

グラフ描画アルゴリズムの性能評価

角 浩二 †
榎原 博之 ‡

田中 寿俊 †
中野 秀男 †

† 大阪大学工学部通信工学科, 吹田市
‡ 関西大学工学部情報処理教室, 吹田市

近年, コンピュータ技術の発展により, その用途は多様化している。その中の一つに、点と線であらわされる図形をグラフとしてモデル化し、描画させるという用途がある。一般のグラフの描画では、「見やすさ」の基準を考える必要があるが、各個人の主観による部分があり、簡単には「見やすさ」の評価をすることは出来ない。

そこで本報告では、グラフの「見やすさ」に対する一般的な基準を考え、定量的に評価することを試みる。さらに、一般グラフを描画する、スプリングモデルに基づいた2つのアルゴリズムとそれらの改良版について、描画したグラフから各アルゴリズムを定量的に評価する。

Performance Evaluations of Graph Drawing Algorithms

Koji Sumi †
Hiroyuki Ebara ‡

Hisatoshi Tanaka †
Hideo Nakano †

† Faculty of Engineering, Osaka University, Suita-shi, 565 Japan
‡ Faculty of Engineering, Kansai University, Suita-shi, 564 Japan

Recently, the applications of computer have increased because of the developing of computer technology. Graph drawing is in one of such applications, where graphs are modeled as pictures represented by points and lines. General graph drawing problems involve *aesthetics*, that is difficult to evaluate because *aesthetics* depends on an individual view.

In this paper, the general aesthetics standards are represented quantitatively. Moreover, two general graph drawing algorithms based on spring model, and their revised algorithms are evaluated.

1 はじめに

近年、CPUの高速化、記憶装置の高密度化、ディスプレイ装置の高精細度化などコンピュータ技術の発達及び、高度情報化社会におけるコンピュータ需要の増加によりコンピュータの用途は多様化している。その中の一つに、電気回路図、フロチャート、PERT図などの点と線で表される図形をグラフとしてモデル化して、ディスプレイ上に描画させるという需要がある。そこでは、単にグラフを描画させるだけでなく、人間が理解し易いように「見やすい」、あるいは「美しい」グラフを自動的に生成して描画させる必要がある。

本報告で取り上げた一般グラフ描画アルゴリズムとは、一般グラフを自動的に見やすく描画するアルゴリズムのことである。様々なグラフに対するレイアウトアルゴリズムについては、文献[5]に述べられている。木、平面グラフ、階層グラフなどのようにグラフを制限すると、これらグラフを見やすく描画するアルゴリズムは、いくつか存在するが、グラフを限定しないで一般グラフを見やすく描画することはなかなか困難である。

また、どのように「見やすさ」を評価するかも、各個人の主観による部分があり、グラフの使用目的にも大きく依存し、簡単には「見やすさ」の評価、ひいては描画アルゴリズムの性能を評価することは出来ない。

一般グラフのレイアウトアルゴリズムについて、Eades[1]とKamada[2]はそれぞれスプリングモデルに基づくアルゴリズムを提案している。本報告ではまず、この2つの一般グラフを描画するアルゴリズムの比較方法について述べる。そのために、グラフの「見やすさ」に対する一般的な基準を文献[3]や文献[5]の記述を参考にして、描画アルゴリズムを定量的に評価する。具体的には、描画されたグラフについて次の各項目を調べ、グラフの「見やすさ」を定量的に表す。

- 枝の交差数
- 枝の長さの一様性
- 枝の長さの総和
- 節点分布の一様性
- 描画面積

さらに、EadesとKamadaのアルゴリズムの改良版を提案し、それらの4つのアルゴリズムの性能を定量的に評価することを試みる。

2 スプリングモデル

スプリングモデルとは、一般グラフの節点を質点に、枝をバネに見立て、バネの力学系を安定にできるように節点を動かすものである。すなわち、二つの節点が離れすぎていると両者を近づけ、近づきすぎていると両者を離すことで節点間距離を一定にするという考え方である。

2.1 Eadesのアルゴリズム

Eadesのアルゴリズムでは、隣接している2節点間においては、対数的なバネの力が働くようにしている。これは文献[1]によると、力学でいうフックの法則をバネの力に適用すると、実験の結果、2節点間距離が長いときにバネの力が強すぎたからである。このバネの力 f_a を定式化すると次のようになる。

$$f_a(d) = C_1 \log\left(\frac{d}{C_2}\right) \quad (1)$$

ここで、 d は隣接節点間の長さ、 C_1, C_2 は定数である（特に、 C_2 は隣接節点間の理想距離であり、 $d > C_2$ のとき $f_a > 0$ で引力となり、 $d < C_2$ のとき $f_a < 0$ で斥力となる）。

また、隣接していない2節点間においては、互いに反発し合うように距離の2乗に反比例するような力が働くようにしている。このバネの力 f_r を定式化すると次のようになる。

$$f_r(d) = \frac{C_3}{d^2} \quad (2)$$

ここで、 d は隣接節点間の長さ、 C_3 は定数である。

全体の手続きの概要を次に示す。

Step 1: 節点をランダムにばらまく¹

Step 2: 以下の操作を M 回繰り返す

(M : 適当な定数)

全ての節点に対し

Step 2-1: ある節点に働く力の合計を計算する

Step 2-2: その節点を合力の向きにその大きさの C_4 倍だけ動かす

Step 3: グラフを描画する

文献[1]によれば、 $C_1 = 2.0, C_2 = 1.0, C_3 = 1.0, C_4 = 0.1$ とするとほとんどのグラフで良い結果が得られ、 $M = 100$ とするとほとんどのグラ

¹2.2節 脚注参照

フでスプリングモデルのエネルギーは収束すると書かれている。本報告では、実際の描画結果から $M = 100$ ではスプリングのエネルギーが収束しないものがあり、 $M = 200$ で全ての描画を行なうことにする。

2.2 Kamada のアルゴリズム

Eades のアルゴリズムが節点にかかる合力を評価するのに対し、Kamada のアルゴリズムは系のエネルギーを直接評価しながらエネルギーが最小になるように節点を動かすものである。

また、Eades はある節点にはその隣接節点だけが重要なので引力は隣接節点間のみ計算していたが、Kamada は隣接していない節点間の理想距離に関する考えを導入した。この考えは描画平面上の節点間の理想的な幾何学距離をグラフ上の距離（最短パスの長さ）とするものであり、アルゴリズムは節点間距離をできるだけこの理想距離になるよう節点を動かす。

この系のエネルギー E を定式化すると次のようになる。

$$E = \sum_{i < j} \left(\frac{1}{2} \right) k_{ij} (|p_i - p_j| - l_{ij})^2 \quad (3)$$

ここで、 p_i はグラフの節点 v_i に対応する描画平面上の点で、 l_{ij} は節点 v_i, v_j 間の理想距離である。 k_{ij} はバネ定数で、理想距離の 2 乗に反比例させている。アルゴリズムはこのように定義された E が最小（あるいは極小）になるよう節点を動かしていく。各節点 p_i の座標を (x_i, y_i) で表すと、極小になる条件は次のようになる。

$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_i} = 0 \quad (i = 1, 2, \dots, n) \quad (4)$$

各ステップで一つの節点（仮に p_m とする）を動かすとき、他の節点の位置を固定し、 E が安定になる位置にその節点を移動させる。具体的には、連立方程式 $\partial E / \partial x_m = \partial E / \partial y_m = 0$ を 2 次元の Newton-Raphson 法で解いて、節点の新しい位置を計算する。

しかし、グラフの規模が大きくなると、Newton-Raphson 法では解が収束しなくなることが、本報告での実験ではかなり生じた。これは、各ステップで一つの節点を動かすとき、初期値が Cauchy の収束条件となっていないからであると考えられ、実際、節点を動かすときの初期位置を変更すると収束することがわかった。したがって、本報告で

は描画アルゴリズム内に、解の収束状態をチェックしランダムに初期値を更新するルーチンを入れることにする。

全体の手続きを以下に示す。

Step 1: 節点をランダムにばらまく²

Step 2: 理想値からのずれを示す Δ_i の中で最大の値が充分小さくなるまで繰り返す

$$\Delta_i = \sqrt{\left\{ \frac{\partial E}{\partial x_i} \right\}^2 + \left\{ \frac{\partial E}{\partial y_i} \right\}^2} \quad (i = 1, 2, \dots, n) \quad (5)$$

Step 2-1: Δ_i の中で最大値を与える節点

$p_m(x_m, y_m)$ を選ぶ

Step 2-2: 連立微分方程式

$$\frac{\partial E}{\partial x_m} = \frac{\partial E}{\partial y_m} = 0 \quad (6)$$

の解 (x_m, y_m) を 2 次元の Newton-Raphson 法で求める

Step 2-3: 得られた解 (x_m, y_m) を節点 p_m の新しい位置とする

Step 3: グラフを描画する

2.3 Eades のアルゴリズムの改良

Eades のアルゴリズムでは隣接節点間のバネの力に対数関数が導入されている。本報告では、発見的に次のようにバネの力を考えてみる。節点間距離が長いとき、バネの引力をフックの法則よりは弱く、対数関数よりは強くなるようにバネの引力に無理関数を導入する。節点間距離が短いとき、バネの斥力をフックの法則や対数関数に比べて強くなるようにバネの斥力に分数関数を導入する。これは、節点間距離が長いときは無理関数が、短いときは分数関数がバネの力を支配するように、さらに節点間距離が理想距離のときバネの力が零となるように次の関数を用意してみる。

$$f_a(d) = \sqrt{\frac{d}{C_2}} - \frac{C_2}{d} \quad (7)$$

ここで、 C_2 は理想距離であり、 $d = C_2$ のとき $f_a = 0$ となる。また、関数は対数関数に対し、 d が大きいときは大きな値をとり、 d が小さいときは小さい値をとる。

²文献 [2] には、節点位置の初期化の方法として同一円上に節点を配置する方法が紹介されている。Kamada の実験では、初期位置は描画結果にあまり影響を与えないことがわかり、簡単に節点を偏りなく配置することができるので、本報告でもこの方法を採用する。

隣接していない2節点間にはたらくバネの力 f_r は次式のようにする。

$$f_r(d) = \frac{C_2}{d} \quad (8)$$

2.4 Kamada のアルゴリズムの改良

Kamada のアルゴリズムでは節点間距離が零になったときでもエネルギーが有限の値をとる。一方、実際の物理学では普通はその距離が零に近づくとき、その系のエネルギーは無限大に発散する。このような差が生じるのは、Kamada のアルゴリズムではエネルギー関数はフックの法則に従ったバネの力だけを考えているからである。すなわち、エネルギー E を $|p_i - p_j|$ で微分すると力 f になるという関係があり、フックの法則では

$$f = \sum_{i < j} k_{ij} (|p_i - p_j| - l_{ij}) \quad (9)$$

であるので、 $|p_i - p_j| = 0$ のときでもエネルギー E は有限となる。

そこで、節点間のバネの力に斥力を加えることで、節点間距離が零のときにエネルギーが発散するような関数を用意してみる。バネの力 f を

$$f = k_{ij} \left(\frac{|p_i - p_j|}{l_{ij}} - \frac{l_{ij}}{|p_i - p_j|} \right) \quad (10)$$

で表す。エネルギー関数 E は f を $|p_i - p_j|$ で積分して次式で表される。

$$E = \sum_{i < j} k_{ij} \left(\frac{1}{2} \frac{|p_i - p_j|^2}{l_{ij}} - l_{ij} \log |p_i - p_j| \right) \quad (11)$$

この関数では節点間距離が零に近づくときエネルギーは無限大に発散する。

3 描画アルゴリズムの定量的評価

文献 [3] には、見やすさの一般的な基準として次のようなものが挙げられている。

- 描画の対称性
- 枝の交差数
- 枝の折れ線数
- 枝の長さの一様性
- 枝の長さの総和
- 節点分布の一様性

- 描画面積

また、文献 [5] には次のような基準が挙げられている。

- 描画の対称性
- 枝の交差数
- 枝の折れ線数
- 枝の長さの一様性
- 節点分布の一様性

本報告では、直線グラフのみを評価対象としているので「枝の折れ線数」を、定量化が困難なことから「描画の対称性」を、それぞれ評価基準から外し、以下のように5つの基準を考える。

- 枝の交差数
- 枝の長さの一様性
- 枝の長さの総和
- 節点分布の一様性
- 描画面積

3.1 枝の交差数

枝の交差数が多いと節点のつながりが分かりにくくなるので、これを評価することはグラフの見やすさを評価することになる。具体的な評価法はグラフの中の全ての枝を総当たりで調べ、交差数を数えていく。

一般的には枝の交差数が少ないほどグラフは見やすいが、必ずしも枝の交差数が少ないからといって見やすいグラフだとは限らず、見やすさを決定づけるものではない。だが、他の評価基準とともに用いて総合的な評価をするときに役立つ。

3.2 枝の長さの一様性

枝の長さが一様であると、グラフのバランスがよくなるのでこれを評価することはグラフの見やすさを評価することになる。

枝の長さの一様性を調べるには、枝の長さに対する分散を求めればよい。枝の本数を n 、各枝の長さを $x_i (i = 1, \dots, n)$ 、枝の平均の長さを \bar{x} とすると枝の長さに対する分散 σ^2 は、次式のようになる。

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (12)$$

枝の長さの一様性は、得られた σ^2 の大小で評価する。 σ^2 が小さいと、枝の長さが一様であることがわかる。

3.3 枝の長さの総和

枝の長さの総和が短いと、節点のつながりが分かりやすくなり、グラフは見やすくなる。したがって、これを評価することはグラフの見やすさを評価することになる。

ただし、例えば相似なグラフにおいては、描画面積の小さいグラフのほうが枝の長さの総和は当然短くなるので、比較するときは相対的な長さというものを考える必要がある。本報告では、各節点間距離の中で最短の長さを単位長さとして、相対的な枝の長さの総和を比較の基準とする。

3.4 節点分布の一様性

節点分布がある部分に片寄っていると、その部分が混みあってグラフがアンバランスになるので、節点分布の一様性を評価することはグラフの見やすさを評価することになる。

本報告では、描画平面を格子状に区切り、その中にある節点数について考える。節点分布が完全に一樣ならば、一つの格子には全節点数/全格子数だけの節点が存在する。節点分布の一様性については、各格子内の節点数の分散を求めることにする。この評価法では、各格子上のどの部分に節点片寄っているかは分からない。また、格子の区切り方が大変細かい場合、異なるグラフから得られる分散の差異が少なくなり、評価が困難となる。逆に、格子の区切り方が大変粗い場合でも、得られる分散が小さくなるので、同様に評価が困難になるので適切に格子を区切らなければならない。以上のような欠点があるが、この評価法は計算が簡単であるので本報告では採用する。

3.5 描画面積

描画面積を比較する場合、各アルゴリズムによって絶対的な描画面積は異なるので、相対的な描画面積というものを考える必要がある。本報告では、描画平面上の各節点間距離の中で最短の長さを単位長さとしたときの相対的な描画面積について比較を行う。描画面積は枝の長さの総和や節点分布に関係するので、これを比較することはグラフの見やすさを評価することになる。

描画面積を決定するには、グラフを囲む凸包の面積を求めるのが一番良い。だが、計算が困難であること、及びグラフは実際には決まった大きさの矩形内に描画することが多いことから次のような面積を採用する。“描画平面上のグラフを囲

む最小の矩形の面積”を、グラフの描画面積とする。評価方法はこのようにして得られた描画面積が小さいほど見やすいグラフであると考えられる。

4 各描画アルゴリズムの評価

各描画アルゴリズムを、節点数と枝数を指定して乱数によって生成されたグラフに適用した結果を以下に示す。なお、より一般的な傾向をみるために、100個のグラフについて得られた結果の平均をとった。

図1から図4は、節点数30、枝数60のグラフの描画結果の一例である。このように図を見ただけでは、どのアルゴリズムが優れているかは判断しにくい。

表1は、グラフの大小を変化させたときの描画結果である。この表から、グラフが大きくなるにつれてEadesのアルゴリズムによる描画は極端に悪くなること、また、グラフの大小に関わらず全体的に改良Kamadaのアルゴリズムによる描画はよいこと、などがわかる。

表2は、グラフの疎密を変化させたときの描画結果である。この表から、グラフが密になるにつれてEadesのアルゴリズム及び改良Eadesのアルゴリズムによる描画は極端に悪くなること、また、グラフの疎密に関わらず全体的に改良Kamadaのアルゴリズムによる描画はよいこと、などがわかる。

5 結論

本報告では、スプリングモデルに基づいたグラフの描画アルゴリズムについて改良を行い、グラフの見やすさに対する定量的な評価を行った。グラフの「見やすさ」というのは簡単には定義することは出来ないが、本報告で提案した5つの評価基準を用いることにより、グラフの見やすさについてある程度定量的な評価を行うことができた。しかし、本報告での評価法にはまだ次のような問題が残っている。

- グラフの見やすさを評価する重要な基準である「描画の対称性」を定量的に表すことができなかった。
- 定量的評価により得られた数値の総合的な判断を、一元的に評価することができなかった。

謝辞

本論文の執筆に当たって適切な御指導を頂いた、大阪大学通信工学科、前田肇教授に深く感謝の意を表します。

参考文献

- [1] P.Eades: A heuristic for graph drawing, Congressus Numerantium, vol.42, pp.149-160(1984).
- [2] T.Kamada: VISUALIZING ABSTRACT OBJECTS AND RELATIONS, World Scientific(1989).
- [3] 鎌田富久: グラフ描画アルゴリズム, bit, Vol.23, No.3, pp.284-290 (March 1991).
- [4] Thomas.M.J.Fruchterman and Edward.M.Reingold: Graph Drawing by Force-directed Placement, Software-Practice and Experience, vol.21, No.11, pp.1129-1164(November 1991).
- [5] P.Eades and R.Tamassia: Algorithms for Drawing Graphs: An Annotated Bibliography, Networks, to appear.
- [6] James.A.McHugh: Algorithmic Graph Theory, Prentice Hall (1990).
- [7] 三浦巧, 田尾陽一 共訳: 計算機のための数値計算法概論, サイエンス社 (1972).
- [8] 野下浩平, 星守, 佐藤創, 田口東 共訳: アルゴリズム第2巻, 近代科学社 (1992).

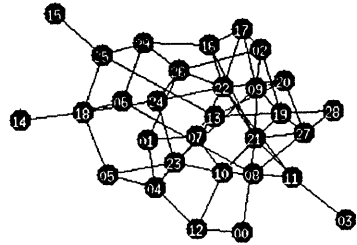


図 1: Eades による描画

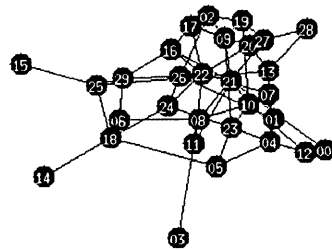


図 2: 改良 Eades による描画

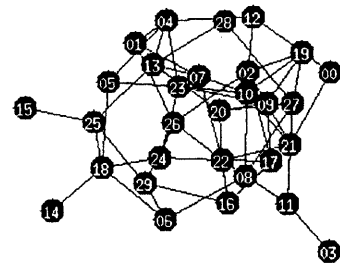


図 3: Kamada による描画

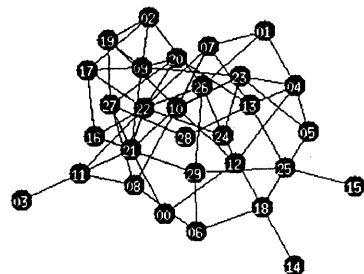


図 4: 改良 Kamada による描画

表 1: グラフの大小を変化させたときの定量的評価

評価項目	Eades	改良 Eades	Kamada	改良 Kamada
節点数 10、枝数 20 のグラフの 100 回平均				
枝交差数	4.560000	4.040000	4.340000	4.870000
枝長分散	0.018266	0.024524	0.010408	0.010103
枝長総和	29.678041	35.920642	32.062723	28.884233
節点分散	0.316000	0.482400	0.301600	0.294400
描画面積	17.170756	27.852539	19.161652	14.804292
節点数 20、枝数 40 のグラフの 100 回平均				
枝交差数	22.420000	21.940000	22.160000	24.040000
枝長分散	0.017906	0.022802	0.007997	0.008267
枝長総和	81.654369	116.072743	96.354557	79.772763
節点分散	0.869600	1.473600	0.860000	0.880800
描画面積	60.117084	157.611831	77.956587	50.478493
節点数 30、枝数 60 のグラフの 100 回平均				
枝交差数	54.190000	52.580000	54.070000	56.980000
枝長分散	0.019850	0.021664	0.007387	0.007502
枝長総和	454.453030	219.414309	184.811619	144.132391
節点分散	3.546400	3.280800	1.938400	1.816800
描画面積	611.545373	335.038058	165.078152	95.700953
節点数 40、枝数 100 のグラフの 100 回平均				
枝交差数	221.02000	201.00000	221.65000	239.56000
枝長分散	0.028472	0.024492	0.012439	0.012249
枝長総和	2982.7467	433.03363	402.51071	307.51602
節点分散	7.924000	6.007200	2.972800	2.972000
描画面積	15612.014	474.15550	253.92643	141.09500
節点数 50、枝数 200 のグラフの 100 回平均				
枝交差数	1844.3600	1339.9900	1490.3400	1564.3200
枝長分散	0.060304	0.039243	0.024392	0.024014
枝長総和	297477.14	960.76772	1054.4345	760.38913
節点分散	80.200000	7.655200	4.320000	4.252000
描画面積	61437581	370.67581	294.84442	149.97380

表 2: グラフの疎密を変えたときの定量的評価

評価項目	Eades	改良 Eades	Kamada	改良 Kamada
節点数 40、枝数 100 のグラフの 100 回平均				
枝交差数	221.02000	201.00000	221.65000	239.56000
枝長分散	0.028472	0.024492	0.012439	0.012249
枝長総和	2982.7467	433.03363	402.51071	307.51602
節点分散	7.924000	6.007200	2.972800	2.972000
描画面積	15612.014	474.15550	253.92643	141.09500
節点数 40、枝数 200 のグラフの 100 回平均				
枝交差数	2124.2600	1589.2000	1803.2600	1903.7500
枝長分散	0.071041	0.054765	0.036118	0.035055
枝長総和	289944.56	814.84736	961.25884	719.26612
節点分散	48.472000	3.330400	2.144000	2.216000
描画面積	5974804.5	179.00992	183.47484	99.502383
節点数 40、枝数 300 のグラフの 100 回平均				
枝交差数	5624.3300	4726.8900	5263.2100	5496.1700
枝長分散	0.076993	0.079759	0.048686	0.046539
枝長総和	703917.52	3953.9336	1608.0311	1170.8290
節点分散	48.431200	2.171200	2.300000	2.508000
描画面積	59067734	2360.2677	176.02960	93.17781
節点数 40、枝数 400 のグラフの 100 回平均				
枝交差数	11115.670	10798.800	10884.450	11571.570
枝長分散	0.081114	0.091228	0.076607	0.067043
枝長総和	1498440.9	16630.603	2308.6717	1675.7567
節点分散	49.510400	5.094400	1.316800	1.817600
描画面積	575850270	107788.80	180.97523	93.992468
節点数 40、枝数 500 のグラフの 100 回平均				
枝交差数	19324.600	19970.700	19087.230	20351.360
枝長分散	0.079887	0.093683	0.082818	0.076737
枝長総和	3973532.4	24376.525	3026.7338	2181.6247
節点分散	49.202400	7.468000	1.600800	1.892800
描画面積	15881439643	35372.405	179.67745	90.935083