

南雲 淳

筑波大学大学院理工学研究科<sup>†</sup>

田中 二郎

筑波大学電子・情報工学系<sup>‡</sup>

## 1 はじめに

データ構造の表現の方法として、辺と節からなるグラフ構造がある。このグラフ構造により表現されたデータをわかりやすく提示するために、グラフ自動描画法が研究されてきている [1]。

また現在、並列論理型言語をグラフによって可視化するビジュアルプログラミングシステムを開発中であり、それにグラフ描画アルゴリズムを応用することを検討している。これは、並列論理型言語での定義節をグラフの節、論理変数を辺で表現することにより可読性の向上を図るものである。

これまでに提案されているグラフ描画アルゴリズムの多くは、グラフの論理的構造を入力として、その構造の最適な配置を求めるといったものであったが、ここでは、特に人間の手によって入力された形の整っていないグラフを、再配置の過程をアニメーションのような表現で見せることによってわかりやすさの向上を図るためのアルゴリズムの改良を提案する。

あるグラフの再配置を行い、入力されたグラフから再配置されたグラフに表示を切り替える場合、再配置されたグラフをいきなり表示するのではなく、元のグラフから再配置されたグラフへ徐々に変化しているように見せるのが認知の連続性の観点から言って好ましい。

アニメーション的な表示をするためには、再配置された結果を用いてコマ割りを行い、それを順次表示する方法と、再配置の計算の経過を順次表示することによりアニメーションとする方法が考えられる。計算過程を表示するだけでスムーズなアニメーションを生成できれば、コマ割りというプロセスが必要なくなり、ビジュアルプログラミングシステムなどのような、アニメーションの最中にも表示されたグラフの直接操作をしたいシステムの制作時に有利であると考えられるため、ここでは後者の方法について研究を行った

\*Improvement of graph drawing algorithm for visual programming system

<sup>†</sup>Jun Nagumo : Master's program in Science and Engineering Univ. of Tsukuba

<sup>‡</sup>Jiro Tanaka : Institute of Information Sciences and Electronics Univ. of Tsukuba

## 2 アルゴリズムの改良・評価

単純無向グラフを再配置するアルゴリズムには直交格子描画や力指向描画と呼ばれるものがある。このうち、力指向描画アルゴリズムには「節の位置についての自由度が高い」「節を少しずつ動かしながら最適位置に近づけていくため、アニメーション化しやすい」などの特徴があるので、アニメーション的表示によりわかりやすさの向上を図るという点から、力指向配置、特にばねモデルと呼ばれるアルゴリズムの中で最も基本的なものであると思われる Eades のアルゴリズム [2] を用いることにする。

Eades のアルゴリズムでは、各節の間に働く力を定義して、その合計値を元に節の移動方向・移動距離を計算し、それを基に節を移動させながら適切な位置に近づけていくアルゴリズムである。以下にその概要を述べる。

辺によって直接接続されている節の間には、その間の距離によって引力あるいは斥力が働き、直接接続されていない節同士には斥力のみが働く。

直接接続された節の間に働く力を  $f_s$ 、そうでない節の間に働く力を  $f_r$  とすると、

$$f_s(d) = C_s \log \frac{d}{d_0}$$

$$f_r(d) = C_r \frac{1}{d^2}$$

のように定義される。

ここで、 $C_s, C_r$  は、引力と斥力のバランスをとるための定数である。また、再配置の際の理想的な辺の長さはプログラマが与えなければならず、これを  $d_0$  とする。 $d$  は節の間の実際の距離である。

このアルゴリズムでは、2 つ以上の節が接近しているときに斥力が大きくなりすぎ、計算の経過を順次表示させると、最初の数ステップの動きが大きくなりすぎ、見苦しい表示になってしまう。

これを解決し、コマ割りせずにアニメーション的表示を可能とするために、直接接続されない節同士に働く力を計算する関数を以下のように変更することによって解決を図った。

$$f_r(d) = \begin{cases} C_r & (d < d_0) \\ C_r \frac{1}{d^2} & (\text{Otherwise}) \end{cases}$$

改良前の  $f_r$  では、常に斥力が節の間の距離の 2 乗に反比例しているため、節同士が接近している状態で斥力を

計算すると大きくなりすぎてしまい、これが節の見苦しい動きの原因となる。斥力を計算する関数を上のように変更することにより、節の間の距離が短いときの斥力を小さくできるので、再配置開始後の数ステップの間に見られる移動の不具合を解消できる [3]。

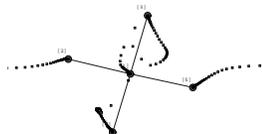


図 1: 改良前の Eades のアルゴリズムによる

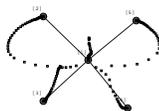


図 2: 改良後の Eades のアルゴリズムによる

図 1,2に、1つの節のごく近くに4つの節を配置した状態から改良前・改良後それぞれのアルゴリズムで再配置をしたときの各節の軌跡を示す。改良前では最初の数ステップの斥力が強すぎるために2つの節が左右に大きく広がってから目標位置に近づいているが、改良後では全ての節が連続的にスムーズに動いているのがわかる。

このアルゴリズムにより生成されるグラフは、Eadesのアルゴリズムによって生成されるグラフと同じものであり、初期配置に依存して起こると思われる不要な辺の交差や配置の「ねじれ」の問題 [4] は解決できない。

この改良による効果を評価するため、様々なグラフについて、初期状態を変化させたときの各節の移動量の平均を調べることにした。

評価のためのグラフの初期状態は、各節を円周上に配置するものとし (配置例を図 5に示す)、配置の際の半径を変化させることによって初期状態を変化させた。また、再配置時の節の間の理想距離は  $\frac{200}{(\text{グラフの半径})}$  [ピクセル] とした。

図 3は、先の図に挙げたグラフの初期配置の幾何学的半径を変化させたときの節の移動量の平均値を示している。グラフの横軸が初期配置の半径、すなわち、初期状態の節の密集の度合 (小さいほど密集している) であり、縦軸が再配置時の節の移動量の平均である。

別の形状のグラフに対して調べた結果を図 4に示す。

これらから、初期状態で節が接近しているときの各節の移動量が小さくなっていることがわかる。これは、節の無駄な移動が減っていることを示しており、アニメーションとして考えたときに、見易くなる方向に改善されていると言える。

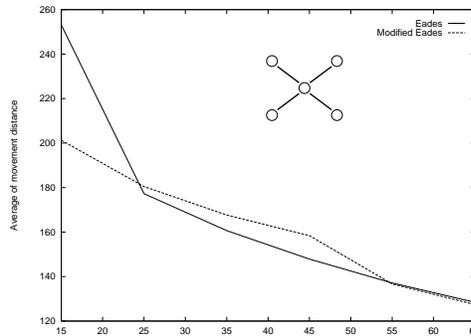


図 3: 各節の平均移動距離

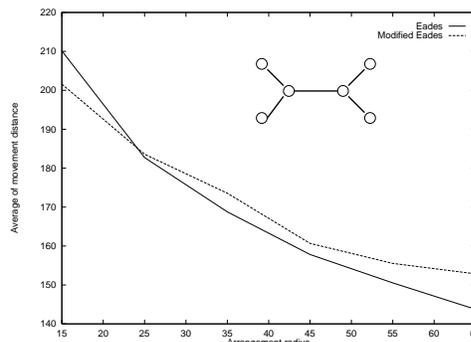


図 4: 各節の平均移動距離 (その 2)

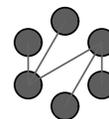


図 5: 評価に用いた初期配置の例

### 3 結論

力指向のグラフ描画アルゴリズムである Eades のアルゴリズムを、再配置の過程を見せることによってアニメーションさせることを前提として改良を行ない、その効果を確認した。グラフ構造の変化をユーザに提示する上で有効な手段になると考えられる。今後は、このアルゴリズムにより並列論理型言語の実行をアニメーション的表示により可視化するシステムを実装する予定である。

### 参考文献

- [1] 杉山公造：グラフ自動描画法とその応用 計測自動制御学会 (1993)
- [2] P.Eades : *A Heuristics for Graph Drawing* Congressus Numerantium, Vol.42, pp.149-160 (1984)
- [3] 南雲淳：ビジュアルプログラミングシステムのためのグラフ描画アルゴリズムの研究 筑波大学情報学類卒業研究論文 (1996) [http://www.softlab.is.tsukuba.ac.jp/iplab/paper/nagumo\\_thesis.ps](http://www.softlab.is.tsukuba.ac.jp/iplab/paper/nagumo_thesis.ps)
- [4] T.Fruchterman and E.Reingold : *Graph Drawing by Force-directed Placement* Software-Practice and Experience, Vol.21, pp.1129-1164 (1991)