

平面的グラフの格子描画アルゴリズム

Algorithms for grid drawings of planar graphs

数学専攻 原 剛士
HARA Tsuyoshi

1 準備

グラフ (graph) G とは、空でない集合 V と、 V の要素の対の集合 E の組 (V, E) のことである。ただし、 V の要素の対 (x, y) は、異なる V の要素 x, y からなり、 (x, y) と (y, x) は区別しない。簡単のため、対 (x, y) を xy と書く。 V の要素を 頂点 (vertex)、 E の要素を 辺 (edge) という。 V を 頂点集合 (vertex set)、 E を 辺集合 (edge set) といい、 $V = V(G)$ 、 $E = E(G)$ と表すこともある。グラフを図として表すのは自然なことで、頂点は点で、辺は頂点同士を結ぶ線で表される。Figure 1.1 はグラフの例である。

$$G = (V, E), V = \{v_1, v_2, v_3, v_4, v_5\}, E = \{v_1v_2, v_1v_5, v_2v_3, v_2v_4, v_2v_5, v_3v_4\}$$

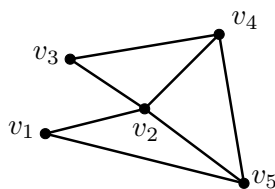
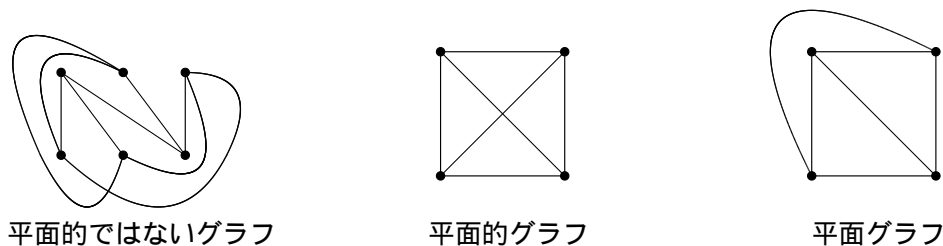


Figure 1.1 グラフ $G = (V, E)$

グラフ G の 位数 (order) とはそのグラフの頂点数のことであり、 $|G| = |V(G)|$ で表される。以降、グラフの位数は n で表すものとする。

平面的グラフ (planar graph) とは辺が互いに交わらないように平面に描くことができるグラフのことである。平面グラフ (plane graph) とは平面的グラフを平面に辺が交わらないように描いた図のことである。Figure 1.2 は平面的グラフの例である。平面グラフは平面を幾つかの領域に分割する。この領域ひとつひとつを 面 (face) という。また、ただひとつの非有界な面のことを 外面 (outer face) という。



平面的ではないグラフ

平面的グラフ

平面グラフ

Figure 1.2 平面的グラフ

パス (path) とは $V(P) = \{v_1, v_2, \dots, v_n\}$ 、 $E(P) = \{v_1v_2, v_2v_3, \dots, v_{n-1}v_n\}$ で与えられるグラフ P のことである。ただし、 $v_i \neq v_j$ ($1 \leq i < j \leq n$) とする。 P をパスとしたとき、グラフ $C := P \cup \{v_nv_1\}$ ($n \geq 3$) を サイクル (cycle) という。Figure 1.3 はパスとサイクルの例である。



Figure 1.3 パス, サイクル

グラフ G の任意の 2 頂点 u, v に対して, u と v を結ぶパスが存在するとき, G は連結 (*connected*) であるという. $k \geq 2$ について, グラフ G が k -連結 (k -*connected*) であるとは, G から任意の k 個未満の頂点を取り除いても, G が連結であるときをいう.

サイクルを含まないグラフを 森 (*forest*) といい, 連結な森は 木 (*tree*) という.

2 格子描画アルゴリズム

平面グラフの 線分描画 (*straight line drawing*) とは頂点を点で, 辺を頂点同士を結び共通の端点以外で互いに交わらない線分で描いたものである. 格子描画 (*grid drawing*) とは, グラフの各頂点を 2 次元格子平面の整数格子の上に配置した線分描画である.

Theorem 2.1 (H. de Fraysseix, J. Pach and R. Pollack [3]) 任意の n 頂点平面的グラフは $(2n - 4) \times (n - 2)$ の格子平面上に格子描画を持ち, それは $O(n^2)$ 時間で得ることができる.

上で示されたアルゴリズムを FPP アルゴリズムと呼ぶことにする. FPP アルゴリズムでは, 頂点に *canonical ordering* と呼ばれる順序をつけ, その順に頂点を格子平面上に描く. 新たな頂点を描く際に, いくつかの頂点の座標をずらし, 隣接頂点の座標から新たな頂点の座標を決定する. この方法は シフト法 と呼ばれる. Figure 2.4 は 6 頂点の平面的グラフを入力とし, FPP アルゴリズムを実行した結果得られた, 8×4 の格子平面上に描かれた格子描画である.

FPP アルゴリズムは $O(n^2)$ のアルゴリズムであったが, M. Chrobak and T. H. Payne [2] は, これを改良した $O(n)$ 時間のアルゴリズムを示した. これを CP アルゴリズムと呼ぶことにする.

CP アルゴリズムでは, グラフの頂点間の関係を二分木を使って表す. さらに各頂点は, 自身の x 座標と, 二分木での親の頂点の x 座標の差 (x -offset) をもつ. この木と, 各頂点の x -offset から, 各頂点の最終的な x 座標を求めることができる. また, ある頂点の x -offset を変えることにより, いくつかの頂点の x 座標をまとめて変えることができるので, 計算時間を減らすことができる. Figure 2.5 は Figure 2.4 のグラフに CP アルゴリズムを適用した際にできる二分木と x -offset, 各頂点の x 座標である.

上で述べた FPP アルゴリズムのシフト法と, CP アルゴリズムでの木と x -offset は後のアルゴリズムでも利用されている.

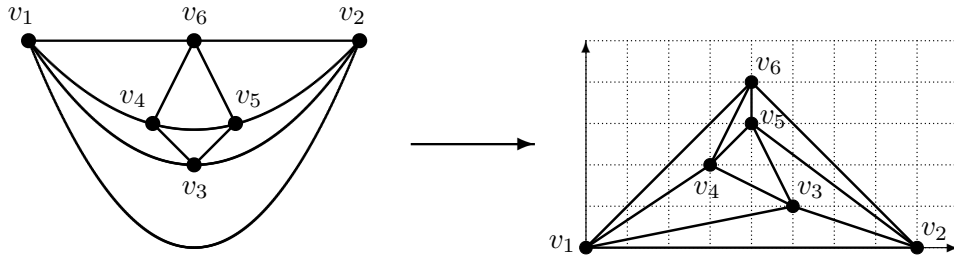
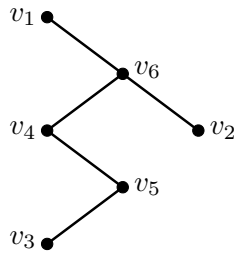


Figure 2.4 FPP アルゴリズムで構築した格子描画



k	親	$\Delta x(v_k)$	$x(v_k)$
1	—	0	0
2	6	4	8
3	5	1	5
4	6	-1	3
5	4	1	4
6	1	4	4

Figure 2.5 Figure 2.4 のグラフに CP アルゴリズムを適用した場合の二分木 T と x -offset, 各頂点の x 座標

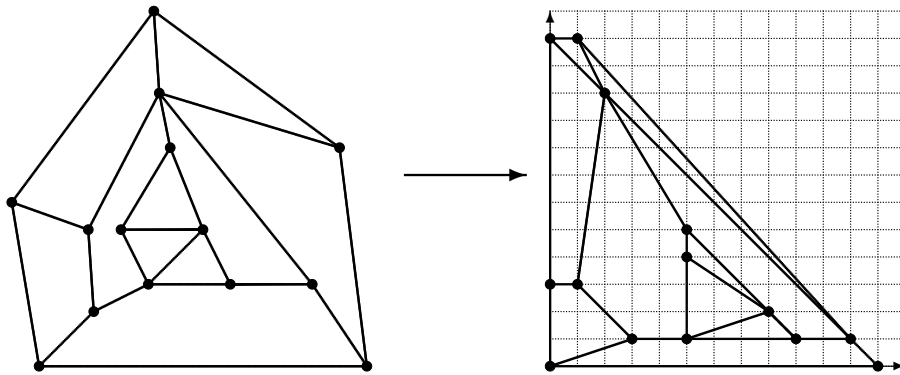


Figure 2.6 CK アルゴリズムで構築した格子凸描画

平面的グラフの格子凸描画とは、グラフの外側を除いた全ての面が凸多角形となるような格子描画のことをいう。M. Chrobak and G. Kant が格子凸描画についてのアルゴリズムを示している。このアルゴリズムを CK アルゴリズムと呼ぶことにする。

Theorem 2.2 (M. Chrobak and G. Kant [1]) 任意の 3-連結平面グラフは $(n-2) \times (n-2)$ の格子平面上に格子凸描画を持ち、それは $O(n)$ 時間で得ることができる。

CK アルゴリズムでも、FPP アルゴリズムのシフト法と、CP アルゴリズムの木の構造と x -offset が利用されている。Figure 2.6 は 14 頂点のグラフに CK アルゴリズムを適用した結果得られた 12×12 の格子平面上に描かれた格子凸描画である。

X. Zhou, T. Hikino and T. Nishizeki [5] はグラフをある条件を満たす 2 つの部分グラフ (bipartition) に分割し, それぞれを CK アルゴリズムを用いて格子描画したものをつなぎ合わせてグラフ全体の描画を得るアルゴリズムを考案した.

Theorem 2.3 (X. Zhou, T. Hikino and T. Nishizeki [5]) G を平面グラフとし, G_1, G_2 を G の任意の bipartition とする. このとき, G は次のような格子描画 D をもつ.

$$\begin{aligned}W(D) &\leq \max\{n_1, n_2\} - 1 \\H(D) &\leq \max\{n_1, n_2\} - 2\end{aligned}$$

ただし, $n_1 = n(G_1)$, $n_2 = n(G_2)$ である. さらに, このような格子描画は $O(n)$ 時間で得ることができる.

参考文献

- [1] M. Chrobak and G. Kant, *Convex grid drawings of 3-connected planar graphs*, **International Journal of Computational Geometry and Applications** **7**, pp. 211-223, 1997.
- [2] M. Chrobak and T. H. Payne, *A linear-time algorithm for drawing planar graph on a grid*, **Information Processing Letters** **54**, pp. 241-246, 1995.
- [3] H. de Fraysseix, J. Pach and R. Pollack, *How to draw a planar graph on a grid*, **Combinatorica** **10**, pp. 41-51, 1990.
- [4] G. Kant, *Drawing planar graphs using the lmc ordering*, **Proc. 33rd Symp. on Foundations of Computer Science**, pp. 101-110, 1992.
- [5] X. Zhou, T. Hikino and T. Nishizeki, *Small grid drawings of planar graphs with balanced partition*, **Journal of Combinatorial Optimization** **24** pp. 99-115, 2012.