

Artisoc で作成する自然ばね長を固定としない 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*

Fruchterman Reingold Algorithm Simulation that the zero-energy length of spring
is not fixed with Artisoc and community extraction

Toshiyuki SUEKI*

Abstract

Over the past few years I have modified the simulation based on the Fruchterman Reingold algorithm with the simulator artisoc. But in the case of Large-scale network, node arrangement is crowded and chaotic in the center portion of the display space. Therefore, I made a modification to lengthen zero-energy length of spring at the center. Furthermore, I made adjustments that nodes with large orders are more centered. As a result, the display of the center part became sparse. However, nodes belonging to different communities are mixedly displayed. In the center portion of the display space, it is not possible to extract community from the display. The node arrangement of peripheral portion is fast fixed. And the central node placement tends to be adjusted afterwards. It is one factor that the central part is crowded and chaotic.

キーワード：pajek ばねモデル

1. ノード位置に基づきばね基本長を変化させる Fruchterman Reingold アルゴリズムによる artisoc シミュレーション

2016年度研究紀要¹⁾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することにより、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。

あとは表示空間の中心部分のノード配置の問題である。ノード数の多いネットワークでは中

*人間健康学部 健康栄養学科

心部分が混雑し、混沌とした表示となる。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。ばね基本長（リンクしたノード間の標準距離）を中心部と周辺部のノードで変化させる修正を加えた。空間の中心に近いノードほど「ばね実際長」を大きくした。表示空間の周辺ではノードが密集して表示され、中心部分ではノードの混雑状況が緩和されることを期待した。

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

2016年度研究紀要¹⁾でも使っているノード数2000のネットワークモデルでシミュレーションを実行してみた。反発力作用域=5.0、ばね自然長=5.0、 $\delta(t)=0.05$ 、 σ 値減少分=0.0001、反発力倍率=0.05、吸引力倍率=2のパラメータ設定¹⁾で実行したものである。

ノードの表示空間中心からの距離に基づいて「ばね実際長」を決める修正を加え何度か実行してみたが、期待した表示とはならなかった。

表示空間中心にノードが集まって混雑する様子に変化はない。図1のように、ノードの中心部への集中傾向がさらに増す状況になることが多かった。

シミュレーションの最初からノードの位置に基づいて「ばね実際長」を変化させては上手くいかないようである。

そこで「ばね実際長」固定のシミュレーションで、ある程度周辺ノードが綺麗にコミュニティ分割された状態のノード配置情報（各ノードのX、Y座標）を書き出し、それを読み込ませてシミュレーションを実行してみた。

図2は、「ばね実際長」固定のシミュレーションを使用し、700ステップ目まで動作させたものである。中心付近のノードは混雑しているが、周辺部ではノードの混雑度合いが緩和され、ある程度コミュニティ分割ができている。このノード配置を初期状態として「ばね実際長」が変化するシミュレーションを実行してみた。

図3は、図2のノード配置から、「ばね実際長」を変化させるシミュレーションを動かして、

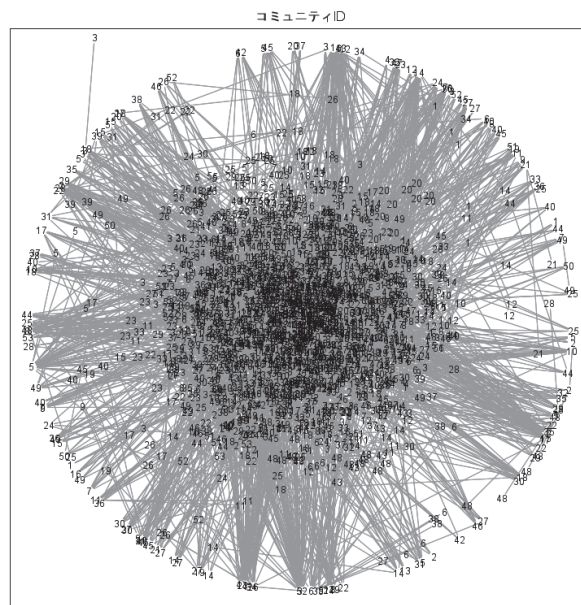


図1

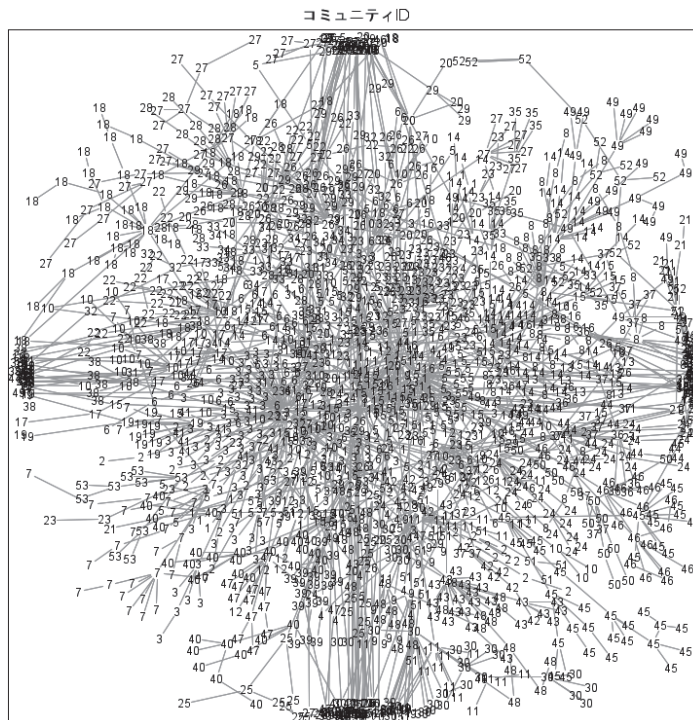


図2

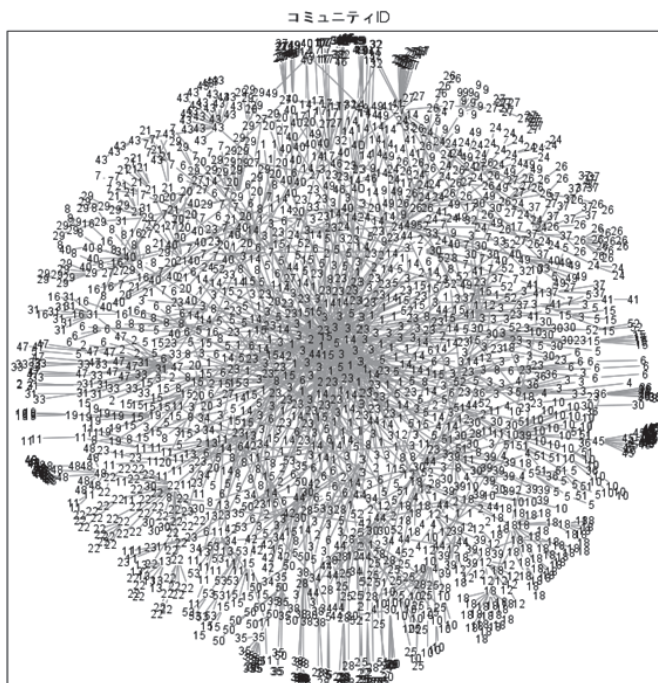


図3

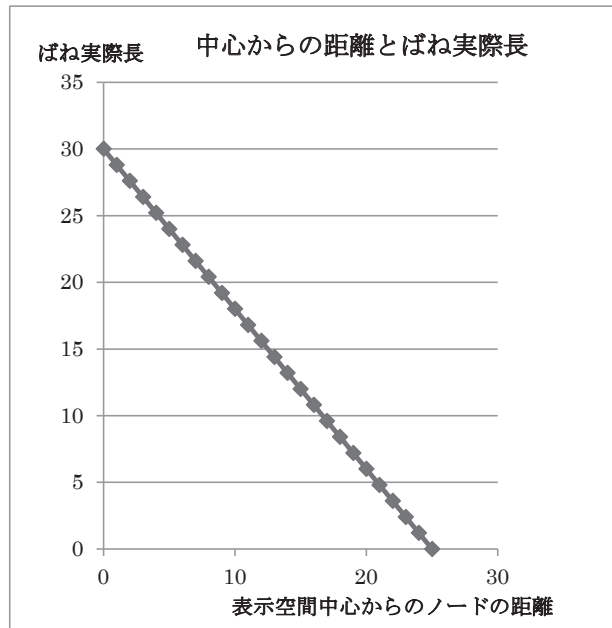


図 4

713ステップ目の表示である。

図 2 と比較すると、中心付近のノードの混雑度がさらに緩和され、周辺に押しやられるノードが増え、周辺でノードが密集する箇所ができてくる。

「ばね実際長」は、ノードの表示空間中心からの距離に基づき、単純に比例的に変化させている。ばね自然長=5.0でのケースでは、「ばね実際長」と表示空間中心からの距離は、図 4 のグラフが示す関係となっている。表示空間の中心付近のノードの場合は、「ばね実際長」 ≈ 30 で計算される。表示空間周辺のノード（中心からの距離 ≈ 25 ）の場合は、「ばね実際長」 ≈ 0 で計算されることになる。

2. 度数に基づきノード配置を微調整を加えた Fruchterman Reingold アルゴリズムによる artisoc シミュレーション

図 3 では、中心部で度数（つながりの数）の大きなノードを含むコミュニティが混在して表

示されている。さらに中心部をすっきりと表示させる工夫が必要である。次数の大きなノードがより中心に配置され、次数の小さなノードは強制的により周辺に押し出される微調整を加えることで、中心部の混雑を強制的に緩和できるのではないかと考えた。各ノードの次数はシミュレーション起動時に調べられ、各ノードに変数として保持されているので簡単に参照できる。最低次数 1 のノードから最高次数を持つノードまで、単純な比例計算で中心からの表示位置を計算し、配置を調整することにした。

ただし、シミュレーションの各ステップで多数のノードの位置調整を行うと、図 5 のようにほとんどのノードが周辺に押しやられてしまう。すべてのノードに対して位置調整を行うような修正はやり過ぎであった。また位置調整も計算通りに配置を変化させるのではなく変化量を緩和することにした。結果として図 6 の表示が得られた。中心からの位置が 23 未満（長方形の表示空間の大きさは縦・横 25 である）であるノー

周辺部のノード配置が早く固定されてしまい、中心部のノード配置はその後で調整される傾向がある。中心部のノード配置は激しく変化するが、周辺部のノードが動かないので、中心部のコミュニティ分割が進行しない傾向が観察される。これはコミュニティ分割が上手くいかない大きな要因と思われる。

3. 結び

ここ数年 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 Artisoc で作成してノード表示を観察している。Pajek にて Fruchterman Reingold コマンドを実行した場合²⁾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、常に綺麗にコミュニティ分割されたノード配置表示が実現されないことが分かった。

そこで基本的な 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に、吸引力と反発力を可変にする修正を加え、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。

次は表示空間の中心部分のノード配置が問題である。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。ばね基

本長（リンクしたノード間の標準距離）を中心部と周辺部のノードで変える修正を加えてみた。表示空間の中心に近いノードほどばね基本長を大きくした。ある程度表示空間周辺のノードが綺麗にコミュニティ分割された状態までステップを進めた後に、ばね基本長をノードの位置に基づいて変更する修正を有効にしてシミュレーションを進行させれば、中心付近のノードの混雑度合がさらに緩和された表示に収束する。

しかし中心部で次数（つながりの数）の大きなノードを含むコミュニティが混在している表示されている。次数の大きなノードがより中心に配置され、次数の小さなノードがより周辺に配置されるなら中心部の混雑を強制的に緩和できるのではないかと考えた。各ノードの次数はシミュレーション起動時に調査し、各ノードに変数として保持しているので簡単に参照できる。最低次数1のノードから最高次数のノードまで、単純な比例計算で中心からの距離を計算し、配置を調整した。すべてのノードについて位置調整を適用すると、多くのノードが表示空間周辺に押しやられてしまい、表示も収束しない。図7のように、一部のノードに絞って位置調整を行うならば、いくらかは中心部分の混雑が緩和された表示にはなった。

基本的な 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に調整を加えれば、ある程度はコミュニティ分割されたノード表示に収束する傾向はある。しかし、周辺のノードは綺麗にコミュニティ分割された表示になるが、周辺部のノード配置が早く固定されてしまい、中心部のノード配置はその後で調整される傾向があり、中心部分のノードが上手くコミュニティ分割されない。またノードが表示空間に均等にばらまかれてしまう。常に他のノード間に反発力と吸引力が働くのでバランス良く配置されることになってしまう。図7の

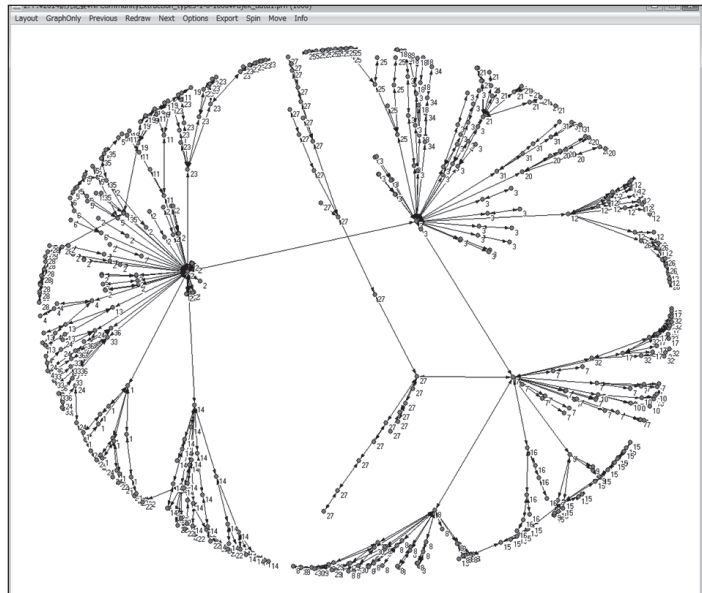


図 7

PajekにてFruchterman Reingold コマンドを実行した場合²⁾のように同じコミュニティに属するノード同士がまとまって表示される状態にはなりにくい。

シミュレーションでは完全にコミュニティ分割されたノード表示に到達するのは難しい。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実行し、ある程度ノード配置が収束して落ち着いた状態に到達させることはできる。しかし、その後周辺のリンクしたノード同士の距離を短く、中心部分では距離を長く調整するなど表示調整を行わないと綺麗にコミュニティ分割されたノード表示にはなりにくいというのが現時点での結論である。

[参考文献・出典]

- 1) 末木俊之 (2016) 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、駒沢女子大学研究紀要、p.153-160
- 2) 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
- 3) 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 つなかりに隠れた現象をひもとく、p51-56、近代科学社、東京
- 4) ウオウター・デノイ、アンドレイ・ムルヴァル、ヴラディミール・パタゲーリ著、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