

遺伝的アルゴリズムを用いたグラフ自動描画

角 浩二^{†*} 榎原 博之^{††} 中野 秀男^{†††} 前田 肇[†]

Automatically Graph Drawing Using Genetic Algorithm

Koji SUMI^{†*}, Hiroyuki EBARA^{††}, Hideo NAKANO^{†††}, and Hajime MAEDA[†]

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。一般グラフの描画アルゴリズムの一つである Kamada アルゴリズムは、グラフについて系のエネルギーを定義することにより、レイアウトの問題をエネルギーの最小化の問題に帰着している。しかし、最小化の手法は Newton-Raphson 法であり局所解を求めているにすぎない。そこで、本論文では遺伝的アルゴリズムを用いてより良い最適解を求めることを試みる。

キーワード 遺伝的アルゴリズム、グラフ、自動描画、スプリングモデル

1. ま え が き

近年のコンピュータ技術の発達により、さまざまな図をディスプレイ上に描画するという需要が高まってきている。これらの図の中には、ネットワーク、電気回路図、フローチャートなど、グラフに抽象化できるものが多く存在する。そこでは、人間が理解しやすいように「見やすい」グラフを自動的に描画させる必要がある。

さまざまなグラフに対する描画アルゴリズムについては、文献[1]で述べられている。木、平面グラフ、階層グラフなどといった特別な構造をもつグラフに対しては、これらのグラフを見やすく描画するアルゴリズムは存在しているが、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。

一般グラフの描画アルゴリズムの一つである Kamada アルゴリズムは、スプリングモデルに基づいたアルゴリズムを提案している[2]~[4]。このアルゴリズムは、グラフについて系のエネルギーを定義することにより、レイアウトの問題をエネルギーの最小化の問題に帰着している。このアルゴリズムは、エ

ネルギーの最小化の手法として、最急降下法である Newton-Raphson 法を用い、エネルギーの局所最適解を求めている。本論文では、より良い最適解を求めるために遺伝的アルゴリズム (GA) を用い、そのときのグラフの描画の構造がどのようになるかを文献[5]の方法を用いて、定量的な評価を行う。

2. スプリングモデル

一般グラフ (無向で多重辺やループのないグラフ) の自動描画法は問題の困難さのためあまり多く提案されていない。これまでに提案された主な描画アルゴリズムとしては、グラフ理論に基づくアルゴリズム[6]とスプリングモデルに基づくアルゴリズム[2]~[5],[7]~[11]などがある。本論文では、スプリングモデルに基づくアルゴリズムについて述べる。

2.1 スプリングモデルのアルゴリズム

スプリングモデルとは、一般グラフの節点を鉄のリング、枝をスプリングとしてとらえ、力学的モデルの系が安定になるように節点を配置するものである。

スプリングモデルに基づいたアルゴリズムを最初に提案したのは Eades[8]であり、spring embedder とされるものである。spring embedder は、ある初期状態のレイアウトから、リング上のスプリングの力によって節点を移動させていく。

Kamada [2]~[4] は Eades のアルゴリズムを改良したアルゴリズム (Kamada アルゴリズム) を提案した。Eades は節点は隣接節点の近くにあることだけが

[†] 大阪大学工学部通信工学科, 吹田市

Faculty of Engineering, Osaka University, Suita-shi, 565 Japan

^{††} 関西大学工学部, 吹田市

Faculty of Engineering, Kansai University, Suita-shi, 564 Japan

^{†††} 大阪市立大学学術情報総合センター, 大阪市

Media Center, Osaka-City University, Osaka-shi, 558 Japan

* 現在, 松下電器産業株式会社

重要であるとしたが、Kamada は隣接しない節点間にも理想距離というグラフ理論上の距離を導入した。理想距離とは節点間の最短経路長であるとし、アルゴリズムは節点間距離ができるだけ理想距離になるように節点を動かす。また、Eades アルゴリズムが各々の節点にかかる合力を評価するのに対し、Kamada 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（あるいは極小）になるように節点を動かすものである。

この系のエネルギー E を定式化すると次のようになる。

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{2} k_{ij} (|p_i - p_j| - l_{ij})^2 \quad (1)$$

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 間の理想距離に比例した値である。 k_{ij} はスプリング定数で、理想距離の 2 乗に反比例させている。このとき、エネルギー E はレイアウトの不均衡の程度とみなすことができ、目指すべきレイアウトを得るには、 E の最小を求めることに帰着される。

その他、エネルギーの定義を行わず、力の相互作用により節点の配置を決めていくという手法 [9] や、「グラフの美しさの基準」を系のエネルギーに付加し、最小値（最適解）を simulated annealing 法で求める手法 [10] などが提案されている。

2.2 Kamada アルゴリズム

前節で挙げたグラフの自動描画アルゴリズムの中では、描画結果の良さから Kamada アルゴリズムがよく研究されている [5], [7], [11]。

Kamada アルゴリズムは前節で述べたように、式 (1) で定義された E が最小（あるいは極小）になるように節点を動かしていく。このとき、Newton-Raphson 法により E の局所最適解を求める。具体的な手法は以下のとおりである。

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。

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

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。

全体の手続きを以下に示す。

- (1) 節点をランダムにばらまく
- (2) 理想値からのずれを示す Δ_i の中で最大の値が十分小さくなるまで繰り返す

$$\Delta_i = \sqrt{\left(\frac{\partial E}{\partial x_i}\right)^2 + \left(\frac{\partial E}{\partial y_i}\right)^2} \quad (i = 1, 2, \dots, n) \quad (3)$$

(2-1) Δ_i の中で最大値を与える節点 p_m を選ぶ

(2-2) 連立微分方程式

$$\frac{\partial E}{\partial x_m} = \frac{\partial E}{\partial y_m} = 0 \quad (4)$$

を Newton-Raphson 法で解く

(2-3) 得られた解を p_m の新しい位置とする

(3) グラフを描画する

2.3 改良 Kamada アルゴリズム

Kamada 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。これは Kamada 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スプリングの力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_i になるという関係があり、フックの法則では

$$f_i = \sum_{i < j} k_{ij} (|p_i - p_j| - l_{ij}) \quad (5)$$

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。このような場合、1本の枝の長さは短くなってしまふ可能性がある。それ故、描画結果において節点が密集してしまうことがある。

そこで、節点間のスプリング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。スプリングの力 f を

$$f = \sum_{i < j} k_{ij} \left(\frac{|p_i - p_j|}{l_{ij}} - \frac{l_{ij}}{|p_i - p_j|} \right) \quad (6)$$

で表す。エネルギー関数 E は f を $|p_i - p_j|$ で積分することによって、次式で表される。

$$\begin{aligned} E &= \sum_{i < j} \int_{l_{ij}}^{|p_i - p_j|} k_{ij} \left(\frac{x}{l_{ij}} - \frac{l_{ij}}{x} \right) dx \quad (7) \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^n k_{ij} \left\{ \frac{1}{2} \frac{(|p_i - p_j|^2 - l_{ij}^2)}{l_{ij}} \right. \\ &\quad \left. - l_{ij} (\log |p_i - p_j| - \log l_{ij}) \right\} \quad (8) \end{aligned}$$

この関数では節点間距離が零に近づくとエネルギーは無限大に発散する。

このように定義されたエネルギーを用いて、Kamada アルゴリズムと同様に E が最小 (あるいは極小) になるよう節点を動かしていくアルゴリズム [5],[7] を改良 Kamada アルゴリズムと呼ぶ。

3. 遺伝的アルゴリズム

遺伝的アルゴリズム (GA) [12] は、自然界における生物進化 (集団遺伝) の原理に着想を得た確率的探索・最適化の一手法である。

3.1 GA の基本構造と操作

まず、解候補を遺伝子的な表現へ、目的関数を評価関数へ変換することで、問題と GA との対応させる。一般には、遺伝子列は 2 進数列で表現するが、他の表現が有効である場合もある。

次に、ランダムに生成された解の集団を初期集団として、選択、交叉、突然変異という GA オペレータを繰り返し適用することで、評価値の高い解が多く含まれていく。選択は、集団における適応度に従って次世代の個体群を確率的に決定するものである。また、交叉は二つの個体間で染色体を組み合わせることによって新しい個体を生成するものであり、突然変異はある確率で遺伝子の値を変えるものである。

また、GA を走らせるにはいくつかのパラメータが必要であり、GA の振舞いを決定するため適切な値に設定することが重要である。

3.2 ハイブリッド GA

一般に GA は最適解の周辺には早く近づくが、局所探索能力が弱いという問題が指摘されている。この問題を解決する有効な手段の一つが、ヒューリスティックアルゴリズムと GA を組み合わせるハイブリッド GA というものである。ハイブリッド GA では、大域的探索を GA で行い、局所探索をヒューリスティックアルゴリズムで行う。

GA のみで探索を行う場合は、探索空間が非常に広く連続的なので、最適解を見つけるのは困難であるが、ハイブリッド GA では、局所探索によって得られた局所解の集合に対して遺伝的操作を行うことになる。従って、GA の探索空間は有限個の局所解となり、その中から最適解を見つけ出すのは、困難ではないのである。

4. 遺伝的アルゴリズムのグラフ描画への適用

2. で挙げた改良 Kamada アルゴリズムは、Newton-Raphson 法によりスプリングエネルギーの局所解を求めているため、大域的な最適解は見つけてはいない。従って GA を利用することにより、より大域的に解を探索できる。

本章では、GA をどのようにしてグラフ描画に適用するか述べていく。

なお、GA を用いたグラフ描画には有向グラフの描画法 [12],[13]、無向グラフの描画法 [14],[15] などが提案されている。前者においては配置に関する制約を評価基準とし、後者においては枝の交差数や対称性などを評価基準としており、いずれも審美的要素が評価の基準として使用されている。

4.1 グラフ描画のための GA の基本構造

グラフ描画のための GA の基本構造は次のようになっている。

- (1) 初期集団の生成
- (2) 終了条件が満たされるまでループ
 - (2-1) 集団内の各個体の適応度の評価
 - (2-2) 選択
 - (2-3) 交叉
 - (2-4) 突然変異
 - (2-5) 局所探索
- (3) 最終調整

一般的な GA と比べて異なっているのは、局所探索と最終調整という新しい操作が導入されていることである。以下では、グラフ描画のために、一般的な GA に対して行った変更について述べる。

4.2 遺伝子の表現と評価関数

一般的な GA では遺伝子型の表現形式としてビット列を用いるが、グラフ描画は節点座標を中心に扱うため、各節点座標は実数表現を用い、配列に格納するという表現形式を用いる。

また、GA の評価関数は、2. で述べた改良 Kamada のエネルギー関数 (8) を用い、個体の適応度にはスケールを施す。すなわち、その世代の集団の中で、最も大きなエネルギーをもつ個体とのエネルギーの差に定数を掛けたものを適応度とすることで、エネルギーの小さい個体ほど、より大きな適応度をもつことになる。

4.3 GA オペレータの設計

GA オペレータの設計の前に、まず初期集団の生成を行う必要がある。Kamada の手法は同一円周上にランダムな順序で節点を配置するという方法をとっているが、ここでは解を集団で扱うことのできる GA の性質を有効に利用するため、同一円周という制約を外し、任意の位置へランダムに節点を配置して複数の初期状態を生成するという方法をとる。

選択に関しては、適応度最大の個体は無条件で次世代に残すエリート保存戦略を用いる。これはその時点で最も良い解が破壊されないという利点があるが、その遺伝子が集団中に急速に広がる危険もある。従って、各個体の適応度に比例した確率で子孫を残すルーレット選択と組み合わせて用いる。

更に、集団内に同じ遺伝子をもつ個体が複数存在する場合、同じ個体の重複抽出が加速されるため、その個体は 1 個しか存在しないものとするシェアリングを用いる。

具体的には、まず (個体数 - 1) 個の個体を、シェアリングされた全個体からルーレット選択で選び、残り 1 個の個体をエリート保存戦略により適応度最大の個体を選んで次世代に残す。

交叉については、遺伝子がグラフの節点の座標をそのまま表していることを考慮して、親の遺伝子の節点の位置に関連してどちらの親から遺伝子を受け継ぐかを定めるようにする。

具体的には、まず親 1 について描画平面を左平面と右平面に、節点がそれぞれの平面に同数含まれるように分ける。左平面を子 1 に右平面を子 2 に受け継がせ、残りの節点の座標を親 2 から受け継ぐ (図 1)。次に、縦方向に関して同様の手順で子を生成する。このようにすることによって、グラフの画面上の構造を次世代に残すことが可能となる。なお、交叉確率は後述する improving rate によって決定される。

突然変異については、より広範囲な探索が行えるよう、選ばれた遺伝子のすべての節点にランダムな座標値を代入する。

4.4 ハイブリッド化

各遺伝子の節点の座標に新しい値が代入されるのは、初期化の時と突然変異のときのみであるため、局所的な探索を効率良く行うことができない。そこで、GA のハイブリッド化を行う。局所探索の方法としては、2. で述べたエネルギー関数の局所解を求める Newton-Raphson 法を用い、他の GA オペレータの実

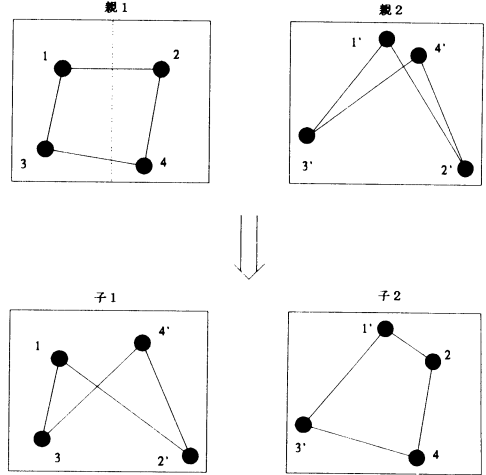


図 1 節点の座標を考慮した交叉
Fig. 1 Crossover considering the coordinates of nodes.

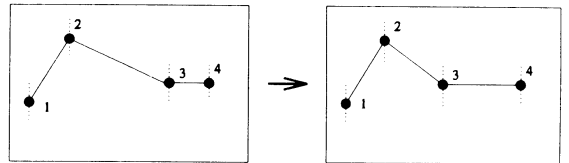


図 2 グラフの構造を考慮した局所探索
Fig. 2 Local search considering the construction of graph.

行後にすべての個体に対してハイブリッド化による局所探索を実行する。

Newton-Raphson 法を用いることにより節点の座標値は更新されるが、座標値が大きく更新された場合には、交叉で受け継いだ親の形質が破壊される可能性がある。従って、保存されたグラフの構造を維持しつつ局所探索を行う必要がある。例えば、図 2 においては、節点 3 は節点 2 と 4 の間に位置しつつ局所探索させたのである。

そこで、エネルギー関数にポテンシャルバリアというものをつ加する。具体的には、描画平面に放物線状のポテンシャルを節点数だけ等間隔に配置したもので、次式のようになる (図 3)。

$$E' = \sum_i \left(\frac{1}{n} A_x x_i'^n + \frac{1}{n} A_y y_i'^n \right) \quad (9)$$

ここで、 n はポテンシャル関数の次数であり、 A_x, A_y はパラメータである。更に、 x_i', y_i' は節点 i の格子内

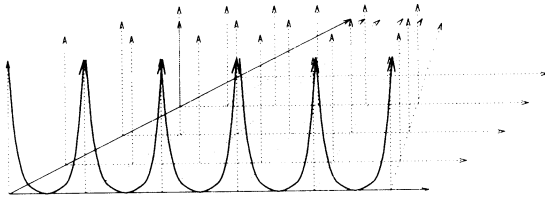


図3 ポテンシャルバリア
Fig.3 Potential barrier.

規格化座標と呼び、

$$x'_i = \frac{x_i - l_i * \text{div}_x}{\text{div}_x} \quad (10)$$

$$y'_i = \frac{y_i - m_i * \text{div}_y}{\text{div}_y} \quad (11)$$

で表され、描画平面を節点数に分割された2次元格子と見て、節点座標の小さい順に節点が1対1に対応する格子における規格化された座標を意味する。なお、 x_i, y_i は節点 i の絶対座標、 $\text{div}_x, \text{div}_y$ は格子幅、 l_i, m_i はソートされた順序である。

このようなポテンシャルバリアを導入することにより、各節点は、従来のエネルギー関数以外に、節点座標の小さい順に対応した各放物曲線の影響を受けることになる。その結果、節点の順序は維持されやすくなり、グラフの構造も保存されやすくなる。但し、ポテンシャルバリアが強すぎると局所探索能力が弱まるので、オリジナルのエネルギー関数とのバランスをとる必要がある。

Kamada アルゴリズムでは、式 (3) の Δ_i が十分小さくなるまでループを繰り返しているが、そのまま GA に適用すると、計算時間がばく大な量になるので、ある程度の回数（節点数の 1/10 から 1/2 回位）とする。

GA の終了後得られた解は、最適解付近の解である可能性がある。そこで、改良 Kamada アルゴリズムを GA の終了後に適用し、解を最適解に近づける（最終調整）。その際、計算時間を考慮し、解の良い順に選ばれた数個の個体に対し最終調整を実行する。実行後、その中で最良のものを選び最終的な最良解とする。

4.5 GA パラメータの設定

終了条件については、計算時間の無駄が生じないように、各世代の最良解が何世代にもわたってその値が改善されない場合とする。

また、GA パラメータに新しく improving rate という「解の改善率」を数値化したものを考え、

$$1 - \frac{E_{na} - E_{nb}}{E_{fa} - E_{fb}} \quad (12)$$

と定義する。 $E_{na}, E_{nb}, E_{fa}, E_{fb}$ はそれぞれ現在の平均、現在の最良、最初の平均、最初の最良の各エネルギー値を表す。解が改善されていくにつれて $E_{na} - E_{nb}$ が小さくなり、improving rate は 1 に近づく。

GA パラメータの一部（交叉確率、エリート保存戦略確率、Newton-Raphson のループ回数）はこの improving rate に支配され、解の改善が進むにつれ解の保存を高くさせるように働く。具体的には解の改善が進むにつれて、交叉確率は低く、エリート保存戦略を行うかどうかの確率は高く、Newton-Raphson のループ回数は多くなるようにしている。

5. 描画アルゴリズムの評価

本章では、2. で述べた改良 Kamada アルゴリズム（改良 Kamada）と、4. で述べた GA を用いた描画アルゴリズム（GA-改良 Kamada）でグラフの描画を行い、その定性的、定量的な評価を行う。

5.1 グラフの定量的評価

グラフの見やすさを示す一般的な基準として、文献 [5] と同様に、以下五つについて考える。

- (a) 枝の交差数
- (b) 枝の長さの一様性
- (c) 枝の長さの総和
- (d) 節点分布の一様性
- (e) 描画面積

具体的には、(b) は枝の長さに対する分散、(d) は描画平面を格子状に区切ったときの各格子内の節点数に対する分散、(e) は描画平面上のグラフを囲む最小の方形の面積である。また、(b)、(c)、(e) では相対的な長さというものを考える必要があるので、各節点間距離の中で最短の長さを単位長さとする。

5.2 描画アルゴリズムの評価

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についての評価を行う。ここで、GA パラメータは、集団数は 100、最大世代数は 900、突然変異の確率は 0.1、終了条件としては、その世代の最良値が 20 世代続いた場合、として描画を行った。

まず、節点 30、枝数 60 のランダムに生成したグラフの描画結果を図 4 と図 5 に示す。図 4 は改良 Kamada、図 5 は GA-改良 Kamada の結果をそれぞれ表している。

次に、グラフのエネルギーの比較と、見やすさの定量的な評価を行うにあたり、より一般的な傾向を見る

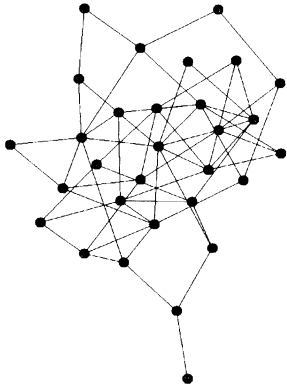


図4 改良 Kamada アルゴリズムの描画例
Fig. 4 Layout example by revised Kamada algorithm.

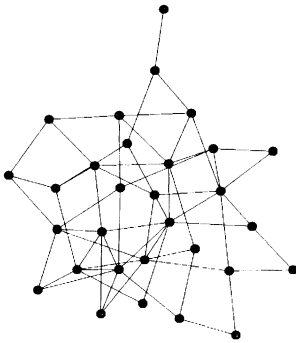


図5 GA-改良 Kamada の描画例
Fig. 5 Layout example by ga-revised Kamada algorithm.

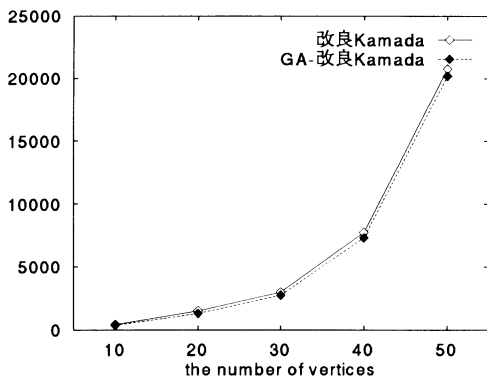


図6 グラフの大小によるエネルギー値の変化
Fig. 6 Variation of energy value with changing graph size.

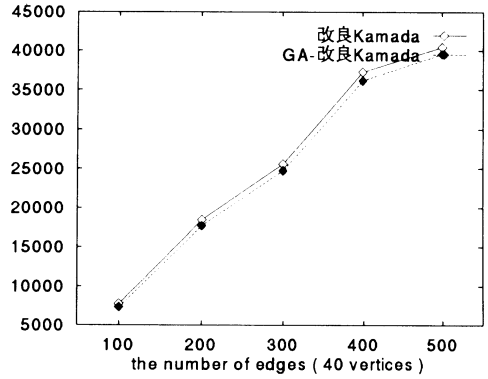


図7 グラフの疎密によるエネルギー値の変化
Fig. 7 Variation of energy value with changing graph density.

表1 グラフの大小を変化させたときの定量的評価
Table 1 Quantitative evaluations with changing graph size.

評価項目	改良 Kamada	GA-改良 Kamada
節点数 10, 枝数 20 のグラフの 100 回平均		
エネルギー	456.014880	382.826815
枝交差数	8.35	6.77
枝長分散	0.012665	0.010737
枝長総和	35.320225	34.606339
節点分散	0.244800	0.244000
描画面積	14.865232	14.915308
節点数 20, 枝数 40 のグラフの 100 回平均		
エネルギー	1522.504021	1323.153332
枝交差数	31.27	25.40
枝長分散	0.008311	0.007366
枝長総和	88.896363	84.903532
節点分散	0.439600	0.434000
描画面積	52.042449	50.664830
節点数 30, 枝数 60 のグラフの 100 回平均		
エネルギー	2997.962945	2759.664913
枝交差数	65.98	58.32
枝長分散	0.007189	0.006465
枝長総和	151.485866	145.297657
節点分散	0.812000	0.846000
描画面積	96.946388	91.678436
節点数 40, 枝数 100 のグラフの 100 回平均		
エネルギー	7759.774286	7329.30297
枝交差数	258.48	238.24
枝長分散	0.012450	0.011511
枝長総和	307.962315	301.009419
節点分散	1.228800	1.092800
描画面積	126.536213	126.009467
節点数 50, 枝数 200 のグラフの 100 回平均		
エネルギー	20765.558980	20187.723225
枝交差数	1643.42	1589.19
枝長分散	0.026296	0.024557
枝長総和	772.577365	751.172501
節点分散	1.371200	1.357600
描画面積	142.373726	138.537415

表2 グラフの疎密を変えたときの定量的評価
Table 2 Quantitative evaluations with changing graph density.

評価項目	改良 Kamada	GA-改良 Kamada
節点数 40, 枝数 100 のグラフの 100 回平均		
エネルギー	7759.774286	7329.302979
枝交差数	258.48	238.24
枝長分散	0.012450	0.011511
枝長総和	307.962315	301.009419
節点分散	1.228800	1.092800
描画面積	126.536213	126.009467
節点数 40, 枝数 200 のグラフの 100 回平均		
エネルギー	18461.117102	17713.413088
枝交差数	2026.42	1925.75
枝長分散	0.036884	0.034052
枝長総和	740.096370	716.931501
節点分散	0.804000	0.836400
描画面積	99.908082	97.125063
節点数 40, 枝数 300 のグラフの 100 回平均		
エネルギー	25600.935034	24739.753242
枝交差数	5820.41	5587.43
枝長分散	0.053865	0.051571
枝長総和	1211.594911	1191.391582
節点分散	0.764800	0.735200
描画面積	94.313509	93.800672
節点数 40, 枝数 400 のグラフの 100 回平均		
エネルギー	37266.730063	36164.134405
枝交差数	12309.12	11770.89
枝長分散	0.086927	0.084565
枝長総和	1718.182583	1706.111838
節点分散	0.736000	0.742400
描画面積	93.056001	94.236470
節点数 40, 枝数 500 のグラフの 100 回平均		
エネルギー	40441.140475	39553.437463
枝交差数	21904.75	20755.19
枝長分散	0.086254	0.084521
枝長総和	2244.918846	2282.210980
節点分散	0.642400	0.690400
描画面積	89.338593	96.497611

ために、それぞれ 100 個のグラフについて得られた結果の平均をとった。

図 6, 図 7 は、エネルギーに関するグラフであり、表 1 はグラフの大小を、表 2 はグラフの疎密を、それぞれ変化させたときのエネルギーを含む定量評価である。エネルギーという観点からは、どの場合でも GA-改良 Kamada は改良 Kamada より改善されている。定量的な評価という観点からも、全体的に見て改善されていることができる。

最後に、GA-改良 Kamada の計算時間は、改良 Kamada と比べて、GA の構造上かなりの計算時間がかかった。例えば、(節点数, 枝数) が (40,100) であるグラフの場合、実行時間は改良 Kamada では数分であるのに対し、GA-改良 Kamada は 1 時間ほどである。

6. む す び

本研究では、スプリングモデルで用いられているエネルギー関数の最適値を求めるために、ハイブリッド GA という手法を用いた。そしてグラフ描画を試み、アルゴリズムの定量的な評価を行った。エネルギーという観点からは、ハイブリッド GA を用いることによって改善がなされた。また、定量的な評価でも、ハイブリッド GA を用いることによって全体的に改善がなされた。

今後、評価関数を自由に変更できる GA の利点を生かし、節点と枝の衝突を防ぐための項目を評価関数に組み込むことによって、更に見やすいグラフの描画を行うことが課題である。

謝辞 本論文の執筆にあたって適切な助言を頂いた大阪大学工学部通信工学科通信システム工学講座、大西克実氏、田中寿俊氏（現、塩野義製薬株式会社）に深く感謝する。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深く感謝する。

文 献

- [1] 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, Oct. 1994.
- [2] 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.
- [3] T. Kamada, "Visualizing Abstract Objects and Relations," World Scientific Publishing, Singapore, 1989.
- [4] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," Information Processing Letters, vol.31, no.1, pp.7-15, 1989.
- [5] 角 浩二, 田中寿俊, 榎原博之, 中野秀男, "グラフ描画アルゴリズムの性能評価," 信学論 (A), vol.J79-A, no.3, pp.680-686, March 1996.
- [6] R. Lipton, S. North, and J. Sandberg, "A method for drawing graphs," Proc. ACM Symp. on Computational Geometry, Baltimore, U.S.A., pp.153-160, June 1985.
- [7] 鈴木和彦, 鎌田富久, 榎本彦衛, "単純無向グラフ自動描画アルゴリズム," コンピュータソフトウェア, vol.12, no.4 pp.45-55, July 1995.
- [8] P. Eades, "A heuristic for graph drawing," Congressus Numerantium, vol.42, pp.149-160, 1984.
- [9] T. Fruchterman and E. Reingold, "Graph drawing by force-directed placement," Software-Practice and Experience, vol.21, no.11, pp.1129-1164, Nov. 1991.
- [10] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing," Technical Report, Dept. of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot, Israel, July 1989.
- [11] 佐野達郎, "グラフの自動描画アルゴリズムについて," 信

学技報, HIP95-19, Nov. 1995.

- [12] 北野宏明編, “遺伝的アルゴリズム,” 産業図書, 東京, 1993.
- [13] 北野宏明編, “遺伝的アルゴリズム 2,” 産業図書, 東京, 1995.
- [14] C. Kosak, J. Marks, and S. Shieber, “A parallel genetic algorithm for network-diagram layout,” Proc. of the 4th International Conference on Genetic Algorithms, pp.458-465, California, U.S.A., 1991.
- [15] A. Markus, “Experiments with genetic algorithms for displaying graphs,” Proc. of 1991 IEEE Workshop on Visual Languages, pp.62-67, Kobe, Japan, 1991.

(平成 8 年 2 月 29 日受付, 7 月 19 日再受付)

前田 肇 (正員)



昭 41 阪大・工・通信卒, 昭 46 同大大学院博士(通信)課程了。同年阪大・基礎工・機械助手。阪大・工・電子助教授を経て, 平 5 年工学部通信工学科教授, 現在に至る。昭 58~59 カナダウォータール大学客員准教授。昭 62 年度, 平元年度計測自動制御学会論文賞, 平元年度システム制御情報学会論文賞, 平 4 年度計測自動制御学会論文賞友田賞, 平 6 年度計測自動制御学会著述賞受賞。システム理論, 制御理論, 信号理論, 回路理論に関する研究に従事。IEEE, 計測自動制御学会, システム制御情報学会各会員。

角 浩二 (正員)



平 6 阪大・工・通信卒。平 8 同大大学院修士(通信)課程了。同年松下電器産業株式会社入社。在学中, グラフレイアウトに関する研究に従事。

榎原 博之 (正員)



昭 57 阪大・工・通信卒, 昭 62 同大大学院博士(通信)課程了。同年阪大・工・助手。平 6 関西大工学部専任講師, 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, 情報処理学会, IEEE, ACM 各会員。

中野 秀男 (正員)



昭 45 阪大・工・通信卒, 昭 50 同大大学院博士(通信)課程了。同年阪大・工・助手。平 3 阪大・工・助教授。平 7 大阪市大学術情報総合センター教授, 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。