

二部グラフにおける
最大ウェイトマッチングの
探索アルゴリズム

青山学院大学 理工学部 物理・数理学科
学籍番号:15118135 荻野晴史
指導教員 西山 享

2022年2月19日

概要

グラフというものは頂点の集合とそれを結ぶ辺の集合の組として与えられる。頂点が2種類に分かれているようなグラフであって同じグループの頂点同士が辺で結ばれないように2つの集合に分けたものを二部グラフといい、この論文では二部グラフを考える。

二部グラフにおけるマッチングとは、辺によって2種類の頂点の間の対応を与えるようなものである。数学的にこれを言い表すと、辺の部分集合 M であり、 M に含まれるどの2つの辺も端点を共有しないようなものということになる。

本研究では各辺にウェイトと呼ばれる正数を与えたとき、マッチングに含まれる辺のウェイトの合計が最も大きくなるマッチング、つまり「最大ウェイトマッチング」の探索アルゴリズムについて考察する。主結果は探索アルゴリズムを与えたことである。

本論文では、まずグラフやマッチングに関する定義をした後、最大マッチングとその探索アルゴリズムについて勉強したことをまとめる。その後、自ら定義したウェイト増加道を利用して、最大ウェイトマッチングの探索アルゴリズムを考案し、それが正しく機能することの証明を行う。その中で、ウェイト増加道や最大ウェイトマッチングの探索アルゴリズムを考案、証明するために考えたアイデアなどを紹介する。

また、卒業研究で学んだ内容の1つである分配関数とカステレイン行列についても本論文で触れる。

目次

1	序論	2
1.1	研究の背景	2
1.2	研究の主結果	2
1.3	本論文の構成	3
2	グラフと二部グラフ	3
2.1	グラフ	3
2.2	道	4
2.3	二部グラフ	4
3	マッチング	5
3.1	マッチング	5
3.2	交互道と交互閉路	6
4	最大マッチングの探索アルゴリズム	6
4.1	最大マッチング	7
4.2	増加道	7
4.3	探索アルゴリズム	8
4.4	探索アルゴリズムが機能することの証明 (定理 17 の証明)	10
5	グラフのウェイト付けとマッチングのウェイト	10
5.1	ウェイト	10
5.2	分配関数とカステレイン行列	11
6	最大ウェイトマッチングとその探索アルゴリズム	14
6.1	最大ウェイトマッチング	14
6.2	ウェイト増加道	15
6.3	最大ウェイトマッチングの探索アルゴリズム	16
7	探索アルゴリズムが機能すること (定理 38) の証明	19
8	結論	20

1 序論

1.1 研究の背景

私が本研究をしようと思った動機は、小学校の時、バスの席決めに時間がかかったことを思い出し、学んでいた教科書の中にマッチングの項目を見つけたからである。

マッチングは2つのもののペアを考えるので、「人」と「席」のペアにすればマッチングを用いて最適に近い席決めができるのではないかと考えた。しかし、安定マッチング(安定結婚)問題として考えようとする、「人」から「席」への評価だけでなく、「席」から「人」への評価を定義する必要があるため、他の観点からマッチングを考えようとした。その結果、ウェイトのという形で「人」から「席」の評価のみを考えることができるのではと思い、本研究で扱う「最大ウェイトマッチング」の考察に至った。

この論文を書くにあたって、[ポリア]や[高橋金久]、[ハーツ]から定義や定理を引用しており、それらが非常に役に立っている。[ポリア]、[高橋金久]、[ハーツ]の著者には大変感謝している。

1.2 研究の主結果

頂点の集合と辺の集合の組み合わせをグラフといい、頂点を同じ種類の頂点同士が辺で結ばれないように2種類に分けられるグラフを二部グラフと言う。2つの辺が頂点を共有しない様な辺の部分集合をマッチングといい、辺の数が最大になるようなマッチングを最大マッチングという。

最大マッチングを求めることは重要であるが一般には難しい問題である。そこで一つでもよいから最大マッチングを求めるアルゴリズムを考える。このようなアルゴリズムは[ポリア]で紹介されている。「最大マッチングの探索アルゴリズム」(定義 16)を学んでみると、「探索アルゴリズムが機能すること」(定理 17)の証明に、辺の数を増やす操作ができる「増加道」(定義 14)や「マッチングの対称差を比べる」といった考え方が大変役に立つことが分かった。

そこで、ウェイトを最大にするようなマッチングを見つけることを卒業論文のテーマに選んだ。まず「最大ウェイトマッチング」(定義 37)を定義し、自ら考えた「ウェイト増加道」(定義 35)を利用して「最大ウェイトマッチングの探索アルゴリズム」(定義 37)を考案した。実際に探索アルゴリズムが機能すること(定理 38)も証明できた。

このとき、教科書[ポリア]で学んだアイデアを用いて、「ウェイト増加道の性質を利用」したり「対称差を比べ」たりと様々な工夫をして探索アルゴリズムを検証することができた。

1.3 本論文の構成

§2 ではグラフおよび二部グラフの基本的性質について述べる. §3 ではマッチングとそれに関する用語を定義する. §4 では最大マッチングの探索アルゴリズムとそれが機能することの証明を [ポリア] に基いて紹介する. §5 では, ウェイトについての定義と, ウェイトに関係することとして, 卒業研究の中で学んだ分配関数とカステレイン行列について触れる. §6 では最大ウェイトマッチングを定義した後, 自ら考えたウェイト増加道を利用し, 最大ウェイトマッチングの探索アルゴリズムを定義する. §7 では本論文の主目標である最大ウェイトマッチングの探索アルゴリズムが機能することの証明を行う. 最後に §8 で, 研究結果のまとめと, 今後の課題について述べる.

2 グラフと二部グラフ

この章では, グラフやマッチングなどについての定義を行う.

2.1 グラフ

グラフとは, いくつかの頂点を辺で結んだものである. 一般的に言ってグラフは頂点の繋がり方を表したものであるが, これをより厳密に考えるため, 頂点の集合と辺の集合の組として考えることにしよう.

定義 1 (グラフ). グラフ G とは, 頂点の集合 V と 2 頂点を結ぶ辺の集合 E の組である. V と E を

$$V = \{v_1, v_2, \dots, v_n\}$$
$$E \subset V \times V \quad (e \in E \text{ を } e = (v_i, v_j) \text{ のように表す})$$

と表せば, グラフ G は

$$G = (V, E)$$

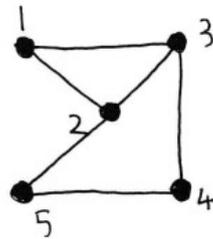
で与えられることになる. この論文では, 図 1 が 5 つの頂点からなるグラフを図で表したものである. 辺の向きについていない無効グラフを考えるため, (v_1, v_2) と (v_2, v_1) は同じものとみなす. また, 同じ頂点を結ぶような辺 (ループ) は考えない. つまり $(v_1, v_2) \in E$ なら $v_1 \neq v_2$ である.

例 2. 図 1 のグラフを $G = (V, E)$ とすると,

$$V = \{1, 2, 3, 4, 5\} \quad E = \{(1, 2), (1, 3), (2, 3), (2, 5), (3, 4), (4, 5)\} \quad (1)$$

となる.

図 1: グラフの例



2.2 道

道はグラフ G の中で、辺を端点で接続した折れ線状の部分グラフのことである。それを厳密に定義すると次のようになる。

定義 3 (道). 道 L とは、隣り合う辺が端点を共有する辺の列であり、

$$L = (e_1, e_2, \dots, e_l) \quad (e_i = (v_i, v_{i+1}) \in E, v_k \in V)$$

と表せる。

定義 4 (単純な道と閉路). (1) 道 L に自己交叉がないとき、 L を**単純な道**という。

(2) 道 L の始点 v_1 と終点 v_{n+1} が一致する場合、 L は**閉路**という。

図 2: 単純な道の例

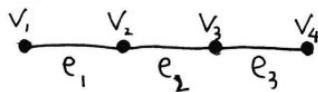
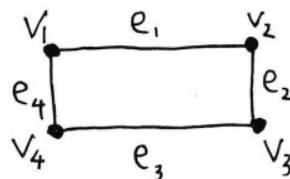


図 3: 閉路の例



グラフでマッチングを考えようとする、頂点の組を考えたいので、ペアとして考えた頂点を区別するために種類を分けて二部グラフを考える必要がある、次節で二部グラフの説明をする。

2.3 二部グラフ

二部グラフは頂点を 2 種類考え、異なる種類の頂点のみが辺で結ばれているような特殊なグラフである。

定義 5 (二部グラフ). 次の 2 つの条件を満たすグラフ G を**二部グラフ**という。

1. 頂点が2種類存在し、それぞれを $B = \{b_1, b_2, \dots, b_{n_1}\}, R = \{r_1, r_2, \dots, r_{n_2}\}$ とすると $V = B \sqcup R$ となる. (B は黒, R は赤をイメージしている)
2. 同じ種類の頂点同士は辺で結ばれない. つまり $E \subset B \times R \sqcup R \times B$ である. (ただし, (b_i, r_j) と (r_j, b_i) は同一視して考える.)

例 6. 二部グラフの例を2つあげる. 図で表すときは B を黒のドット, R を白のドットで描く.

図 4: 二部グラフの例 (1)

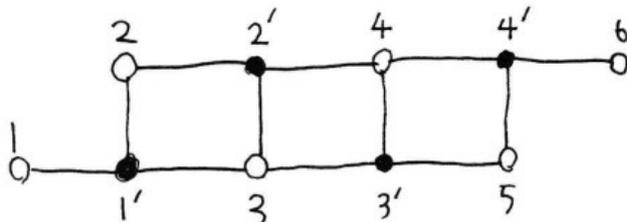
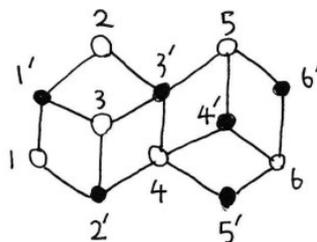


図 5: 二部グラフの例 (2)



3 マッチング

マッチングとは2種類の頂点に対応づけることある。「ルームメイト問題」のように1つの集合からペアを選ぶものもあるが、この論文では2つの集合からペアを選ぶマッチングを考え、それを二部グラフを用いて表現する.

3.1 マッチング

定義 7 (マッチング). 二部グラフ $G = (V = B \sqcup R, E)$ のマッチング M とは, E の部分集合であって, M のどの2つの辺も端点を共有しないようなものである. つまりマッチングを

$$M = \{e_1, e_2, \dots, e_m\} \quad (e_i \in E)$$

と表すと, $e_i, e_j (i \neq j)$ は端点を共有していない. また, 1つの辺からなる集合や空集合もマッチングと考える.

例 8. 図 6 の二部グラフでは $M = \{(1', 3), (2', 4), (4', 5)\}$ がマッチングの例である.

以降この論文では, マッチングに含まれる辺を二重線で表す.

図 6: マッチングの例 (1)

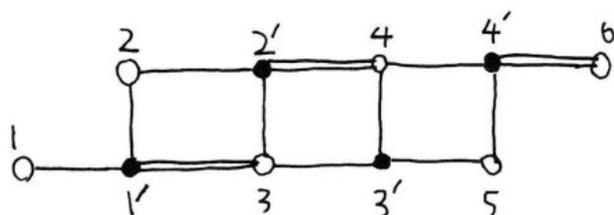
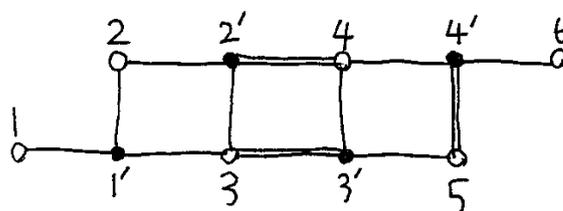


図 7: マッチングの例 (2)



定義 9 (自由頂点). 自由頂点とは、 M に含まれている辺の端点でない頂点のことである。マッチングに含まれる辺の端点の集合を $V(M)$ と書くと、 $V \setminus V(M)$ はマッチング M に関する自由頂点の集合となる。

図 6 では $1, 2, 3', 6$ が自由頂点である。

定義 10 (完全マッチング). マッチング M に含まれる辺がグラフ G 内の全ての頂点を端点としているとき、 M は完全マッチングという。つまり、 $V(M) = V$ となっているとき M は完全マッチングである。グラフ G の完全マッチング全体の集合を $\mathcal{M} = \mathcal{M}(G)$ で表す。

3.2 交互道と交互閉路

二部グラフ G に対してマッチング M を一つ定めたとき、 M に関係するような特殊な道を考えることができる。

定義 11 (交互道と交互閉路). 二部グラフ G に対してマッチング M を一つ定める。

- (1) M に含まれる辺と含まれない辺が交互に現れる単純な道を M に関する交互道という。
- (2) M に含まれる辺と含まれない辺が交互に現れる閉路を M に関する交互閉路という。閉路も単純な道も道の 1 つなので閉路は道であるが、交互道は単純な道の 1 つなので、交互閉路は交互道ではないことに注意する。

例 12. 図 7 のマッチングで考えた交互道と交互閉路の例をあげる。

4 最大マッチングの探索アルゴリズム

マッチング M は様々な選び方ができるが、この章では M に含まれる辺の数が最大になるようなマッチング (最大マッチング) を求めるアルゴリズムを紹介する。このアルゴリズムは教科書 [ポリア] で勉強したものだが、実際に最大マッチングを探索できることの証明も紹介する。

図 8: 交互道の例

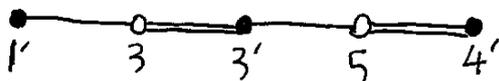
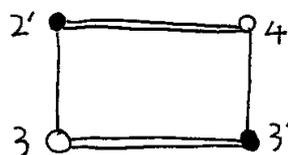


図 9: 交互閉路の例



4.1 最大マッチング

まず 2 部グラフにおける最大マッチングを定義しよう。

定義 13 (二部グラフにおける最大マッチング). 二部グラフ $G = (B \sqcup R, E)$ のマッチング M であって、辺の数が最大になるようなものを**最大マッチング**という。最大マッチングは 2 部グラフ G に対して多数存在する。

4.2 増加道

最大マッチングは一度の計算で求めることが難しい。そこで、マッチング M をとったとき、 M より多くの辺を持つマッチングを探し出すアルゴリズムが必要である。そのアルゴリズムのキーポイントになるものが**増加道**である。

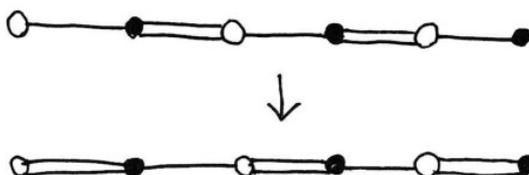
定義 14 (増加道). 二部グラフ G に対してマッチング M を一つ定めたとき、道の始点と終点が M に関する**自由頂点**である**交互道**を M に関する**増加道**という。

図 10: 増加道の例



観察 15 (増加道の性質). マッチング M に対して増加道が存在する場合、増加道内のマッチングに含まれる辺、含まれない辺を入れ替えることにより、マッチングに含まれる辺の数が 1 だけ大きくなるようにマッチングを更新できる。

図 11: 増加道を利用したマッチングの更新の例



4.3 探索アルゴリズム

増加道の性質を利用して最大マッチングの探索アルゴリズムを定義しよう.

定義 16 (最大マッチングの探索アルゴリズム).

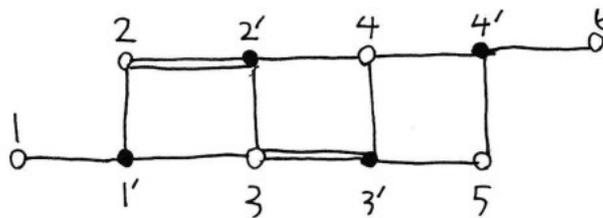
1. 二部グラフ G に対して任意のマッチング M を選ぶ.
2. M に関する 増加道を探す.
3. 増加道の性質 (観察 15) を利用し, 辺の本数が 1 大きくなるように M を更新し, それを M' とする.
4. 手順 2, 3 を M' に対して繰り返して, 増加道が存在しなくなるまで続ける.

定理 17. 最大マッチングの探索アルゴリズム (定義 16) で得たマッチングは最大マッチングである.

例 18 (探索アルゴリズムの例). 図 4 の二部グラフ G における最大マッチングを定義 16 のアルゴリズムで探索する.

1. 図 12 のように, 任意のマッチングとして $M_1 = \{(2, 2'), (3, 3')\}$ を選ぶ.

図 12:



2. このとき, $L_1 = ((1', 2), (2, 2'), (2', 3), (3, 3'), (3', 4))$ は M_1 に関する増加道である.

図 13: 赤色の辺が M_1 に関する増加道 L_1

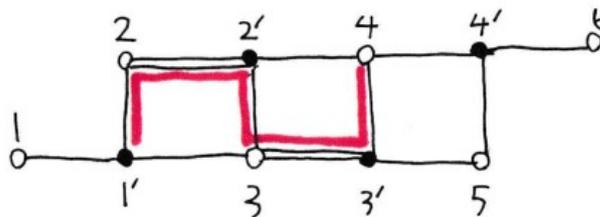
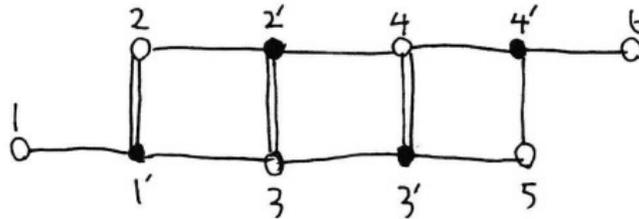
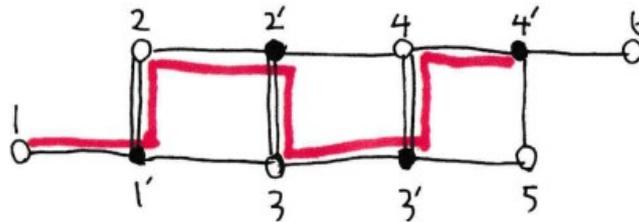


図 14:



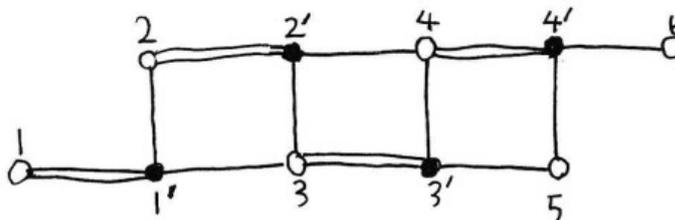
3. L_1 を利用して図 14 のように $M_2 = \{(2, 1'), (3, 2'), (4, 3')\}$ とマッチングを更新する.
4. このとき, $L_2 = ((1, 1'), (1', 2), (2, 2'), (2', 3), (3, 3'), (3', 4), (4, 4'))$ は M_2 に関する増加道である.

図 15: 赤色の辺が M_2 に関する増加道 L_2



5. L_2 を利用して図 16 のように $M_3 = \{(1, 1'), (2, 2'), (3, 3'), (4, 4')\}$ とマッチングを更新する.

図 16:



6. M_3 に関する増加道は存在しないため, M_3 は G における最大マッチングである,

4.4 探索アルゴリズムが機能することの証明 (定理 17 の証明)

定理 17 を証明するためには、定義 16 の探索アルゴリズムで得たマッチングが最大マッチングであることを示す必要がある。そのために、まず最大マッチングを 1 つとして M_0 とする。 M_0 は必ず存在するが、比較のために用いるので、具体的でなくてよい。この M_0 と探索アルゴリズムで得たマッチング M の辺の数を比較する。そのために対称差

$$M \ominus M_0 = (M \cup M_0) - (M \cap M_0) \quad (2)$$

に含まれる M に含まれる辺と M_0 に含まれる辺の数を比較する。

補題 19. 2 つのマッチング M, M' の対称差を $M \ominus M' = (M \cup M') - (M \cap M')$ とする。 $M \ominus M'$ に含まれる辺からなる道は M, M' 双方にとって交互道、または交互閉路である。

(証明). $\forall v \in V$ は M, M' に含まれる辺のそれぞれ高々 1 つの端点なので、 $M \ominus M'$ に含まれる辺の高々 2 つの端点である。よって $M \ominus M'$ に属する辺からなる道は分岐が存在しないので単純な道、もしくは閉路になる。また、マッチングに含まれる辺は同じ頂点を共有しないので、それらは全て M, M' にとっての交互道、または交互閉路になる。 \square

補題 19 と定義 14 を利用して定理 17 の証明をする。

定理 17 の証明. 探索で得たマッチングを M , 最大マッチングを M_0 とする。補題 19 より、 $M \ominus M'$ に含まれる辺は M, M_0 双方にとっての交互道、または交互閉路を形成する。

M はアルゴリズムの終了条件より増加道を持たない。 M_0 は辺の数が最大なので増加道を持たない。 M, M' 双方にとって増加道でない交互道には M に含まれる辺と M_0 に含まれる辺が同数現れるため、 $M \ominus M'$ に含まれる M に含まれる辺と M_0 に含まれる辺は同数である。

$M \cap M_0$ に M に含まれる辺と M_0 に含まれる辺が同じ数含まれることは自明であるので、 M と M_0 の辺の数は等しい。よって探索で得たマッチング M は最大マッチングである。 \square

5 グラフのウェイト付けとマッチングのウェイト

5.1 ウェイト

定義 20 (ウェイト). グラフのウェイトは各辺 $e \in E$ に正数 $w(e) > 0$ を与えたものである。これは関数 $w : E \rightarrow \mathbb{R}_{>0}$ と同一視できる。さらに、マッチング M のウェイト $w(M)$ を

$$w(M) = \sum_{e \in M} w(e)$$

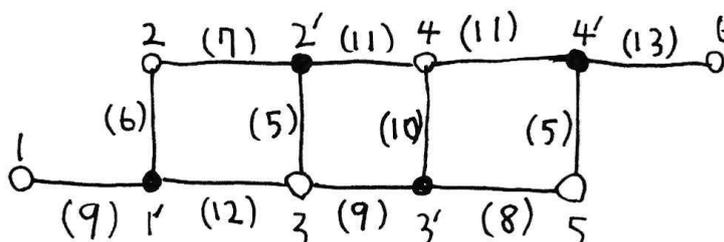
で、道 L のウェイト $w(L)$ を

$$w(L) = \sum_{e \in L} w(e)$$

で定義する.

例 21 (ウェイトを付けた二部グラフの例). 括弧内の数字がウェイトである. 例えば $w((1,1')) = 9$ であり, 道 L を $L = ((1',3), (3,2'), (2',4))$ とすると, $w(L) = 12+5+11 = 28$ である.

図 17: ウェイト付き二部グラフの例



5.2 分配関数とカステレイン行列

再々ウェイトマッチングを探索することとは直接的な関係はないが, ウェイトについて考えるときに, 分配関数を考えることは重要である. 卒業研究である行列を用いて分配関数を計算する方法について学んだのでここにその内容を書いておく.

定義 22 (平面的なグラフ). 二部グラフとは限らない一般のグラフ G が平面上で辺が交差しないように描くことができるとき, G を平面的なグラフという.

定理 23. グラフ G が 5 次の完全グラフ K_5 の細分, または 3 次の完全二部グラフ $K_{3,3}$ の細分を部分グラフとして含むとき, G は平面的でない. 逆にそのような細分が部分グラフに含まれていなければ G は平面的なグラフである.

定理 24 (Kuratowski). G は平面的でないなら, グラフ G は 5 次の完全グラフ K_5 の細分, または 3 次の完全二部グラフ $K_{3,3}$ の細分を部分グラフとして含む.

これらの定理は [ハーツ, 定理 9.1.1, 9.1.2] から引用した

定義 25 (マッチングのウェイト). マッチング M の積ウェイト $w(M)$ を

$$w(M) = \prod_{e \in M} w(e)$$

と定義する.

本論文で扱うマッチングのウェイト $w(M)$ は和で定義されているが、積ウェイト w とは異なる定ことに注意する。

定義 26 (分配関数). グラフの**分配関数** Z は完全マッチング M の重みを用いて

$$Z := \sum_{M \in \mathcal{M}} w(M)$$

と定義する。

この節に限り $B = R = n$ と書く。カステレイン行列 K は(平面的)な二部グラフの分配関数 Z を計算するための行列である。

定義 27. 辺に対して $\varepsilon(w_i, r_j) = \pm 1$ を選んで、

$$k_{ij} = \begin{cases} \varepsilon(w_i, r_j)w(w_i, r_j) & (w_i, r_j) \in E \\ 0 & (w_i, r_j) \notin E \end{cases} \quad (3)$$

とする。 $K = (k_{ij})_{i \leq i, j \leq n}$ をカステレイン行列と呼ぶ。

定理 28. 平面的な二部グラフ G に対して $\varepsilon(b_i, r_j)$ を適切に調整すると、

$$Z = |\det K| \quad (4)$$

(証明).

$$\det K = \sum_{\sigma \in S_n} \text{sgn}(\sigma) k_{1, \sigma(1)} k_{2, \sigma(2)} \cdots k_{n, \sigma(n)}$$

$\sigma \in S_n$ を一つ決める。

$$M = \{(w_1, b_{\sigma(1)}), (w_2, b_{\sigma(2)}), \dots, (w_n, b_{\sigma(n)})\}$$

と書くと、 K の定義より、

$$M \notin E \implies k_{1, \sigma(1)} k_{2, \sigma(2)} \cdots k_{n, \sigma(n)} = 0$$

となる。 $M \in E$ となるもののみが残る。つまりゼロではない行列式 $\det K$ の項の内 $M \in \mathcal{M}$ の重みに符号をつけたものになる。 G が平面的であれば、 $\varepsilon(b, r)$ をうまく選んで

$$\det K = \sum_{M \in \mathcal{M}(G)} \pm w(M) \text{ となる。}$$

□

$\sigma \in S_n$ に対して ε を適切に選ぶことができるための条件を考えよう。

$$\varepsilon(\sigma) = \varepsilon(1, \sigma(1)) \varepsilon(2, \sigma(2)) \cdots \varepsilon(n, \sigma(n))$$

とすると,

$$\det K = \sum_{\sigma \in S_n} = \text{sgn}(\sigma)k_{1,\sigma(1)}k_{2,\sigma(2)} \cdots k_{n,\sigma(n)} = \sum_{\sigma \in S_n} = \text{sgn}(\sigma)\varepsilon(\sigma)w(M)$$

となるので $\text{sgn}(\sigma)\varepsilon(\sigma)$ が σ によらないとき,

$$\det K = \sum_{M \in \mathcal{M}(G)} \pm w(M) = \pm Z$$

となる. ではどの様なときに $\text{sgn}(\sigma)\varepsilon(\sigma)$ が σ によらないのかを考える必要があるが, それにはマッチングの回転を利用する.

定義 29 (マッチングの回転). マッチング M に関する交互閉路に対して M に含まれる辺と M に含まれない辺を入れ替えて $M \rightarrow M'$ と更新する操作を**マッチングの回転**という.

定理 30. マッチングの回転によってマッチングを同じ大きさの他のマッチングに変換できる.

これは [高橋金久, §11,3] で証明されている.

マッチングの回転によって $\text{sgn}(\sigma)\varepsilon(\sigma)$ が変化しないならば, $\text{sgn}(\sigma)\varepsilon(\sigma)$ が σ によらない. 元のマッチングを M に関する交互閉路を

$$C = \{(b_{i_1}, r_{j_2}), (b_{i_2}, r_{j_2}), \dots, (b_{i_c}, r_{j_c}), \}$$

とし, C に対するマッチングの回転後のマッチング M' に関する交互閉路を

$$C' = \{(b_{i_1}, r_{j_c}), (b_{i_2}, r_{j_1}), \dots, (b_{i_c}, r_{j_{c-1}}), \}$$

とする.

C, C' に対する置換をそれぞれ σ, σ' とする,
このとき,

$$\sigma = \begin{pmatrix} i_1 & i_2 & \cdots & i_n \\ j_1 & j_2 & \cdots & j_n \end{pmatrix} \quad (5)$$

$$\sigma' = \begin{pmatrix} i_1 & i_2 & \cdots & i_n \\ j_c & j_1 & \cdots & j_{c-1} \end{pmatrix} \quad (6)$$

となる.

$$\sigma = (j_1, j_2, \dots, j_n)\sigma' \quad (7)$$

だから

$$\text{sgn}(\sigma) = (-1)^{n-1}\text{sgn}(\sigma') \quad (8)$$

である.

$$\operatorname{sgn}(\sigma)\varepsilon(\sigma) = \operatorname{sgn}(\sigma')\varepsilon(\sigma') \quad (9)$$

となるためには

$$\operatorname{sgn}(\sigma)\varepsilon(\sigma) = (-1)^{n-1}\operatorname{sgn}(\sigma')\varepsilon(\sigma) = \operatorname{sgn}(\sigma')\varepsilon(\sigma') \quad (10)$$

でなくてははいけない. 従って

$$\varepsilon(\sigma')\varepsilon(\sigma) = (-1)^{n-1} \quad \therefore \varepsilon(C) = (-1)^{n-1} \quad (11)$$

でなければならない. よって, 任意の交互閉路 C に対して $\varepsilon(C) = (-1)^{n-1}$ となるように ε を定めれば $\operatorname{sgn}(\sigma)\varepsilon(\sigma)$ が σ によらないので,

$$|\det K| = Z \quad (12)$$

が成り立つ.

補題 31. G が平面的な二部グラフなら (12) 式が成り立つように $\varepsilon(e)$ ($e \in E$) を選ぶことができる. [高橋金久, §11,5]

補題 31 によって, 次の定理が成り立つことが分かる.

定理 32 (カステレインの定理). 二部グラフ G が平面的であるなら $\operatorname{sgn}(\sigma)\varepsilon(\sigma)$ が σ によらないように $\varepsilon(w_i, r_j)$ を選ぶことができ, そのような $\varepsilon(w_i, r_j)$ を利用してカステレイン行列 K を定めれば,

$$Z = |\det K|$$

となる.

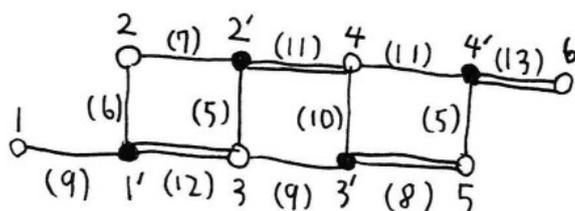
6 最大ウェイトマッチングとその探索アルゴリズム

6.1 最大ウェイトマッチング

定義 33 (最大ウェイトマッチング). マッチング M のウェイト $w(M)$ が最大になるようなマッチング M を**最大ウェイトマッチング**という. 二部グラフ G を決めたとき, 最大ウェイトマッチングは一般には複数存在する.

例 34. 図 17 の二部グラフ G における最大ウェイトマッチング M は, $M = \{(1, 3'), (4, 2'), (5, 3'), (6, 4')\}$ であり, $w(M) = 44$ である.

図 18: 図 17 の二部グラフ G における最大ウェイトマッチングの例



以下では本論文の主目標である、最大ウェイトマッチングを求めるアルゴリズムの考案と、それが機能することの証明を行う。

6.2 ウェイト増加道

最大マッチングを求める時には増加道の性質を利用したので、最大ウェイトマッチングを求めるときにも増加道のようにウェイトを増加させる操作ができる道を定義できれば最大ウェイトマッチングを求められるのではないかと考えた。

そこで、ウェイトを増加させる操作ができる道について考察する。ウェイトを増加させる操作ができる条件は

- マッチング M に含まれる辺と M に含まれない辺を入れ替える操作ができること
- 入れ替える操作をすることで M に含まれる辺のウェイトの合計が増えること

の2つである。

この条件を満たすようにウェイトを増加させる操作ができる道を次のように定義する。

定義 35 (ウェイト増加道). M をマッチングとする. 次の条件 (A), (B), (C) を満たす交互道

$$L = (e_1, e_2, \dots, e_n), \quad e_i = (v_i, v_{i+1}) \quad (1 \leq i \leq n) \quad (13)$$

および条件 (C) のみを満たす交互閉路を**ウェイト増加道**と定義する。

(A) $e_1 = (v_1, v_2) \in M$ または v_1 が M に関する自由頂点

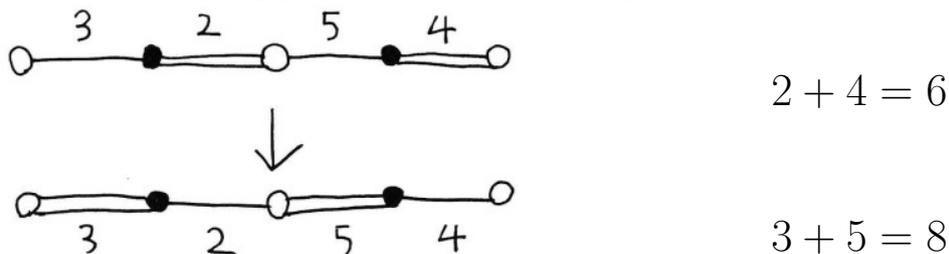
(B) $e_n = (v_i, v_{i+1}) \in M$ または v_{n+1} が M に関する自由頂点

(C) $w(L) > 2 \sum_{e \in M \cap L} w(e)$

ウェイト増加道は、条件 (A),(B) によってマッチングに「含まれる辺と含まれない辺を入れ替える操作」が可能になっていて、条件 (C) によって「入れ替える操作をすることでマッチングに含まれる辺のウェイトの合計が増える」様になっているのでウェイトを増加させる操作ができる様になっている。

観察 36 (ウェイト増加道の性質). マッチング M に対してウェイト増加道が存在する場合, 増加道内のマッチングに含まれる辺, 含まれない辺を入れ替えることにより, マッチングに含まれる辺のウェイトの合計が大きくなるようにマッチングを更新できる.

図 19: ウェイト増加道を利用したマッチングの更新の例



上図の例ではマッチングに含まれる辺のウェイトが $2 + 4 = 6$ から $3 + 5 = 8$ と増加している.

6.3 最大ウェイトマッチングの探索アルゴリズム

ウェイト増加道を利用することで, すでに紹介した最大マッチングの探索アルゴリズム (定義 16) に近い形で最大ウェイトマッチングの探索アルゴリズムを定義することができる.

定義 37 (最大ウェイトマッチングの探索アルゴリズム).

1. 二部グラフ G に対して任意のマッチング M を選ぶ.
2. M に対する ウェイト増加道を探す.
3. ウェイト増加道の性質 (観察 36) を利用し, マッチングのウェイトがより大きくなるように M を更新する.
4. 手順 2, 3 を繰り返し, ウェイト増加道が存在しなくなるまで続ける.

次の定理が本論文の主定理である.

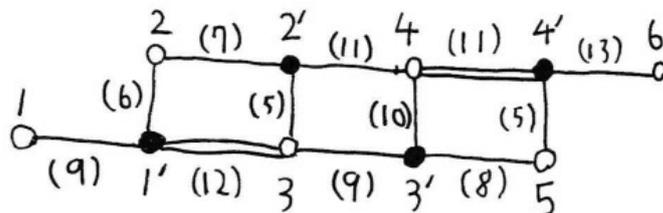
定理 38. 定義 37 の探索で得たマッチングは最大ウェイトマッチングである.

証明の前に, 最大ウェイトマッチングの探索アルゴリズムを実際に使った例を考えてみよう.

例 39 (定義 37 の探索アルゴリズムの使用例).

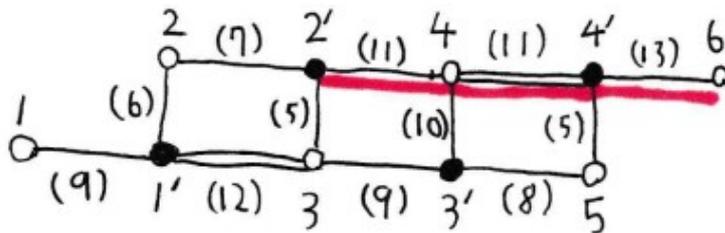
図 17 の二部グラフ G における最大ウェイトマッチングを探索する.

図 20:



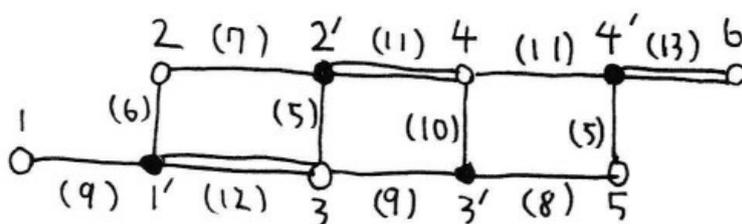
1. 図 20 のように、任意のマッチングとして $M_1 = \{(3, 1'), (4, 4')\}$ を選ぶ.
 $(w(M_1) = 23)$
2. このとき、 $L_1 = ((2', 4), (4, 4'), (4', 6))$ は M_1 に関するウェイト増加道である.

図 21: 赤色の辺が M_1 に関するウェイト増加道 L_1



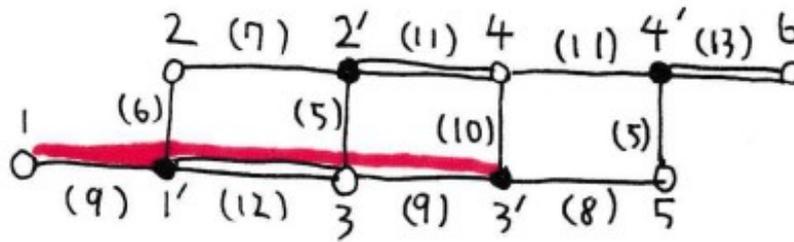
3. L_1 を利用して図 22 のように $M_2 = \{(1', 2), (2', 3), (3', 4)\}$ とマッチングを更新する.
 $(w(M_2) = 36)$

図 22:



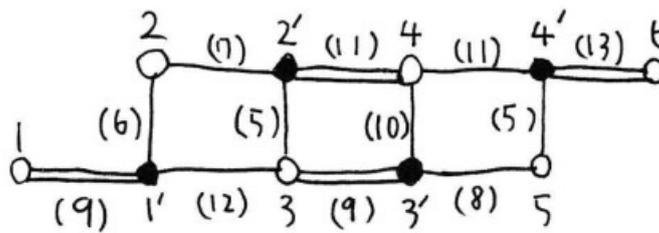
4. このとき、 $L_2 = ((1, 1'), (1', 3), (3, 3'))$ は M_2 に関する増加道である.

図 23: 赤色の辺が M_2 に関するウェイト増加道 L_2



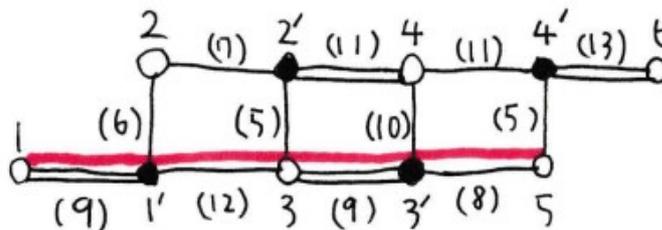
5. L_2 を利用して図 24 のように $M_3 = \{(1, 1'), (3, 3'), (4, 2'), (6, 4')\}$ とマッチングを更新する. ($w(M_3) = 42$)

図 24:



6. このとき, $L_3 = ((1, 1'), (1', 3), (3, 3'), (3', 5))$ は M_3 に関する増加道である.

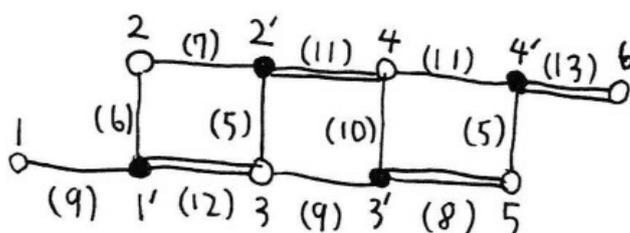
図 25: 赤色の辺が M_3 に関するウェイト増加道 L_3



7. L_2 を利用して図 26 のように $M_4 = \{(1, 3'), (4, 2'), (5, 3'), (6, 4')\}$ とマッチングを更新する. ($w(M_4) = 44$)

8. M_4 に関する増加道は存在しないため, M_4 は G における最大ウェイトマッチングである. 同時に最大ウェイトが 44 であることがわかった.

図 26:



7 探索アルゴリズムが機能すること (定理 38) の証明

定理 38 を証明しよう。

最大マッチングの探索アルゴリズムの証明のときと同じように、最大ウェイトマッチング M_0 をとり、探索で得たマッチング M と M_0 のウェイトを比べる。このとき M_0 は必ず存在するが、比較のために用いるので具体的でなくてよい。

定理 38 の証明.

探索で得たマッチングを M , 最大ウェイトマッチングを M_0 とする。 M と M_0 の対称差 $M \ominus M_0$ に属する辺からなる連結成分をそれぞれ L_1, L_2, \dots, L_n とすると、補題 19 より、 $L_i (1 \leq i \leq n)$ は M, M_0 双方にとっての交互道、または交互閉路を形成する。

そこでまず、 L_i が単純な道ならば、ウェイト増加道の条件 (A), (B) を満たすことを示そう。 $L_i = (e_1, e_2, \dots, e_n), e_1 = (v_1, v_2), e_n = (u_1, u_2) \quad (1 \leq i \leq n)$

とする。 L_i が単純な道であり、 $e_1 \notin M$ であるとする、つまり $\forall v \in V$ に対して $(v, v_1) = e_0 \notin M$ を背理法で示す。

$(v, v_1) = e_0 \in M$ と仮定すると、 $e_0 \notin M \ominus M_0, \therefore e_0 \in M \cup M_0$ となるが、 L_i は交互道であるため $e_1 = (v_1, v_2) \in M_0$. よって定義 7 より $e_0 \notin M_0$ となり、 $e_0 \in M \cap M_0$ と矛盾する。 よって $\forall v \in V$ に対して $(v, v_1) = e_0 \notin M$ であり、 $e_1 \notin M$ であるとき、 v_1 は M にとって自由頂点である。

よって、 L_i は単純な道であるとき M にとってウェイト増加道の条件 (定義 35)(A) を満たす。

条件 (B) についても条件 (A) と同じように満たすことを示すことができる。

L_i は同様に M_0 に対してもウェイト増加道の条件 (A), (B) を満たすことに注意しよう。

さて、アルゴリズムの終了条件より M に関するウェイト増加道が存在せず、 M_0 も最大ウェイトを持つのでウェイト増加道が存在しない、つまり L_i は M に対しても M_0 に対してもウェイト増加道ではない。 よって、 L_i は単純な道であっても閉路であってもウェイト増加道の条件 (C)

$$w(L) > 2 \sum_{e \in M \cap L} w(e) \quad (14)$$

が M, M_0 双方に対して成り立たないことになる.

従って, 繁雑なので L_i を L と書いておくと

$$\begin{aligned} \sum_{e \in M \cap L} w(e) &\leq \frac{w(L)}{2}, \quad \sum_{e \in M_0 \cap L} w(e) \leq \frac{w(L)}{2} \\ w(L) &= \sum_{e \in M \cap L} w(e) + \sum_{e \in M_0 \cap L} w(e) \end{aligned}$$

より

$$\sum_{e \in M \cap L} w(e) = \sum_{e \in M_0 \cap L} w(e) = \frac{w(L)}{2}$$

である. 各 L_i の和をとると

$$\begin{aligned} W((M_0 \ominus M) \cap M) &= \sum_{i=1}^n \left(\sum_{e \in M \cap L} w(e) \right) \\ &= \sum_{i=1}^n \left(\sum_{e \in M_0 \cap L} w(e) \right) = W((M_0 \ominus M) \cap M_0) \end{aligned} \tag{15}$$

であるため.

$$\begin{aligned} w(M) &= w(M \cap M_0) + w(M \cap (M \ominus M_0)) \\ &= w(M \cap M_0) + w(M \cap (M_0 \ominus M_0)) = w(M_0) \end{aligned} \tag{16}$$

である. □

8 結論

本研究では, 二部グラフとマッチングについて考え, 最大マッチングの探索アルゴリズムに用いた考え方を一般化することで, 最大ウェイトマッチングの探索アルゴリズムを考案し, それが機能することを証明することができた.

ウェイト増加道を定義することでアルゴリズムを得ることができたが,

1. ウェイト増加道を総当たり以外の効率的な方法で探索するアルゴリズムの考案
2. 総当たりによるウェイト増加道の探索と, ウェイト増加道を用いずに総当たりで最大ウェイトマッチングを探索する計算量の比較

が今後の課題である.

また, 考案したアルゴリズムが機能することを証明できたので, アルゴリズムを実現するプログラムを製作することも考えたが, 時間の制約で実現しなかった.

卒業研究発表会での質問とそれに対する回答

- アルゴリズムはどのくらいの大きさのグラフで試してみたか (松本裕行先生)
→ 質問を受けた時点で用意ができていなかったなので, 頂点の数が合計 10 個付近のものを手計算で試した.
- ウェイト増加道を効率よく探索する方法は見つかってないが, 増加道の探索方法は存在するのか (市原直幸先生)
→ 効率のいい方法はまだ見つかっていない. 総当たりすれば見つかるが, それは効率的とは言えない.

最後に, 本研究を行うにあたって指導してくださった西山先生に感謝を申し上げます.

参考文献

[ポリア] G. ポリア・R.E. タージャン・D.R. ウッズ (今宮淳美 訳), 組み合わせ論入門, 近代科学社, 1986/9/10.

[高橋金久] 高橋金久, 線形代数と数え上げ, 精興社, 2012/6/30.

[ハーツ] N. ハーツフィールド・G. リンゲル (鈴木晋一 訳), グラフ理論入門, サイエンス社, 1990/12/01.