

## グラフ描画アルゴリズムの性能評価

その他のタイトル	Performance Evaluations of Graph Drawing Algorithms
著者	角 浩二, 田中 寿俊, 榎原 博之, 中野 秀男
雑誌名	電子情報通信学会論文誌A
巻	J79-A
号	3
ページ	680-686
発行年	1996-03
権利	copyright (C)1996 IEICE
URL	<a href="http://hdl.handle.net/10112/1873">http://hdl.handle.net/10112/1873</a>

グラフ描画アルゴリズムの性能評価

角 浩二<sup>†</sup> 田中 寿俊<sup>†\*</sup> 榎原 博之<sup>††</sup> 中野 秀男<sup>††</sup>

Performance Evaluations of Graph Drawing Algorithms

Koji SUMI<sup>†</sup>, Hisatoshi TANAKA<sup>†\*</sup>, Hiroyuki EBARA<sup>††</sup>, and Hideo NAKANO<sup>††</sup>

あらまし 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、点と線で表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できない。そこで本論文では、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。更に、一般グラフを描画する、スプリングモデルに基づいた二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。

キーワード グラフ、レイアウト、スプリングモデル、性能評価

1. ま え が き

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および、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必要がある。そこでは、単にグラフを描画させるだけでなく、人間が理解しやすいように「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[10]。

本論文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。さまざま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できない。

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。本論文ではまず、この二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],[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

更に、EadesとKamadaのアルゴリズムの改良版を提案し、それらの四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。

<sup>†</sup> 大阪大学工学部通信工学科, 吹田市  
Faculty of Engineering, Osaka University, Suita-shi, 565  
Japan

<sup>††</sup> 関西大学工学部情報処理教室, 吹田市  
Faculty of Engineering, Kansai University, Suita-shi, 564  
Japan

<sup>†††</sup> 大阪市立大学工学部, 大阪市  
Faculty of Engineering, Osaka City University, Osaka-shi,  
Japan

\* 現在, 塩野義製薬株式会社

## 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、枝をばねに見立て、ばねの力学系を安定にす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、対数的なばね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ばねの力に適用すると、実験の結果、2 節点間距離が長いときにばねの力が強すぎたからである。このばねの力  $f_a$  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。このばねの力  $f_r$  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<sup>(注1)</sup>

Step 2: 以下の操作を  $M$  回繰り返す ( $M$ : 適当な定数)

すべての節点に対し

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。本論文では、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、 $M = 200$  ですべての描画を行うことにする。

### 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。 $k_{ij}$  はばね定数で、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、本論文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。従って、本論文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位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。

(注 1): 2.2 脚注参照。

Step 1: 節点をランダムにばらまく<sup>(注2)</sup>

Step 2: 理想値からのずれを示す  $\Delta_i$  の中で最大の値が十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:  $\Delta_i$  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ばねの力に対数関数が導入されている。本論文では、発見的に次のようにばねの力を考えてみる。節点間距離が長いとき、ばね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ばねの引力に無理関数を導入する。節点間距離が短いとき、ばね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ばね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ばねの力を支配するように、更に節点間距離が理想距離のときばね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。

隣接していない 2 節点間に働くばねの力  $f_r$  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ばね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、節点間のばね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。ばねの力  $f$  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。

なお、Kamada のアルゴリズムの改良は、文献[9]において同様の方法で、独立に提案されている。また、この文献では枝と節点の collision を回避するために、枝と節点の間に斥力を定義するということが提案されている。

### 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。

- (a) 描画の対称性
- (b) 枝の交差数
- (c) 枝の折れ線数
- (d) 枝の長さの一律性
- (e) 枝の長さの総和
- (f) 節点分布の一律性
- (g) 描画面積

(注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、本論文でもこの方法を採用する。

また、文献 [5] には先に挙げた基準のうち、(a)(b)(c)(d)(f) が挙げられている。

本論文では、直線グラフのみを評価対象としているので (c) を、定量化が困難なことから (a) を、それぞれ評価基準から外し、(b)(d)(e)(f)(g) の五つの基準を考える。

### 3.1 枝の交差数

枝の交差数は、具体的にはグラフの中のすべての枝を総当りで調べる。

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、他の評価基準と共に用いて総合的な評価をするときに役立つ。

### 3.2 枝の長さの一意性

枝の長さの一意性を調べるには、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  $\sigma^2$  は、次式ようになる。

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

枝の長さの一意性は、得られた  $\sigma^2$  の大小で評価する。 $\sigma^2$  が小さいと、枝の長さが一様であることがわかる。

但し、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ばらつきは当然小さくなるので、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。本論文では、各節点間距離の中で最短の長さを単位長さとして、相対的な枝の長さに対する分散を比較の基準とする。

### 3.3 枝の長さの総和

この比較項目においても、相対的な長さというものを考える必要がある。「枝の長さの一意性」の場合と同様に、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 節点分布の一意性

本論文では、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様ならば、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。節点分布の一意性については、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が偏っているかはわからない。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論文では採用する。

### 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。本論文では、「枝の長さの一意性」「枝の長さの総和」の場合と同様に、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よい。だが、計算が困難であること、およびグラフは実際には決まった大きさの方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方形の面積」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。なお、表 1 から表 3 の項目の中にある「実行時間」とは、ワークステーション NEWS 上で節点の配置のみを計算するために必要とする CPU 時間 (秒) である。

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、図 1 から図 4 の定量評価である。この表からは、改良 Kamada のアルゴリズムが五つの評価項目のうち三つの項目で最も良い値をとっていることがわかる。それ故、このアルゴリズムがよいと判断できる。

表 2 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、グラフの大小にかかわらず、「枝の交差数」を除く項目において最も良い値をとっていることから改良

Kamada のアルゴリズムによる描画は良いこと、などがわかる。

表 3 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 のアルゴリズムおよび改良 Eades のアルゴリズムによる描画は極端に悪くなること、グラフの疎密にかかわらず、「枝の交差数」「節点の分散」を除く項目において最も良い値をとっていることから改良 Kamada のアルゴリズムによる描画は良いこと、などがわかる。

実行時間については短い順に、Eades, 改良 Eades, Kamada, 改良 Kamada である。これは前者二つはすべての節点を一度に動かすのに対し、後者二つは一つの節点だけを一度に動かすことにより実行時間の差が生じている。更に、各改良アルゴリズムはエネルギーの計算式が複雑になったため、実行時間が長くなった。

### 5. む す び

本論文では、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できないが、本論文で提案した五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

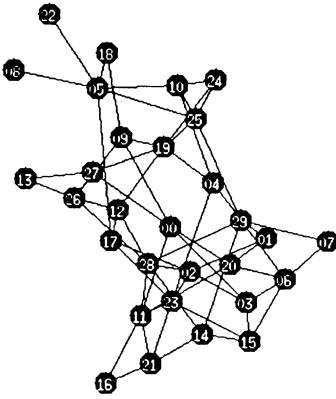


図 1 Eades による描画  
Fig.1 Layout by Eades' algorithms.

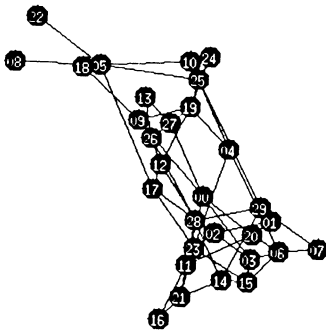


図 2 改良 Eades による描画  
Fig.2 Layout by revised Eades' algorithms.

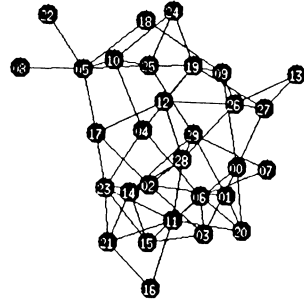


図 3 Kamada による描画  
Fig.3 Layout by Kamada's algorithms.

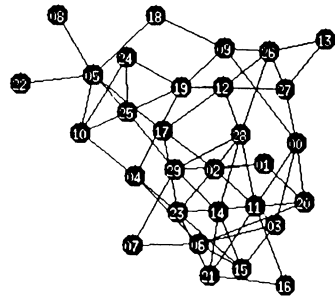


図 4 改良 Kamada による描画  
Fig.4 Layout by revised Kamada's algorithms.

表 1 図 1 から図 4 の定量的評価  
Table 1 Quantitative evaluations of Fig.1 to Fig.4.

評価項目	Eades	改良 Eades	Kamada	改良 Kamada
枝交差数	56	48	58	53
枝長分散	0.815326	3.206545	0.659629	0.374609
枝長総和	138.37597	197.44418	186.66950	148.00756
節点分散	2.400000	4.000000	2.240000	2.080000
描画面積	100.26446	202.621000	167.98009	91.88925
実行時間 (秒)	2.0000000	2.0000000	7.0000000	6.0000000

表2 グラフの大小を変化させたときの定量的評価  
Table 2 Quantitative evaluations with changing graph size.

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	5.070000	4.200000	4.550000	5.260000
枝長分散	0.206906	0.558691	0.127143	0.102421
枝長総和	30.896656	36.569877	33.091045	30.503388
節点分散	0.288000	0.408800	0.292800	0.294400
描画面積	17.407465	28.872189	18.934996	15.342599
実行時間 (秒)	0.1500000	0.2000000	0.1300000	0.1900000
節点数 20, 枝数 40 のグラフの 100 回平均				
枝交差数	22.680000	21.160000	21.180000	24.640000
枝長分散	0.543123	2.224064	0.340924	0.221661
枝長総和	81.961859	116.93290	96.973018	81.237831
節点分散	0.920000	1.592000	0.908800	0.862400
描画面積	65.046500	173.73007	83.21661	54.198264
実行時間 (秒)	0.7000000	0.7700000	1.4900000	2.0000000
節点数 30, 枝数 60 のグラフの 100 回平均				
枝交差数	56.500000	50.520000	52.820000	54.960000
枝長分散	1.010343	4.589145	0.613578	0.338943
枝長総和	146.77522	230.24822	183.29955	140.34952
節点分散	1.897600	3.395200	1.952000	1.772800
描画面積	116.03080	388.68289	166.78318	92.963264
実行時間 (秒)	2.7000000	3.0000000	10.700000	21.500000
節点数 40, 枝数 100 のグラフの 100 回平均				
枝交差数	210.12000	202.35000	215.65000	232.69000
枝長分散	316826.45	6.060524	1.433767	0.742814
枝長総和	4012.0373	436.95977	396.91680	300.40363
節点分散	7.477600	6.199200	3.176000	3.031200
描画面積	7981.5398	474.58003	249.93974	137.32579
実行時間 (秒)	4.8000000	5.8000000	36.000000	53.200000
節点数 50, 枝数 200 のグラフの 100 回平均				
枝交差数	1832.1500	1323.8300	1475.0300	1556.8100
枝長分散	8565147705	6.958049	3.376641	1.679072
枝長総和	796887.49	950.587026	1032.01523	748.120464
節点分散	82.405600	7.031200	4.216800	4.032000
描画面積	125113319	363.549024	282.840472	144.214199
実行時間 (秒)	7.7000000	8.8000000	110.80000	144.20000

うことができた。しかし、本論文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** 本論文の執筆にあ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。また、多くの有益なコメントをして頂いた査読者の方々に感謝致します。

文 献

[1] P. Eades, "A heuristic for graph drawing," *Congressus Numerantium*, vol.42, pp.149-160, 1984.

[2] T. Kamada, "VISUALIZING ABSTRACT OBJECT-SAND 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] T.M.J. Fruchterman and E.M. Reingold, "Graph drawing by force-directed placement," *Software-Practice and Experience*, vol.21, no.11, pp.1129-1164, Nov. 1991.

[5] G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis, "Algorithms for drawing graphs: An annotated bibliography," *Computational Geometry*, vol.4, no.5, pp.235-282, 1994.

[6] J.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.

[9] 鈴木和彦, 鎌田富久, "グラフ構造理解のための単純無向グラフの自動描画アルゴリズム," 第35回プログラムシンポジウム, pp.71-82, 1994.

[10] 杉山公造, "グラフ自動描画法とその応用-ビジュアル

表3 グラフの疎密を変えたときの定量的評価  
Table 3 Quantitative evaluations with changing graph density.

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	210.12000	202.35000	215.65000	232.69000
枝長分散	316826.45	6.060524	1.433767	0.742814
枝長総和	4012.0373	436.95977	396.91680	300.40363
節点分散	7.477600	6.199200	3.176000	3.031200
描画面積	7981.5398	474.58003	249.93974	137.32579
実行時間 (秒)	4.8000000	5.8000000	36.000000	53.200000
節点数 40, 枝数 200 のグラフの 100 回平均				
枝交差数	2087.7400	1596.5300	1780.1900	1887.7800
枝長分散	1034646822	4.450763	3.011927	1.585792
枝長総和	308205.10	808.46395	961.55722	712.02474
節点分散	49.100000	3.199200	2.277600	2.340000
描画面積	13947709	174.68614	182.92039	98.703564
実行時間 (秒)	5.0000000	6.3000000	38.900000	62.800000
節点数 40, 枝数 300 のグラフの 100 回平均				
枝交差数	5531.5400	4745.7800	5282.2200	5546.9200
枝長分散	1303368348	78.612193	3.983242	2.199005
枝長総和	1117008.9	4069.07580	1558.1384	1182.9988
節点分散	50.481600	2.348000	2.222400	2.433600
描画面積	6573271301	2551.53018	165.86291	95.614054
実行時間 (秒)	5.1000000	6.5000000	42.500000	67.100000
節点数 40, 枝数 400 のグラフの 100 回平均				
枝交差数	11203.900	10743.480	10825.680	11501.700
枝長分散	4656499290	4749.14920	5.287049	2.693683
枝長総和	2773833.2	13734.104	2285.05809	1669.8360
節点分散	50.617600	6.528000	1.336800	1.831200
描画面積	12277107558	22041.271	178.58061	93.017057
実行時間 (秒)	5.2000000	7.3000000	45.100000	71.800000
節点数 40, 枝数 500 のグラフの 100 回平均				
枝交差数	20016.860	19971.440	19024.650	20305.910
枝長分散	8552431826	1239.2326	6.191679	3.0478123
枝長総和	4143849.0	20714.127	3028.1376	2169.1232
節点分散	46.467200	8.585600	1.583200	1.8192003
描画面積	5463563100	23286.7153	180.86207	89.875279
実行時間 (秒)	5.5000000	8.3000000	53.900000	79.300000

“ヒューマンインタフェースのための基礎技法,” 計測自動制御学会出版, 1993.

(平成 7 年 4 月 17 日受付, 8 月 21 日再受付)

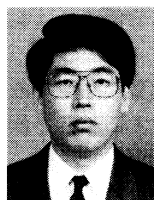
### 角 浩二 (学生員)

平 6 阪大・工・通信卒。現在、同大学院修士課程在学中。グラフィックアウトに関する研究に従事。



### 田中 寿俊

平 5 阪大・工・通信卒。平 7 同大学院修了。同年塩野義製薬株式会社。在学中、グラフィックアウトに関する研究に従事。



### 榎原 博之 (正員)

1958 生。1982 阪大・工・通信卒。1987 同大学院博士 (通信) 課程了。同年阪大工学部助手。1994 年関西大工学部専任講師となり現在に至る。組合せ最適化問題、計算幾何学、並列アルゴリズム等の研究に従事。工博。IEEE, ACM 各会員。

### 中野 秀男 (正員)

昭 45 阪大・工・通信卒。昭 50 同大学院博士 (通信) 課程了。同年阪大通信工学科助手。平 3 阪大助教授。平 7 大阪市立大学教授となり現在に至る。離散最適化問題の近似解法の評価、情報セキュリティ、ソフトウェア工学等の研究に従事。工博。情報処理学会、日本 OR 学会、ソフトウェア科学会、IEEE, ACM 各会員。

