 동형을 보장하지 않음
- 그래프 표준형 (graph canonical form)
 - 그래프 표준형이 같음은 동형을 보장함

최신 알고리즘

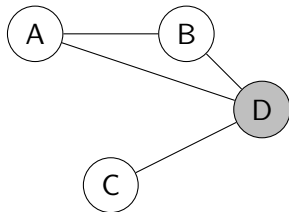
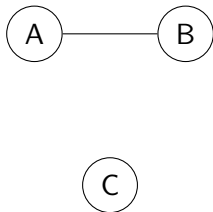
- CRaB
 - G. Gu et al. (2021) "Scalable Graph Isomorphism: Combining Pairwise Color Refinement and Backtracking via Compressed Candidate Space," in ICDE
- DviCL
 - C. Lu et al. (2021) "Graph Iso/Auto-morphism: A Divide-&-Conquer Approach," in SIGMOD

- 표준형을 구하는 대신, 백트래킹(backtracking)을 통해 문제 해결
- 안정된 색칠을 이용하여 탐색공간을 줄임
- 색이 같은 정점이 한 쌍만 존재하는 경우를 먼저 대응시킨 뒤 백트래킹 적용

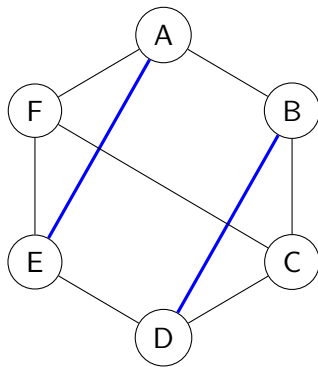
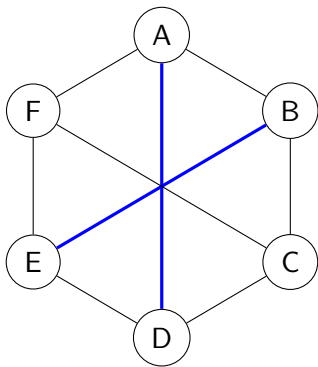
- 안정된 색칠을 기반으로 분할 정복 기법 사용
- 나뉜 부분 그래프의 크기가 1이 아닐 때에는 기존 알고리즘 (nauty, Traces) 사용
- 표준형을 구한 뒤 비교하는 방법으로 그래프 동형을 부차적으로 판별

실험 방법

- 주어진 그래프를 연결 그래프로 변환
 - 연결 그래프가 아닐 때, 정점을 하나 추가하여 모든 정점과 연결
 - 이를 통해 색칠에 영향 없이 연결 그래프로 만들 수 있음



실험 방법

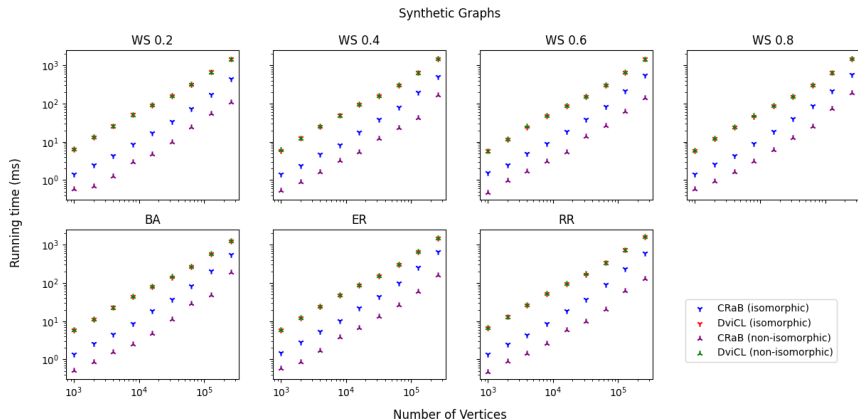


실험 방법

- 생성된 쿼리에 대해 CRaB와 DviCL을 각각 실행하여 소요시간 측정
- DviCL의 경우 표준형을 생성한 뒤 비교하는 과정까지 소요시간에 포함

실험 결과 - 합성 그래프

● 정점 개수에 따른 실행 시간



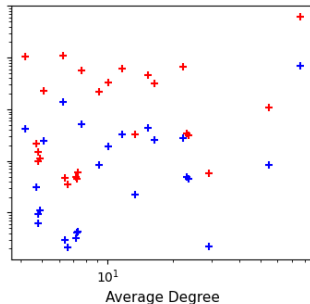
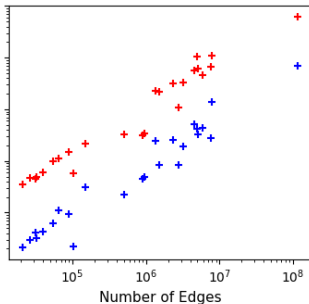
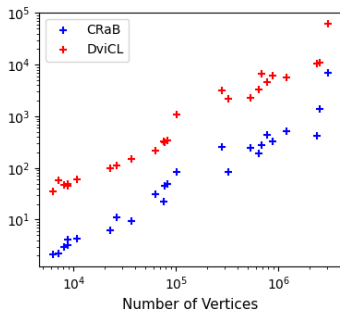
실험 결과 - 합성 그래프

- 두 알고리즘 모두 실행 시간이 정점 개수에 대해 선형에 가깝게 나타남
- CRaB가 DviCL에 비해 빠르게 동작

실험 결과 - 실제 그래프

- 비동형 쿼리에 대한 실험 결과

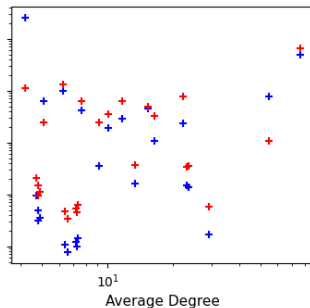
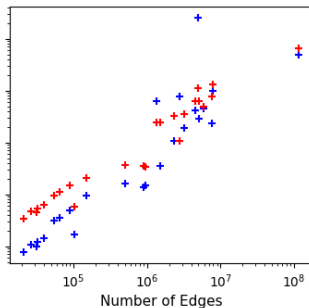
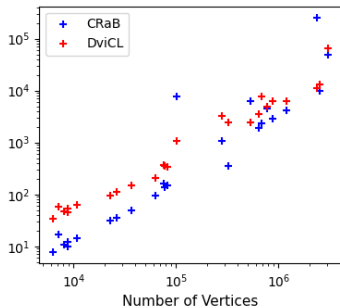
Real Graphs, Non-isomorphic Queries



실험 결과 - 실제 그래프

- 동형 쿼리에 대한 실험 결과

Real Graphs, Isomorphic Queries



실험 결과 - 실제 그래프

- 정점 개수와 간선 개수가 증가함에 따라 수행 시간도 대체적으로 증가
- 평균 차수는 수행 시간에 영향을 주지 않음
- 대체로 CRaB가 DviCL에 비해 빠르게 동작

- 비동형 쿼리에서 CRaB가 DviCL에 비해 매우 빠르게 동작
 - CRaB는 안정된 색칠이 서로 다를 때 즉시 종료
 - DviCL은 표준형을 모두 생성한 후 종료
- 동형 쿼리에서 CRaB가 DviCL에 비해 빠르게 동작
 - 동형 쿼리에서는 항상 안정된 색칠이 같기 때문에 CRaB에서 추가적인 시간 소요

- Buzznet, Delicious, WikiTalk에서 CRaB의 실행 속도가 특징적으로 느림
 - DviCL: 최악의 경우 시간 복잡도 (worst case time complexity)가 정점에 대해 지수적
 - CRaB: 최악의 경우 정점 개수에 대해 지수 시간이 걸리는 백트래킹 기법 사용
- 정점과 간선 수가 적을 때에는 CRaB가 DviCL에 비해 매우 빠르게 동작
- 정점과 간선 수가 커짐에 따라 CRaB와 DviCL 사이의 격차 감소

결론

- 합성 그래프와 실제 그래프 모두에서 CRaB가 대체적으로 DviCL보다 빠르게 동작
- 비동형인 쿼리에서 CRaB가 빠르게 종료
- 정점과 간선 수에 대해 선형에 가까운 실행 시간
- 특징적인 예외 데이터: Buzznet, Delicious, WikiTalk

감사합니다.