Revenue Maximization of Influence Diffusion in Social Networks

515030910603　张嘉懿

Final Report

Report outline

① Background

② Related Work

③ Model and Solution

④ Experiments

Report outline

Background

Information Spread over Social Network

- **Diffusion by social interaction**

Virus

Rumor

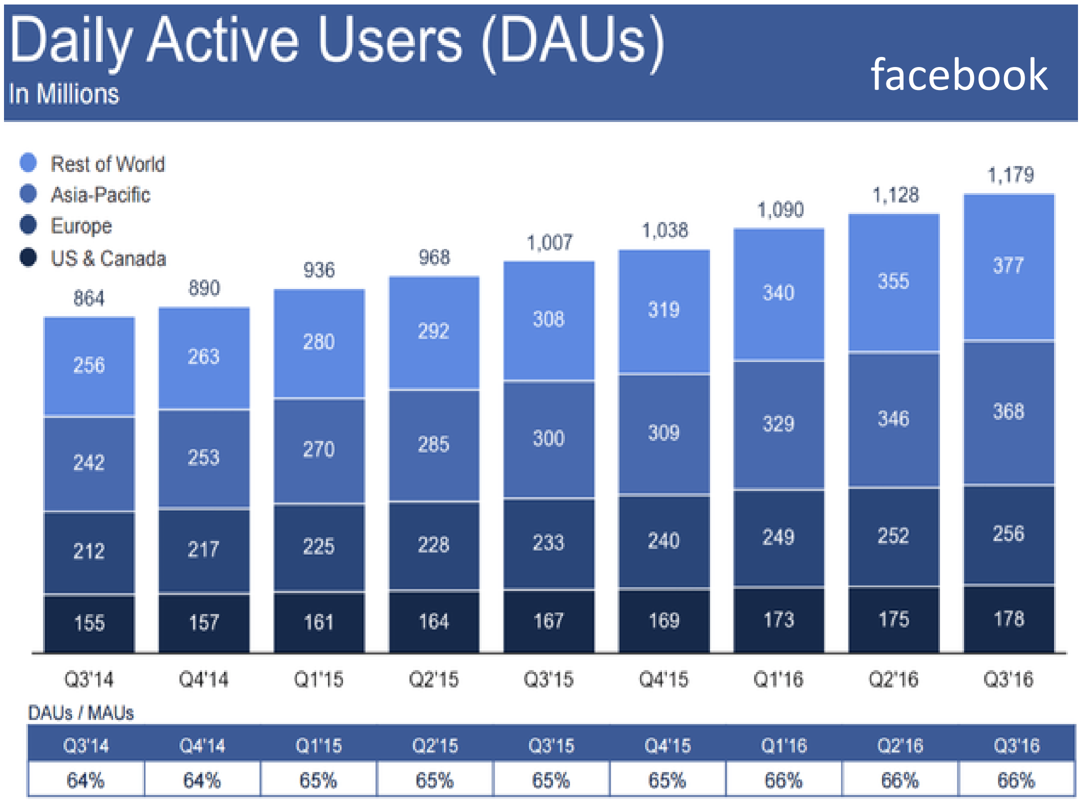
Political Party

- **Online social network of unprecedented scale**

Such as Facebook and Twitter connects a huge number of people at a single time

Background

Advertising opportunity via social network



How to maximization the influence?

Survey: 1,713 companies in USA [Salseforce 2015]

72% will increase cost for social marketing

78% have a dedicated social media team

70% believe social marketing is core

Report outline

1. Background
2. Related Work
3. Model and Solution
4. Experiments

Seeding Problem

New products product promotion

Find people as seed nodes

Spread information in social network

Continue until all information or meet

people get time deadline

Some people are influenced by the information

target

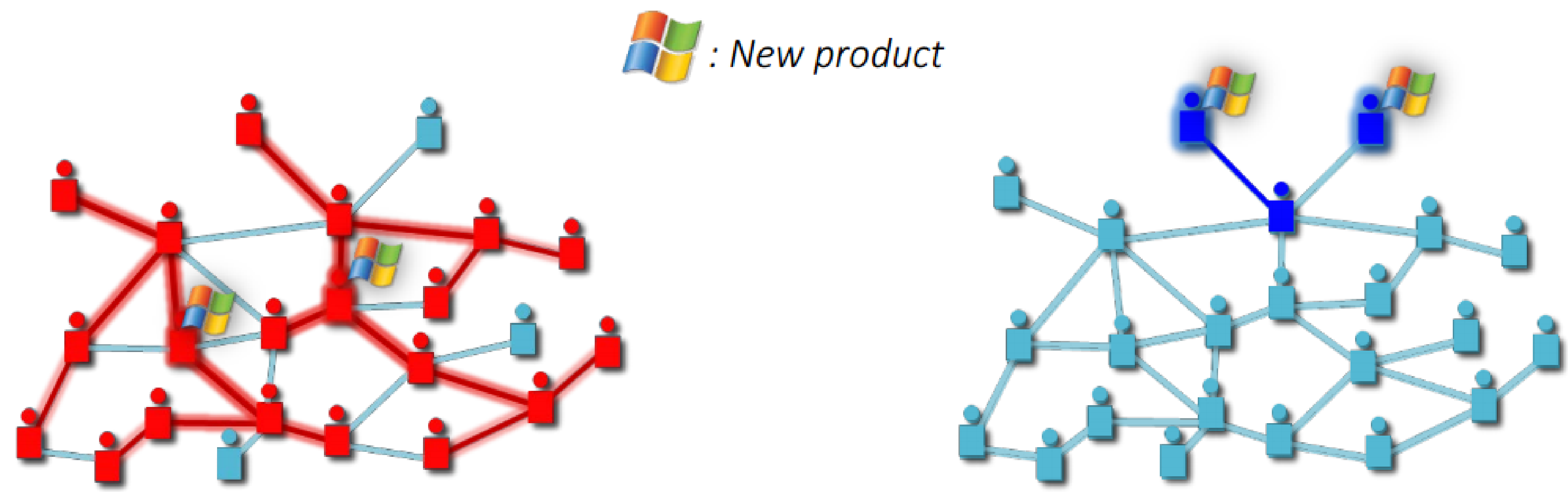
Reach maximum customers

Core problem : how to choose seed set?

Seeding Problem

Influence spreads through seed nodes in the network

Great difference between different seeding strategies.

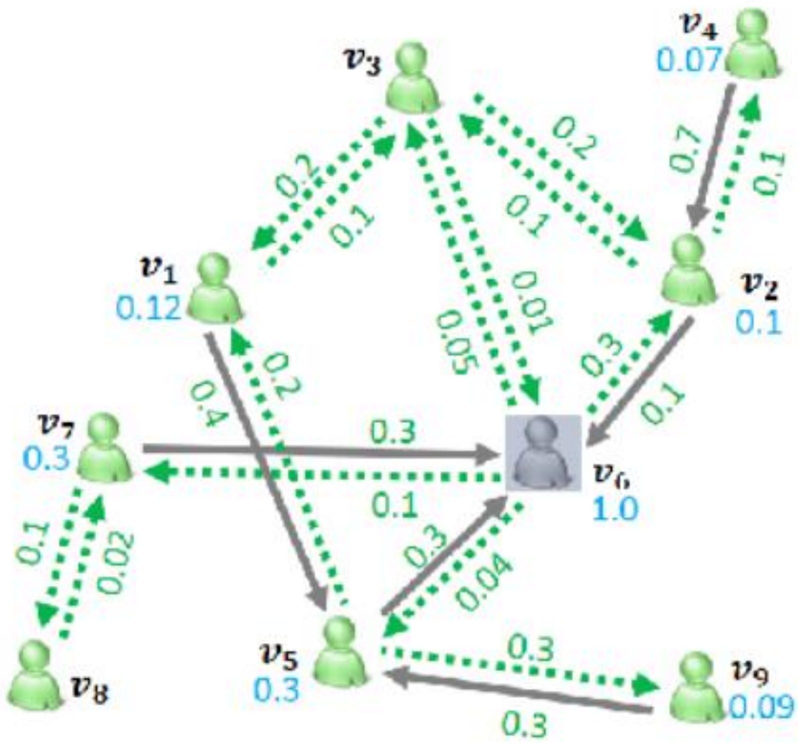


A good seeding influences over 17 people.

A bad seeding influences over only 1 person.

Seeding Problem Model

Social network Structure



- **Consider people as nodes, connection as edges**
- **Social network as Graph G=(V,E)**
  V --- nodes   E --- edges
- **Activation probability or weight on edges**

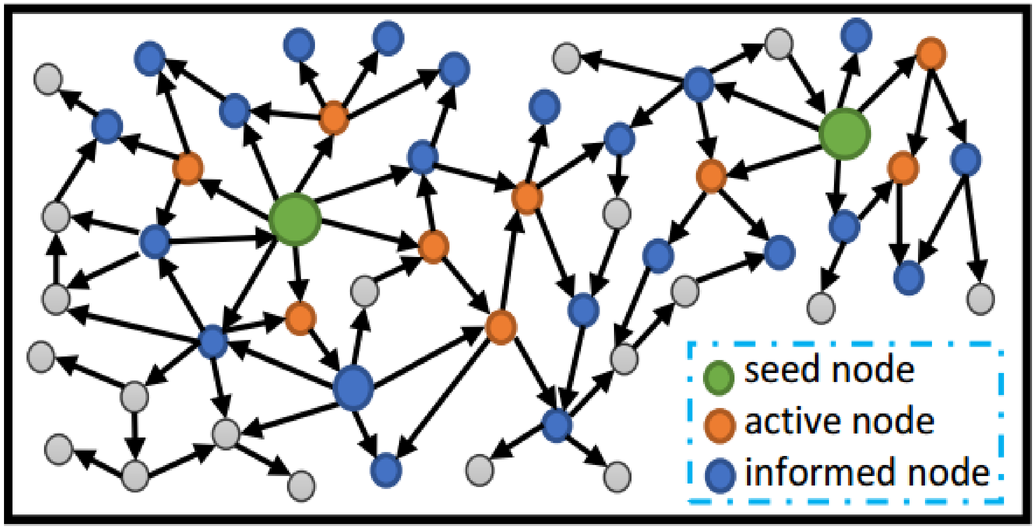
Seeding Problem Model

Two main models

- **Linear threshold model (LT)**
- **Independent cascade model (IC)**

Types of nodes

- **Seed** --- nodes being chosen to start diffuse
- **Active** --- adopting the new information or product
- **Informed** --- get the information but not spread
- **Inactive**



Linear threshold model (LT)

- Node receives information from threshold number of nodes then it will change state and start to diffuse influence.

Independent cascade model (IC)

- Works based on activation probability of other neighbor nodes and node have single chance to get activated.

My consider

- All the existing works focus on the maximization of the total active nodes in the event lifespan or in a limited time period.
- But as a company, their paramount goal is to maximize revenue.

How to maximize revenue in Influence Propagation?

Report outline

Assumptions

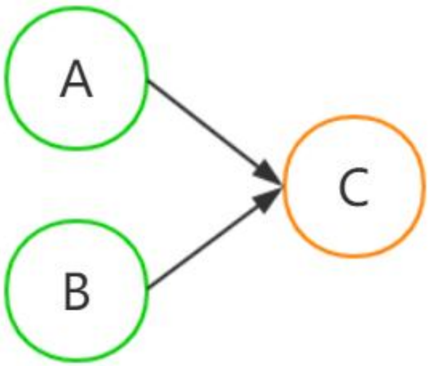
- ●Closed world --- people can only be influenced by other members of the network and information spreads because of informational cascades .

- ●Each seed node has identical cost and the revenues are different according to the type of nodes.

- ●Using IC to express the diffusion of the information in my model.

- ●Actived users immediately become seeds and have the same willingness of diffusion.

Toy Example



- cost(A) > cost(B)

  before : A or B randomly

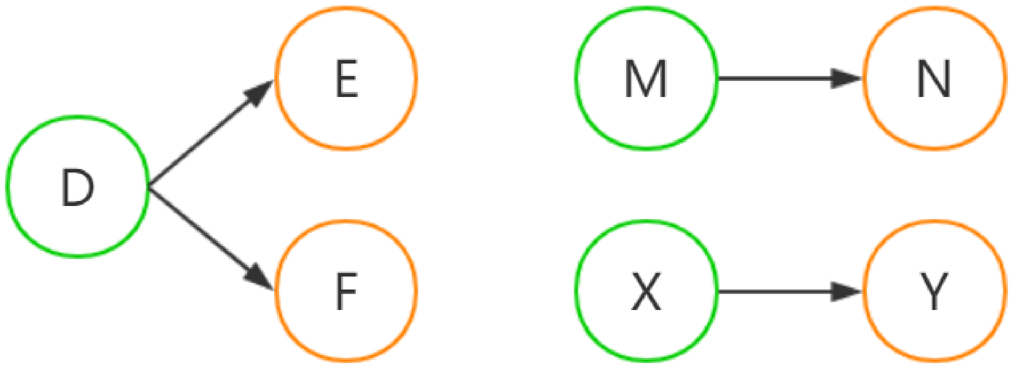
  revenue : choose B

- budget 2 node , payoff = 1
- cost(D) = 5
- cost(M) = cost (X) = 1

  before :

  D and M or X randomly

  revenue : choose M and X



Model

Social network Structure

- **Social network as Graph G=(V,E,P)**

  V --- nodes   E --- edges P --- diffusion probability matrix

- **Node set :**

  S --- seed nodes   A --- active nodes I --- informed nodes

- **Notation**

  Seed cost : c   active node payo p_a **informed node pay** p_i **ff:**

  Revenue : $r(S) = p_a * |A| + p_i * |I| - c * |S|$ v $|S| \leq k$

  (k is the maximum seed set size within budget)

- **Target** $\arg\max_{S} r(S)$

ALGORITHM

Algorithm 1: GeneralGreedy Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev
initialize $S = \emptyset$ and R;
for $i = 1$ *to* k **do**
    for *each node* $n \in V \backslash S$ **do**
        $s_n = 0$;
        for $i = 1$ *to* R **do**
            $s_n + = |Random(S \cup \{n\})|$;
        end
        $s_n = s_n / R$;
    end
    $S = S \cup \{\arg\max_{n \in V \backslash S}\{s_n\}\}$;
end
$rev = r(S)$;
return S,rev;

ALGORITHM

Algorithm 2: ImproveICGreedy Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev

```
initialize S = ∅ and R;
for i = 1 to k do
    for each node n ∈ V\S do
        s_n = 0;
    end
    for i = 1 to R do
        compute G' by removing each edge from G with
        probability 1 − p;
        compute N_{G'}(S);
        compute |N_{G'}(n)| for all n ∈ V;
        for each node n ∈ V\S do
            if n ∉ N_{G'}(S) then
                s_n += |N_{G'}(n)|;
            else

            end
        end
    end
    s_n = s_n/R for all n ∈ V\S;
    S = S ∪ {arg max_{n∈V\S}{s_n}};
end
rev = r(S);
return S,rev;
```

ALGORITHM

Algorithm 3: Revenue Maximization Greedy Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev

initialize $S = \emptyset$;
initialize $max\_r = 0$;
initialize $S' = \emptyset$;
for *each node n in V* **do**
    compute $\Delta(n) = r(n)$;
    $flag_n = 0$;
end
while $|S'| < k$ **do**
    $n = \arg\max_{n \in V \setminus S'} \Delta(n)$;
    if $flag_n == |S'|$ **then**
        $S' = S' \cup n$;
        if $r(S') > max\_r$ **then**
            $S = S'$;
            $max\_r = r(S')$;
        else
            continue;
        end
    else
        compute $\Delta(n) = r(S' \cup n) - r(S')$;
        $flag_n = |S'|$;
    end
end
$rev = max\_r$;
return S,rev;

ALGORITHM

Algorithm 4: Degree Rank Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev

initialize $S = \emptyset$;
initialize $C = \emptyset$;
for *each node n in V* **do**
  $Degree(n) = OutDegree(n)$;
end
while $k > 0$ **do**
  while $|S| < k$ **do**
    $n = \arg\max_{n \in V \setminus S} Degree(n)$;
    $S = S \cup n$;
    $C = C \cup OutNeighbour(n)$;
    for *each node n in* $V \setminus S$ **do**
      $Degree(n) = OutDegree(n) - |C \cap OutNeighbour(n)|$;
    end
  end
  k = k-1;
end
$rev = r(S)$;
return S,rev;

Algorithm 5: Degree Discount Algorithm

Input: seed budget k, $G = (V, E, P)$
Output: seed set S,revenue rev

initialize $S = \emptyset$;
for *each node n in V* **do**
  compute $Degree(n)$;
  $dDgree(n) = Degree(n)$;
  $flag_n = 0$;
end
for $i = 1$ *to* k **do**
  $u = \arg\max_n \{dDegree(n) | n \in V \setminus S\}$;
  $S = S \cup u$;
  for *each neighbour n of u and* $n \in V \setminus S$ **do**
    $flag_n = flag_n + 1$;
    $dDegree(n) = Degree(n) - 2flag_n - (Degree(n) - flag_n)flag_n p$;
  end
end
$rev = r(S)$;
return S,rev;

Report outline

Experiments Setting

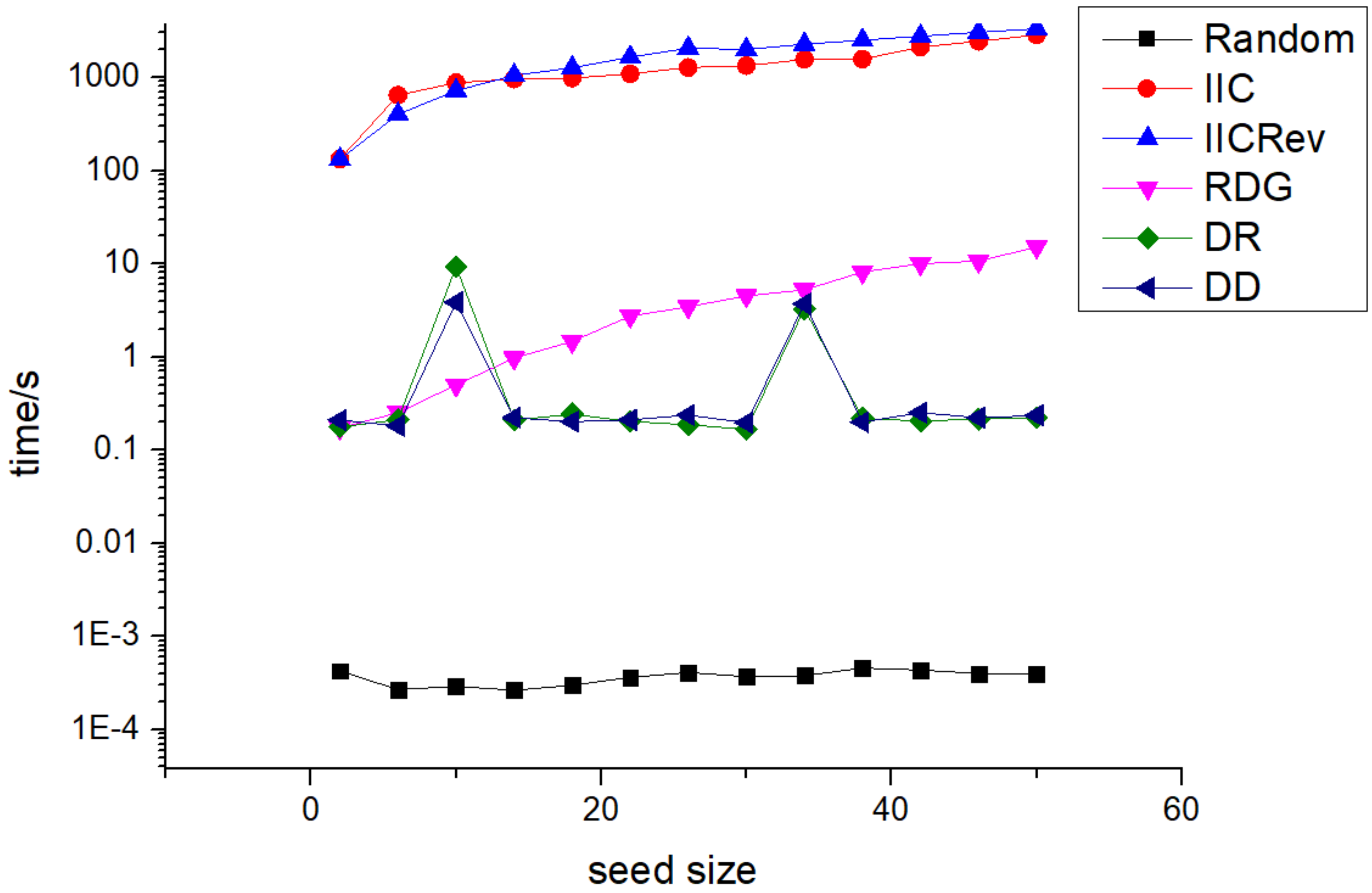
Real - life network

collaboration graph of High Energy Physics – Theory section, from year 1991 to year 2003.
It has 15233 nodes and m = 58891 edges.

Set of algorithms

- **Random ---- baseline comparison, simply select k random nodes in the graph.**
- **IIC ---- The ImprovedICGreedy algorithm in Algorithm 2.**
- **IICRev ---- Change the heuristic function in Algorithm 2 and make it related to the revenue.**
- **RDG ---- Revenue Difference Greedy heuristic algorithm in Algorithm 3.**
- **DR ---- Degree Rank heuristic algorithm in Algorithm 4.**
- **DD ---- Degree Discount heuristic algorithm in Algorithm 5.**

Time Complexity



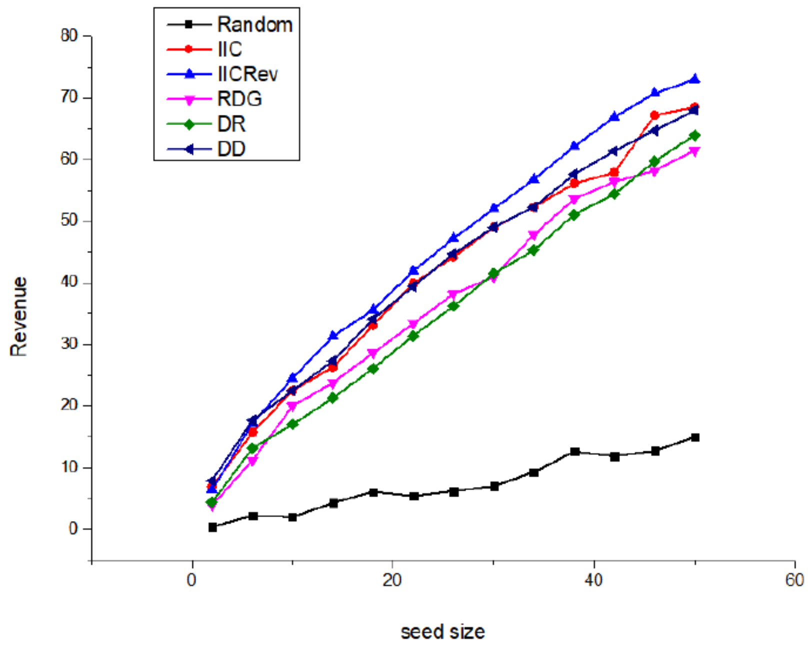
Log y-axis !!!

Random > DD ≈ DR > RDG > IIC > IICRev

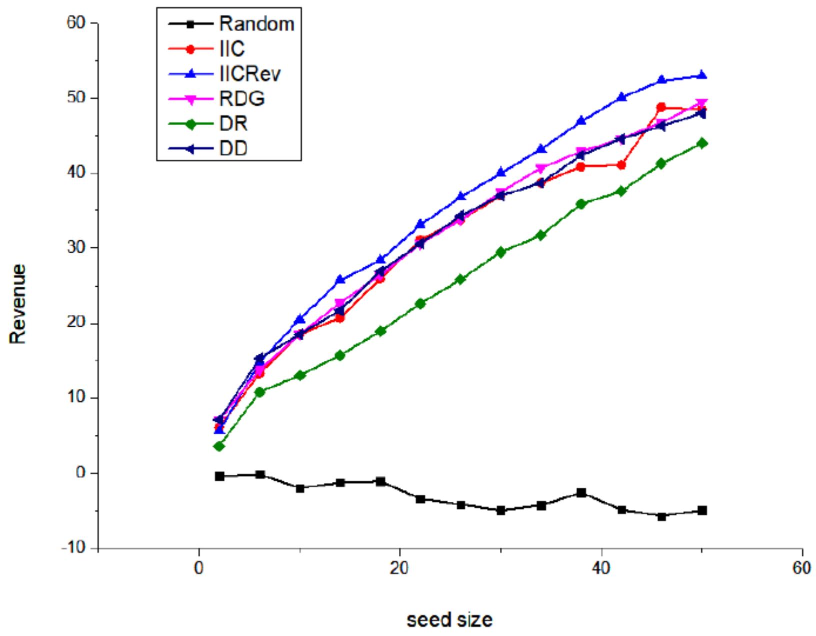
Revenue and Seed Set Size

cost of seed node = 0.5

payoff of informed node = 0.5 payoff of active node = 1



(a) seed node cost = 0.8

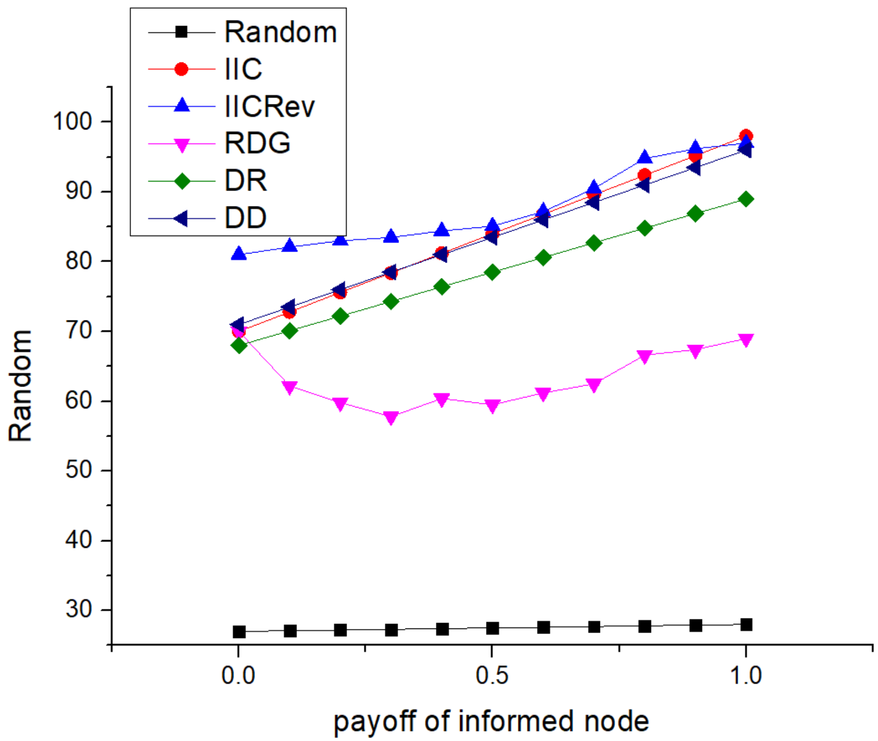


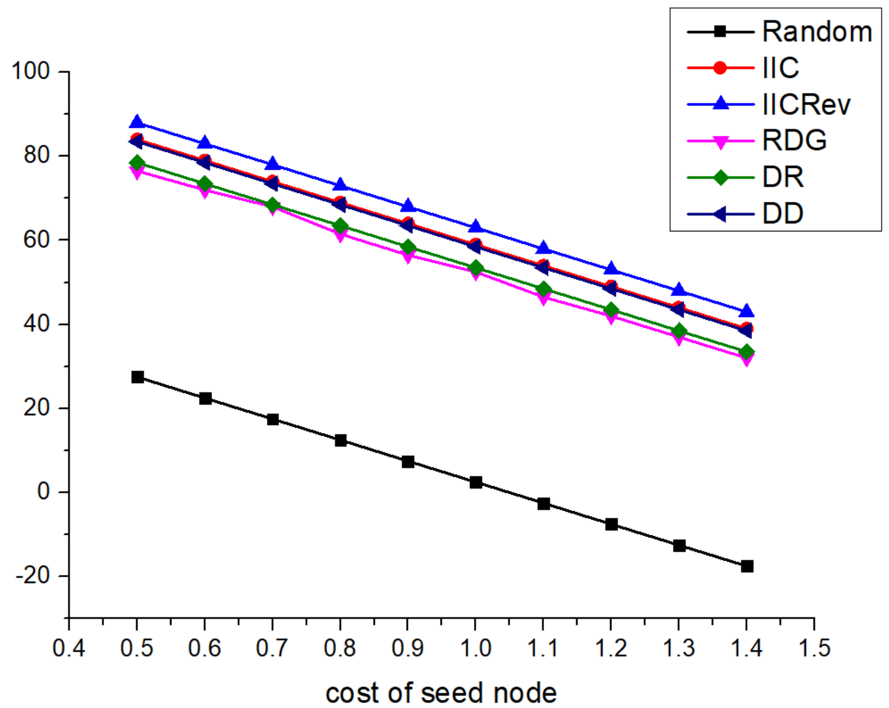
(b) seed node cost = 1.2

IICRev > IIC ≈ DD > RDG > DR > Random

Revenue and Payoff/Cost Value

payoff of active node = 1   seed size = 50





IICRev > IIC ≈ DD > RDG ≈ DR > Random

Conclusion

- IICRev

very simple network
strong computation ability
no requirement for time

Tractable → **DD**

High revenue → **DD**

- DD/RDG

relaxation on revenue maximization
high complexity network
DD has lower time complexity

Thank You

515030910603 张嘉懿

Final Report