4H-05

外平面グラフに対する Koller の $L(2,1)$ ラベリングアルゴリズムの計算時間解析 *

山中　寿登 †　　　　小野　廣隆 ‡
† 九州大学経済学部　　　　‡ 九州大学大学院経済学研究院

1　はじめに

グラフの頂点への非負整数の割り当てをラベリングという．本研究では距離 2 以下の頂点同士のラベル値に制約がある $L(2,1)$ ラベリング問題（使用ラベル値範囲の最小化）を扱う．

この問題は平面グラフに対しては次数を制限しても NP 困難であり, 直並列グラフに対しても NP 困難であるが, 木グラフに対しては線形時間アルゴリズムが知られている (詳しくは [2] の調査を参照されたい).

本研究では, これらの中間に位置するグラフクラスである外平面グラフに対する Koller の多項式時間アルゴリズムの計算量 [3] について考察する. Koller のアルゴリズムは計算時間 $O(n \cdot \Delta^{4\Delta+7})$ で動く A アルゴリズム (ただし, Δ はグラフの最大次数) と, $\Delta \geq 1000$ かつ $\frac{\ln n}{\ln \Delta} < 0.01\Delta$ を満たすグラフに対して,$O(n^{36})$ で動く B アルゴリズムからなっている. B アルゴリズムが適用できない場合, A アルゴリズムを適用することになるが, $\Delta \leq 1000$ のときは線形時間, $\Delta \geq 1000$ のときは $O(n^{407})$ であるため, 総合的に多項式時間を達成している. しかし，この $O(n^{407})$ は多項式ではあるものの次数が非常に大きく現実的ではない.

本研究では, まず Koller のアルゴリズムを整理, 再構築することにより, 同様のアルゴリズムが $\frac{\ln n}{\ln \Delta} > 0.25\Delta$ のとき $O(n^{13.5} \cdot \Delta^6)$ で, それ以外のとき $O(n^{9+\varepsilon} \cdot \Delta^{15} \log n)$ で実行可能であることを示す（ただし ε は微小な正定数）. さらに, このアルゴリズムに, 木に対して提案された蓮沼らの線形時間の (近似) 貪欲アルゴリズム [1] を拡張したものを組み合わせることにより, 外平面グラフに対する最適な $L(2,1)$ ラベリングが $O(n^{16.5+\varepsilon} \log n)$ 時間で求めることができることを示す.

2　準備

2.1　グラフラベリング

グラフラベリングについて次のように定義する.

定義 1　グラフ $G = (V, E)$ の $L(2,1)$ ラベリングとは関数 $L : V \to N \cup \{0\}$ で以下を満たすもののことを言う: u と v が隣接するとき, $|L(u) - L(v)| \geq 2$, u と v が共通の隣接する頂点を持つとき $L(u) \neq L(v)$.

最適な $L(2,1)$ ラベリングとは使用ラベル値範囲を最小にするもののことを言い, その使用ラベル値範囲を σ で表す. σ の下限としては以下が知られている.

命題 1　任意のグラフに対して, $\sigma \geq \Delta + 2$ が成立する.

2.2　外平面グラフ

外平面グラフとは, すべての頂点が外面に面しており, 辺は交差しない平面描画を持つグラフである. 外平面グラフの $L(2,1)$ ラベリングについて以下が成立する.

命題 2　([3]) 外平面グラフにおいて, 一般に $\sigma \leq \Delta + 8$ が, $\Delta \geq 9$ の場合は $\sigma \leq \Delta + 3$ が成立する.

3　Koller のアルゴリズム

1 節で述べたように Koller のアルゴリズムは A アルゴリズムと B アルゴリズムからなる. これらは共に分割統治法に基づいており, 下記のステップを再帰的に繰り返す. その説明のため以下の 3 つを定義する.

- $G_1 = \{v_2, v_3, \ldots, v_i\}$($v_i$ はグラフ G 上のある頂点)
- $G_2 = \{v_i, v_{i+1}, \ldots, v_n, v_1\}$
- $N_{G_x}[v]$: グラフ G_x における頂点 v とそれに隣接する頂点を表わす頂点集合

A アルゴリズムを以下で定義する.

Step 1. 外平面グラフ G を G_1 と G_2 に分割する.

Step 2. $N_{G_1}[v_2, v_i]$ と $N_{G_2}[v_1, v_i]$ に $\{0, 1, \cdots, \Delta + y\}$ からラベルを割り当てる (ただし, $y \in \{1, 2, \cdots, 7\}$).

Step 3. $L(2,1)$ 制約を満たしているかを確認するために G_1 と G_2 のラベリングされた頂点を見比べる.

ここで, $|N_{G_1}[v_2, v_i]|, |N_{G_2}[v_1, v_i]| \leq 2\Delta + 2, \sigma \leq \Delta + 8$ に注意すると, 頂点数 n, 最大次数 Δ のグラフに対する A

---

*本研究は旭硝子財団, JSPS 科研費 JP26540005, MEXT 科研費 JP24106004 の助成を受けた.
A Computational Time Analysis of Koller's L(2,1)-labeling Algorithm for Outerplanar graphs
†Hisato YAMANAKA　†School of Economics, Kyushu University
‡Hirotaka ONO　‡Faculty of Economics, Kyushu University

アルゴリズムの計算量 $F(n,\Delta)$ は, 以下のように見積もることができる.

$$\begin{aligned} F(n,\Delta) &= O(n\cdot(\sigma^{2\Delta+2})^2\cdot(2\Delta+2)^2) \\ &= O(n\cdot(\Delta+8)^{4\Delta+4}\cdot(2\Delta+2)^2) \end{aligned} \qquad (1)$$

Koller は [3] で式 (1) を $O(n\cdot\Delta^{10\Delta})$ と評価している. 1 節で述べたように, A アルゴリズムは $\Delta \leq 1000$ または $\frac{\ln n}{\ln\Delta} \geq 0.01\Delta$ のときに実行される. 前者の場合, その計算量は線形であるが, 後者の場合, $n \geq \Delta^{0.01\Delta}$ から $O(n\cdot\Delta^{10\Delta}) = O(n^{1001})$ としている.

しかし, そもそも式 (1) を $O(n\cdot\Delta^{10\Delta})$ と見積もるのは過剰であり, 二項定理を用いることで $O(n\cdot\Delta^{4\Delta+7})$ と評価できる. この評価を用いると, 上記条件下での A アルゴリズムの計算量は $O(n^{408})$ となる.

B アルゴリズムは, 基本的には A アルゴリズムと同じスキームで動くが, Step 2 において, n,Δ により定義されるパラメータ k によって使用するラベル値を制限する点と, 仮想的にワイルドカードとして機能するラベル h を用いる点が異なる. アルゴリズム B の Step 2 は以下のように定義される.

Step 2. $N_{G_1}[v_2]$ と $N_{G_2}[v_1]$ に $\{0,1,\ldots,k-1\}\cup\{\Delta+2-k,\Delta+3-k,\ldots,\Delta+1\}\cup\{h\}$ からラベルを割り当てる. ただし, $k=\lceil\frac{2\ln n}{\ln 0.1\Delta}+2\rceil$.

ここで, 仮想ラベル h は, それぞれ実態的には $\{k,k+1,\ldots,\Delta+1-k\}$ のうちのいずれかの値をとるラベルであり, それを考慮した実行可能性は保証するものとする. このようなラベリングを H-ラベリングと呼ぶ. つまり, B アルゴリズムは $L(2,1)$ ラベリングを求めるアルゴリズムではなく, H-ラベリングを求めるアルゴリズムとなっている. Koller は以下の定理を示した.

補題 1 ([3]) $\Delta \geq 1000$, $\frac{\ln n}{\ln\Delta} < 0.01\Delta$ を満たすグラフにおいて, H-ラベリングが存在するとき, かつそのときに限り, $\sigma=\Delta+2$ の $L(2,1)$ ラベリングが存在する.

A アルゴリズムのと同様の解析により, B アルゴリズムの計算時間 $F(n,\Delta)$ は以下のようになる.

$$\begin{aligned} F(n,\Delta) &= O(n\cdot((\Delta+2)^{2k})^2\cdot(2\Delta+2)^2\cdot\Delta) \\ &= O(n\cdot\Delta^3\cdot(\Delta+2)^{4k}) \end{aligned}$$

Koller はこれを k については $k \leq \frac{4\ln n}{\ln\Delta}$, また $(\Delta+2)^{2k} \leq (\Delta^2)^{8\ln n/\ln\Delta} = \Delta^{\ln n^{16}/\ln\Delta} = n^{16}$ として, その計算量を $O(n^{36})$ としている.

4 アルゴリズムの改善及び拡張

4.1 外平面グラフ版貪欲アルゴリズム

紙面の都合上, 結果のみ記す.

定理 1 $\Delta > \sqrt{n+\frac{273}{16}}+\frac{19}{4}$ を満たす外平面グラフに対して, 最適な $L(2,1)$ ラベリングを線形時間の貪欲法により求めることができる.

このアルゴリズムの存在から, 対象とする外平面グラフは $\Delta = O(n^{1/2})$ に限定することができる.

4.2 A アルゴリズムの改善

A アルゴリズムの Step 2. を以下のように調整することにより, その計算量を削減することができる: $|N_{G_1}[v_i]| = x, |N_{G_2}[v_i]| = y$(ただし $x+y \leq \Delta$) とし, v_i へのラベリングをしてから $N_{G_1}[v_2,v_i]$ と $N_{G_2}[v_1,v_i]$ にラベリングする. このとき, その計算時間は $O(\sigma\cdot\sigma^{\Delta+x+1}\cdot\sigma^{\Delta+y+1})$ 時間すなわち $O(\sigma^{3\Delta+3})$ となる. これにより全体の計算時間 $F(n,\Delta)$ は以下のようになる.
$F(n,\Delta) = O(n(\Delta+8)^{3\Delta+3}(2\Delta+2)^2) = O(n\Delta^{3\Delta+6})$.

4.3 補題の一般化

以下のように補題 1 に対してパラメータ α,β,t を導入し一般化した補題を証明することができる.

補題 2 $\alpha \leq \frac{\beta-t}{4\beta}\left(1-\frac{6}{10^t}-\frac{19}{10^\beta}\right), \beta > t$ を満たす定数 $\alpha > 0$, $\beta \geq 1$, $t \geq 1$ が与えられたとする. $\Delta \geq 10^\beta$, $\frac{\ln n}{\ln\Delta} < \alpha\Delta$ を満たすグラフにおいて, $k=\lceil\frac{2\ln n}{\ln 0.1^t\Delta}+2\rceil$ とした H-ラベリングが存在するとき, かつそのときに限り, $\sigma=\Delta+2$ の $L(2,1)$ ラベリングが存在する.

本補題の条件式を満たすとき, B アルゴリズムを実行し, 満たさないとき, A アルゴリズムを実行することにすると, 前節の貪欲アルゴリズムを利用することにより, 全体としての計算量を正定数 $\varepsilon > 0$ に対して, $O(n^{16.5+\varepsilon}\log n)$ とする α,β,t が存在する.

定理 2 外平面グラフの最適 $L(2,1)$ ラベリングのための $O(n^{16.5+\varepsilon}\log n)$ 時間アルゴリズムが存在する ($\varepsilon > 0$).

参考文献

[1] T. Hasunuma, T. Ishii, H. Ono, Y. Uno, “An $O(n^{1.75})$ algorithm for L(2,1)-labeling of trees”, *Theoretical Computer Science*, 410 (38): 3702–3710, 2009.

[2] T. Hasunuma, T. Ishii, H. Ono, Y. Uno, “Algorithmic aspects of distance constrained labeling: a survey”, *International Journal of Networking and Computing*, 251–259, 2014.

[3] A. E. Koller, *The frequency assignment problem*, PhD thesis, Brunel University, School of Information Systems, Computing and Mathematics, 2005.