

群れアルゴリズムを応用したグラフ描画法

臼杵正郎・杉山公造

北陸先端科学技術大学院大学

グラフを良いレイアウトで描画するために、鳥や魚が群れを構成する時のアルゴリズムを応用した。グラフの頂点を群れの個体、グラフの辺を個体間の関係の有無(仲間であるか、いなか)とみなして、群れアルゴリズムを適用した。群れアルゴリズムでは、各個体が群れ全体を把握するのではなく、局所的な周囲の状況を認識し、単純なルールの下に行動することによって、全体として群れが構成される。今回提案するアルゴリズムにおいても、各頂点が周囲の状況にあわせて単純なルールに従い移動することによって、全体を良いレイアウトに導くものとなっている。それによって、有向辺の向きを局所的にあわせ、辺を一定の長さに近づけることが可能になった。

Graph Drawing based on Flocking Algorithm

Masao Usuki and Kozo Sugiyama

Japan Advanced Institute of Science and Technology (JAIST)

We develop graph drawing algorithms based on a flocking algorithm which simulates birds or fishes behavior in making a group. Considering a node of a graph as an individual of a flock and an edge of a graph as a relationship among group's individuals, a flocking model is applied to graph drawing. As a flock of birds or fishes adapts with surrounding circumstances, a layout of diagram is constituted with movements of each node corresponding to surrounding circumstances. Our algorithm enables to align directions of directed edges locally and uniform lengths of edges.

1. はじめに

グラフ構造を含んだ情報は非常に多くあり、辺に方向性のある有向グラフを含んだものも多い。これらの構造を、人間にとって把握しやすい形にするために、図として描画することは頻繁に行われている。最近の例ではUMLのユースケース図やクラス図、バイオ分野の物質間の流れである代謝系やシグナル伝達系の図などが挙げられる。

しかし、グラフを描画する作業は、人間にとって大変なものである。そのため、グラフを自動描画するための研究が多くなされている。ただし、そのほとんどが単純な形のグラフの描画方法である。単純無向グラフや単純有向グラフ、それから木や平面グラフなど限定された形のものが多い。

実際には、頂点や辺に複数の意味属性が付属したグラフ構造を用いることが多い。例えば、UMLのクラス図では無向辺だけでなく、有向辺も使用されており、さらに細かく複数の種類に分類されている。また、バイオの代謝系やシグナル伝達系の図でも、複数の種類の辺や頂点を用いられている。このような複雑な構造を持ったものを自動描画するためには、その構造にあわせた柔軟で、拡張が容易な描画アルゴリズムが必要である。

また、描画するに当たって、どのようなレイアウトにすれば、構造が把握しやすくなるのか考慮しなくてはならない。レイアウトが悪いと、構造を把握しにくくなるだけでなく、間違った意味を伝達してしまうことさえある。良いレイアウト構造を得るための美的基準を作成し、それを満たすためのアルゴリズムを考えなければならない。

本研究では描画対象を、無向辺と有向辺が混在したグラフとした。これを自動描画するためのアルゴリズムとして、鳥や魚の群れの行動モデルを応用したものを考案した。特に有向辺を見やすくするためのアルゴリズムを導入している。

本論文の構成は、この節を含め6節から構成される。2節では既存研究と美的基準について、3節では群れアルゴリズムについて説明し、4節で本研究の成果であるアルゴリズムについて説明する。5節で評価、6節でまとめと今後の課題について記述する。

2. 既存研究と美的基準

本研究のアルゴリズムと類似した既存研究としては、スプリング・モデルとマグネティック・スプリング・モデルが挙げられる。

2.1 スプリング・モデル

スプリング・モデル[2]は、無向グラフを自動描画するためのアルゴリズムであり、頻繁に用いられている。バネのフックの法則を模した仮想的な物理系を仮定しており、頂点間にスプリングをつないだものとみなしている。なお、あくまで仮想的なものであり、忠実な物理モデルにしたがったものではない。頂点間には2種類のスプリングが使用されており、隣接頂点間では引力と斥力がかかっている。非隣接頂点間では斥力だけがかかっている。それによって、隣接頂点同士が近くに配置される。なおこのアルゴリズムでは、逐次的に頂点を移動していくことにより、徐々に良いレイアウトに近づけていくアプローチを取っている。このモデルで、改善される美的基準は以下の4つである。

- (1) 辺の長さが均一
- (2) 辺の交差数が最小
- (3) 対称性を顕示
- (4) 頂点を均一に分配

2.2 マグネティック・スプリング・モデル

マグネティック・スプリング・モデル[4]はスプリング・モデルに磁力を導入したアルゴリズムである。辺を磁針とみなし、グラフを描画するフィールドを磁場として、辺が磁場に従って向きを変化する。向きを変えることによって辺をある特定の向きにそろえることができる。ただし、ここでも仮想的な物理系を想定しているため、磁場に人工的な性質を加えることが出来る。単純に北(上)に向くだけでなく、ある点を中心に同心円状に磁場が働くものなどが考えられている。このモデルで改善される美的基準はスプリング・モデルの4つの美的基準と、以下の1つである。

- (5) 辺が特定の向きまたは方向に従う

2.3 既存研究の問題点

スプリング・モデルでは、有向辺の向きをそろえる描画ができない。有向辺を無向辺と考えアルゴリズムを実行することはできる。しかしその時には、無向グラフとしては同じ構造でも、有向グラフとして捕らえた時には違いがあるものでも、同じようにしか描画することができず、有向辺の向きやパスの流れを把握しにくい。

また、マグネティック・スプリング・モデルでは、有向辺全体をある特定の方向へそろえるのだが、局所的には特定の方向へそろえない方が良い場合もあり、柔軟性に欠ける。その特定の方向についても人がアルゴリズムによって決める必要があり、データ構造にあわせて最適なものを選ぶことは困難である。

また、スプリング・モデルとマグネティック・スプリング・モデルでは、2つの隣接頂点があり、一方の頂点がその他の頂点群に囲まれている場合、隣接頂点間の辺が極端に短くなる(図1:悪い例)。これは、隣接頂点間に掛かる斥力と非隣接頂点間に掛かる斥力が同じで、隣接頂点間には引力が働くため、その部分だけが短くなってしまう。そこで、隣接頂点間に掛かる斥力を、非隣接頂点間に掛かる斥力より大きくすれば、解決すると考えられる。

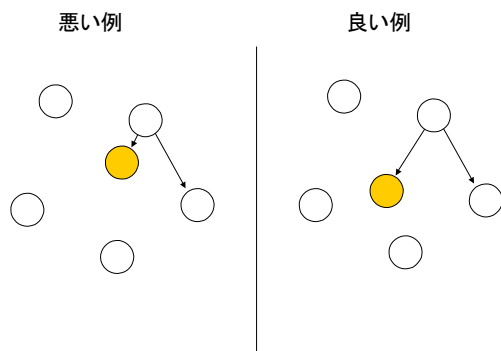


図1：非隣接頂点によって接続頂点間の距離が左右される例

2.4 美的基準

本研究では、以上の問題点を改善するため以下のような美的基準を設定した。

- (1) 頂点間の距離を一定以上に保つ
- (2) 隣接頂点間の距離を一定以上に保つ。その距離は非隣接頂点間以上にする。
- (3) 隣接頂点を近くに配置
- (4) 辺の交差を少なく
- (5) 各頂点に接続している有向辺の向きをそろえる
- (6) 対称性を顕示

上記(1)~(3)によって、辺を一定の長さに近づけることができる。また、(5)によって、有向辺の向きやパスの流れを見やすくする。

これら美的基準を改善するためのアルゴリズムを作成するために群れアルゴリズムを応用した。次節ではそれについて説明する。

3. 群れアルゴリズム

ここでは、本研究のアルゴリズムに応用した C. W. Reynolds の群れアルゴリズム[5]について説明する。これは、鳥や魚が群れを構成する時に必要なルールに関するものである。

3.1 群れの特徴

自然の鳥や魚の群れ(flock or school)から観察される特徴は以下の3つである

- (1) 明確なグループを形成する
- (2) 互いにぶつからないように等しい感覚をとる傾向がある。
- (3) 群れは群れとして全体的な方向を持つ傾向がある

全体的な方向を持つことと、グループ化していることから、群れは1つの意思の下に動いているように見える。しかし、群れの構成形態や構成している個体の数が違っていてもうまくいくことや、リーダーのような存在がいなくても良いことから、1匹1匹の行動から群れは構成され、各個体は自分の周りの局所的な世界認識に基づいて行動していると考えられる。

3.2 群れアルゴリズム

Craig W. Reynolds は、群れを構成するためのアルゴリズムを考案した。そのアルゴリズムでは、各個体が基本的な3つのルールに基づいておのお

の行動することによって全体として群れが構成されるようになる。

- (1) Separation(独立)・・・お互いぶつからないようにある一定の距離を保つこと
- (2) Cohesion (結合)・・・群れの中心に進もうとすること
- (3) Alignment (同調)・・・群れと、スピードと方向を合わせようとする

なお、ルールの中でいう群れとは、各個体から見る事ができる範囲にいる、仲間の個体群のことである。図2は3つのルールによるそれぞれの移動を説明したものである。灰色の円の中心にいる

黒い魚が各ルールによって、どの方向に動くのかを示している。灰色の円は黒い魚の視界範囲を示しており、黒い魚から出ている矢印は各ルールに従った移動ベクトルを示している。実際には各ルールからベクトルを生成し、それを合成することによって得たベクトルにしたがって移動する。

図3は実際に群れのルールを適用した魚の群れ行動をシミュレーションした3次元CGの様子である。

本研究では、群れを構成するための3つのルールを応用することによって、グラフを自動描画するためのアルゴリズムを考案する。

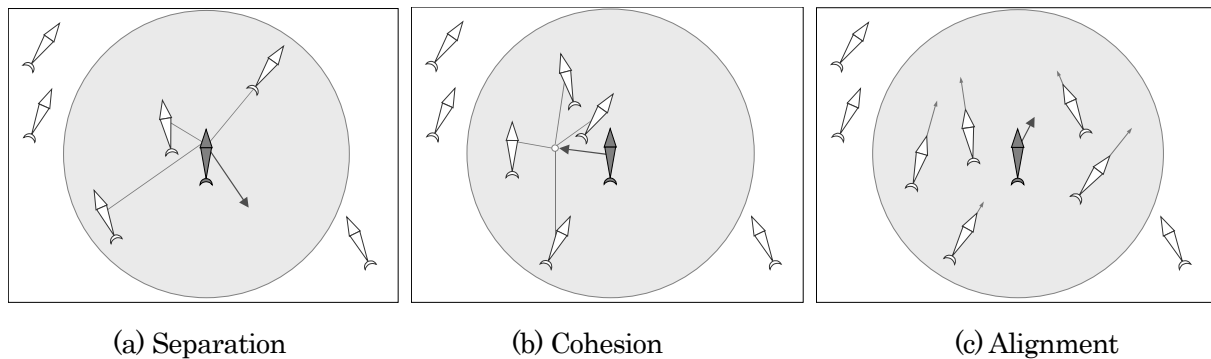


図2：群れアルゴリズムの基本ルール3種

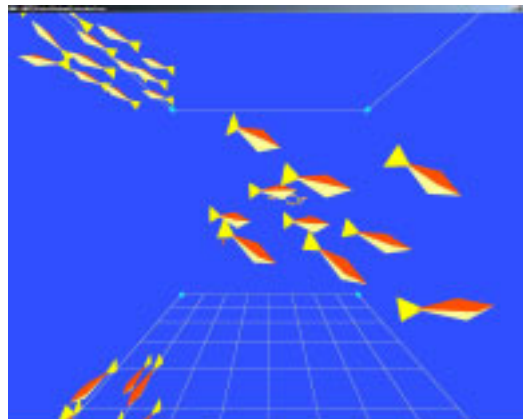


図3：魚の群れ(Fish School)のシミュレーションの様子

4. グラフ描画アルゴリズム

グラフを描画するために前節で挙げた3つのルールを応用し、描画アルゴリズムを構築した。

4.1 アルゴリズムの概要

本研究のアルゴリズムでは、スプリング・モデルのように逐次的に動かす方法を用いた。ステップごとに、各頂点の移動ベクトルを計算し、そのベクトルに従って現在位置から動かす。移動ベクトルはルールごとに作成し合成することによって頂点の移動ベクトルとする。それを繰り返すこと

によって、良い描画状態に近づけていく。なお、繰り返し回数はあらかじめ決めておく。

逐次的描画手法では、初期配置によって描画結果が変わってくるが、今回のアルゴリズムではランダム配置とし、同じグラフ構造を持っていても毎回違う描画がされる手法を採用した。

4.2 群れとグラフの対応関係

群れアルゴリズムを応用するために、群れの要素とグラフの要素の対応関係について説明する。

群れの各個体を、グラフの各頂点として扱う。また、群れの仲間を、辺で接続した頂点同士とみなし、さらに各個体の移動方向を、各頂点に接続した有向辺の平均ベクトルと考える。以上を前提として群れアルゴリズムを応用して、新しいアルゴリズムを構築した。

4.3 群れアルゴリズムの応用

群れアルゴリズムの基本ルール3つから応用した3つの移動ベクトルについて説明する。

- (1) Separation・・・頂点から、一定以上はなれるためのベクトル
- (2) Cohesion・・・隣接頂点に近づくためのベクトル
- (3) Alignment・・・有向隣接頂点の平均有向ベクトルと、その間にある有向辺の向きがそろう位置へ移動するためのベクトル

この3つのベクトルを求めるための計算式は式1～3の通りである。これらは、ある頂点 v の移動ベクトルを求めるための式である。 V は全頂点の集合を指し示している。 C は頂点 v に隣接している頂点の集合を指し、 D は頂点 v に有向辺によって隣接している頂点の集合を指し示している。

式1 Separation(v) と式2 Cohesion(v) については、C. W. Reynolds の群れアルゴリズムと同じである。しかし、式3 Alignment(v) に関しては、変更を加えている。群れのルール Alignment の特徴は、周囲の群れに同調して合わせることである。したがって変更するに当たっては、おのおのが周囲に合わせるという考え方を重視して移動ベクトルを考案した。図4は、頂点 v が、頂点 v と有向辺によって接続している、ある頂点 x との関係から、ど

のようにして Alignment(v) を計算するのかについての説明図である。ここでは頂点 v は source 側の場合を扱っている。頂点 x の有向辺の平均ベクトル \vec{A}_x と、頂点 x との間の辺の方向を合わせるため

には、頂点 v は図4の理想位置 α にあると良い。そこで、理想位置 α の位置に移動するためのベクトルを Alignment(v) とした。

$$Separation(v) = \frac{1}{|V'|} \sum_{x \in V'} \frac{k - |\vec{xv}|}{|\vec{xv}|} \vec{xv}$$

$$V' = \left\{ x \in V \mid |\vec{xv}| \leq k \right\}$$

式1 : Separation ベクトル

$$Cohesion(v) = \frac{1}{|C|} \sum_{x \in C} \vec{vx}$$

式2 : Cohesion ベクトル

$$Alignment(v) = \frac{1}{|D|} \sum_{x \in D} \begin{cases} \vec{vx} - \frac{|\vec{vx}|}{|\vec{A}_x|} \vec{A}_x & \text{if } v \text{ is source} \\ \vec{vx} + \frac{|\vec{vx}|}{|\vec{A}_x|} \vec{A}_x & \text{if } v \text{ is target} \end{cases}$$

\vec{A}_x は頂点 x に接続している有向辺の平均ベクトル

式3 : Alignment ベクトル

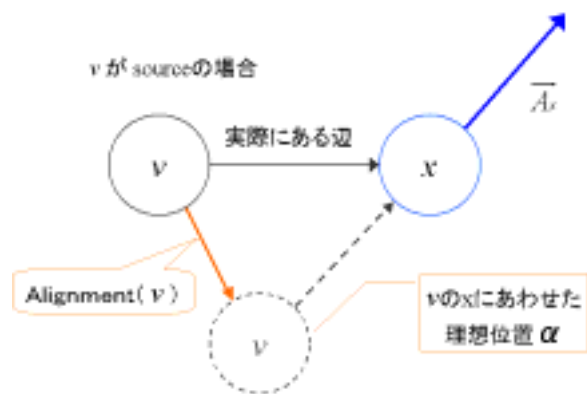


図4 : Alignment(v) の説明をするための図

4.4 その他のルール

その他に、2.3 節の図1のような問題点を改善するルールを作成した。隣接頂点間を非隣接頂点間より大きく取るために、 $Separation(v)$ を改良した $Adjacent(v) : Separation(\text{from adjacent nodes})$ を導入している。

Adjacent: 隣接頂点から一定以上離れる。ここで言っている一定以上の距離は $Separation$ で定義した距離 k より大きくしなければならない。

$$Adjacent(v) = \frac{1}{|C'|} \sum_{x \in C'} \frac{p - \left| \frac{\vec{vx}}{|\vec{vx}|} \right|}{\left| \frac{\vec{vx}}{|\vec{vx}|} \right|} xv$$

$$C' = \left\{ x \in C \mid \left| \frac{\vec{vx}}{|\vec{vx}|} \right| \leq p, p > k \right\}$$

式4 : Adjacent ベクトル

4.5 ベクトル合成

式1～4で挙げた計算式により得たベクトルを合成する。その時、各ベクトルに適当なパラメータを設定する。そして、各ベクトルにそれぞれのパラメータを掛けたベクトルを合成する。それをパラメータの合計で割ったベクトルを頂点 v の移動ベクトルとする。

全ての頂点に対して移動ベクトルを計算し、逐次的に動かすことによってレイアウトを改善することになる。

5. 評価と描画例

アルゴリズムの導入により、2.4 節で述べた美的基準に沿ってレイアウトが改善されたかどうか実験を行った。

5.1 アルゴリズム比較のための設定

マグネティック・スプリング・モデルと比較す

	Separation	Cohesion	Alignment	Magnetic	Adjacent
Random	0	0	0	0	0
Spring	6	2	0	0	0
Magnetic	6	2	0	2	0
Flock	6	2	2	0	0

表1 : Adjacent を使用しない時のパラメータ

るために、式5を追加しベクトル合成することにした。

$$Magnetic(v) = \frac{1}{|D|} \sum_{x \in D} \begin{cases} \frac{\vec{vx} - \frac{\vec{vx}}{|\vec{vx}|} \vec{S}}{|\vec{S}|} & \text{if } v \text{ is source} \\ \frac{\vec{vx} + \frac{\vec{vx}}{|\vec{vx}|} \vec{S}}{|\vec{S}|} & \text{if } v \text{ is target} \end{cases}$$

\vec{S} は、ある特定の方向を持ったベクトル。

今回の実験では画面下方向のベクトル

式5 : Magnetic ベクトル

式1～5の各ベクトルのパラメータを変更することによって、スプリング・モデル、マグネティック・モデルと同等のアルゴリズムが設定可能である。スプリング・モデルでは、**Separation** と **Cohesion** 以外のパラメータを0にすれば良い。マグネティック・スプリング・モデルでは、**Alignment** を0に設定する。

5.2 実験設定

実験には、頂点数30、辺数40の有向グラフをランダムに生成したものを使用した。初期配置はランダム配置である。比較対象は以下の4つ。

- (1) **Random** : ランダムに配置した初期状態
- (2) **Spring** : スプリング・モデル
- (3) **Magnetic** : マグネティック・スプリング・モデル
- (4) **Flock** : 群れアルゴリズムを応用したモデル

上記の比較対象それぞれについて、Adjacentベクトルがある時とない時での比較も行った。

実験でのパラメータの設定は、表1 Adjacent を使用しない場合、表2 Adjacent を使用する場合の2種類用意した。式1、4に出てくるパラメータは、それぞれ $k=2, p=3$ と設定した。

	Separation	Cohesion	Alignment	Magnetic	Adjacent
Spring	3	3	0	0	2
Magnetic	3	3	0	2	2
Flock	3	3	2	0	2

表2：Adjacentを使用する時のパラメータ

以上の計7つの場合について実験した。各比較対象において500ステップ動かした後の状態を比較した。それぞれ10回試行して、以下の5つの比較項目について平均を取っている。

- (1) 交差数：エッジの交差数
- (2) ずれ角の平均(radian)：各頂点に接続している有向辺の平均角度とその頂点に接続している有向辺とのずれ角の平均
- (3) ずれ角の分散：上記のずれ角の分散
- (4) 辺長の平均：辺の長さの平均
- (5) 辺長の変動係数：辺の長さの変動係数(%)

5.4 実験結果

実験の結果は表3、表4のとおりである。各アルゴリズムともに、辺の交差数を減少させることに成功している。また、MagneticとFlockでは、ずれ角平均・分散ともに減少してレイアウトが改善していることが分かる。

しかし、Adjacentベクトルを使用していない時には、辺長の変動係数がすべてのアルゴリズムにおいて大きい。特にMagneticとFlockにおいては、辺の向きをそろえる力のため頂点の子が有向辺の平均ベクトルの向きに集まりやすく、辺を極端に短くしている例が多く見られた。Adjacentベクトルを用いると、すべてのアルゴリズムにおいて、辺長の変動係数が小さくなり、辺の長さを一定にする効果が得られた。また、Adjacentベクトルを用いることによって、ずれ角の平均・分散も減少した。これは、頂点間の斥力のためのSeparationの影響が弱くなったためだと考えられる。

MagneticとFlockの比較であるが、交差数については、Magneticの方が低いが、ずれ角の平均・分散ともにFlockの方が減少している。つまり、局所的な方向の流れはFlockの方がそろっている結果となった。

	交差数	ずれ角平均	ずれ角分散	辺長の平均	辺長の変動係数
Random	160.7	0.989645575	1.682066368	5.22742783	48.72879947
Spring	8.2	0.942763138	1.487449653	1.42732944	29.2470744
Magnetic	7.9	0.781361783	1.112172537	1.24138433	46.86753498
Flock	9.6	0.747269395	0.948283109	1.3322458	43.35170929

表3：Adjacentを使用しない時の結果

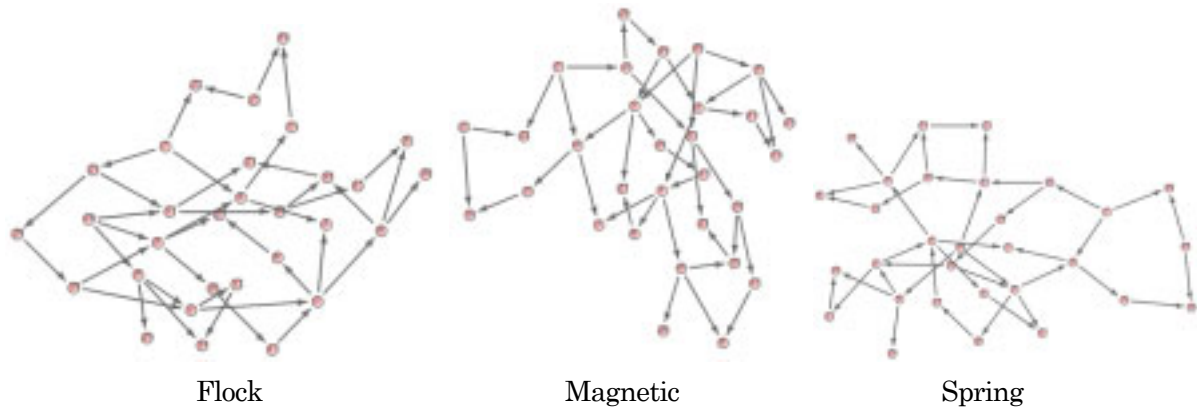
	交差数	ずれ角の平均	ずれ角の分散	辺長の平均	辺長の変動係数
Spring	13.2	0.882933703	1.234427542	1.49854733	12.47717256
Magnetic	11.3	0.763260879	1.028603876	1.39104152	17.68492403
Flock	13.2	0.712375308	0.84196649	1.38426133	15.68555909

表4：Adjacentを使用する時の結果

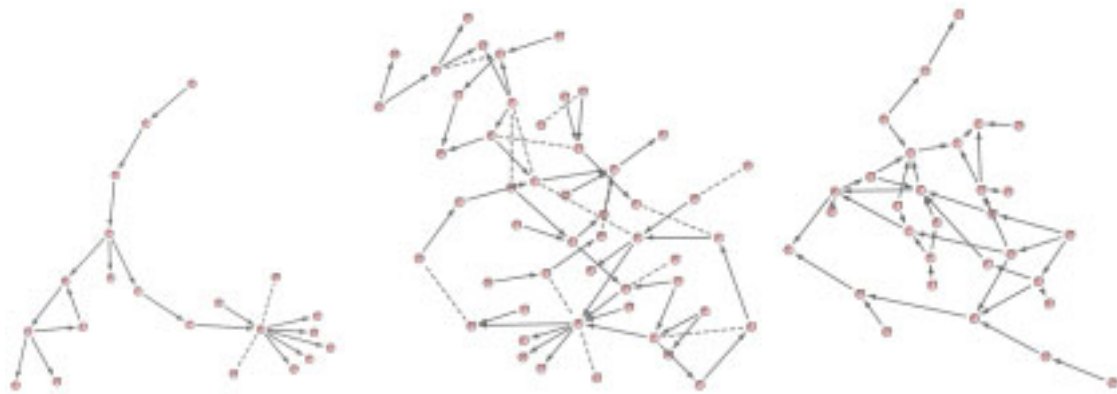
5.5 描画例

以下に描画例を示し、次節のまとめにて、評価項目だけでは分からない問題点を指摘する。

モデル比較：頂点 30, 辺 40 の同一グラフ, Adjacent:あり, 評価実験とパラメータ同じ



条件比較：Flock モデルでの比較. 一番右のみ Adjacent なし. 評価実験とパラメータ同じ



頂点 21, 辺 21, 有向辺 90% 頂点 50, 辺 60, 有向辺 80% 頂点 30, 辺 40, 有向辺 100%
 図 5：描画結果(点線は無向辺)

6. まとめ

頂点ごとにルール別にベクトルを求めて合成ベクトルにしたがって移動させることにより, 局所的な最適化を促し, 全体としての描画を良い状態へ導くことが可能となった. 各頂点に与えるルールの追加(新しいベクトル作成)などによる, 新たなアルゴリズムの作成も容易である.

描画例を見てみると, 局所的な流れはそろっているが, 群れの仲間と見なした隣接頂点だけでは, 本来の群れアルゴリズムよりも数が少なく, 全体としての方向があまり見えてこなかった. 群れの仲間として, 描画配置が実際に近い頂点を仲間と見なしたり, 2つ以上先の隣接関係を仲間とみなしたりするなど, 仲間とみなす頂点の数を増やすことにより描画の改善が見込まれる. また, 今回手動でパラメータを決定したが最適な値をどのように決めるかなどを今後の課題としたい.

参考文献

- [1] P. Eades and R. Tamassia, Algorithms for drawing graph: An annotated bibliography, Tech. Rep. CS-89-09, Dept. of Computer Science, Brown Univ., 1989.
- [2] P. Eades: A Heuristic for Graph Drawing, CONGRESSUS NUMERANTIUM, Vol.42, pp.149-160, 1984.
- [3] 杉山公造：『グラフ自動描画法』, 計測自動制御学会, 1993.
- [4] Sugiyama, K. and K. Misue: Graph Drawing by the Magnetic Spring Model, Journal of Visual Languages and Computing, 6, pp.217-231, 1995.
- [5] Craig W. Reynolds: Flocks herds, and schools: A distributed behavioral model, Computer Graphics, Vol.21, No.4, 21-35, 198.