

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[†]

[†]大阪大学工学部通信工学科, 吹田市

^{††}関西大学工学部, 吹田市

^{†††}大阪市立大学工学部, 大阪市

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過ぎない. そこで, 本報告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.

Automatically Graph Drawing Using Genetic Algorithm

Koji Sumi[†] Hiroyuki Ebara^{††}
Hideo Nakano^{†††} Hajime Maeda[†]

[†]Faculty of Engineering, Osaka University, Suita-shi, 565 Japan

^{††}Faculty of Engineering, Kansai University, Suita-shi, 564 Japan

^{†††}Faculty of Engineering, Osaka-City University, Osaka-shi, Japan

Recently, the applications of computer have increased because of the development of computer technology. Graph drawing is one of such applications, where graphs are modeled as pictures represented by points and lines. Kamada Algorithm, which is one of the general graph drawing algorithm, transforms the layout problem into the energy minimizing problem by defining the system energy of graph. But its minimizing technique is Newton-Raphson, which finds merely a local optimum. In this paper, we try to find a better local optimum by using genetic algorithm.

1 序論

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、人間が理解し易いように「見やすい」グラフを自動的に描画させる必要がある。

様々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][3][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。本報告では、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構造がどのようなになるかを文献 [5] の方法を用いて、定量的な評価を行なう。

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][3][4][5][7][8][9][10][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 といわれるものである。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][3][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。アルゴリズムは、節点間距離ができるだけこの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見なすことができ、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、(1) 式で定義された 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通りである。

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、一本の枝の長さは短くなってしまふ可能性がある。それゆえ、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。

$$E = \sum_{i < j} \int_{l_{ij}}^{|p_i - p_j|} k_{ij} \left(\frac{x}{l_{ij}} - \frac{l_{ij}}{x} \right) dx \quad (7)$$

$$= \sum_{i=1}^{n-1} \sum_{j=i+1}^n k_{ij} \left\{ \frac{1}{2} \frac{(|p_i - p_j|^2 - l_{ij}^2)}{l_{ij}} - l_{ij} (\log |p_i - p_j| - \log l_{ij}) \right\} \quad (8)$$

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。

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 アルゴリズムとよぶ。

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、遺伝子列は二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 の振る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 で行ない、局所探索をヒューリスティックアルゴリズムで行なう。

GA のみで探索を行なう場合は、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

を行なうことになる。したがって、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章で挙げた改良 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。したがって 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。

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 に対して行な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章で述べた改良 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リングを施す。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持つ個体とのエネルギーの差に定数をか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持つことになる。

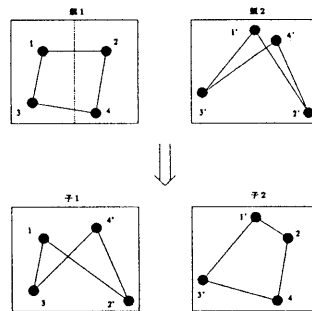


図 1: 節点の座標を考慮した交叉

4.3 GA オペレータの設計

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よ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。したがって、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る。さらに、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。

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について節点を横方向の座標の大きさでソートし、座標値が中央値未満の節点を子1に、中央値以上の節点を子2にそれぞれ受け継がせ、残りの節点の座標を親2から受け継ぐ(図1)。次に、節点を縦方向の座標の大きさでソートし、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。

突然変異については、より広範囲な探索が行なえるよう、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時のみであるため、局所的な探索を効率良く行なうことが出来ない。そこで、GA のハイブリッド化を行なう。局所探索の方法としては、2章で述べたエネルギー関

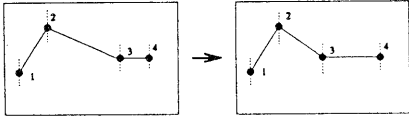


図 2: グラフの構造を考慮した局所探索

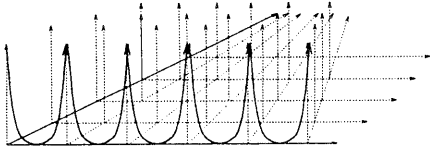


図 3: ポテンシャルバリア

数の局所解を求める Newton-Raphson 法を用い、他の GA オペレータの実行後に全ての個体に対して実行する。

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、交叉により保存された親の構造を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なう必要がある。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いのである。

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付加する。具体的には、描画平面に放物曲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。さらに、 x_i', y_i' は節点 i の格子内規格化座標とよび、

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

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

で表され、 x_i, y_i は節点 i の絶対座標、 $\text{div}_x, \text{div}_y$ は格子幅、 m, n 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、 i 番目の大きさの座標をもつ節点は、 i

番目の放物曲線に従うことになる。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。ただし、ポテンシャルバリアが強過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取る必要がある。

Kamada アルゴリズムでは、(3) 式の 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、計算時間が莫大な量になるので、ある程度回数 (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、計算時間のむだが生じないように、各世代の最良解が何世代にも渡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) はこれに支配され、解の改善が進むにつれ解の保存を高くさせるようにはたらく。

5 描画アルゴリズムの評価

本章では、第 2 章で述べた改良 Kamada アルゴリズム (改良 Kamada) と、第 4 章で述べた 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ない、その定性的、定量的な評価を行なう。

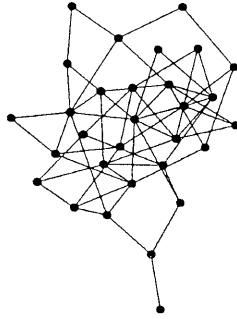


図 4: 改良 Kamada アルゴリズムの描画例

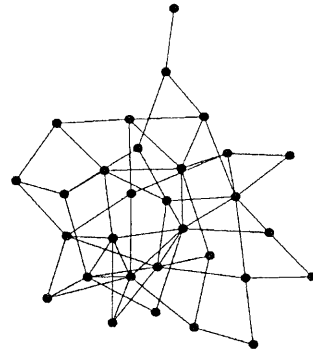


図 5: GA-改良 Kamada の描画例

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 5 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

また、それぞれの評価アルゴリズムについては文献 [5] で述べられている。

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なう。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、として描画を行なった。

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なうにあたり、より一般的な傾向を見るために、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。なお、図 6, 7, 表 1, 2 において、ExtKamada と ga-ExtKamada はそれぞれ改良 Kamada と GA-改良 Kamada を表している。

図 6, 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

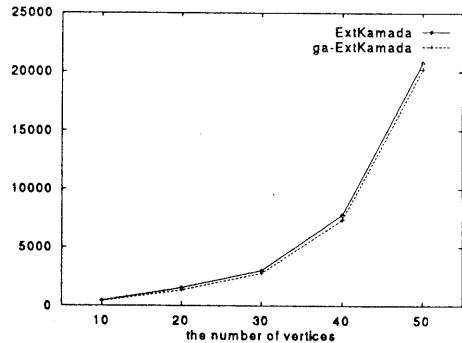


図 6: グラフの大小によるエネルギー値の変化

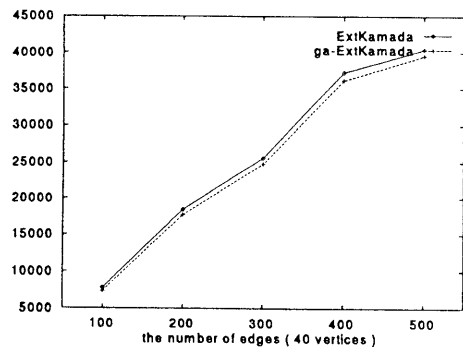


図 7: グラフの疎密によるエネルギー値の変化

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	ExtKamada	ga-ExtKamada
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	8.350000	6.770000
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	31.270000	25.400000
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	65.980000	58.320000
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	258.480000	238.240000
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	1643.420000	1589.190000
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

評価項目	ExtKamada	ga-ExtKamada
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	258.480000	238.240000
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	2026.420000	1925.750000
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	5820.410000	5587.430000
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	12309.120000	11770.890000
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	21904.750000	20755.190000
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

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、全体的にみて改善されているといえることができる。

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 結論

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 の利点を活かし、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、さらに見やすいグラフの描画を行なうことが課題である。

謝辞

本論文の執筆に当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。

参考文献

- [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 (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 (1991)
- [3] T.Kamada: *Visualizing Abstract Objects and Relations, A Constraint-Based Approach*, 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, pp7-15 (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, 情報処理学会 研究報告, アルゴリズム 41-8 (1994)
- [6] R.Lipton, S.North, and J.Sandberg: A Method for Drawing Graphs, Proc. ACM Symp. on Computational Geometry, pp.153-160, (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 (1995)
- [8] P.Eades: A Heuristic for Graph Drawing, *Congressus Numerantium*, vol.42, pp.149-160 (1984)
- [9] T.Frunchterman and E.Reingold: Graph Drawing by Force-Directed Placement, *Software-Practice and Experience*, vol.21, no.11, pp.1129-1164 (1991)
- [10] R.Davidson and D.Harel: Drawing Graphs Nicely Using Simulated Annealing, Technical Report, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot (1989)
- [11] 佐野達郎: グラフの自動描画アルゴリズムについて, 電子情報通信学会 信学技報, pp1-6, HIP95-19
- [12] 北野宏明 編: 遺伝アルゴリズム, 産業図書, 1993