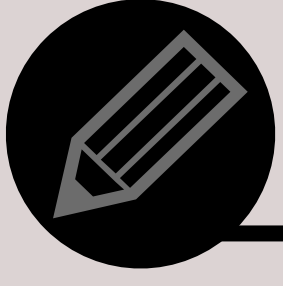
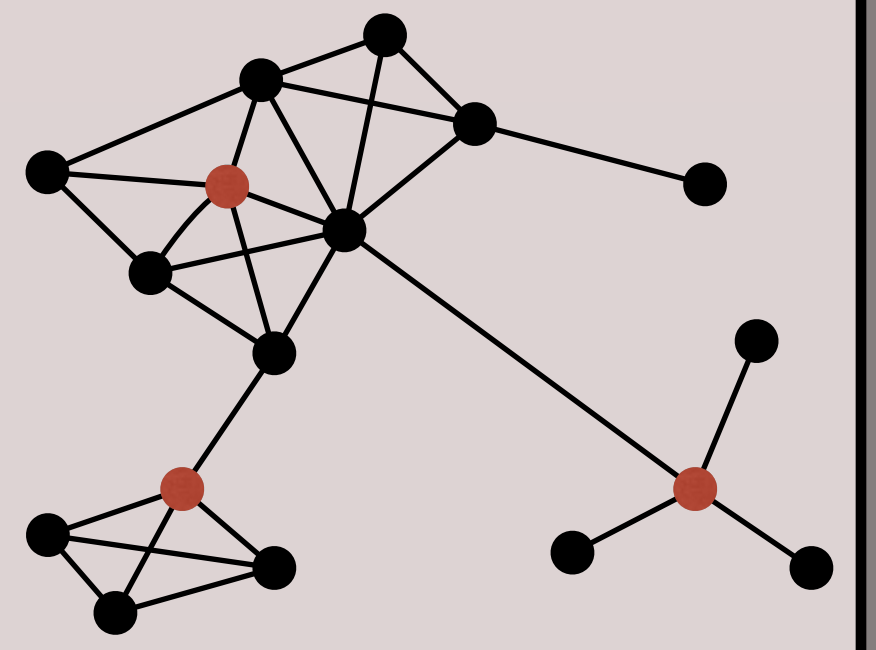




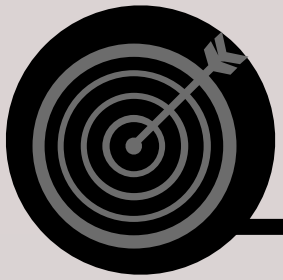
Diffusion Time Minimization on Social Networks

김성창, 김정훈, 유건식
지도 교수님: 양성봉
연세대학교 공과대학 컴퓨터과학 전공



Introduction: 확산시간 최소화

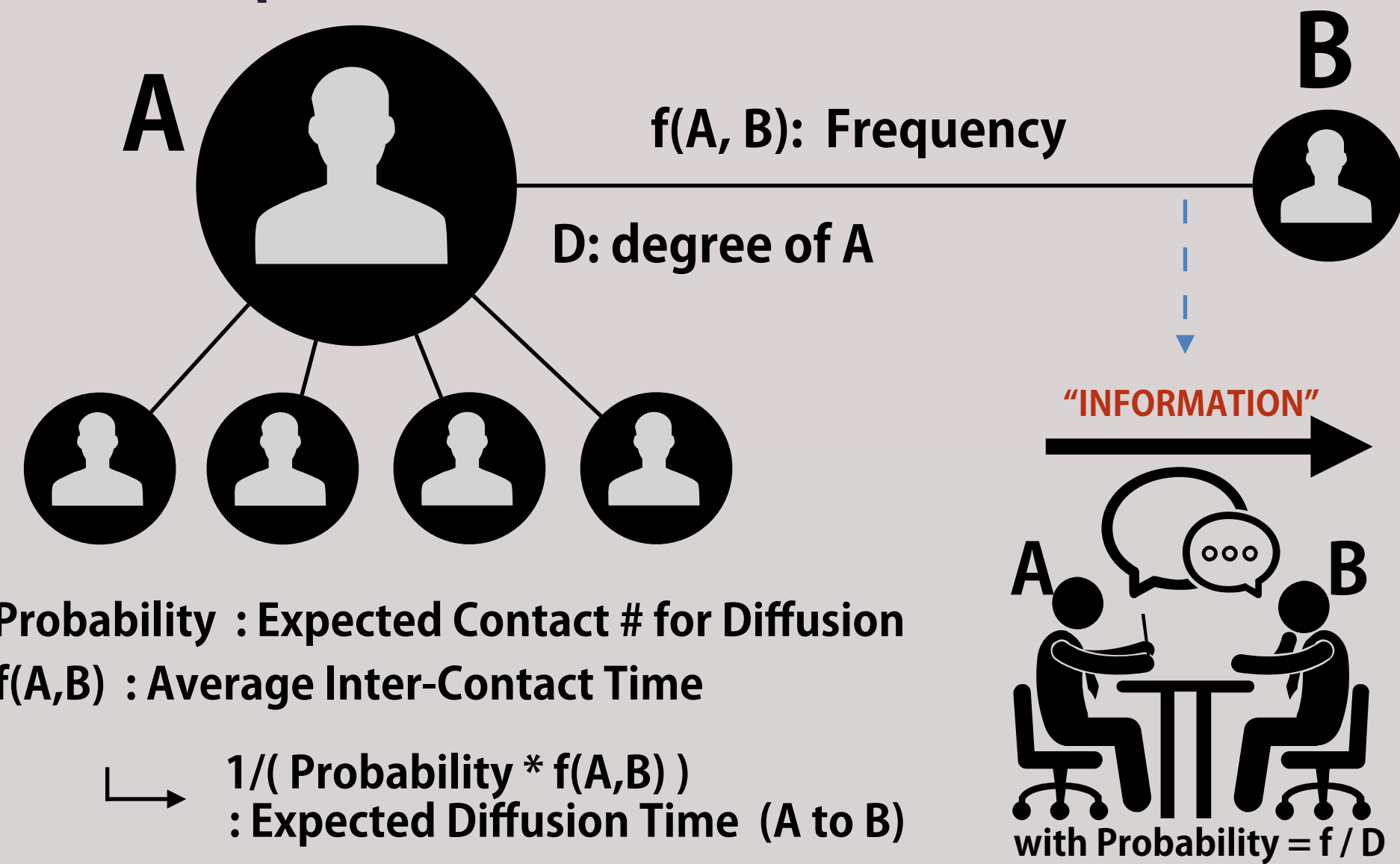
네트워크 연결을 제공하는 SNS 서비스가 증가함에 따라 이로부터 생성되는 네트워크형 데이터를 사용하고자하는 시도가 있어왔다. 이에 따라 본 연구는 고정된 네트워크 상에서 정보 확산(Information Diffusion)의 효과를 극대화 할 수 있는 최초 노드 집합의 선택 알고리즘에 대해 연구했다. 기존 커뮤니티(Community Detection) 기반의 알고리즘이 가지고 있는 계산 비용 문제를 해결하면서, 중심성(Centrality) 척도가 복수 노드 선별에서 갖는 단점을 보완하기 위해 트리 구조의 관점에서 본 문제에 접근 했다.



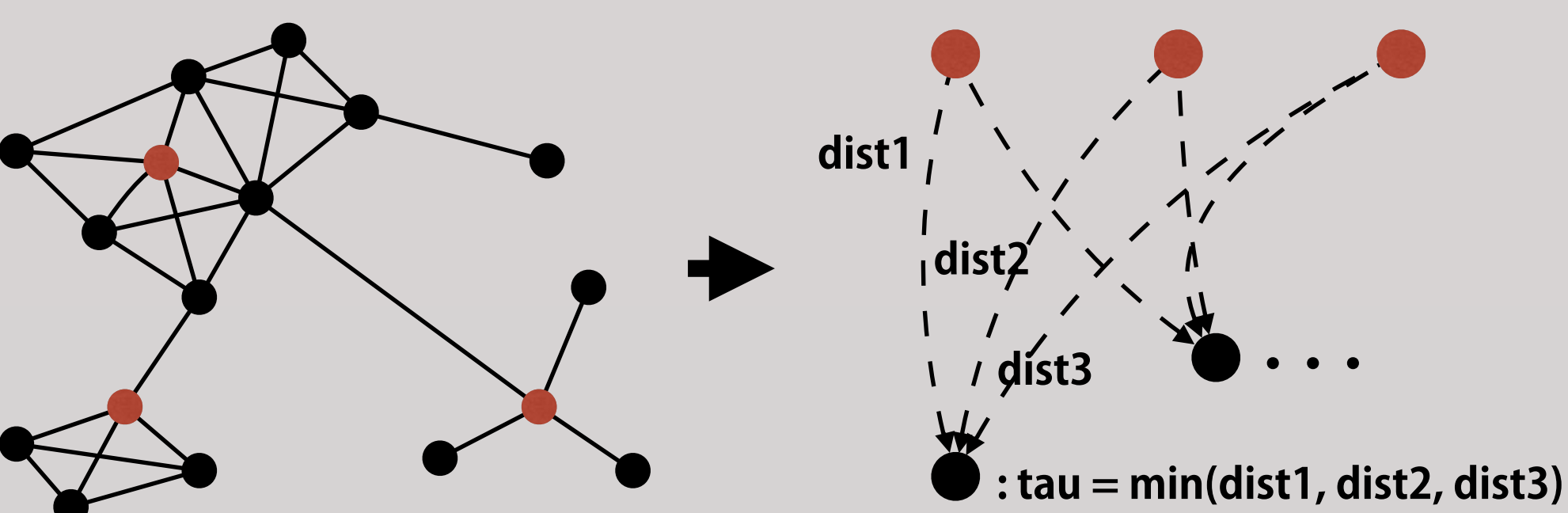
Assumption & Problem Statement

정보 확산 행태에 대한 가정으로 본 연구는 확률적 확산 모형을 택했으며, 이는 1:1 접촉에 의해 정보확산이 이루어진다는 가정이다. 이 가정을 통해 정보확산 문제를 연결 사이에 확정적인 비용(Deterministic Cost)이 존재하는 그래프 문제로 변경했고 이로써 해결하고자했던 확산 시간 최소화 문제에 트리와 최단거리 알고리즘을 응용할 수 있게 되었다.

1. Assumption: Probabilistic Diffusion Model



2. Diffusion Time Minimization



MINIMIZE Complete Diffusion Time = $\max(\tau \text{ for all nodes })$



Our Approach: Pseudo Code

Algorithm 2 Tuning Diffusion Nodes by Root Shifting

```
Input: F
Output: F'
1: for each tree T in F do
2:   flag ← True
3:   repeat
4:     r ← T.getRoot()
5:     D ← {v | v is a direct child of r}
6:     n(1) ← v ∈ D with Max |(u, v, f(v))|
7:     n(2) ← v ∈ D \ {n(1)} with Max |(u, v, f(v))|
8:     if |(r, n(1), f(n(1)))| > |(r, n(2), f(n(2)))| + |(n(1), r)|
9:       then
10:        T.setRoot(n(1))
11:     else
12:       flag ← False
13:   until flag = True
14: end for
```

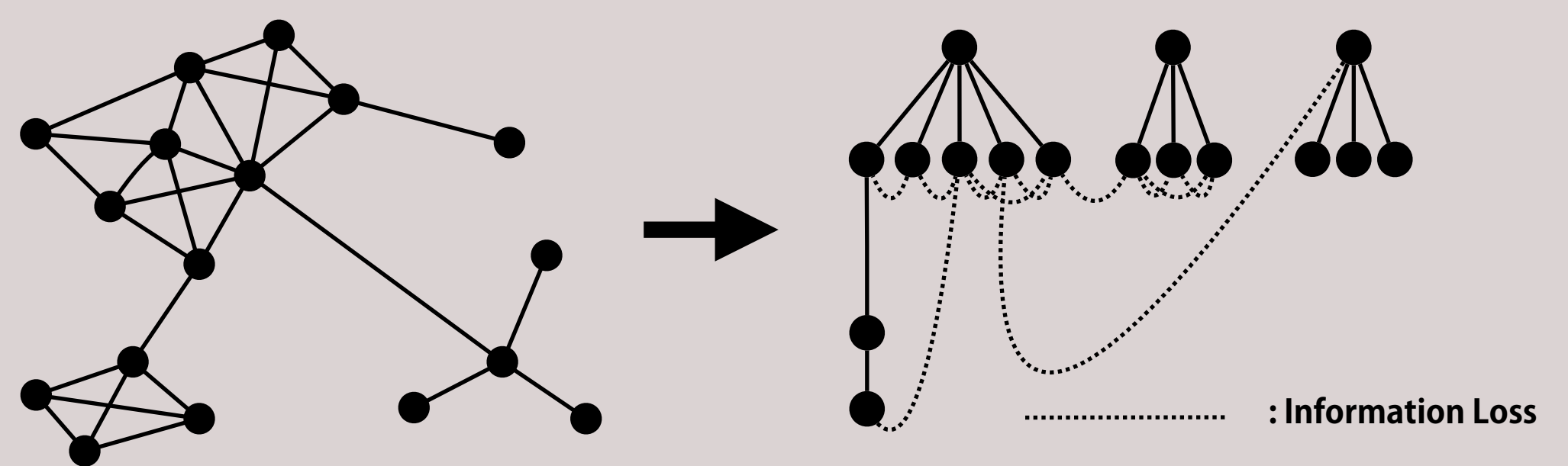
Algorithm 1 Expanding a Forest by Multiple Dijkstra

```
Input: G, K
Output: F
1: for each node r in K do
2:   Queue.enqueue(r)
3:   while Queue is not empty do
4:     u ← Queue.dequeue()
5:     activate u
6:     for each v in Neighbor(u) do
7:       if w(v) > w(u) + t(u, v) then
8:         w(v) ← w(u) + t(u, v)
9:       end if
10:      if v is not activated then
11:        Queue.enqueue(v)
12:      end if
13:    end for
14:  end while
15: end for
```

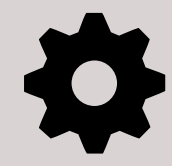


Our Approach: Descriptions

Step 1) Tree Expand

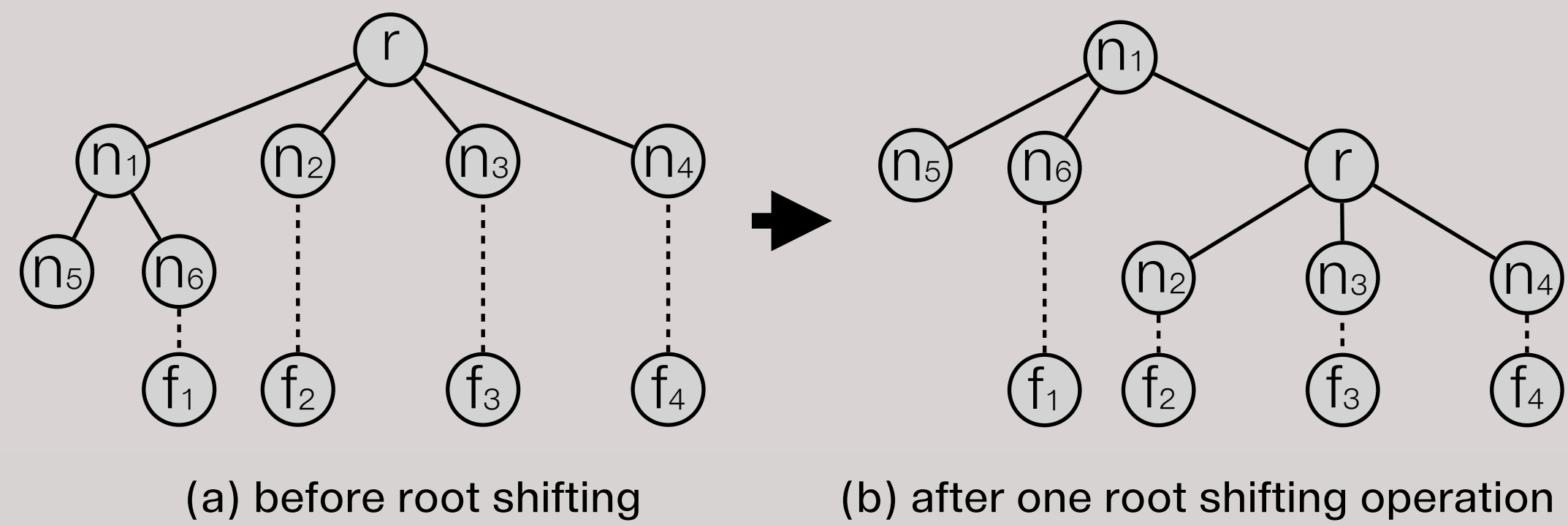


기존의 커뮤니티(Community Detection)나 클러스터링(Clustering) 기반의 방법론이 갖는 가장 큰 문제점은 계산 비용이 매우 높다는 점이다. 본 연구에서는 주어진 문제를 효과적으로 해결하기 위해 주어진 네트워크 구조를 복수의 트리 형태로 재구성하였고, 이 과정에서 다익스트라를 이용하여 정보손실을 최소화하고자 하였다.

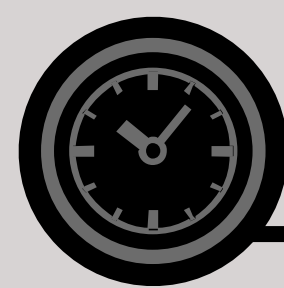


Step 2) Root Tuning

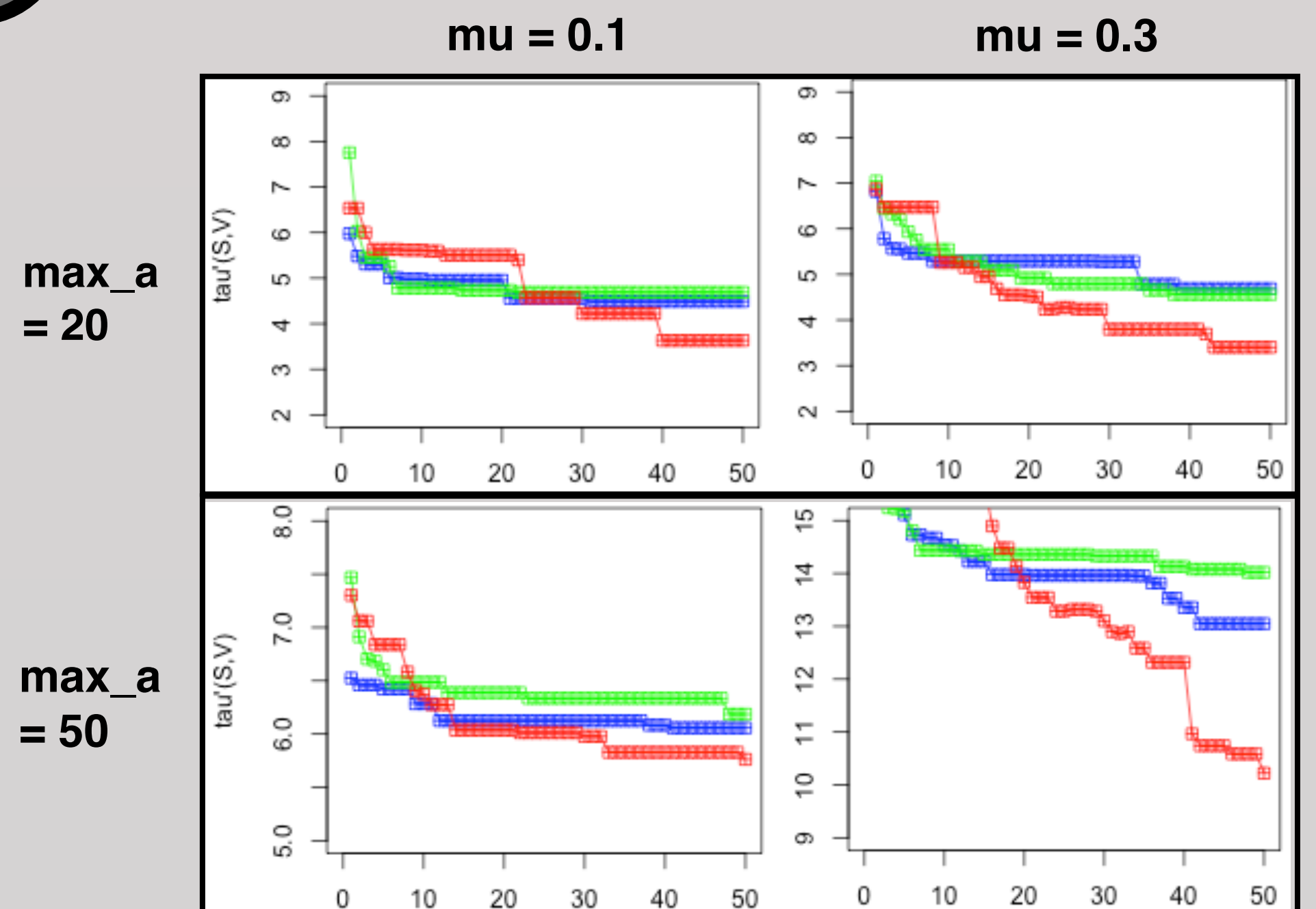
Shifting Condition: $|(r, n[1])| > |(r, n[2])| + t(n[1], r)$



네트워크 분석에 있어, 중심성 척도(Centrality Measure)는 복수의 확산 노드를 선택하는 환경에서 확산 효과가 중첩되는 부분(Overlap)을 적절하게 고려하지 못하기 때문에 본 연구가 해결하려는 문제에 효과적인 노드를 선별하지 못하는 문제가 있었다. 이에 본 연구는 트리를 구성함으로써 주어진 문제를 적절하게 분리해내고 분리된 문제에 대해서 부분 최적화를 하여 전체 문제를 해결하는 방식을 택했다.



Evaluation & Further Study



Reference

- [1] A Lancichinetti, S Fortunato, and F Radicchi, "Benchmark graphs for testing community detection algorithms"
- [2] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities"
- [3] <http://santo.fortunato.googlepages.com/benchmark.tgz>
- [4] Z. Lu, Y Wen and Guohong Cao, "Information Diffusion in Mobile Social Networks: The Speed Perspective"