第1章　はじめに

生体分子ネットワーク構造や, プログラムのフローチャート図など, データ間の関連性がある構造を表現するために, しばしばグラフ構造が用いられる. グラフとは有限のノード集合と, ノード間を結合するエッジ集合からなるデータ構造である. グラフ表現されたデータから人間が情報を得るには, グラフを適切に視覚化し, ノード間の関係やノードの情報が分かりやすく見えていることが望ましい. しかし, グラフを分かりやすく視覚化することは難しく, グラフが大規模になると人間が手作業で視覚化することは困難である. 任意のグラフからグラフの二次元描画を生成する問題は, グラフ描画問題と呼ばれ, 今日までに多くの手法が提案されている.

グラフ描画は, 人間が理解しやすいことが望ましいが, グラフごとにノードやエッジに与えられる意味は異なり, 人によって見易さの好みが異なるため, グラフ描画に最も良いという基準を与えることは出来ない. そのため, グラフの特徴などを考慮した適切なグラフ描画を考えていく必要がある.

現在, グラフ描画には力学モデルによる方法などが用いられている. グラフを力学的モデルとみなしてシミュレーションを行い, 系が安定するまで計算することで, グラフノード間が均衡した状態になるようなグラフ描画を求めることができる.

しかし, ノードの情報に着目したい場合, グラフ描画上にノード情報を表示させるために, ノード間に一定以上の空間を確保したいケースが考えられる. 従来のグラフ描画手法では, グラフ描画にノード間に一定以上の空間を保障することはできない.

この目的のために, 多項式時間で動作するグラフノードの二次元点集合近似照合による格子状配置アルゴリズムが提案された.[1] しかし, 二次元点集合の近似照合は時間においても領域においても計算量の次数が高く, 現実的なサイズのグラフ描画に対して直接適用することができなかった.

そこで本研究では, 二次元点集合近似照合における格子点集合 (テキスト) が, 格子状であることに着目し, 二次元点集合近似照合を, グラフノード点集合 (パターン) の部分集合を包含する格子点集合に対して軸平行な最小の矩形を部分点集合の境界矩形とし, 空集合からパターン全体になるまで, サイズが 1 ずつ増加するような真に包含関係にある点集合列に対応する矩形拡張の列を考え, 矩形拡張列を最小の平行移動によって格子点集合上に格子配置する問題にみなせることを利用する.

以下では, 第 2 章において, 任意の点集合に対する境界矩形および境界矩形拡張列と, 境界矩形拡張列の格子配置における編集距離の定義を行い, 境界矩形の編集距離の再帰的定義と, 時間・空間ともに $O(n^2)$ の計算量で実行可能な動的計画法 (DP) による解法を紹介する. 第 3 章では, 境界矩形拡張列の格子配置によるグリッドグラフ描画アルゴリズムについて示した後, アルゴリズムの実行時間と, いくつかのグリッドグラフ描画の修正法を示し, グリッド

グラフ描画の結果を示す. 第 4 章では結論と, 今後の課題について述べる.

第2章　境界矩形拡張列の格子配置問題とその解法

本章では, 境界矩形拡張列の格子配置について述べる.

2.1　境界矩形拡張列とその性質

格子配置のための二次元点集合では, パターンとなるグラフノードを表す二次元点集合に対して, テキストは格子状点集合 $\mathrm{Grid}(d) = \{(dx, dy) | x, y \in \mathbb{N}\} (d \in \mathbb{Z})$ と定義する.

任意の二次元点集合 $S = \{(x, y) | x, y \in \mathbb{Z}^{+2}\}$ に対して, S の点を全て包含する格子状点集合に対して平行な境界矩形 R を与えることができる. R は, 対応する S の点によって矩形の左下と右上を定める点または点対によって, 矩形を表現することができる. すなわち S の部分境界矩形は, S から矩形の上下左右を定める点の組み合わせの数だけ存在する.

2.2　境界矩形拡張列の格子状配置における編集距離

2.3　解法

第3章　実験結果

3.1　グラフの格子状配置アルゴリズム

3.2　配置にかかる計算時間

3.3　グラフに入力される重複した点の対策

アルゴリズムでは考慮しないことにしたが, 実用上では入力される二次元点集合に重複点が与えられることが考えられる. X 軸, Y 軸列を安定ソートするなどして, 重複点に一意な順序関係を与えてやることで, 格子状配置アルゴリズムを適用させることは容易であるが, 冗長な形状に展開されることもある. 元々, 重複点が与えられるということは, グラフノード間に密接な関係があると考えることが妥当であるので, 隣接関係を維持しつつグリッド状に展開する方法をアルゴリズムの拡張として組み込んだ.

重複点は一つの点とみなしてアルゴリズムの計算を行い, 重複点を境界矩形に追加して拡張を行う際, 重複点を最小の境界矩形に展開し, そのサイズを境界矩形拡張の編集距離に追加することで処理した.

3.4　伸長したグラフの畳み込み処理

アルゴリズムでは, 計算量の次数を下げるために右上方向にのみ拡張をするように提案した. そのため, グリッド幅に対して密な入力が与えられた場合, グリッドグラフ描画は大きく右上方向に伸長した形状になる. 過度に伸長したグラフは線形に近づき, 通常の矩形ディスプレイに表示するには形状が不適切である. そのため, 出力が矩形に近づくようにグラフノード間の不必要な空白を畳み込み, グリッドグラフ描画の幅と高さを小さくする処理を後処理として行った.

3.5　実験結果

第4章　終わりに

参考文献

[1] 吉田英聡, 二次元点集合近似照合によるグラフの格子状配置アルゴリズム, 電子情報通信学会総合大会講演論文集 2008 年 情報・システム (1), "S-17"-"S-18", 2008-03-05