

## Artisoc で作成する反発力・吸引力可変な Fruchterman Reingold アルゴリズムシミュレーションとコミュニティ抽出について

末 木 俊 之\*

### Repulsive and attractive force variable Fruchterman Reingold Algorithm Simulation with Artisoc and community extraction

Toshiyuki SUEKI\*

In 2015, I have created a simulation based on the Fruchterman Reingold algorithm with the simulator artisoc for network drawing. In the case of Large-scale network, node arrangement is chaotic. And it is not possible to extract community from the display. Even in the case of small network, I noticed that the network drawing is not sometimes preferable. By observation, simple unfavorable factors were revealed. If there are two non-adjacent nodes in the vicinity of a node, the line connecting the two nodes may acts like a wall. Node may become impossible to move beyond the wall. Repulsive force acting between non-adjacent nodes and attractive force acting between adjacent nodes are balanced. Node does not move at all. Adjacent nodes should be close. In early steps of simulation, strong attracting force should be necessary. So I modified the simulation. The new simulation can vary strength of repulsive force and attractive force. By operating the simulation at setting strong attractive force, tendency to separate communities has increased. Although in the center portion of the display space node arrangement is chaotic, in the peripheral portion node arrangement has improved.

#### 1. Fruchterman Reingold アルゴリズムによる artisoc シミュレーションの改造

2015年度研究紀要<sup>1)</sup>では、Fruchterman Reingold アルゴリズムに基づくネットワーク表示シミュレーションを Artisoc で作成した。1ステップの計算処理毎に、全てのノードについて自分に作用している力(全ての反発力と吸引力の合力)を計算し、その力に基づいて次の表示位置(座標)を決定してノードを表示する。しかしシミュレーションを実行した結果、期待

した大まかなコミュニティ分割を確認できるノード配置は得られなかった。ノード数が少ないネットワークのケースでは、コミュニティ分割された綺麗なノード配置の表示となる。しかしノード数が多いとコミュニティ分割を目視で確認できる表示とはならない。リンクは交錯し、ノード配置は混沌として美しさにも欠ける。

ノード数の少ないモデルにおいても何度もシミュレーションを実行しているうちに、コミュニティが綺麗に分離するノード配置に収束しな

---

\*人間健康学部 健康栄養学科

いケースが多いことに気づいた。図1の例は、ノード数34、コミュニティ数2のネットワーク<sup>2)</sup>で、パラメータの初期値として、反発力作用域=5.0、ばね自然長=5.0、 $\delta(t) = 0.05$ 、 $\sigma$ 値減少分=0.0001、反発力倍率=1で実行したシミュレーション<sup>1)</sup>である。図1ではコミュニティ番号0とコミュニティ番号1に属するノードが綺麗に分離していない。図1右上のコミュニティ番号1に属するノードを例に考えると、コミュニティ番号0に属する近傍の2つのノードが壁になり、同コミュニティに属する他の仲間のノードに近づけない状態にあることが観察できる。近傍の2つのノードに対する反発力の合力ベクトル（図中で太い矢印で示した力ベクトル）と、仲間のノードに対する吸引力が釣り合っている状態にあると考えられる。図1右上の1の数値が表示されているノードと0の数値が表示されているノード間には点線が引いてあるが、これは反発力の説明のための補助線であり、実際にはノード間リンクは存在せず、ノード間に吸引力は働かない。反発力のみ働いている。このことから、シミュレーションの初期には、

反発力よりも吸引力が重要であると考えられる。まず同一コミュニティに属するノードは集まらなければならない。そのためには吸引力を反発力よりも強くする必要がある。また反発力および吸引力を計算する対象とするノードの範囲（近傍）を広げることも考えられる。図1で注目したコミュニティ番号1に属するノードでは、近傍の2つのノード（リンクしていないノード）を反発力計算の対象としている。近傍の範囲を広げれば、反発力の合力ベクトルの方向・強さは図中の太い矢印の反発力ベクトルとは違ってくるはずである。しかし、近傍の範囲を広げるとシミュレーションが極端に遅くなるので現実的では無い。

以上のことから、吸引力および反発力が可変なシミュレーションに改造して、シミュレーション実行初期に吸引力を強めてステップを実行し、ノード配置の様子を観察することにした。

具体的には、図2のように、コントロールパネルに、吸引力倍率、反発力倍率を変化させるためのスライドバーを追加し、その数値に基づいてシミュレーションが実行されるように改造

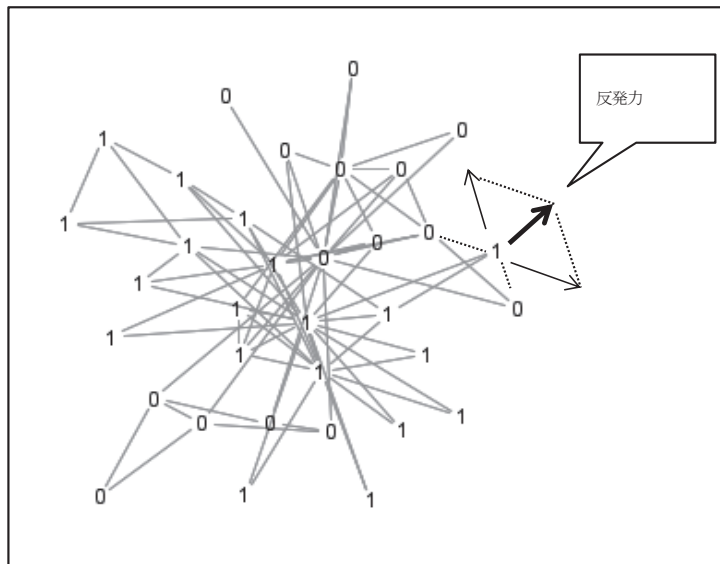


図1



図 2

した。例えば、吸引力倍率 = 2 の場合には、吸引力の強さが基本的な Fruchterman Reingold アルゴリズムに基づく吸引力の 2 倍の強さとなる。

## 2. 吸引力・反発力可変な Fruchterman Reingold アルゴリズム artisoc シミュレーション実行結果

### 2-1. ある大学の空手部部員の友人関係を表現したグラフモデルのケース

参考にした書籍で扱われていたノード数34、コミュニティ数2のネットワーク<sup>2)</sup>について、Artisocで作成したFruchterman Reingoldアルゴリズムシミュレーションを実行した。図1

もこのネットワークモデルのものである。

図1と同様、反発力作用域 = 5.0、ばね自然長 = 5.0、 $\delta(t) = 0.05$ 、 $\sigma$  値減少分 = 0.0001のパラメータ設定<sup>1)</sup>で実行したものである。反発力と吸引力を変えて2種類実行した。

(1) 反発力倍率 = 0.2、吸引力倍率 = 2 に設定して実行したケース

シミュレーションを10回実行し、すべてノード配置が綺麗にコミュニティ分割された表示に収束するのを確認した。収束するまでのステップ数はそれぞれ、50、73、108、58、40、27、43、31、52、81であった。平均56.3ステップである。図3はそのうちの1つの表示例である。

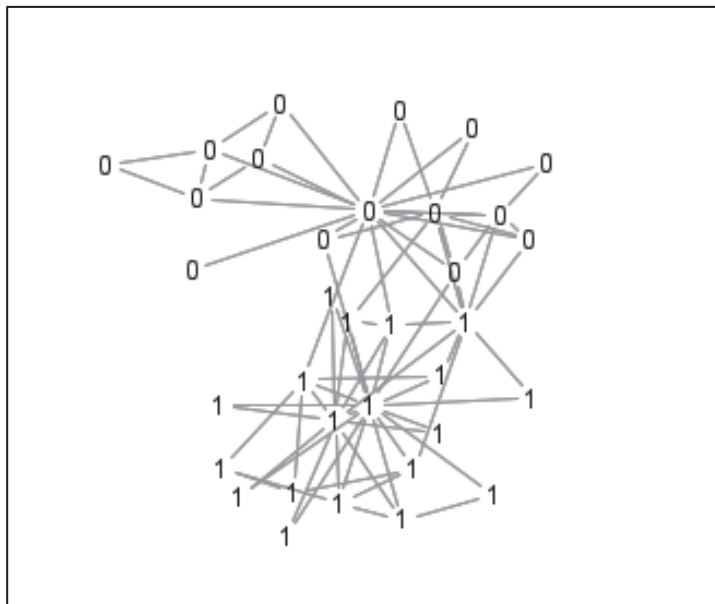


図 3

(2) 反発力倍率 = 1、吸引力倍率 = 1 に設定して実行したケース

これは基本的な Fruchterman Reingold アルゴリズムシミュレーションと同じ結果となるはずである。10回実行した。そのうち半分は綺麗にコミュニティ分割された表示に収束した。コミュニティ分割されずに収束した場合はシミュレーションを打ち切った。収束するまでのステップ数はそれぞれ、133、114、分割せず (120ステップで打ち切り)、54、40、分割せず (151ステップで打ち切り)、分割せず (156ステップで打ち切り)、分割せず (143ステップで打ち切り)、46、分割せず (105ステップで打ち切り) であった。

コミュニティ分割された表示に収束したケースの平均ステップ数は、77.4である。

結果、吸引力を強めないと綺麗にコミュニティ分割された表示に収束させることは難しい

ことが明白となった。

## 2-2. BAモデル ( $m_0=0$ , $m=1$ 、ノード数2000、リンク数3998) 表示結果

ノード数が多いネットワークモデルでも同じシミュレーションを実行してみた。2014年度研究紀要<sup>3)</sup>にて Pajek で Fruchterman Reingold 自動レイアウト表示を行った同じネットワークモデルを使った。反発力作用域 = 5.0、ばね自然長 = 5.0、 $\delta(t) = 0.05$ 、 $\sigma$  値減少分 = 0.0001 のパラメータ設定<sup>1)</sup> で実行したものである。反発力と吸引力を変えて2種類実行した。

(1) 反発力倍率 = 0.2、吸引力倍率 = 2 のケース

図4は1000ステップ目の表示である。500ステップ頃までは激しく表示位置が変化するノードが確認されたが、その後はノードの移動はほとんどなくなって落ち着いてくる。

2015年度研究紀要<sup>1)</sup>に記載されている同じ

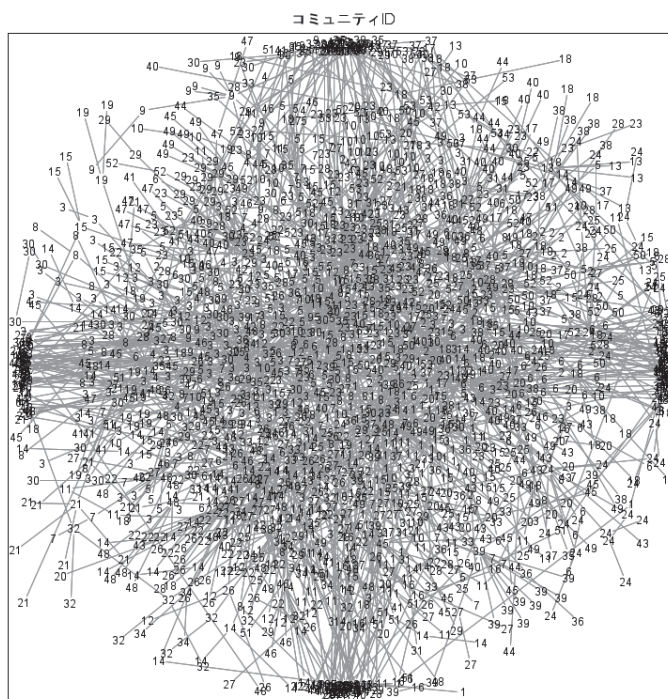


図4

ネットワークモデルに基づく表示図は、今回のシミュレーションでは「反発力倍率 = 1、吸引力倍率 = 1」のケースに相当する。それと比較すると、図4では、中心から離れた周辺ノードのコミュニティ分離度合いが良好になっている。主に周辺ノードの混雑度合いが緩和され、近傍に入るノード数が減り計算時間が減り、各ステップの実行速度も速くなっている。

(2) 反発力倍率 = 0.05、吸引力倍率 = 2のケース

(1) より、さらに反発力を弱めたケースである。

図5は、163ステップ目の表示である。

周辺ノードのコミュニティの分離度合いは良好に見える。ただし中心付近のノードの混雑度合いが高い。

図6は、400ステップ目の表示である。

図5と比較すると、中心付近ノードの混雑度合いが緩和され、周辺に押しやられるノードが増えているように見える。

図7は、700ステップ目の表示である。

図6と比較すると、中心付近のノードの混雑度合いがさらに緩和され、周辺に押しやられるノードが増えているように見える。逆に周辺でノードが密集している箇所ができてくる。

### 3. 結び

2015年度研究紀要<sup>1)</sup>では、Fruchterman Reingold アルゴリズムに基づくノード配置シミュレーションを Artiscoc で作成した。Pajek にて Fruchterman Reingold コマンドを実行した場合<sup>3)</sup>のように、シミュレーションを実行すると綺麗にコミュニティ分割されたノード配置表示となることを期待したが、そうではなかった。ノード数が多いケースでは特に混沌として、リンクが錯綜した表示となる。ノード数が少ない場合は綺麗にコミュニティ分割されたノード配置表示が実現された。しかし、その後何度もシミュレーションを実行した結果、ノード数が少ないネットワークでも、常に綺麗にコ

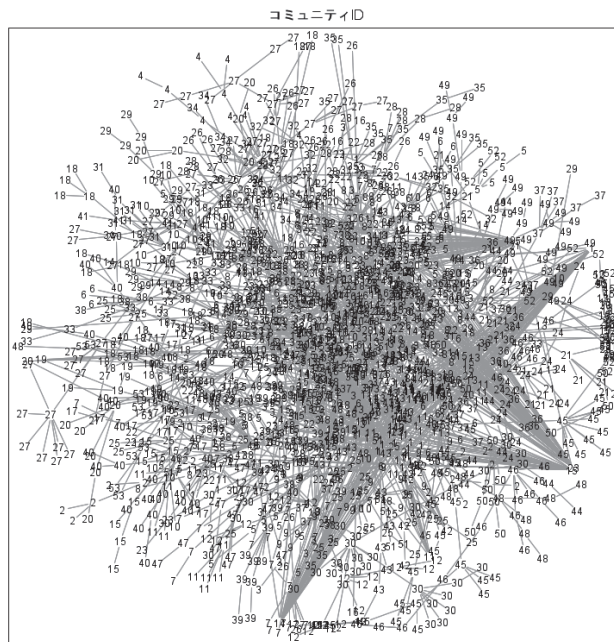


図5

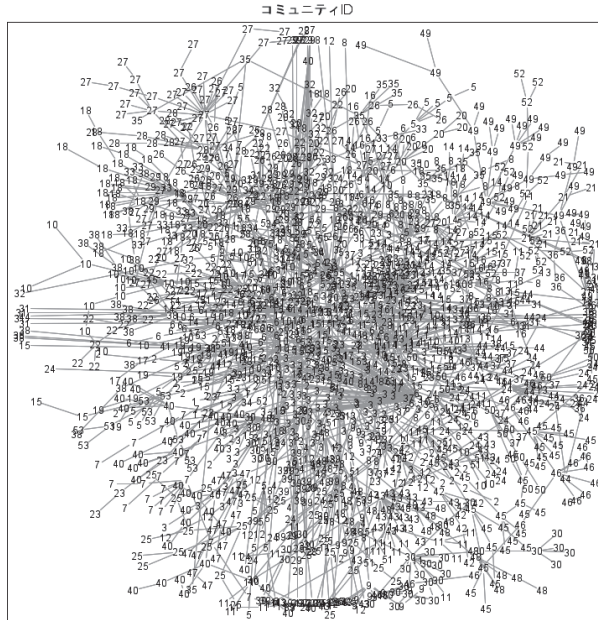


図 6

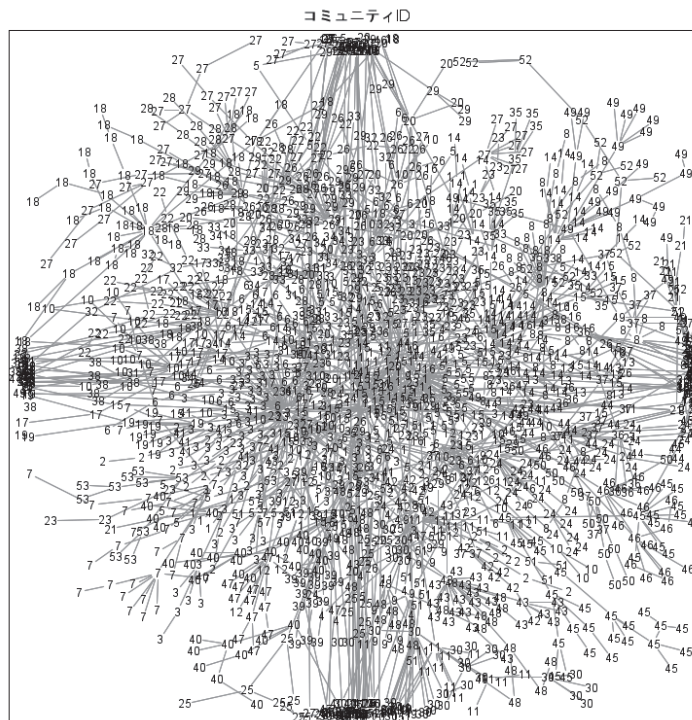


図 7

コミュニティ分割されたノード配置表示が実現されるわけではなかった。

ノード数が少ないネットワークのシミュレーションの観察により、その1つの理由が理解できた。近傍のリンクしていないノードとの間には反発力が発生する。リンクしない2つのノードが近傍に存在すると、その2つのノードを繋ぐラインが壁のようになって、その壁の反対側にあるリンクした仲間のノードに近づくことができないことがある。仲間のノード間に働く吸引力と、リンクしないノードとの間に働く反発力が相殺して動けない状態となって収束してしまっている。とても単純な理由である。確かに基本的な Fruchterman Reingold アルゴリズムに基づく計算だけでは上手くいかないはずである。やはり工夫が必要であった。

シミュレーションでは、近傍のリンクしていないノードを反発力計算の対象としている。近傍の範囲を広げれば、計算対象のノードが増え、反発力の合力は違った結果となる。しかし、近傍の範囲を広げるとシミュレーションスピードが極端に遅くなるので現実的では無い。同一コミュニティに属するノードは集まらなければならない。そのためには強力な吸引力が必要であろう。そこで、単純に吸引力を強めてシミュレーションを実行することを考えた。特にシミュレーションの初期には、反発力よりも強い吸引力が必要ではないだろうか。

以上のことから、2015年度研究紀要<sup>1)</sup>に作成した Fruchterman Reingold アルゴリズムに基づくノード配置シミュレーションを吸引力および反発力が可変なシミュレーションに改造した。そして吸引力を強めてシミュレーションを実行してみた。

その結果ノードの少ないネットワークでは、常に綺麗にコミュニティ分割されたノード配置表示が実現されるようになった。ノード数の多

いネットワークでも、表示空間の中心から離れた周辺部に、コミュニティごとにまとまった綺麗な表示が見られるようになった。基本的な Fruchterman Reingold アルゴリズムにおいて、吸引力を強くすることで綺麗にコミュニティ分割されたノード配置表示に一步近づくことが確認できた。

後は表示空間の中心部分のノード配置の問題であろう。ノード数の多いネットワークでは中心部分が混雑し、混沌とした表示となる。中心部分に来るリンクするノード同士は重なり合うほどごく近くに配置されることを許すような工夫も必要であろう。中心部分のノードと周辺に配置されたノードの扱いが同じでは上手くいきそうにない。吸引力と反発力の強さ、ばね基本長（リンクしたノード間の標準距離）などを中心部と周辺部のノードで変えるなどさらに基本的な Fruchterman Reingold アルゴリズムに修正を加える必要がある。

#### [参考文献・出典]

- 1) 末木俊之 (2015) Artisoc で作成する基本的 Fruchterman Reingold アルゴリズムシミュレーションとコミュニティ抽出について、駒沢女子大学研究紀要、p.143-151
- 2) 林幸雄・大久保潤・藤原義久・上林憲行・小野直亮・湯田聡夫・相馬亘・佐藤一憲著 (2007) ネットワーク科学の工具箱 つながりに隠れた現象をひもとく、p51-56、近代科学社、東京
- 3) 末木俊之 (2014) コミュニティ抽出と Pajek の自動レイアウトについて、駒沢女子大学研究紀要、p.241-254
- 4) ウオウター・デノーイ、アンドレイ・ムルヴァル、ヴラディミール・バタゲーリ著、安田雪監訳 (2009) Pajek を活用した社会ネットワーク分析、学校法人東京電機大学

東京電機大学出版局、東京

- 5) 山影進 (2007) 人工社会構築指南 artisoc  
によるマルチエージェント・シミュレーション入門、有限会社書籍工房早山、東京
- 6) Michael Kaufmann, Dorothea Wagner  
(2001) Drawing Graphs : Methods and  
Models, p71-77. Springer-Verlag, Berlin