Voting System of Two Candidates

Yangrui HU*

hyr1996@mail.ustc.edu.cn

August 16, 2016

I. Introduction

In order to understand various real system, the study of complext network is developed. For example, suppose that there are 100 people and two parties. Each person can support one party. At first, there are 50 people supporting Party R and 50 people supporting Party B. The deadline of voting is fixed. People can discuss with friends and change their opinions before deadline. Here is a question: if Party R wants to win, which distribution of initial supporters is best? Random network is usually used to simulate real network.

In this paper, I use random network model to simulate voting dynamics and solve this question. My work can be divided into three parts. Firstly, I study the effects of probability p and system temperature T in this voting system and find the optimum values of p and T. Secondly, I use Metropolis Algorithm to simulate voting dynamics and probe the relationships between different kinds of triangles (and quadrangles) in the network and the final number of Party R supporters. Finally, I use genetic algorithm to optimize random network and make Party R always win.

In this section, I review briefly the Erdos-Renyi model and the properties which I will probe in my first part of simulation. In the second section I will elaborate the algorithm used in the simulation of my frst part of work. The simulation results of voting system will be demontrated in third section. Next, I will state the general idea, algorithm to optimize ramdon network and simulation results.

*A thank you or further information

i. The Erdos-Renyi model

Erdos and Renyi defined a random graph as N nodes conected by n edges which chosen randomly from the all possible edges in their first article on random graphs. The number of possible edges equals $\frac{N(N-1)}{2}$. Thus there are $C^n_{\frac{N(N-1)}{2}}$ possible graphs and every realization is equiprobable.

A binomial model is an alternative and equivalent definition. We have N nodes and every pair of nodes being connected with probability p. With different probability p, we will get different network.

ii. Clustering coefficient

When we consider a node in the random graph and its neighbors, the probability that two of these neighbors are connected is equal to the probability that two randomly selected nodes are connected. If two of these neighbors are connected, these two points and the node we consider constitude a triangle. We can also define clustering coefficient as follow:

$$C_{node} = \frac{the\ real\ number\ of\ triangles}{the\ maximum\ number\ of\ triangles} \tag{1}$$

According to the idea of Monte Carlo, when we calculate the average clustering coefficient of all nodes with definition above,the result should be rounded to the probability p. This parameter reflects the topological property of voting system.

iii. Triangles in the network

As we mentioned in the last subsection,there are several triangles in the network.When we think about the problem of voting, each node has two choices:Party R or Party B. So, there are four kinds of triangles: zero node, one node,two nodes,three nodes supporting Party R(or Party B) in the triangle.

We can use different color to represent different choice.If the node chooses Party R, we can color this node with red. If the node supports Party B, we can color this node with blue. The number of different kinds of triangles reflects the topological property of color distribution in the voting system.

iv. Quadrangles in the network

Similarly, we can also define quadrangle in the network.When two neighbors(node B,node C) of a node(node A) have the same neighbor(node D) and at the same time there is no link between node B and node C,node A and node D, a quadrangle appears.There are six kinds of quadrangles: quadrangle with zero red node(Quadrangle0),quadrangle with one red node(Quadrangle1), quadrangle with two red nodes,quadrangle with three red nodes(Quadrangle3) and quadrangle with four red nodes(Quadrangle4). For the squadrangle with two red nodes,there are two kinds of compositions:the colors of two neighbors are same(Quadrangle2a) and the color of two neighbors are different(Quadrangle2b). Fig.1 is an explaination of the different kinds of quadrangles with two red nodes.The number of different kinds of quadrangles also reflects the topological property of color distribution in the voting system.

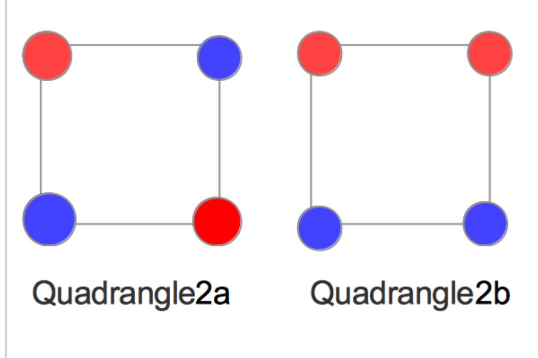

Figure 1: *two kinds of quadrangle with two red nodes*

II. Algorithm

i. Modeling Algorithm

Firstly,transform the problem into physical model. Use points to represent people, red and blue to represent two parties. If two people are friends, there will be a link between two corresponding nodes. So a social network can be demonstrated by an abstract network. This network evolves following a specific dynamics. People talk with their friends about their choices.If most of their friends support the opposite party,people may change their opinions. This process is similar to Ising Model[1], so we can use the idea of Ising Model to deal with this problem.

The voting dynamics can be concluded as follow:

- σ_i represents the color of i-th node,$\sigma_i = 1(red), \sigma_i = -1(blue)$
- Hamiltonian of voting system: $H = \sum_{<i,j>} -J\sigma_i\sigma_j$
- Discussion: consider i-th node and its friends each time. According to their σ value, i-th node converts its σ_i with probability $P(i)_{convert}$.
- After 10,000 times discussion,compare the number of nodes with $\sigma = 1$ and $\sigma = -1$.

ii. Simulation Algorithm

I use Metropolis Algorithm to simulate voting dynamics. Metropolis Algorithm is a kind of Markov Chain Monte Carlo(MCMC) algorithm. According to the master equation[2], we can get Detailed balance[3].

[1] An Introduction to Thermal Physics - Daniel Schroeder
[2] https://en.wikipedia.org/wiki/Master_equation
[3] https://en.wikipedia.org/wiki/Detailed_balance

Detailed balance equation:

$$\frac{P(x \to x')}{P(x' \to x)} = \frac{P(x')}{P(x)} \tag{2}$$

$P(x \to x')$ is the probability of that state x evolves to state x′.$P(x' \to x)$ is the probability of that state x′ evolves to state x.$P(x)$ is the probability of state x and $P(x')$ is the probability of state x′.

Divide the evolution probability into two parts:

$$P(x \to x') = g(x \to x')P_{convert}(x \to x') \tag{3}$$

$g(x \to x')$ is suggestion probability and $P_{convert}(x \to x')$ is convertion probability. Then we can get:

$$\frac{P_{convert}(x \to x')}{P_{convert}(x' \to x)} = \frac{P(x')g(x' \to x)}{P(x)g(x \to x')} \tag{4}$$

Choose appropriate convertion probability that satisfies Detailed Balance condition:

$$P_{convert}(x \to x') = min(1, \frac{P(x')g(x' \to x)}{P(x)g(x \to x')}) \tag{5}$$

Simulation algorithm can be concluded as follow:

- choose a node randomly,and calculate its local energy E_{old}
- convert its σ value and calculate its new local energy E_{new}
- accept new state with probability $min(1, e^{-(E_{new}-E_{old})/k_B T})$

III. Simulation Results

I use N_red to represent the number of nodes whose σ value equals 1,and N_blue to represent the number of nodes whose σ value equals -1 after fixed simulation step. N_red and N_blue demonstrate the speed of dominance of one party. There are several influensing factors whose effect I should research. In this section, I will talk about some simulation results of these factors.

i. Effect of parameter p

Probability p is an important parameter of the random ER network. As I mentioned in the introduction, p equals clustering coefficent. So I can detect the margin of error of Monte Carlo method in this simulation by clustering coefficient. The relation between p and N_red is shown as Fig.1. Parameter β is seted as 1.0, and simulation step is seted as 100.

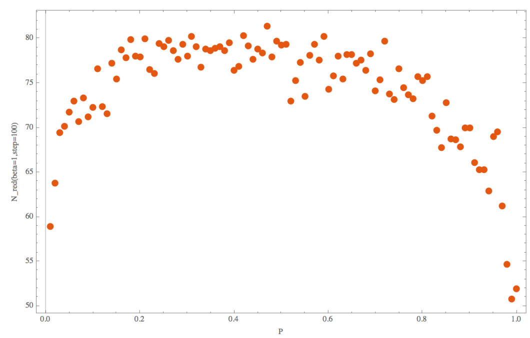

Figure 2: *p - N_red($\beta = 1.0$,step=100)*

Looking at Fig.2 we can find that when p=0.0 0.2, the speed of red dominance increased gradually, when p=0.2 0.6, the speed of red dominance is steady, and when p=0.6 1.0, the speed of red dominance decreases gradually. So the best probability p is from 0.2 to 0.6. I choose p=0.3 in the following simulations.

Fig.3 and Fig.4 are about average clustering coefficient of red nodes and blue nodes respectively. From these two figures, we can see the margin of error of Monte Carlo simulation is ±0.005 for the degree of node.

ii. Effect of parameter T

Temperature T is an important parameter in Ising model. In the voting system, temperature T represent the effect of noise, which means other factor rather than friends(neighbors), such as the effect of newspaper,televation, broadcast and so on. So temperature T is another effect we should consider. Fig.5 and Fig.6 are relation between temperature and average magnetisation, N_red respectively.I set the period of evolution as 100 steps.

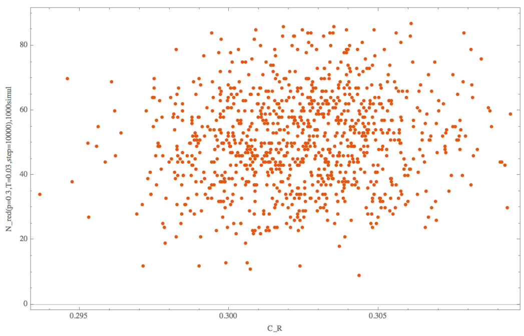

Figure 3: *C_R - N_red(p=0.3,T=0.03,step=10000)*

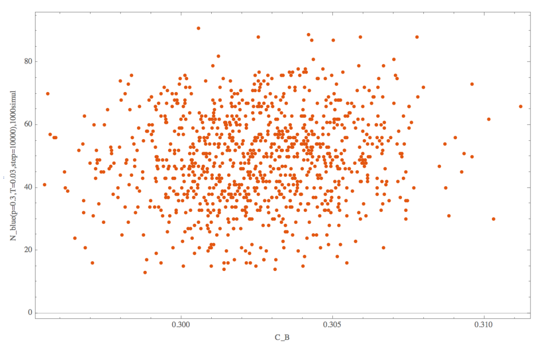

Figure 4: *C_B - N_blue(p=0.3,T=0.03,step=10000)*

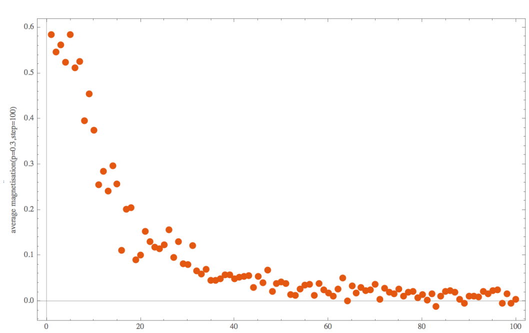

Figure 5: *T - average magnetisation(p=0.3,step=100)*

Magnetisation is defined as $M = \sum_i \sigma_i$ in the Ising Model. It relates with N_red and N_blue, so its shape is similar with Fig.5.

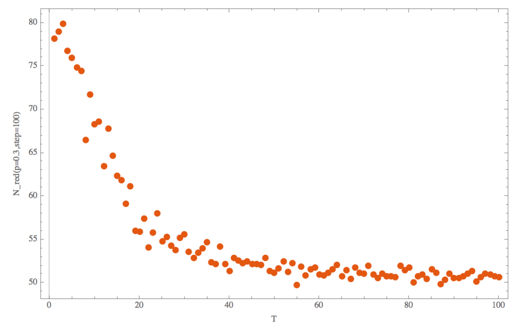

Figure 6: *T - N_red(p=0.3,step=100)*

From these two figures, the speed of red dominance reduces following temperature incresing. If temperature is high enough, spontaneous magnetization will not occur. In the other word, when temperature is high enough, no party will dominate. In the simulation following, I choose temperature as 0.03.

iii. Effect of Triangles

As I mentioned in the section of introduction, there are four kinds of triangles in the network. Fig.7 to Fig.10 are relation between the number of different kinds of triangles in the network and N_red. I simulate 1000 independent random networks and set the period of evolution as 10000 steps.

From these four figures, it seems that there is linear relationship between the number of different kinds of triangles and red dominance speed. And it can be conclude that the good initial color distribution contains high ratio of triangles with 2 and 3 red nodes and low ratio of triangles with 1 and 0 red nodes, if the dominance color is red.

iv. Effect of Quadrangles

As I mentioned in the section of introduction, there are six kinds of quadrangles in the network. Fig.11 to Fig.17 demonstrate the relation-

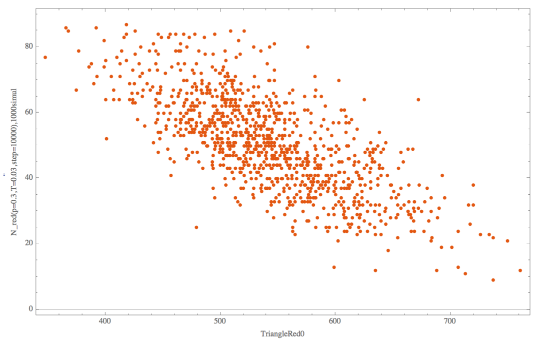

Figure 7: *Triangle with zero red node - N_red(p=0.3,T=0.03)*

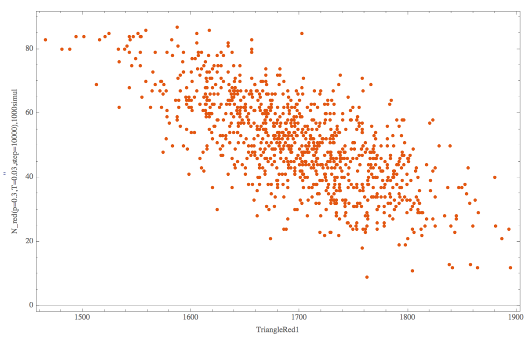

Figure 8: *Triangle with one red node - N_red(p=0.3,T=0.03)*

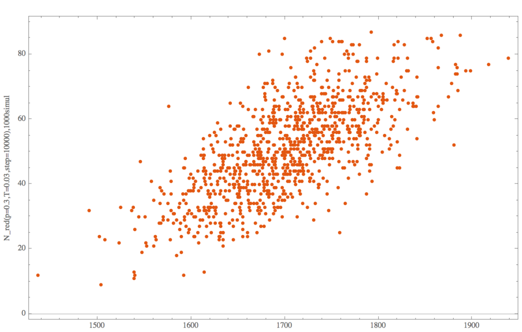

Figure 9: *Triangle with two red nodes - N_red(p=0.3,T=0.03)*

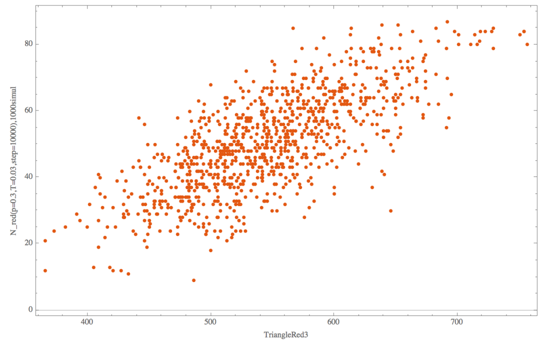

Figure 10: *Triangle with three red nodes - N_red(p=0.3,T=0.03)*

ship between the number of different kinds of quadrangles in the network and N_red.I simulate 1000 independent random networks and set the period of evolution as 10000 steps.

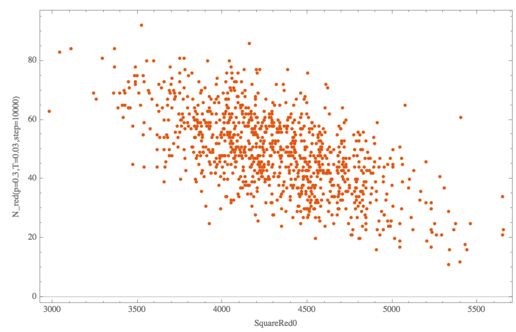

Figure 11: *Qudrangle0 - N_red(p=0.3,T=0.03,step=10000)*

From these six figures, it also seems that there is linear relationship between the number of different kinds of quadrangles and red dominance speed except quadrangles with two red nodes. And it can be conclude that the good initial color distribution contains high ratio of quadrangles with 3 and 4 red nodes and low ratio of triangles with 1 and 0 red nodes, if the dominance color is red. As for quadrangles with two red nodes, there is no obvious relationship between them and red dominance speed. Additionally, from the simulation results, there is no difference between the results

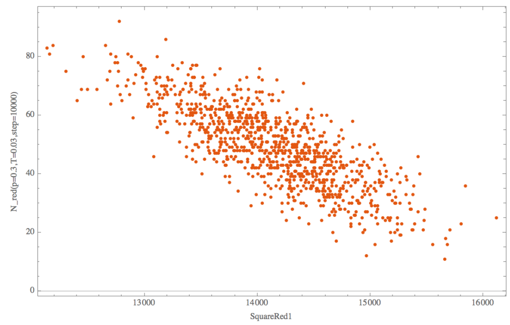

Figure 12: *Qudrangle1* -
N_red(p=0.3,T=0.03,step=10000)

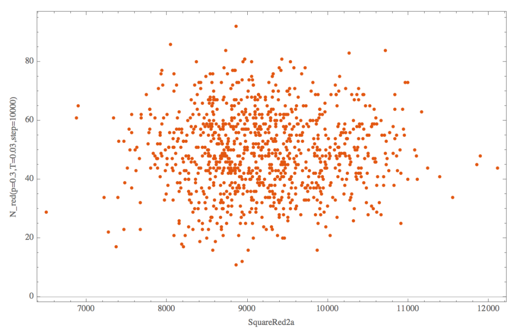

Figure 13: *Qudrangle2a* -
N_red(p=0.3,T=0.03,step=10000)

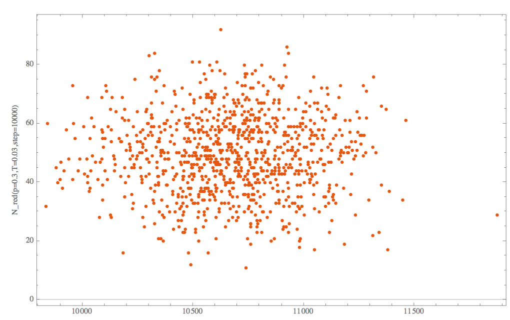

Figure 14: *Qudrangle2b* -
N_red(p=0.3,T=0.03,step=10000)

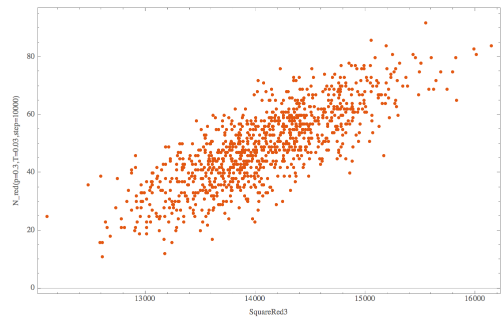

Figure 15: *Qudrangle3* -
N_red(p=0.3,T=0.03,step=10000)

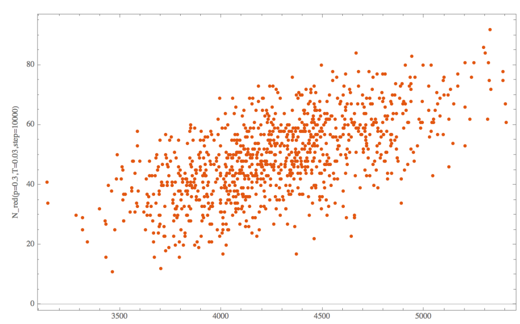

Figure 16: *Qudrangle4* -
N_red(p=0.3,T=0.03,step=10000)

of two kinds of compositions. It is not difficult to understand. There is no color preference in the quadrangle with two red nodes, so it doesn't contribute to red dominance.

v. Conclusion

In the simple voting dynamics, the links between different nodes are steady and will not change by time. At first, there are 50 people supporting Party R and 50 people supporting Party B. The deadline of voting is fixed. People can discuss with friends and change their opinions before deadline.

If Party R wants to win, the social network and the distribution of its initial supporters should have several properties:

- The average degree in the social network should be from 20 to 60, if the total number of nodes is 100.
- Temperature of voting system should be very low, about 0K to 1K.
- The initial color distribution should have high ratio of triangles with 2 and 3 red nodes.
- The initial color distribution should have low ratio of triangles with 1 and 0 red nodes.
- The initial color distribution should have high ratio of quadrangles with 3 and 4 red nodes.
- The initial color distribution should have low ratio of quadrangles with 1 and 0 red nodes.

IV. Optimize random network

i. General Idea

From the third section, I get some conclusions about several properties of network with which Party R could win. For a random network with random color distribution, it would not have these good properties. So if I want Party R to win in any network, I should change the network following some rules that make this network have good propeties. We can get the rules easily from the good properties.For example, a good network should have high ratio of triangles with 2,3 red nodes and have low ratio of triangles with 0,1 red nodes, thus, I should reduce the number of triangles with 0,1 red nodes and increase the number of triangles with 2,3 red nodes. Similarily,for every property I can get a corresponding rule.

Generally speaking, I can optimize random network using any rule. For example, I can just consider reducing the number of triangle with zero red node. Genetic algorithm is a basic and convinient algorithm to evolve network. So, in this paper, I choose one kind of genetic algorithm to reduce the number of triangle with three blue nodes in any random network.

ii. Genetic Algorithm

iii. Simulation Results

In the first place, the rewiring algorithm used in my work can be concluded as follow:

- calculte the distance matrix.(d_{ij} =the least number of links between node i and node j)
- choose one node(node i) randomly
- if this node and two neighbors form a triangle with 3 blue nodes, break one link and rewire this link to the node whose distance to node i is farthest.

I simulate 100 independent random networks and rewire them using algorithm I illustrate above.Then calculate their N_red both with rewiring and without rewiring and compare the results. The step of rewiring is seted as 5,10,15,20 and 50. For each value, the result is as Fig.17 to Fig.21.

The blue curves in the figures are the results of rewiring networks,and red curves are the results of not rewiring networks.In these six figures, the values of N_red in rewiring networks are always higher than the values in not rewiring networks. That is to say that rewiring method always works and increases the number of Party R supporters. The red

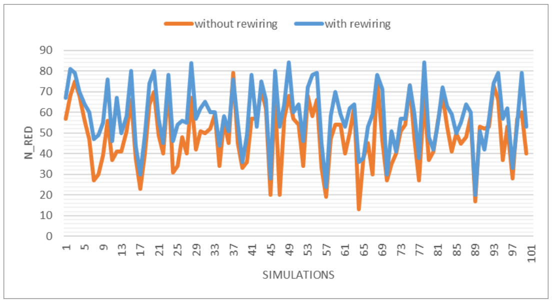

Figure 17: *Comparison: Rewiring v.s NOT rewiring(step=5)*

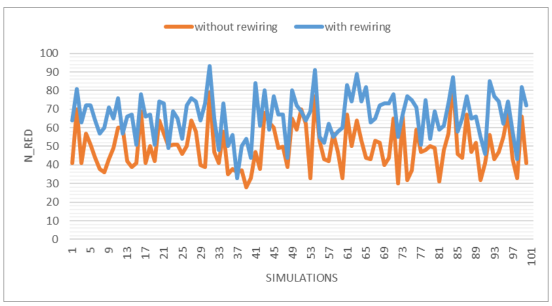

Figure 18: *Comparison: Rewiring v.s NOT rewiring(step=10)*

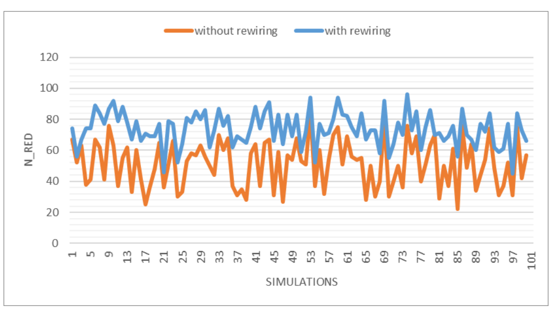

Figure 19: *Comparison: Rewiring v.s NOT rewiring(step=15)*

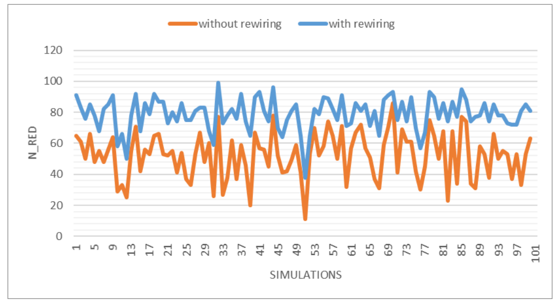

Figure 20: *Comparison: Rewiring v.s NOT rewiring(step=20)*

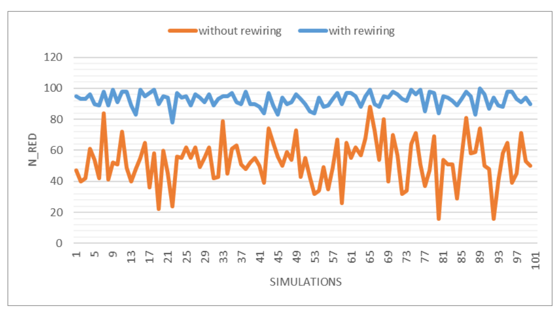

Figure 21: *Comparison: Rewiring v.s NOT rewiring(step=50)*

curves fluctuate around 50 since that colors distribute randomly and the probability of Party R or Party B winning is both 0.5 theoretically.

Additionally, compare six figures and we can see that obviously,the result will be better if the value of rewiring step is more. When I choose rewiring step equals 50, which means that I choose one node randomly and rewire its links following my rule and repeat this process 50 times, N_red is very large and approches 100. So Party R always wins when rewiring step equals 50.Through these simulations I can say that the rewiring method does optimize the random network.

References

[1] D.Schroeder,An Introduction to Thermal Physics

[2] R.Albert and A.L.Barabasi, Statistical mechanics of complex networks,REVIEWS OF MODERN PHYSICS, VOLUME 74, JANUARY 2002

[3] Y.Yu and K.Y.Szeto,Minimize the average mean first passage of random walk in complex networks by genetic algorithm