

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工 学研究科 情報・通信工学 専攻 博士前期課程		
氏 名	高木 雄大	学籍番号	1531060
論 文 題 目	ランダムグラフ上の最長路問題に対する発見的アルゴリズム		
<p>要 旨</p> <p>本研究では最長路問題について扱う。最長路問題とは、入力グラフに対する部分グラフの中で辺数が最大のオイラーグラフまたは準オイラーグラフを求める問題である。</p> <p>ランダムグラフ上の最長路問題は高い辺密度においては高確率で最大流問題から最長路が誘導されることが示唆されている。しかし、辺密度の低いグラフにおいてはその確率が大きく減少していることも観察されている。そこで、本研究では最長路問題の計算複雑性が NP 困難であることを示し、その問題に対する最大流問題と強連結成分分解を用いた発見的アルゴリズムを提案する。提案アルゴリズムは、まず入力グラフの強連結成分分解を求める。次にそれによって得られた分割の順序関係の順に各強連結成分における最大流問題を解く。このとき、2 番目以降の強連結成分について考える場合、それより前の強連結成分における流れも用いることができる場合が存在することに注意して最大流問題を解く。例えば、2 番目の強連結成分の中に 1 番目の強連結成分のみを通して辿り着ける頂点が存在する場合、その流れが使えることを踏まえた上で最大流問題を解く。そして、最終的に得られた最大の流れを解として出力する。ただし、提案アルゴリズムは必ずしも最長路問題の最適解を出力するものではない。そこで本論文では、提案アルゴリズムが多項式時間で動作することを証明した後、最長路が得られる確率を実験により求めた。</p> <p>実験では、頂点数が 10, 20 の場合については辺密度を 0.01 から 0.50 まで 0.01 刻みで、頂点数 30, 40, 50 の場合については辺密度を 0.005 から 0.25 まで 0.005 刻みで、それぞれ 1000 個のランダムグラフをデータとして用意し、最大流問題を一度だけ解いて最長路が得られるデータの割合と提案アルゴリズムを適用した場合に最長路が得られるデータの割合をそれぞれ調べた。実験の結果、最大流問題を一度だけ解いた場合に最長路が得られた割合で最も低かった値はそれぞれ、頂点数が 10 のときは 0.484%、20 のときは 0.382%、30 のときは 0.271%、40 のときは 0.242%、50 のときは 0.218% という結果が得られた。それに対し、提案アルゴリズムを適用した場合に最長路が得られたデータの割合で最も低かったときの値はそれぞれ、頂点数が 10 のときは 0.987%、20 のときは 0.973%、30 のときは 0.958%、40 のときは 0.956%、50 のときは 0.938% という結果が得られた。これにより、提案アルゴリズムを適用する場合は最大流問題を一度だけ解く場合に対し、最長路が得られる確率が大きく改善することがわかった。</p> <p>以上より、本研究の提案アルゴリズムは NP 困難な最長路問題をランダムグラフ上においては多項式時間で高確率に解けるアルゴリズムであることが示唆された。</p>			

電気通信大学 情報理工学研究科 情報・通信工学専攻
2016年度 修士論文

ランダムグラフ上の最長路問題に対する
発見的アルゴリズム

平成 29 年 3 月 13 日

学籍番号 1531060

高木 雄大

指導教員 岡本 吉央

概要

ランダムグラフ上の最長路問題は高い辺密度においては高確率で最大流問題の解から最長路が誘導されることが示唆されている。しかし、辺密度の低いグラフにおいてはその確率は大きく減少していることも観察されている。そこで、本研究では最長路問題の計算複雑性がNP困難であることを示し、その問題に対する最大流問題と強連結成分分解を用いた発見的アルゴリズムを提案する。ただし、このアルゴリズムは必ずしも最長路問題の最適解を出力するものではない。そこで本論文では、提案アルゴリズムが多項式時間で動作することを証明した後、最長路が得られる確率を実験により求めた。実験の結果、提案アルゴリズムにより最長路を得られる確率は頂点数が10のときは0.484%以上、20のときは0.382%以上、30のときは0.271%以上、40のときは0.242%以上、50のときは0.218%以上という結果が得られた。これにより提案アルゴリズムは、NP困難な最長路問題をランダムグラフ上においては多項式時間で高確率に解けるアルゴリズムであると考えられる。

目次

1	はじめに	4
2	本研究に至った背景	5
2.1	最長しりとり問題に関する研究 [2]	5
2.1.1	最長しりとり問題の定義	5
2.1.2	最長しりとり問題のモデル化	6
2.1.3	最長しりとり問題の定式化	7
2.1.4	最長しりとり問題の実験	13
2.2	最長路問題の性質 [1]	15
3	最長路問題の計算複雑性の証明	17
3.1	最大オイラー部分グラフ問題	18
3.2	最長路問題への多項式時間還元	18
4	提案アルゴリズム	21
4.1	アルゴリズムの着想	24
4.2	アルゴリズムの概要	26
4.3	アルゴリズムの動作例	27
4.4	アルゴリズムの計算量	36
5	実験	38
5.1	ランダムグラフの作成	38
5.2	連結成分の探索	38
5.3	実験 1	39
5.3.1	実験結果	40
5.3.2	考察	41
5.4	実験 2	42
5.4.1	実験結果	43
5.4.2	考察	47
6	まとめ	48

1 はじめに

本研究では後述する最長路問題を扱う。この問題はランダムグラフ上で高い辺密度においては最大流問題に帰着できることが示唆されている。しかし、辺密度の低いグラフにおいては最長路が得られる確率が大きく減少していることが観察されている [1]。そこで本研究では、まず最長路問題の計算複雑性が NP 困難であることを示し、その問題に対する最大流問題と強連結成分分解を用いた発見的アルゴリズムを提案する。ただし、このアルゴリズムは必ずしも最長路問題の最適解を出力するものではない。そこで最長路を求められる確率を実験により求める。

最長路問題

入力: 有向グラフ $G(V, A)$ 。ただし、多重辺と自己ループを許す。

出力: グラフ $G'(V, A')$, $A' \subset A$ 。

制約:

- G' はオイラーグラフまたは準オイラーグラフである。
- 任意のオイラーグラフおよび準オイラーグラフ $G''(V, A'')$, $A'' \subseteq A$ に対し、 $|A'| \geq |A''|$ を満たす。

オイラーグラフとは、オイラー閉路を持つグラフのことである。ここで、オイラー閉路とはグラフ上の全ての辺をちょうど一度だけ通る閉路のことである。また、準オイラーグラフとは、グラフ上の全ての辺を通るようなオイラー閉路以外のオイラー路を持つグラフのことである。ここで、オイラー路とはグラフ上の全ての辺をちょうど一度だけ通る道のことである。

図1はオイラーグラフの例であり、図2は準オイラーグラフの例である。図1において $v_1, v_2, v_3, v_1, v_4, v_3, v_5, v_1$ という順で頂点を辿ると図1のすべての辺をちょうど一度ずつ通る。そのため、この有向道はオイラー路でありかつ始点と終点と同じであるため、これを特にオイラー閉路と言う。また図2において $v_1, v_2, v_3, v_1, v_4, v_3$ という順で頂点を辿ると図2のすべての辺をちょうど一度ずつ通る。そのため、この有向道はオイラー路である。しかし、始点は v_1 で終点は v_3 となっているため、これはオイラー閉路ではない。ゆえに図1のグラフはオイラーグラフであるのに対し、図2のグラフは準オイラーグラフとなる。

本論文では、まず第2章で本研究に至るきっかけとなった論文である「最長しりとり問題の解法 [2]」について紹介する。次に最長路問題の NP 困難であることを第3章で示した後、第4章で辺密度の低いグラフに対しても最長路が得られる確率が高くなるようなアルゴリズムの提案を行い、その計算量を示す。さらに提案アルゴ

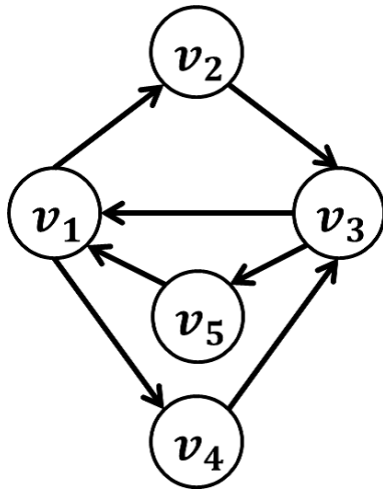


図 1: オイラーグラフ

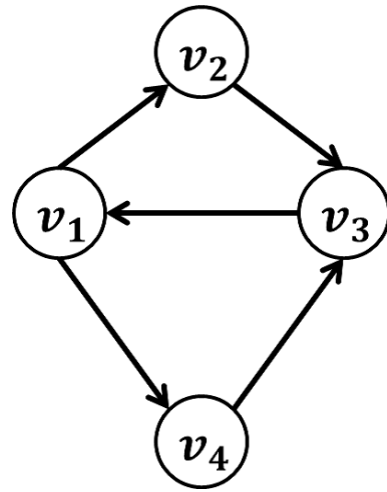


図 2: 準オイラーグラフ

リズムの最長路が得られる確率を検証した実験を第 5 章で示す。

2 本研究に至った背景

本章では、本論文に取り組むきっかけとなった研究 [1, 2] について、説明を行っていく。まず、最長路問題の研究に至る背景となった最長しりとり問題の研究 [2] について説明を行い、次に最長路問題の性質を調べた研究 [1] について説明する。

2.1 最長しりとり問題に関する研究 [2]

文献 [2] では最長しりとり問題をネットワーク表現によりモデル化し、最大流問題として解を求めている。

2.1.1 最長しりとり問題の定義

まず、最長しりとり問題は次のように定義されている。

最長しりとり問題

しりとりは、ある単語から始めて、その単語の終了文字が次の単語の開始文字と一致するように単語を提示していき、次に提示する単語がない場合に終了するゲームである。同じ単語が2度以上しりとりで現れてはいけない。単語はすべてひらがなで表記されているものとする。最長しりとり問題は、与えられた単語群(辞書)において、しりとりとなる最も単語数の多い単語列を作成する問題と定義する。つまり、最長しりとり問題における“長さ”は単語数として定義する。

2.1.2 最長しりとり問題のモデル化

次に、この最長しりとり問題のモデル化を行っていく。与えられた辞書における単語間の関係を有向グラフとして表現し、1つのしりとりをオイラー路(すべての有向辺をちょうど1回だけ使う路)に対応付け、辞書から構成される有向グラフにおける最大準オイラー部分グラフ(準オイラー部分グラフは、あるグラフ G の部分グラフの中で、辺がオイラー路になっているものである)を求める問題としてモデル化する。また、その有向グラフの頂点間の多重有向辺を取り除き、多重有向辺の数を有向辺の容量としたネットワーク上でのネットワークフロー問題としてのモデル化も示す。

しりとりでは、開始文字または終了文字になる文字だけが重要であるので、開始文字または終了文字となる文字の集合を V とおく。ここでは、文字を数字に対応付けて表現し、文字数 n の集合を $V = \{1, 2, \dots, n\}$ と表す。有向辺の集合 A は単語の集合で、 $A = B_{11} \cup B_{12} \cup \dots \cup B_{nn}$ と表される。ただし、 B_{ij} は文字 i で始まり、文字 j で終わるすべての単語の集合である。つまり、 B_{ij} のすべての要素は $i \in V$ から $j \in V$ への有向辺となる。以上によって表されるグラフでは、オイラー路を持つ部分グラフがしりとりを表現する。よって、このグラフにおいて最長しりとり問題は次のようにモデル化される。

最長しりとり問題のグラフ表現によるモデル

有向グラフ $G = (V, A)$ の準オイラー部分グラフの中で有向辺の数が最大となるものを見つける問題。

さらに、しり通りの長さに着目する際には、単語の違いを考慮しなくても良いので、前述の集合 B_{ij} で必要な情報はサイズのみである。このため、文字 i から j へのすべての有向辺を1つの有向辺 a_{ij} で置き換えた集合 $A' = \{a_{11}, a_{12}, \dots, a_{nn}\}$ と、単語数 $f_{ij} = |B_{ij}|$ の集合 $F = \{f_{11}, f_{12}, \dots, f_{nn}\}$ を用いたネットワークを考える。ここ

で、 f_{ij} は有向辺 a_{ij} の容量と考えると、例えば頂点 i から頂点 j への有向辺の変換は図3のようになる。

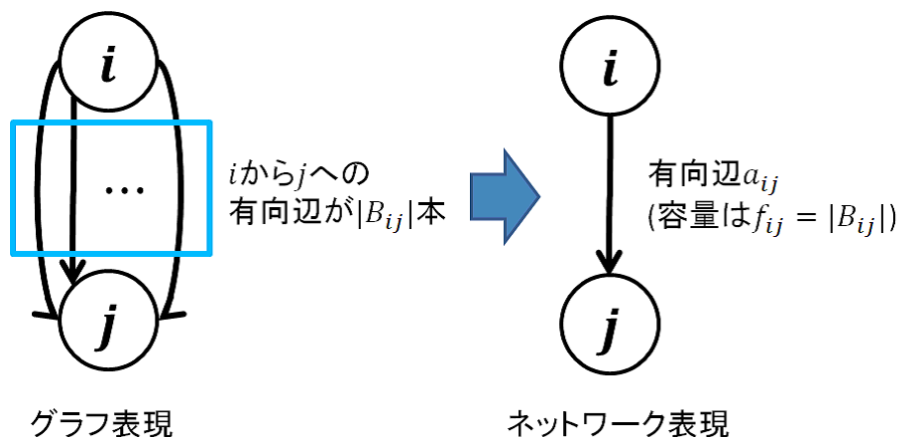


図 3: グラフ表現からネットワーク表現への変換

このネットワークにおいて、最長しりとり問題は次のようにモデル化される。

最長しりとり問題のネットワーク表現によるモデル

ネットワーク $N = (G = (V, A'), F)$ において、任意の頂点を始点と終点にした場合に構成されるフローネットワークの中で、各有向辺に流れる流量の和を最大にする問題。

フローネットワークや流れは次のように定義する。

「フローネットワーク」とは、グラフ理論における有向グラフの一種である。フローネットワークにおける各有向辺のことを始点から終点の方向への「流れ」と言い、そのときの流れの量のことを「流量」と呼ぶ。またフローネットワークを構成する際には元となるネットワークが存在し、そのネットワークの各有向辺には容量が設定されている。流れが満足すべき制約条件は、まずフローネットワークの各有向辺の流量はその容量を超えないこと。さらに、始点 (source) と終点 (sink) を除く任意の頂点 v において v に流入する流量の和と v から流出する流量の和が常に等しいことである。

2.1.3 最長しりとり問題の定式化

任意の頂点を始点と終点にした場合に構成されるフローネットワークを考える代わりに、補助ネットワーク N' 上での s を始点、 t を終点とした場合のフローネット

ワークを考える。補助ネットワーク N' とは、ネットワーク N に対してスーパーソース s とスーパーシンク t を付加したもので、頂点 s から N のすべての頂点へ、 N のすべての頂点から頂点 t へ、容量 1 の有向辺が存在する。この操作の例が図 4 である。図 4 の左のグラフに s と t を加えると図 4 の右のようなグラフが得られる。

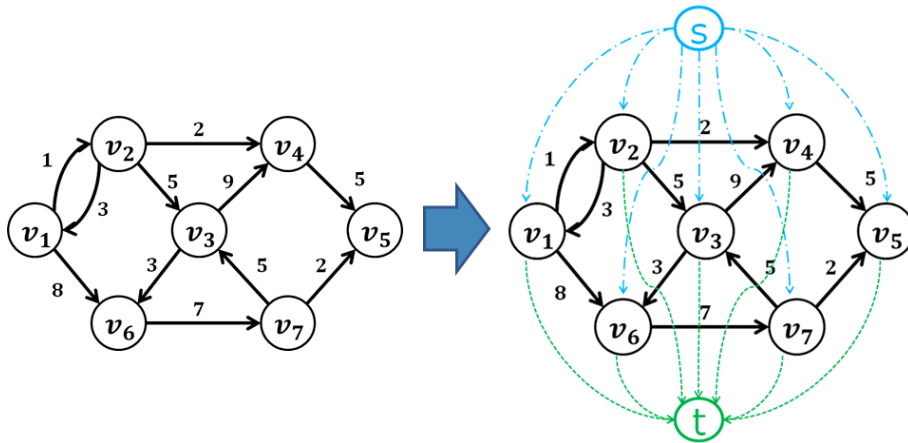


図 4: s と t を加える操作の例

このネットワークを用いて、最長しりとり問題を整数計画問題として解くために必要な条件は、大きく分けて次の 2 つからなる。

- (1) 頂点 s, t を除き、各頂点に入るフローと出るフローが等しい。頂点 s では出ていくフローが 1 だけ多く、頂点 t では入ってくるフローが 1 だけ多い。
- (2) フローの値が正である有向辺により誘導される部分グラフが連結である。

この条件を式で記述したとき、文字の種類 n に対し、(1) の条件は $n + 2$ 個の制約式であるが、(2) の条件については $O(2^n)$ 個の制約式が必要になる。

前述の定式化では、小規模なデータしか扱うことができない。そこで、緩和問題を繰り返し解くことにより、最長しりとり問題を解決する。まず、フローネットワークとしての条件 (上記の (1)) だけを考慮し、各有向辺を流れるフローの総和を最大化する緩和問題 (RP_0) (この問題は始点から流れるフローを 1 に制限した上で全ての有向辺に流れるフローを最大化する最大流問題と等しい) を考える。 x_{ij} は辺 (i, j) を流れる流量を表す変数である。

$$\begin{aligned}
(RP_0) \text{ 最大化} \quad z_0 &= \sum_{i \in V \cup \{s\}, j \in V \cup \{t\}} x_{ij} \\
\text{条件} \quad \sum_{j \in V} x_{sj} &= 1, \\
\sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} &= 0 \quad \forall i \in V, \\
\sum_{j \in V} x_{jt} &= 1, \\
0 \leq x_{ij} &\leq f_{ij}, \quad \forall i \in V, \forall j \in V, \\
0 \leq x_{sj} &\leq 1, \quad \forall j \in V, \\
0 \leq x_{jt} &\leq 1, \quad \forall j \in V.
\end{aligned}$$

問題 (RP_0) は、最長しりとり問題の緩和問題となっているので、その最適値 z_0 は最長しりとり問題の最適値の上界値を与える。 (RP_0) は、よく知られた整数定理 [3] の成り立つ制約条件を持ち、各 f_{ij} は整数なので、 (RP_0) の最適解も整数解として求まる。よって、仮に、 (RP_0) の解によって誘導されるグラフが連結であるならば、最長しりとり問題は解けたことになる。

それに対し、 (RP_0) の解によって誘導されるグラフが、複数の連結成分に分かれた場合は、分枝操作を行うことによって解を求める。 (RP_0) の制約条件より頂点 s と t をともに含む連結成分が必ず存在する。そこで、 s, t を含む連結成分に含まれる頂点の集合を V_0^* とし、最長しりとり問題の許容解集合を 2 つに分割する以下の問題を考える。

- (1) 頂点 $i \in V_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移が少なくとも一つはある最長しりとり問題。
- (2) 頂点 $i \in V_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移が一つもない最長しりとり問題。

この 2 つの問題の最長しりとり問題の内、求まる最適値が大きいほうの最適解が、元の最長しりとり問題の最適解となる。

(2) は (RP_0) の頂点集合 $V \cup \{s, t\}$ を V_0^* に制限したものに相当する。よって、新たに問題を解くことなく、 (RP_0) の解から x_{ij} ($\forall i \in V_0^*, \forall j \in V_0^*$) を取り出せば (2) の最適解が構成可能である。このとき目的関数値 z'_0 は

$$z'_0 = \sum_{i \in V_0^*, j \in V_0^*} x_{ij}$$

として求まる (問題 (RP_k) に対しては $z'_k = \sum_{i \in V_k^*, j \in V_k^*} x_{ij}$)。 z'_0 は最長しりとり問題の最適値の下界値を与える。

(1) に関しては、「頂点 $i \in V_0^*$ から頂点 $j \in V \setminus V_0^*$ への単語の遷移が少なくとも一つはある」という条件

$$\sum_{i \in V_0^*, j \in V \setminus V_0^*} x_{ij} \geq 1$$

を (RP_0) に加えて得られる問題 (RP_1) を考える。 (RP_1) を解き、同様の分枝操作を行う。以上を再帰的に繰り返し、アルゴリズムを構成する。 k 回目の再帰で解く問題 (RP_k) は次のようになる

$$\begin{aligned} (RP_k) \text{ 最大化} \quad z_k &= \sum_{i \in V \cup \{s\}, j \in V \cup \{t\}} x_{ij} \\ \text{条件} \quad \sum_{j \in V} x_{sj} &= 1, \\ \sum_{j \in V} x_{ij} - \sum_{j \in V} x_{ji} &= 0 \quad \forall i \in V, \\ \sum_{j \in V} x_{jt} &= 1, \\ \sum_{i \in V_l^*, j \in V \setminus V_l^*} x_{ij} &\geq 1, \quad l = 0, 1, \dots, k-1, \\ 0 \leq x_{ij} &\leq f_{ij}, \quad \forall i \in V, \forall j \in V, \\ 0 \leq x_{sj} &\leq 1, \quad \forall j \in V, \\ 0 \leq x_{jt} &\leq 1, \quad \forall j \in V, \\ x_{ij} \in \mathbb{Z}, \quad &\forall i \in V \cup \{s\}, \forall j \in V \cup \{t\}. \end{aligned}$$

V_l^* は、 (RP_l) の解における s, t を含む連結成分の頂点集合である。 (RP_k) においては、整数条件を取り除くと、最適基底解の整数性は保証されない。よって線形計画問題として定式化できないため、この方法では多項式時間で解けるとは限らない。

以上の分枝限定法を行うアルゴリズムの動作を図5から図8を用いて説明する。

図1は (RP_0) を解いても最長しりとり問題の最適解が得られないグラフの例である。なお図中の有向辺の容量はすべて1とし、 s, t に隣接する頂点は v_1, \dots, v_9 であるため、 s, t に接続する有向辺の本数が多い。そのため図中に記載するとグラフが見にくくなってしまったため、図5に限り s, t に接続する有向辺は省略する。

まず、 (RP_0) の最適解によって誘導されるグラフは図6の実線矢印となる (点線部分は元のグラフには存在するが、最適解には存在しない有向辺である)。ここで、

誘導されたグラフが連結ではないので、分枝操作が行われる。このとき、 $V_0^* = \{s, v_1, v_2, v_3, v_4, v_5, t\}$ となる。

次に、図7の薄い線の矢印のどれかが結ばれる条件を (RP_0) に加え、新たな問題 (RP_1) とする。

最後に (RP_1) の最適解によって誘導されるグラフが図8の実線矢印となる。このグラフは連結であるので、最長しりとり問題の解が得られたことになる。

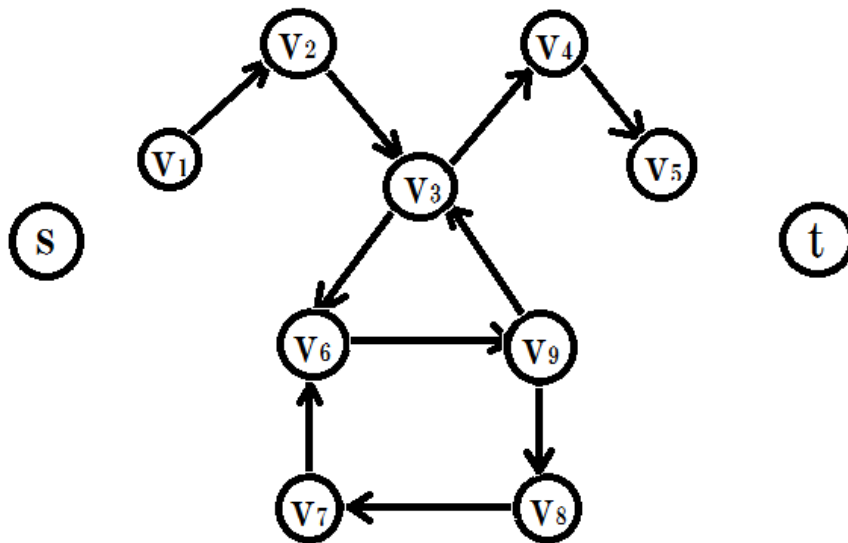


図 5: (RP_0) で終わらないグラフの例

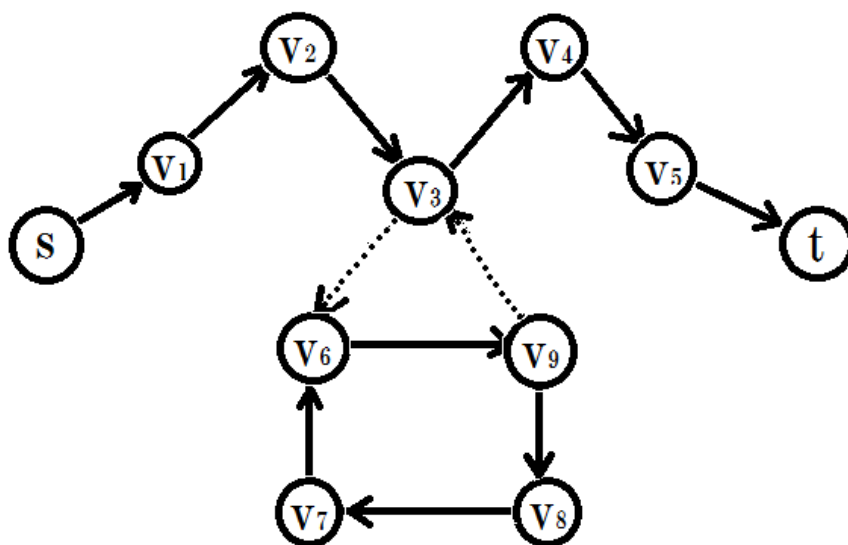


図 6: (RP_0) の解

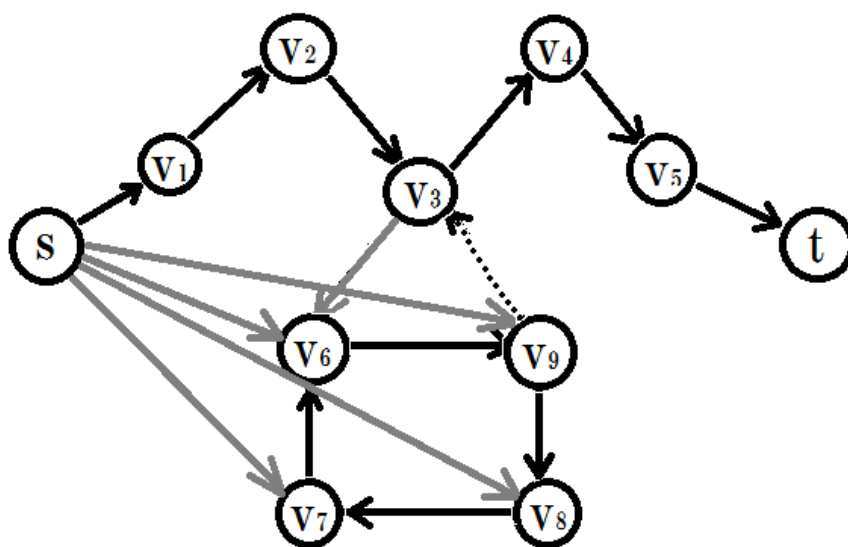


図 7: (RP_1) に加えられる条件

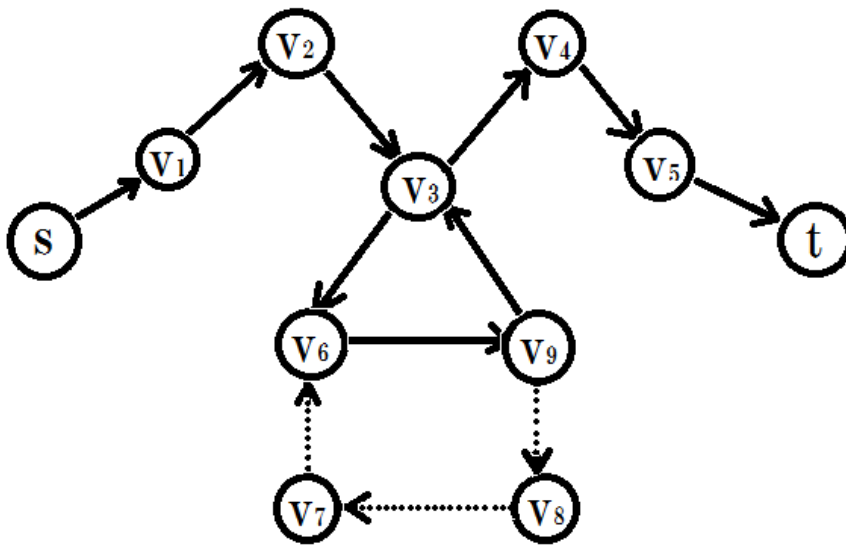


図 8: 最適解

2.1.4 最長しりとり問題の実験

文献 [2] において、実験は数種の実際に存在する辞書に対して行われているが、ここでは、広辞苑を用いた実験のみを紹介する。広辞苑を用いたデータとしては、見出し語が清音で始まる単語を取り出し、見出し語の重複を省いたデータ「任意の品詞」と名詞である見出し語すべてを使ったデータ「名詞だけ」の2つを用いている。

実験環境は、実装に Windows XP 上の Cygwin (ver.2.218) で gcc (ver.3.2) を用い、整数計画法のソルバとして、GNU GLPK (ver.4.2) を使用している。実験に使用した PC は、Xeon 2.8GHZ、Dual Processor、2GB Memory である。

表 1: データの特徴 [2]

	任意の品詞	名詞だけ
単語数	137335	192687
文字種類数	70	70
開始文字種類数 (S)	44	70
終了文字種類数 (T)	70	70
有向辺数 (A)	2876	4205
$\frac{A}{ST}$	93%	86%
有向辺数に対する最長しりとり長 (L)	1844	3907
$\frac{L}{A}$	64%	92%

まず、広辞苑の見出し語より取り出した単語のデータの特徴を表1に示す。表1において、「有向辺数」は頻度が1以上であった有向辺数を表す。 $\frac{A}{ST}$ は単語を割り付けることが可能な有向辺数に対する実際に存在した有向辺の割合を示し、共に高い割合となっている。「有向辺数に対する最長しりとり長」は、有向辺の頻度を1(開始文字と終了文字のペアが同じ単語の使用を一度のみに制限した場合)の最長しりとりの長さを表す。

表 2: 文字頻度による最長しりとり長(任意の品詞)[2]

単語数 (W)	最長の長さ (L)	割合 (L/W)	IP 適用回数
137335	56519	41%	1
68417	27718	41%	1
33497	13339	40%	1
16079	6183	38%	1
7406	2654	36%	1
3219	1016	32%	1
1265	303	24%	1
444	70	16%	1
124	15	12%	1

表 3: 文字頻度による最長しりとり長 (名詞のみ)[2]

単語数 (W)	最長の長さ (L)	割合 (L/W)	IP 適用回数
192687	86788	45%	2
95255	42236	44%	1
46595	20077	43%	1
22351	9115	41%	3
10361	3792	37%	1
4535	1372	30%	1
1853	401	22%	1
697	91	13%	1
231	18	8%	1

広辞苑を用いた実験結果を表2と表3に示す。表2と表3の2行目以降は、それぞれ一つ前の行の辞書に対するグラフの有向辺の容量 (開始文字と終了文字のペアが同じ単語の数) を半分 (ただし小数点以下切り捨て) にすることで辞書を構成し、それに対する最長しりとり問題を解いた場合の実験結果を示している。割合は最長しりとりを構成する単語数の総単語数に対する割合を表す。IP 適用回数は、前述した分枝操作が行われた回数を示す。例えば、IP 適用回数が1の場合は、 (RP_0) で連結したグラフが得られたことを示している。

表2と表3より、ほとんどの場合、緩和問題 (RP_0) を解いただけで、つまり最大流問題を解くだけで最長しりとりの解が求められている分かる。よって、実在する辞書で最長しりとりを求める問題には、一般の最長路問題とは異なり最大流問題を解くだけで最適解が得られやすい特徴があると考えられる。

そこで結果および辞書のデータより、最長路問題は辺密度が高いグラフにおいて最大流問題に帰着できる特徴があると考え、そのことを実験により確かめた [1]。次にその実験の概要と結果を示す。

2.2 最長路問題の性質 [1]

最長路問題の性質として、高い辺密度において最大流問題の解から最長路が得られやすいという性質があると考えられるため、様々な辺密度のグラフについて最大流問題を解き、その解が最長路となっているかどうかの確認を行う。それにより、最長路問題が最大流問題として帰着できるかどうかを調べた。ここで、最大流問題の解が最長路となっているかどうかは得られた解が連結であるかどうかを調べるこ

によって判定することができる。つまり、最大流問題の解が連結であれば、それは最長路になる。

ここで、最大流問題の解が非連結になるグラフを例に最長路にならないことの説明を行う。図9のグラフについて最大流問題を解くと図10のようになる。図10のグラフの辺における $\frac{1}{3}$ などの表記法は分母が入力グラフの容量を表しており、分子が最大流問題の最適解におけるその辺の流量を表している。また、頂点 v_6 に入ってくる矢印と v_5 から出て行く矢印に点線のものが存在するが、これは最大流問題の解において、頂点 s から次に流れる頂点は v_6 であること、また頂点 t へ流れるのは頂点 v_5 からであることを表している。このとき図10において、始点 s から終点 t への道となっている連結成分と v_1 と v_2 の閉路により構成されている連結成分の二つに分かれている。つまり、 s から v_1, v_2 への道は存在しないため、解における辺 $(v_1, v_2), (v_2, v_1)$ は道として用いることができない。よって、これは最長路とはなっていない。そのため図9のグラフは最大流問題を解くことによって最長路は得られないグラフということになる。

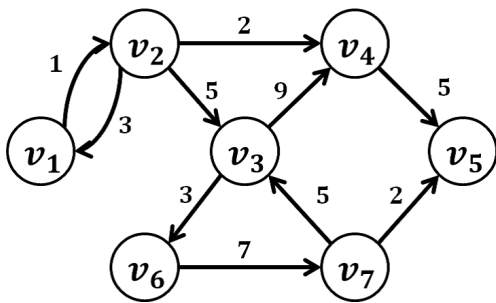


図 9: 入力グラフ

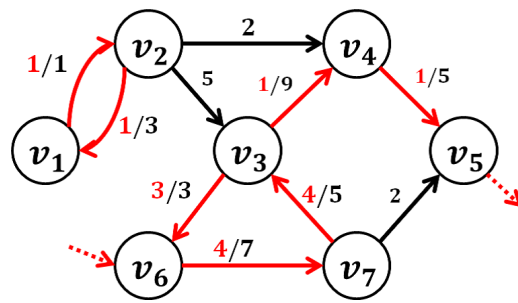


図 10: 最大流問題の解

次に実験の説明を行う。実験内容として、まずランダムグラフを作成する。このとき、作成されたグラフは必ずしも連結であるとは限らない。次にそのグラフについて最大流問題を解く。最後にその解が連結であるかどうかを調べ、連結なグラフが得られた場合は最長路が得られたことになり、実験ではその数と辺密度の関係をまとめた。

実験データとして頂点数を10, 辺密度を0.01から0.60の値を0.01刻みとして、それぞれ2000個ずつ用意し、これらのデータに対しそれぞれ最大流問題を解いた。

図11が頂点数10, データ数2000の実験結果である。横軸が辺密度、縦軸が2000個のデータに対し最大流問題を解いた結果、最長路が得られたデータの数である。これにより、高い辺密度では最長路問題は最大流問題に帰着できると考えられるが、低い辺密度ではその限りではない。

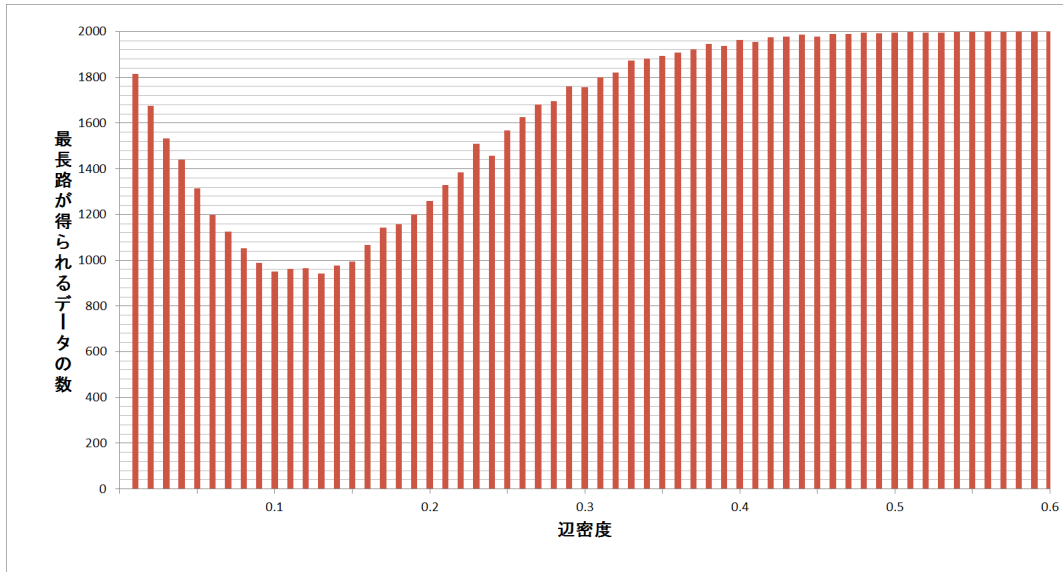


図 11: 頂点数 10 の最長路が得られる確率

そこで本研究では、最長路が得られる確率を改善する発見的アルゴリズムの提案を行う。そのアルゴリズムを説明するにあたり、まずはアルゴリズムの有用性を示すために最長路問題の計算複雑性の証明を行う。

3 最長路問題の計算複雑性の証明

最長路問題の計算複雑性の話に入る前に一般的な最長路問題である辺重みを最大にする最高路問題と本研究で扱う最長路問題の違いについて話す。

辺重みを最大にする最長路問題は次のような2つの場合について考えられる。

1. 入力グラフが辺重み和が正になる閉路を含む場合、その閉路を使えば使うほど目的関数値を上げることができるため最適値は無限に大きな値になる。
2. 1.のような閉路を含まない場合、最適解は必ず閉路を含まない道、つまりハミルトン路になる。そのため、辺重みを最大にする最長路問題は辺重み最大ハミルトン路問題と等価になる。

それに対し、本研究で扱う最長路問題は入力グラフにおける最大のオイラー路を求める問題である。ここで、本研究で扱う最長路問題において入力グラフの辺重み

は全て1であるとみなせるが、各辺が使用できるのは一度ずつであるため最適値を無限に大きくすることはできない。よって、辺重みを最大にする最長路問題とは異なり解に閉路が含まれる場合も存在する。

以上より、これら二つの問題は異なる問題となる。

次に本研究における最長路問題の計算複雑性を証明するために、最大オイラー部分グラフ問題の定義を行う。この問題がNP困難 [4, 5] であり、最長路問題へ多項式時間還元できることから最長路問題がNP困難であることが導かれる。

3.1 最大オイラー部分グラフ問題

最大オイラー部分グラフ問題の定義は次のようになる。

最大オイラー部分グラフ問題

入力: 有向グラフ $G(V, A)$

出力: $G'(V, A'), A' \subseteq A$

制約:

- G' はオイラーグラフである。
- 任意のオイラーグラフ $G''(V, A'')(A'' \subseteq A)$ に対し、 $|A'| \geq |A''|$ を満たす。

3.2 最長路問題への多項式時間還元

最大オイラー部分グラフ問題のインスタンスとして与えられるグラフ $G(V, A)$ に対し、次のような操作を行う。ただし、 $V = \{v_1, v_2, \dots, v_{|V|}\}$ とする。

1. G を $|V|$ 個並べ、以降 i 番目の G を G_i と書くこととする。
2. 全ての $i \leq |V| - 1$ について、 G_i における v_i を始点、 G_{i+1} における v_{i+1} を終点とした有向辺を追加する。 G_i と G_{i+1} はこの有向辺でのみ連結している。
3. 頂点 s, t を追加する。
4. 頂点を $|A|$ 個、辺を $|A| + 1$ 本追加することで、 s から G_1 における v_1 への有向道を作成する。
5. 頂点を $|A|$ 個、辺を $|A| + 1$ 本追加することで、 $G_{|V|}$ における $v_{|V|}$ から t への有向道を作成する。

この操作を行うと図 12 のようにグラフが得られる。この操作によって得られたグラフを G_l と書くことにする。

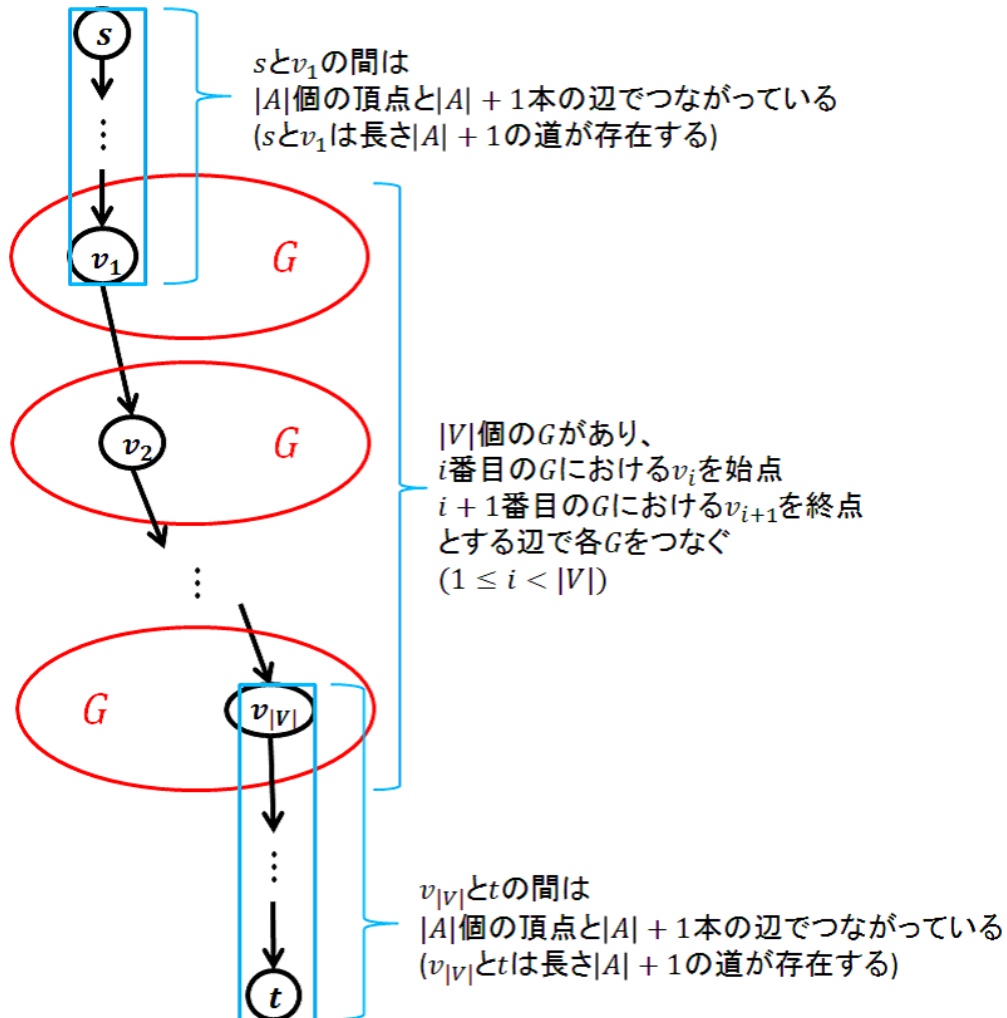


図 12: 多項式時間還元

この操作は $|A|$ に関する多項式回のステップで完了するものである。また、 G_l は G が $|V|$ 個、各 G をつなぐ辺が $|V| - 1$ 本、 s および t と G をつなぐ道を加えることによって増加する頂点数と辺数は $2(|A| + 1)$ である。よって、 G_l は頂点数が $|V|^2 + 2|A| + 2$ 、辺数が $|V| \cdot |A| + 2|A| + |V| + 1$ となり、 G_l の頂点数および辺数は $|V|$ と $|A|$ に関する多項式になる。

次に G_l に対し最長路問題を解くことで最大オイラー部分グラフ問題の最適解を得る方法を示す。まず、 G_l 上で最長路問題を解く。その結果得られた最適解は必ず始

点が s 、終点が t となっており、全ての G_i を通る。また、 G_l における G_i の頂点集合 V_i に対し、最適解における V_i が誘導する部分グラフを考えると、それは必ず v_i を通る G のオイラー部分グラフの中で最大のものとなっている。よって、全ての V_i が誘導する部分グラフの辺数を比較し、その中で最大のものを選べば、それは G における最大オイラー部分グラフ問題の最適解となっている。

以上より、最大オイラー部分グラフ問題は最長路問題へ多項式時間還元可能であるため本研究における最長路問題は NP 困難な問題である。

そこで本研究では最長路問題に対し、計算時間が多項式の発見的アルゴリズムを提案する。

4 提案アルゴリズム

最長路問題に対する発見的アルゴリズムを提案する。最長路問題のアルゴリズムの概要を示す前に、アルゴリズム内で扱う事柄について説明を行う。

まず、[2]における最大流問題を元にした頂点重み付き最大流問題と終点固定の頂点重み付き最大流問題という二つの問題を本アルゴリズムで扱うので、その定義を行う。続いて有向グラフにおける概念の一つである強連結成分分解について説明する。

最大流問題の定義は次のようになる。

最大流問題

入力: 有向グラフ $G(V, A)$ 、各辺 $a \in A$ の容量 $c(a) \in \mathbb{N}$ 、
各頂点 $v \in V$ の重み $w(v) \in \mathbb{N}$ 。

出力: フローネットワーク $F = \{f(a) \in \mathbb{Z} \mid a \in A'\}$ 。

制約: G に始点 s と終点 t を加えたグラフ $G'(V', A')$ について考える。

$(V' = V \cup \{s, t\}, A' = A \cup \{(s, v), (v, t) \mid v \in V\})$ 、 $\forall v, c((s, v)) = c((v, t)) = 1$

1. 各辺を流れる流量は0以上、容量以下である。 $(\forall a \in A', 0 \leq f(a) \leq c(a))$
2. 始点 s においては流出量の総和が1である。 $(\sum_{v \in V} f((s, v)) = 1)$
3. 終点 t においては流入量の総和が1である。 $(\sum_{v \in V} f((v, t)) = 1)$
4. 始点、終点以外の任意の頂点では流入量と流出量の総和が等しくなる。
 $(\forall u \in V, \sum_{v \in V \cup \{t\}, (u, v) \in A'} f((u, v)) - \sum_{v \in V \cup \{s\}, (v, u) \in A'} f((v, u)) = 0)$
5. 1 から 4 を満たす任意のフローネットワーク $F' = \{f'(a) \in \mathbb{Z} \mid a \in A'\}$ に対し、 F の流量の総和は F' の流量の総和以上である。このとき s の次に流れる頂点が v となる場合、 $w(v)$ だけ流量の総和が増加するものとする。
 $(\sum_{v \in V} w(v)f((s, v)) + \sum_{a \in A} f(a) \geq \sum_{v \in V} w(v)f'((s, v)) + \sum_{a \in A} f'(a))$

最大流問題の定式化は次になる。 x_{ij} は辺 (i, j) を流れる流量を表す変数とする。

$$\begin{aligned}
 \text{最大化} \quad z &= \sum_{j \in V} w(j) \cdot x_{sj} + \sum_{\substack{i \in V, j \in V \cup \{t\}, \\ (i,j) \in A'}} x_{ij} \\
 \text{制約} \quad \sum_{j \in V} x_{sj} &= 1, \\
 \sum_{\substack{j \in V \cup \{t\}, \\ (i,j) \in A'}} x_{ij} - \sum_{\substack{j \in V \cup \{s\}, \\ (j,i) \in A'}} x_{ji} &= 0, \quad \forall i \in V, \\
 \sum_{i \in V} x_{it} &= 1, \\
 0 \leq x_{ij} &\leq c((i, j)), \quad \forall (i, j) \in A, \\
 0 \leq x_{sj} &\leq 1, \quad \forall j \in V, \\
 0 \leq x_{it} &\leq 1, \quad \forall i \in V.
 \end{aligned}$$

次に終点を固定した場合の最大流問題を定義する。この問題は前述した最大流問題において入力グラフの頂点集合 V の中から終点が選ばれ、その代わりに加えられる頂点に t が存在しないような問題になる。

終点固定の最大流問題

入力: 有向グラフ $G(V, A)$ 、各辺 $a \in A$ の容量 $c(a) \in \mathbb{N}$ 、
各頂点 $v \in V$ の重み $w(v) \in \mathbb{N}$ 、終点 $t \in V$ 。

出力: フローネットワーク $F = \{f(a) \in \mathbb{Z} \mid a \in A'\}$ 。

制約: G に始点 s を加えたグラフ $G'(V', A')$ について考える。

$(V' = V \cup \{s\}, A' = A \cup \{(s, v) \mid v \in V\})$ 、 $\forall v, c((s, v)) = 1$

1. 各辺を流れる流量は 0 以上、容量以下である。 $(\forall a \in A', 0 \leq f(a) \leq c(a))$
2. 始点 s においては流出量の総和が 1 である。 $(\sum_{v \in V} f((s, v)) = 1)$
3. 終点 t において流入量の総和は流出量の総和より 1 多い。
 $(\sum_{v \in V \cup \{s\}, (v, t) \in A'} f((v, t)) - \sum_{v \in V, (t, v) \in A'} f((t, v)) = 1)$
4. 始点、終点以外の任意の頂点では流入量と流出量の総和が等しくなる。
 $(\forall u \in (V - \{t\}), \sum_{v \in V, (u, v) \in A'} f((u, v)) - \sum_{v \in V \cup \{s\}, (v, u) \in A'} f((v, u)) = 0)$
5. 1 から 4 を満たす任意のフローネットワーク $F' = \{f'(a) \in \mathbb{Z} \mid a \in A'\}$ に対し、 F の流量の総和は F' の流量の総和以上である。このとき s の次に流れる頂点が v となる場合、 $w(v)$ だけ流量の総和が増加するものとする。
 $(\sum_{v \in V} w(v)f((s, v)) + \sum_{a \in A} f(a) \geq \sum_{v \in V} w(v)f'((s, v)) + \sum_{a \in A} f'(a))$

終点固定の最大流問題の定式化は次になる。

$$\begin{aligned}
 \text{最大化} \quad z_t &= \sum_{j \in V} w(j) \cdot x_{sj} + \sum_{\substack{i \in V, j \in V \cup \{t\}, \\ i, j \in A'}} x_{ij} \\
 \text{制約} \quad \sum_{j \in V} x_{sj} &= 1, \\
 \sum_{\substack{j \in V \\ (i, j) \in A'}} x_{ij} - \sum_{\substack{j \in V \cup \{s\}, \\ (j, i) \in A'}} x_{ji} &= 0, \quad \forall i \in (V - \{t\}), \\
 \sum_{\substack{i \in V \cup \{s\}, \\ (i, t) \in A'}} x_{it} - \sum_{\substack{j \in V, \\ (t, j) \in A'}} x_{tj} &= 1, \\
 0 \leq x_{ij} &\leq c((i, j)), \quad \forall (i, j) \in A, \\
 0 \leq x_{sj} &\leq 1, \quad \forall j \in V.
 \end{aligned}$$

続いて強連結成分分解の定義を行う。

強連結成分分解

グラフ $G(V, A)$ の強連結成分分解とは、以下を満たす頂点集合 V の分割 V_1, V_2, \dots, V_k である。分割とは、 $V_1 \cup V_2 \cup \dots \cup V_k = V$ かつ $\forall i, j, V_i \cap V_j = \emptyset$ を満たす V_1, V_2, \dots, V_k のことである。各 V_i (ただし $1 \leq i \leq k$) のことを強連結成分と言う。

- 全ての V_i において、任意の 2 頂点 $v, u \in V_i$ に対し v から u への有向道および u から v への有向道が G に存在する。
- 上の条件を満たす 2 頂点は必ず同じ集合 V_i に属する。

図 13 は強連結成分の例である。図における左のグラフが入力グラフ、右のグラフが入力グラフに対する強連結成分分解を求めた結果であり、 V_1, V_2, V_3, V_4 はそれによって得られた分割である。

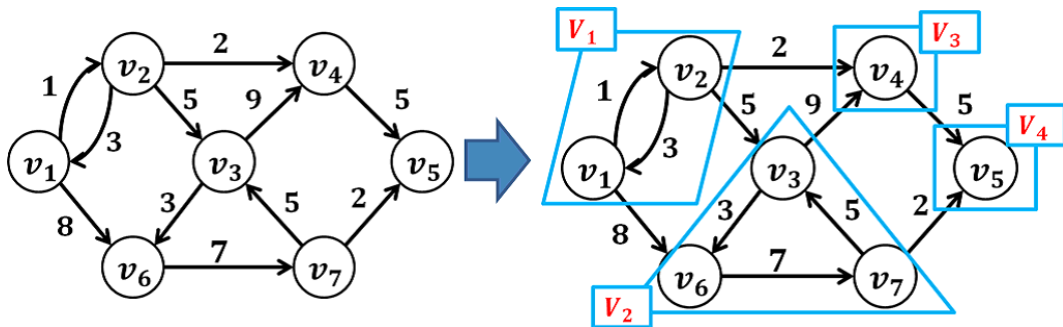


図 13: 強連結成分分解

以上がアルゴリズム中で用いるものの定義である。

次にこれらを用いて最長路問題に対する発見的アルゴリズムを設計するに至った背景を示し、その概要およびアルゴリズムの計算量を示す。

4.1 アルゴリズムの着想

まず最大流問題を解いた結果、最長路が得られなかった場合のグラフについて、そのときの入力グラフとそのグラフに対する最大流問題の最適解を観察する。図 14 は最大流問題の最適解が最長路にならないグラフの例である。各辺の値は容量を表す。

図 14.2 において v_6 が孤立点となっておりかつ自己ループを持っている。つまり v_{10} から v_1 へ至る道と v_6 が持つ自己ループは非連結になっているため 16 ページで

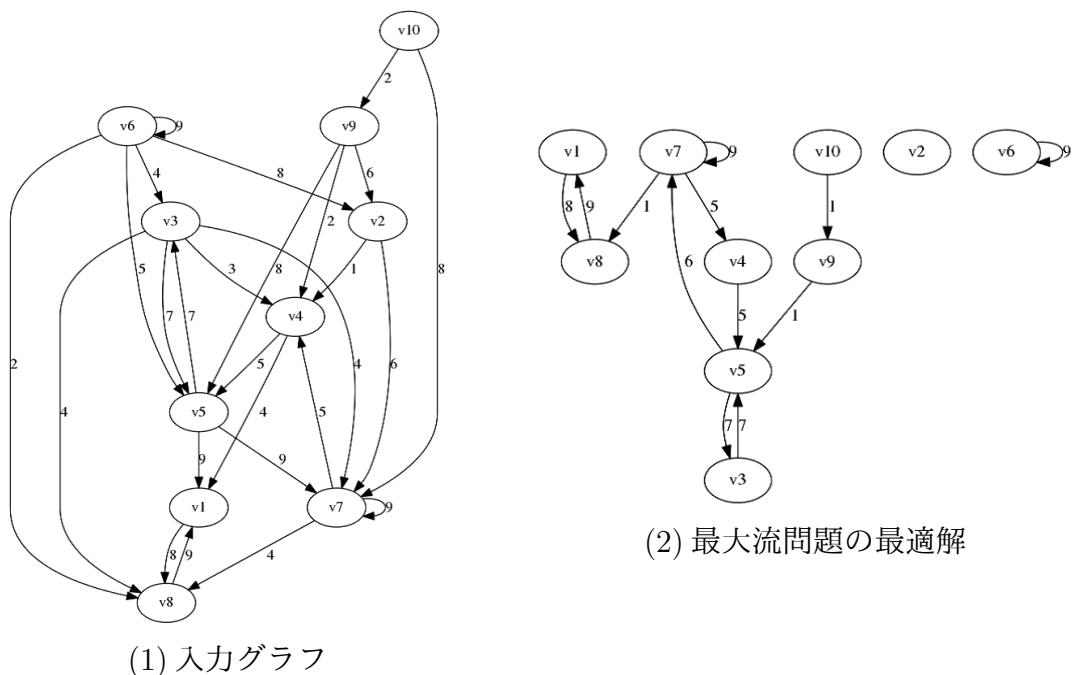


図 14: 最長路が得られないグラフの例

説明したようにグラフ中の全ての辺を通る道は存在しない。よって、この例の最大流問題の最適解は最長路となっていない。そこで最大流問題の最適解において孤立点になってしまう原因を考えるため、入力グラフにおける強連結成分分解を求め観察した。図 14.1 における強連結成分分解を求めると

$$\{v_6\}, \{v_{10}\}, \{v_9\}, \{v_2\}, \{v_3, v_4, v_5, v_7\}, \{v_1, v_8\}$$

が得られる。また図 14.2 において孤立点となっているのは v_2 と v_6 である。そして図 14.1 において v_2 と v_6 はそれぞれその頂点のみの強連結成分である。これらを見比べると、最大流問題の最適解において孤立点になるのは入力グラフにおいて小さい強連結成分に含まれていることが原因の一つであると考えられる。そこで強連結成分分解で得られた分割 V_1, V_2, \dots, V_k について、最大流問題は各 V_i が誘導する部分グラフ上でのみ解き、 V_i における最適解を用いる場合は必ず s, t 道に含まれるようにアルゴリズムを設計することで最長路が得られる確率が上がるのではないかと考えた。

4.2 アルゴリズムの概要

$G(V, A)$ は入力として与えられた有向グラフとする。ただし多重辺、自己ループを許す。また、アルゴリズム内で用いる変数を次のように定義する。

- $w(v)$ を頂点 $v \in V$ の重みとする。
- $c(a)$ を辺 $a \in A$ の容量とする。
- $L_s[v]$ を頂点集合 V_i まで考えたとき V_1, \dots, V_i を用いて $v \in V_{i+1} \cup \dots \cup V_k$ へ至る最長路に用いる辺集合とする。
- L を暫定的な最長路に用いる辺集合とする。
- l を L に保存されている道の長さとする。

Step 1

G における V の強連結成分分解 V_1, \dots, V_k を求める。このとき順序関係

$$V_i \succeq V_j \Leftrightarrow \forall u \in V_i \text{ から } \forall v \in V_j \text{ への有向道が存在}$$

が得られる。ただし、 $V_i \succeq V_j \Rightarrow i \leq j$ を満たすものとする。

Step 2

$i := 1, L := \emptyset, l := 0$ と初期化する。さらに全ての頂点 $v \in V$ について、 $w(v) := 1, L_s[v] := \{(s, v)\}$ と初期化する。

Step 3

V_i が誘導する部分グラフ上で最大流問題を解く。

ただし、 $\forall v, u \in V, c((v, u)) = (v \text{ から } u \text{ への有向辺の本数})$ とする。このとき、得られた最適値 z^* 、最適解 L^* に対し、 $l < z^*$ が成り立つとき

$$\begin{aligned} l &:= z^*, \\ L &:= (L^* - \{(s, v_s)\}) \cup L_s[v_s] \end{aligned}$$

と更新する。ただし、 v_s は最大流問題の解において $f((s, v_s)) = 1$ となっている頂点を表している。また、最大流問題の解における辺 (v, u) の流量は最長路問題における v から u への多重辺を用いる本数に対応するものとして更新を行う。

Step 4

$T_i = \{v \in V_i \mid \exists u \in V_{i+1} \cup \dots \cup V_k, (v, u) \in A\}$ とする。 T_i は V_i において V_i 以外の強連結成分への辺が存在する頂点の集合を表している。

Step 5

任意の $v \in T_i$ について、 v を終点に固定して、 V_i が誘導する部分グラフ上で終点固定の最大流問題を解く。このときの最大流問題の最適値を z_i^* 、最適解を L_v^* とする。このとき、 $(v, u) \in A$ を満たす任意の $u \in V_{i+1} \cup \dots \cup V_k$ に対し、 $w(u) < z_i^* + 1$ が成り立つとき

$$w(u) := z_i^* + 1,$$

$$L_s[u] := (L_i^* - \{(s, v_s)\}) \cup L_s[v_s] \cup \{(v, u)\}$$

と更新する。ただし、 v_s は最大流問題の解において $f((s, v_s)) = 1$ となっている頂点を表している。また、最大流問題の解における辺 (v, u) の流量は最長路問題における v から u への多重辺を用いる本数に対応するものとして更新を行う。

Step 6

もし $i \geq k$ であれば全ての強連結成分について探索し終えたので、アルゴリズムを終了し辺集合 L を出力。そうでない場合、 i を 1 増加させ Step 3 へ戻る。

以上が提案アルゴリズムの概要である。

4.3 アルゴリズムの動作例

続いて最大流問題の解から最長路が得られないグラフである図 15 とその解である図 16 を例に提案アルゴリズムの動作を説明する。

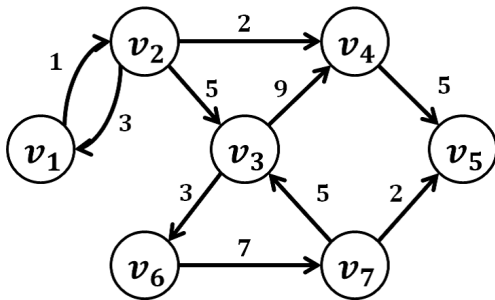


図 15: 入力グラフ (図 9 の再掲)

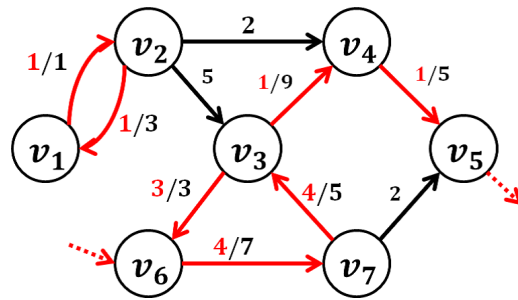


図 16: 最大流問題の解 (図 10 の再掲)

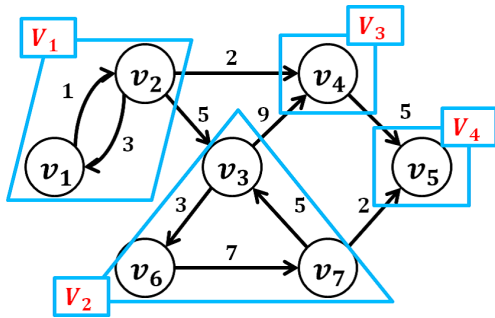


図 17: 強連結成分分解

[Step 1]

入力グラフの強連結成分分解を求める。
この入力の場合は図 17 の V_1, V_2, V_3, V_4 が
分割として得られる。Step 2 へ。

[Step 2]

変数をそれぞれ

$$i := 1,$$

$$L := \emptyset,$$

$$l := 0$$

と初期化する。また任意の頂点 v_i について

$$w(v_i) := 1,$$

$$L_s[v_i] := \{(s, v_i)\}$$

と初期化する。Step 3 へ。

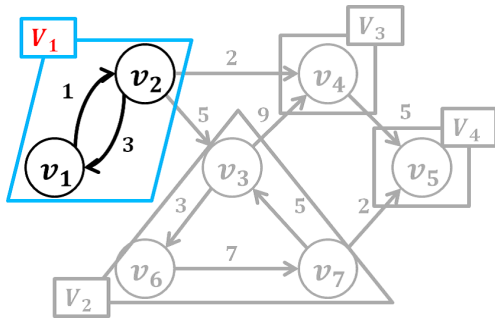


図 18: V_1 が誘導する部分グラフ

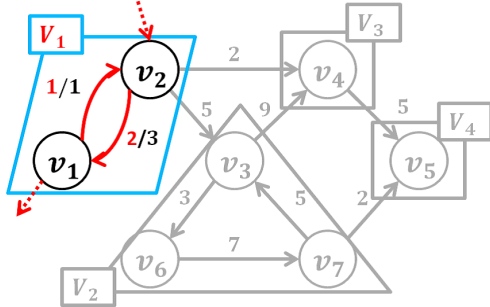


図 19: V_1 の最大流

[Step 3]

V_1 が誘導する部分グラフ上で最大流問題を解く。図 18 においては $V_1 = \{v_1, v_2\}$ について考える。解いた結果、図 19 のようなグラフが解として得られる。このとき、最適解 L^* と最適値 z^* はそれぞれ

$$L^* := \{(s, v_2), (v_2, v_1), (v_1, v_2), (v_2, v_1), (v_1, t)\},$$

$$z^* := 5$$

となっている。ここで、 L^* の中には (v_2, v_1) が二つ存在しているように見えるが、最長路問題においては多重辺が存在するので異なる有向辺であり区別する。この最適値と暫定値 $l = 0$ を比較すると、 $l < z^*$ となる。よって

$$L := (L^* - \{(s, v_2)\}) \cup L_s[v_2]$$

$$= \{(v_2, v_1), (v_1, v_2), (v_2, v_1), (v_1, t)\} \cup \{(s, v_2)\}$$

$$= \{(s, v_2), (v_2, v_1), (v_1, v_2), (v_2, v_1), (v_1, t)\},$$

$$l := 5$$

と更新する。Step 4 へ。

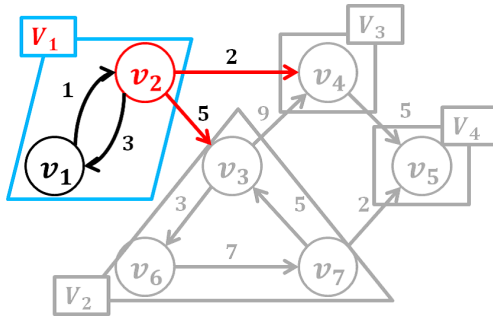


図 20: 頂点集合 T_1

[Step 4]
 T_1 を求める。図 20 においては

$$T_1 := \{v_2\}$$

となる。Step 5 へ。

[Step 5]
 v_2 を終点に固定した最大流問題を解くと
 図 21 のような結果になるため

$$L_t^* := \{(s, v_2), (v_2, v_1), (v_1, v_2)\},$$

$$z_t^* := 3$$

が最適解として得られる。

このとき、 $(v_2, v_3), (v_2, v_4) \in A$ なので、
 $z_t^* + 1 = 4$ と $w(v_3) = 1, w(v_4) = 1$ を比較
 すると、それぞれ $w(v_3) < z_t^* + 1, w(v_4) <$
 $z_t^* + 1$ となる。よって

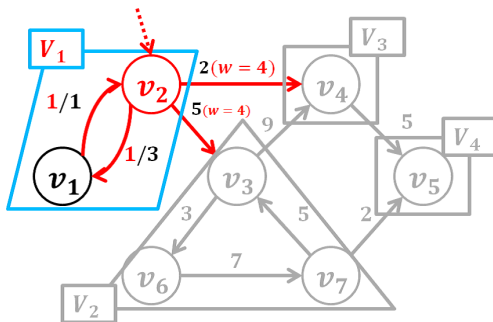


図 21: 終点 v_2 の最大流

$$L_s[v_3] := \{(s, v_2), (v_2, v_1),$$

$$(v_1, v_2), (v_2, v_3)\},$$

$$w(v_3) := 4,$$

$$L_s[v_4] := \{(s, v_2), (v_2, v_1),$$

$$(v_1, v_2), (v_2, v_4)\},$$

$$w(v_4) := 4$$

と更新する。Step 6 へ。

[Step 6]
 $i = 1, k = 4$ より $i < k$ なので、 $i := 2$ と
 更新して Step 3 へ戻る。

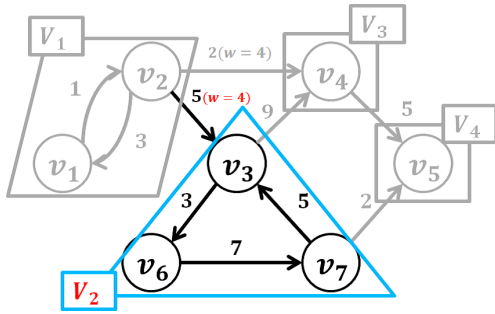


図 22: V_2 が誘導する部分グラフ

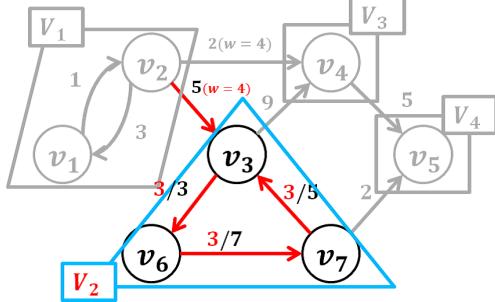


図 23: V_2 の最大流

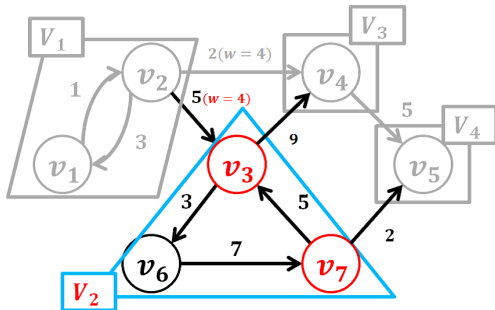


図 24: 頂点集合 T_2

[Step 3]

V_2 が誘導する部分グラフ上で最大流問題を解く。図 22 においては $V_2 = \{v_3, v_6, v_7\}$ について考える。解いた結果、図 23 のようなグラフが解として得られる。このとき最適値 z^* は

$$z^* := 14$$

となる。この最適値と暫定値 $l = 5$ を比較すると、 $l < z^*$ となる。よって

$$L := (L^* - \{(s, v_3)\}) \cup L_s[v_3],$$

$$l := 14$$

と更新する。Step 4 へ。

[Step 4]

T_2 を求める。図 24 においては

$$T_2 := \{v_3, v_7\}$$

となる。Step 5 へ。

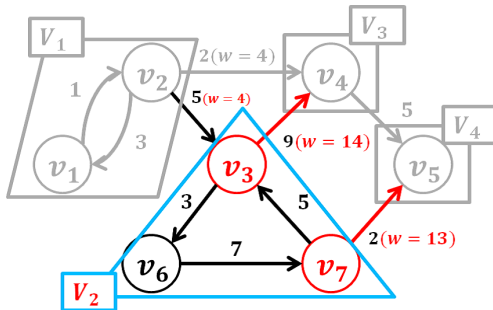


図 25: 終点固定の最大流を解いた結果

[Step 5]

v_3, v_7 を終点に固定した最大流問題をそれぞれ解く。まず、 v_3 を終点とした場合の最適値 z_t^* は

$$z_t^* := 13$$

となる。このとき、 $(v_3, v_4) \in A$ なので、 $z_t^* + 1 = 14$ と $w(v_4) = 4$ を比較すると $w(v_4) < z_t^* + 1$ となる。よって

$$L_s[v_4] := (L_t^* - \{(s, v_3)\}) \cup L_s[v_3] \cup \{(v_3, v_4)\},$$

$$w(v_4) := 14$$

と更新する。次に v_7 を終点に固定した場合の最適値 z_t^* は

$$z_t^* := 12$$

となる。このとき $(v_7, v_5) \in A$ なので、 $z_t^* + 1 = 13$ と $w(v_5) = 1$ を比較すると $w(v_5) < z_t^* + 1$ となる。よって

$$L_s[v_5] := (L_t^* - \{(s, v_3)\}) \cup L_s[v_3] \cup \{(v_7, v_5)\},$$

$$w(v_5) := 13$$

と更新する。更新した結果は図 25 のようになる。Step 6 へ。

[Step 6]

$i = 2$ より、 $i < k$ なので $i := 3$ と更新して Step 3 へ戻る。

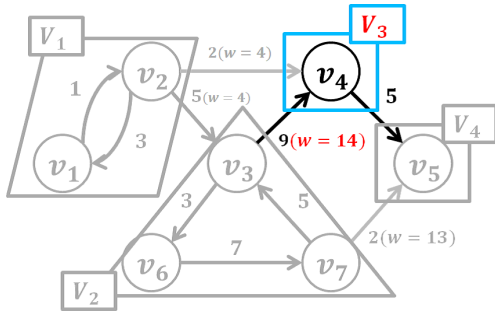


図 26: V_3 が誘導する部分グラフ

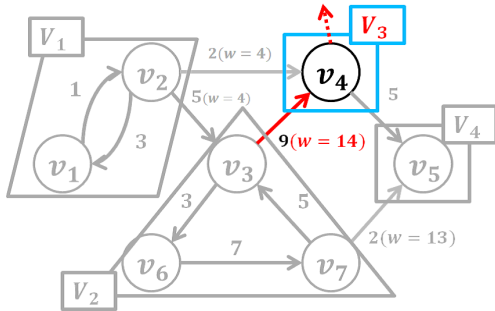


図 27: V_3 の最大流

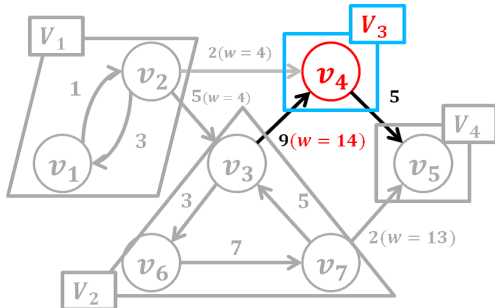


図 28: 頂点集合 T_3

[Step 3]

V_3 が誘導する部分グラフ上で最大流問題を解く。図 26 では $V_3 = \{v_4\}$ について考える。解いた結果、図 27 のようなグラフが解として得られる。このとき最適値 z^* は

$$z^* := 15$$

となる。この最適値と暫定値 $l = 14$ を比較すると $l < z^*$ となる。よって

$$L := (L^* - \{(s, v_4)\}) \cup L_s[v_4],$$

$$l := 15$$

と更新する。Step 4 へ。

[Step 4]

T_3 を求める。図 28 においては

$$T_3 := \{v_4\}$$

となる。Step 5 へ。

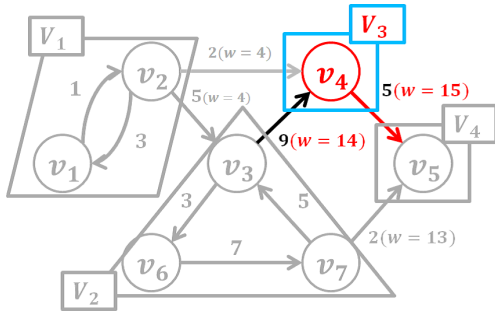


図 29: 終点固定の最大流を解いた結果

[Step 5]

v_4 を終点に固定した最大流問題を解く。 v_4 を終点とした場合の最適値 z_t^* は

$$z_t^* := 14$$

となる。このとき、 $(v_4, v_5) \in A$ なので、 $z_t^* + 1 = 15$ と $w(v_5) = 13$ を比較すると $w(v_5) < z_t^* + 1$ となる。よって

$$L_s[v_5] := (L_t^* - \{(s, v_4)\}) \cup L_s[v_4] \cup \{(v_4, v_5)\},$$

$$w(v_5) := 15$$

と更新する。更新した結果は図 29 のようになる。Step 6 へ。

[Step 6]

$i = 3$ より、 $i < k$ なので $i := 4$ と更新して Step 3 へ戻る。

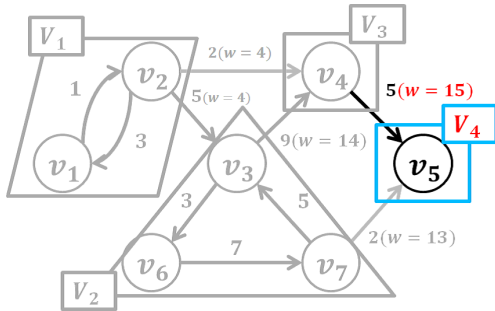


図 30: V_4 が誘導する部分グラフ

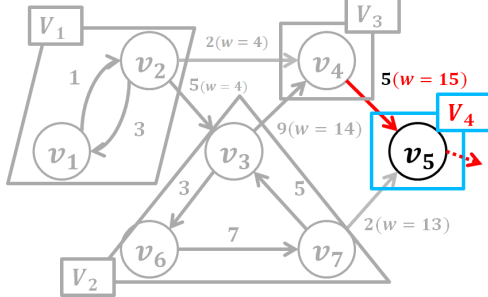


図 31: V_4 の最大流

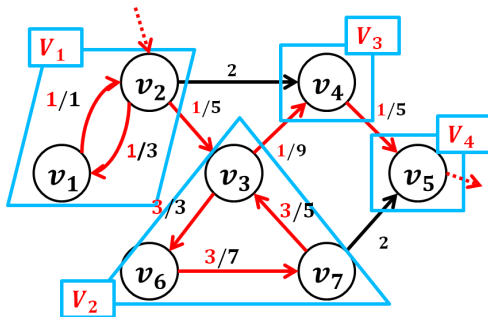


図 32: 出力するときの L

[Step 3]

V_4 が誘導する部分グラフ上で最大流問題を解く。図 30 では $V_4 = \{v_5\}$ について考える。解いた結果、図 31 の解が得られる。このとき最適値 z^* は

$$z^* := 16$$

となる。この最適値と暫定値 $l = 15$ を比較すると $l < z^*$ となる。よって

$$L := (L^* - \{(s, v_5)\}) \cup L_s[v_5],$$

$$l := 16$$

と更新する。Step 4 へ。

[Step 4]

T_4 を求める。図 30 においては

$$T_4 := \emptyset$$

となる。Step 5 へ。

[Step 5]

$T_4 = \emptyset$ なので、Step 6 へ。

[Step 6]

$i = k$ より、 L を出力しアルゴリズムを終了する。出力されるとき L は図 32 のようになる。

4.4 アルゴリズムの計算量

提案アルゴリズムの計算量の見積もりを行う。提案アルゴリズムでは、まず強連結成分分解を求める。強連結成分分解を求めるアルゴリズムは計算量が $O(|V| + |A|)$ のものが存在する。また、各 V_i において、最大流問題を 1 回解き、終点固定の最大流問題を $|T_i|$ 回解く。そこで頂点集合 V 、最大の辺容量 C 、最大の頂点重み W の入力グラフにおける最大流問題を解くときの計算量を $f(|V|, C, W)$ とする。このとき、最大流問題は線形計画問題として定式化できるため最悪の場合でも多項式時間アルゴリズムが存在する。よって $f(|V|, C, W)$ は最悪の場合でも $|V|^l \log^m C \log^n W$ (ただし l, m, n は定数) になる。このとき、提案アルゴリズムの計算量は

$$\begin{aligned} (\text{アルゴリズムの計算量}) &= O(|V|^2) + \sum_{i=1}^k (|T_i| + 1) f(|V_i|, C, W) \\ &\leq O(|V|^2) + \sum_{i=1}^k (|T_i| + 1) f(|V|, C, W) \\ &\leq O(|V|^2) + (|V| + k) f(|V|, C, W) \quad (\because \sum_{i=1}^k |T_i| \leq |V|) \\ &\leq O(|V|^{l+1} \log^m C \log^n W) \\ &\quad (\because k \leq |V|, f(|V|, C, W) \leq |V|^l \log^m C \log^n W) \end{aligned}$$

となる。これにより C, W は入力によって決まるので、計算量を見積もる上では各強連結成分のサイズ $|V_i|$ が重要になると考えられる。ここで、ランダムグラフにおける強連結成分のサイズは辺数によって異なる値に概収束することが知られており [6]、それは次のようになる。

辺数に対する強連結成分のサイズ [6]

有向グラフ $G(V, A)$ 、 $\epsilon > 0$ を定数、 ω を $\omega(|V|) \rightarrow \infty, (|V| \rightarrow \infty)$ を満たす任意の関数とすると

1. $\epsilon < 1, |A| = (1 - \epsilon)|V|$ のとき、各強連結成分は長さが $\omega(|V|)$ より小さい閉路になる。つまり、任意の i に対し $|V_i| < \omega(|V|)$ となる。
2. $|A| = (1 + \epsilon)|V|$ のとき、各強連結成分は、 $\alpha|V|$ (α は定数) の大きな強連結成分になるか、または $\omega(|V|)$ より小さい閉路になる。つまり、ある i に対し $|V_i| = \alpha|V|$ となり、その i を除く任意の j においては $|V_j| < \omega(|V|)$ となる。
3. $\frac{|A|}{|V|} \rightarrow \infty$ のとき、 $(1 - o(1))|V|$ のサイズの強連結成分は一つだけ存在する。

これにより、アルゴリズムの計算量は次のように概収束する。

1. $\epsilon < 1, |A| = (1 - \epsilon)|V|$ のとき、

$$\begin{aligned} (\text{アルゴリズムの計算量}) &= \sum_{i=1}^k (|T_i| + 1) f(|V_i|, C, W) \\ &< \sum_{i=1}^k (|T_i| + 1) f(\omega(|V|), C, W) \\ &= O(|V| f(\omega(|V|), C, W)). \end{aligned}$$

ここで、 $\omega(|V|)$ は $\omega \rightarrow \infty$ を満たす任意の関数であるため、 $\forall i, |V_i| < \omega(|V|)$ より $|V_i| < \log |V|$ を満たす。よって、アルゴリズムの計算量は

$$\begin{aligned} (\text{アルゴリズムの計算量}) &= O(|V| f(\omega(|V|), C, W)) \\ &= O(|V| \log^l |V| \log^m C \log^n W) \\ &\leq O(|V|^{l+1} \log^m C \log^n W). \end{aligned}$$

2. $|A| > |V|$ のとき、

$$\begin{aligned} (\text{アルゴリズムの計算量}) &= O(|V| f(|V|)) \\ &\leq O(|V|^{l+1} \log^m C \log^n W). \end{aligned}$$

提案アルゴリズムの計算量はこのようになる。以上より、提案アルゴリズムは任意のグラフに対し最悪の場合でも多項式時間で動作するが、辺密度が低いランダムグラフ上においては計算量の頂点数に関する項が $|V| \log^l |V|$ となるため、 $|V|^{l+1}$ の場合よりも良い計算時間で動作する。

5 実験

提案アルゴリズムから最長路が得られる確率を検証するため、実際にランダムグラフを複数作成し提案アルゴリズムを適用する実験を行い、そのときに最長路が得られる確率を観察する。まず、実験1では様々な辺密度のグラフについて「最大流問題を解いた場合に最長路が得られる確率」と「提案アルゴリズムを適用した場合に最長路が得られる確率」を求めて比較し、その差を観察することで性能を検証する。次に実験2では頂点数を変えた場合にもアルゴリズムを適用することにより最長路が得られる確率が改善されるかどうか確認するため、頂点数 20, 30, 40, 50 の場合について実験1と同様の実験を行い、その結果を観察した。ここで本研究ではランダムグラフ上の最長路問題を考える。また、解が最長路になっているかどうかの確認は、解によって誘導されるグラフが連結であるかどうかによって決まる。そのため本章の最初では、ランダムグラフの作成方法と連結成分の探索を行うためのアルゴリズムの説明を行う。

5.1 ランダムグラフの作成

ランダムグラフは、頂点数を n 、辺密度を p とすると、 $[n^2p]$ 本の辺をどこにつけるかを無作為に決めることで作る。また各辺は容量 (多重辺の本数) を持ち、値は 1 から 9 の整数から無作為に決めることで作成する (全ての辺について $\frac{1}{9}$ の確率で 1、 $\frac{1}{9}$ の確率で 2、 \dots 、 $\frac{1}{9}$ の確率で 9 となる)。このとき、作成されるグラフは必ずしも連結であるとは限らない。

5.2 連結成分の探索

最大流問題を解くことで得られた解により誘導されるグラフの連結成分に含まれる頂点を幅優先探索で求め、一つの連結成分から構成されているならば “yes”、複数の連結成分に分かれたならば “no” を出力するプログラムを設計した。次にアルゴリズムを示す。ここで、幅優先探索に用いているアルゴリズムは文献 [7]、P82 の BFS を参考に作成している。

連結成分探索アルゴリズム

Algorithm

任意の頂点 $v \in V$ に対して $\text{Discovered}[v]$ を false とする

s を $L[0]$ に加える

層カウンター i を 0 にする

While $L[i] \neq \text{空}$

$L[i+1]$ を空とする

For 各頂点 $u \in L[i]$

u に接続する各辺 (u, v) を考える

If $\text{Discovered}[v]=\text{false}$ **then**

$\text{Discovered}[v]$ を true とする

v をリスト $L[i+1]$ に加える

Endif

Endfor

 層カウンター i を 1 増やす

Endwhile

ステータス変数 $status$ を true とする

任意の頂点 $v \in V$ について考える

If $\text{Discovered}[v]=\text{false}$ かつ v に接続する辺が存在する **then**

$status$ を false とする

Break

Endif

If $status = \text{true}$ **then**

 “yes” と出力

Else “no” 出力

Endif

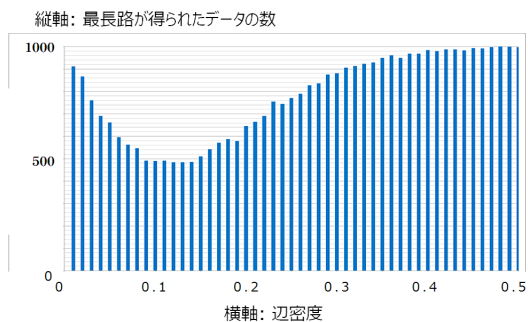
5.3 実験 1

まず、様々な辺密度のグラフについて「最大流問題を解いた場合に最長路が得られる確率」と「提案アルゴリズムを適用した場合に最長路が得られる確率」を求めて比較し、その差を観察することで提案アルゴリズムの性能を検証する。

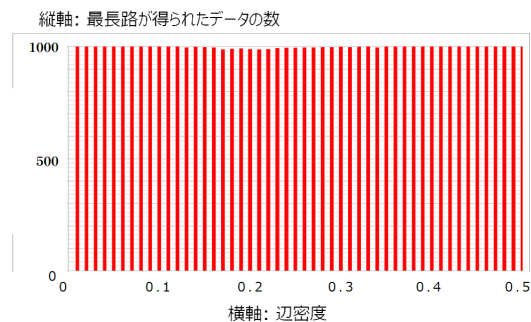
実験データとして頂点数を 10、辺密度を 0.01 から 0.50 の値を 0.01 刻みでそれぞれ 1000 個のランダムグラフを作成する。

5.3.1 実験結果

実験結果は最大流問題のみの場合が図 33.1、提案アルゴリズムを適用した場合が図 33.2 のようになった。横軸が辺密度、縦軸が各辺密度において最長路が得られたデータの数を表している。

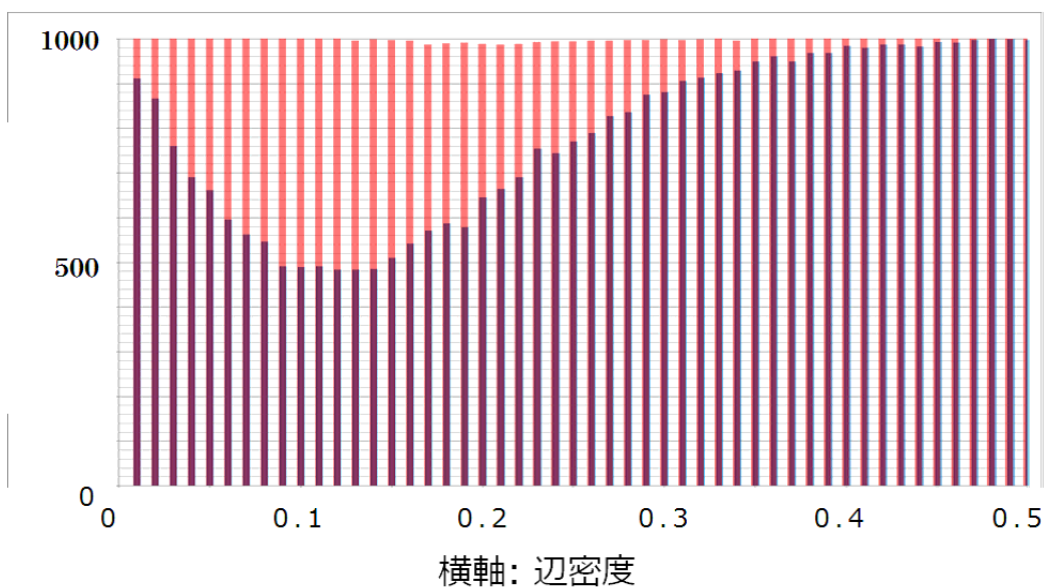


(1) 最大流問題を解いた結果



(2) 提案アルゴリズムを適用した場合

縦軸: 最長路が得られたデータの数



(3) 図 33.1 と 33.2 を重ねた図

図 33: アルゴリズム適用の有無による実験結果の比較

5.3.2 考察

図 33.1 より最大流問題を解いた場合は辺密度 0.4 以下で解ける確率が低くなっていることがわかる。それに対し、図 33.2 より提案アルゴリズムを適用した場合は大きく改善されていることがわかる。しかし、アルゴリズムを適用した場合でも最長路が得られないようなグラフも存在した。図 34 は、そのようなグラフの例であり、各辺の値は多重辺の本数を表している。

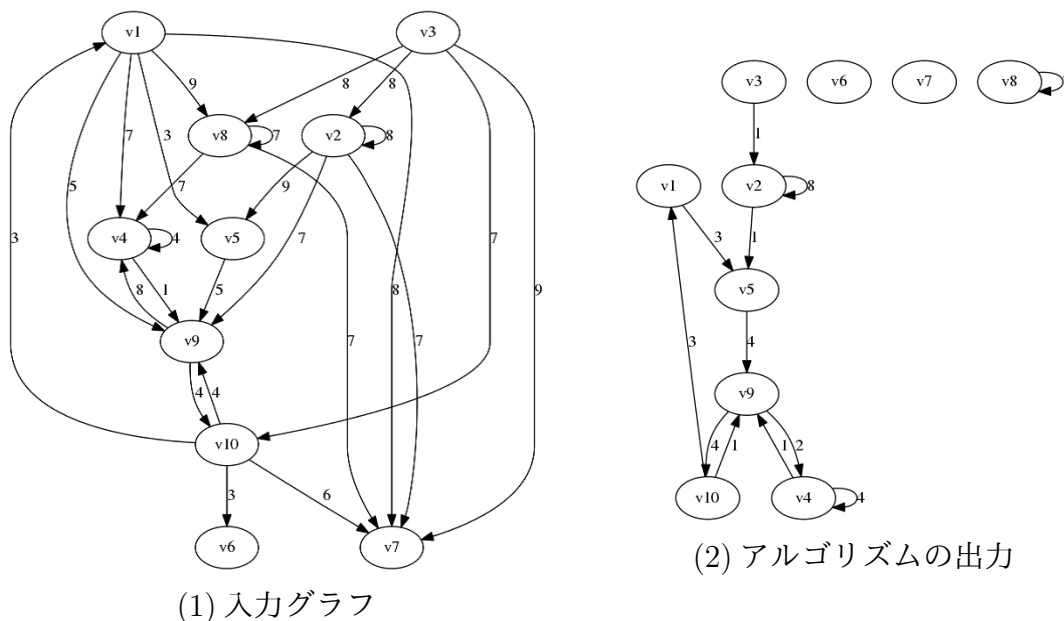


図 34: 最長路が得られないグラフの例

図 34 の例では、出力において頂点 v_8 が自己ループを持つ孤立点となっているため、これは最長路ではない。また入力グラフにおける強連結成分分解を求めると

$$\{v_3\}, \{v_2\}, \{v_1, v_4, v_5, v_8, v_9, v_{10}\}, \{v_6\}, \{v_7\}$$

が得られる。これにより、 v_8 が大きい強連結成分に含まれていることがわかる。それにもかかわらず最大流問題の最適解において孤立点になってしまうので、本研究における提案アルゴリズムでは対応できず、最長路が得られなかったと考えられる。

そこで様々な最長路が得られなかったグラフにおいて、 v_8 と同じように孤立点となるような頂点を観察したところ、 s から t への道に含まれない頂点 v の特徴の一つとして、次のようなことが考えられる。

1. v は大きい強連結成分に含まれる。以降その強連結成分を V_i とする。

2. V_l が誘導する部分グラフにおいて、次の頂点集合 V_p, V_n について考える。

$$\begin{aligned} V_p &= \{u \in V_l \mid (u, v) \in A\}, \\ V_n &= \{u \in V_l \mid (v, u) \in A\}. \end{aligned}$$

これらの頂点集合のサイズ $|V_p|, |V_n|$ が小さい。図 34 の v_8 においては、 $|V_p| = 1$, $|V_n| = 1$ である。

3. v への辺 (u_p, v) を持つ頂点 u_p について、 u の入次数が出次数に比べ小さい。
図 34 において u_p に対応する頂点は v_1 と v_3 である。 v_1 の入次数は 3、出次数は $5 + 7 + 3 + 9 + 8 = 32$ であり、 v_3 の入次数は 0、出次数は $8 + 8 + 7 + 9 = 32$ なので、ともに入次数に比べ出次数が非常に大きい。
4. v からの辺 (v, u_n) を持つ頂点 u_n について、 u_n の出次数が入次数に比べ小さい。
図 34 において u_n に対応する頂点は v_4 と v_7 である。 v_4 の出次数は 1、入次数は $7 + 7 + 8 = 22$ であり、 v_7 の出次数は 0、入次数は $6 + 7 + 8 + 7 + 9 = 37$ なので、ともに出次数に比べ入次数が非常に大きい。

これらの内、3 と 4 のどちらか一方の特徴を持ち、さらに 1 と 2 の特徴を持っているような頂点が存在するグラフは提案アルゴリズムを適用しても最長路が得られない確率が高いと考えられる。

5.4 実験 2

実験 1 の結果より、提案アルゴリズムによって最長路が得られる確率が改善されることが示唆された。次に頂点数が変化した場合にも提案アルゴリズムによる改善が見られるか検証する実験を行う。

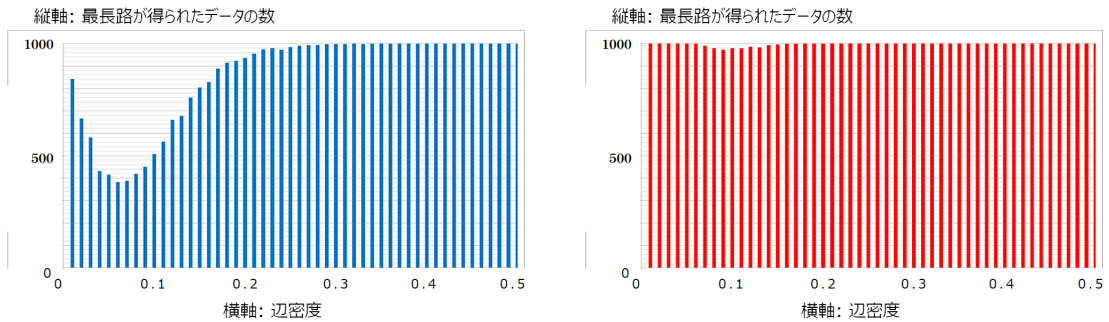
実験データとして頂点数を 20、30、40、50 の場合について実験を行った。

頂点数 20 の場合は、辺密度を 0.01 から 0.50 の値を 0.01 刻みでそれぞれ 1000 個のランダムグラフを作成した。

頂点数 30 以上の場合は、辺密度を 0.005 から 0.250 の値を 0.005 刻みでそれぞれ 1000 個のランダムグラフを作成した (頂点数が 30 以上のとき、辺密度が 0.25 以上のデータについては、ほとんど最長路が求められる示唆が得られている [1] ため、辺密度 0.25 以上は省略する)。

5.4.1 実験結果

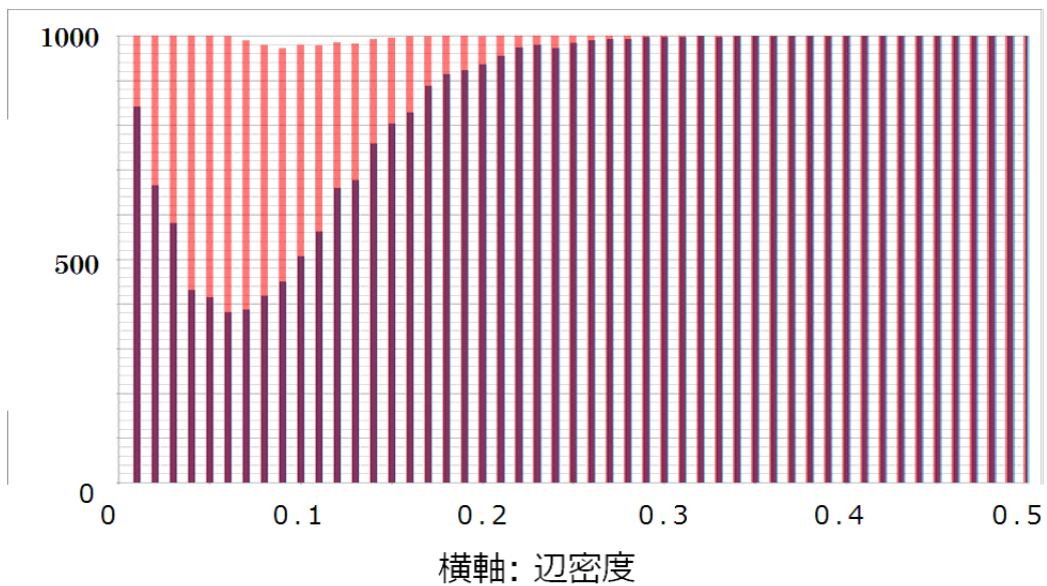
実験の結果は図 35, 36, 37, 38 のようになった。横軸が辺密度、縦軸が各辺密度において最長路が得られたデータの数を表している。また図 35 が頂点数を 20 としたときの結果、図 36 が頂点数を 30 としたときの結果、図 37 が頂点数を 40 としたときの結果、図 38 が頂点数を 50 としたときの結果をそれぞれ表している。



(1) 最大流問題を解いた結果

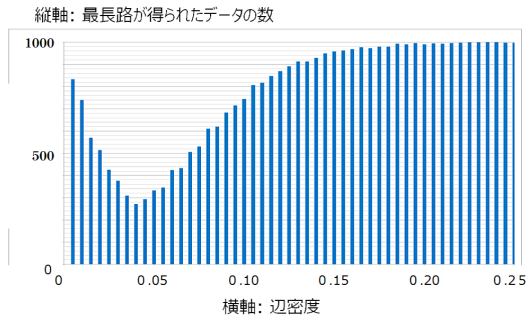
(2) 提案アルゴリズムを適用した場合

縦軸: 最長路が得られたデータの数

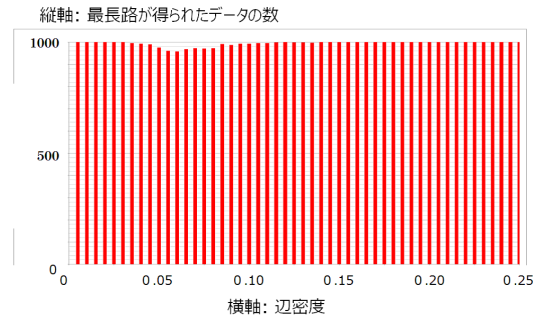


(3) 図 35.1 と 35.2 を重ねた図

図 35: アルゴリズム適用の有無による実験結果の比較 (頂点数 20)

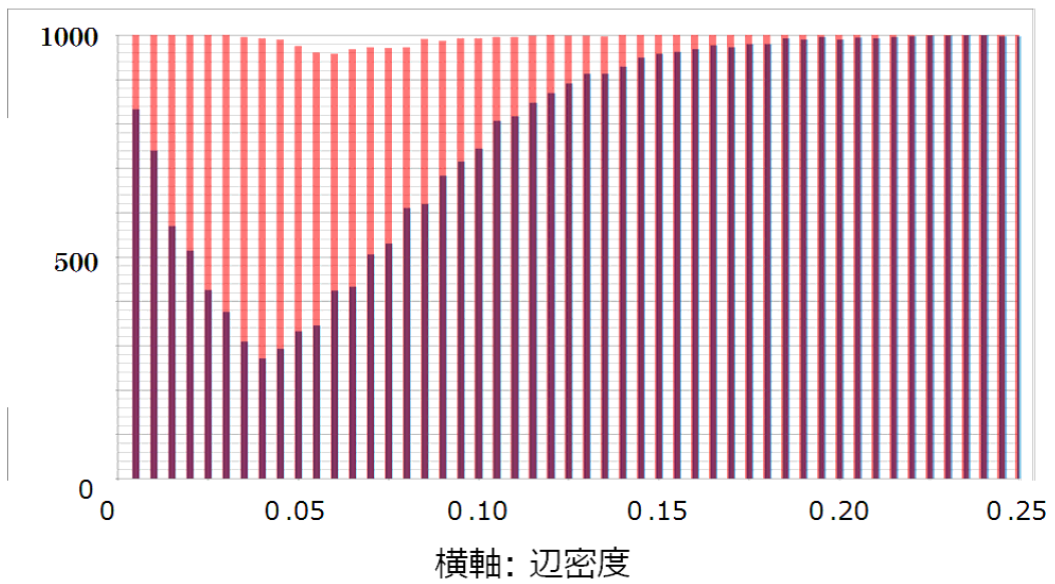


(1) 最大流問題を解いた結果



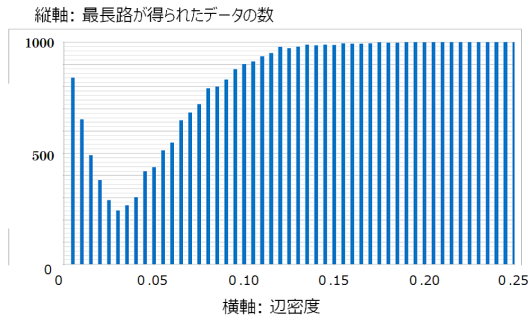
(2) 提案アルゴリズムを適用した場合

縦軸: 最長路が得られたデータの数

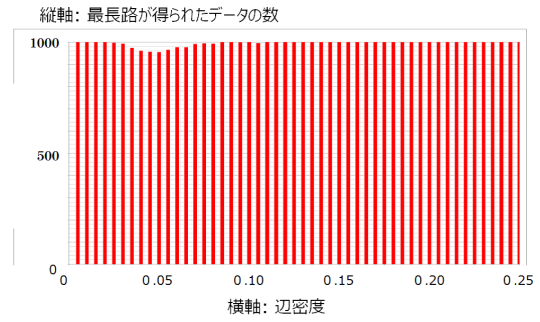


(3) 図 36.1 と 36.2 を重ねた図

図 36: アルゴリズム適用の有無による実験結果の比較 (頂点数 30)

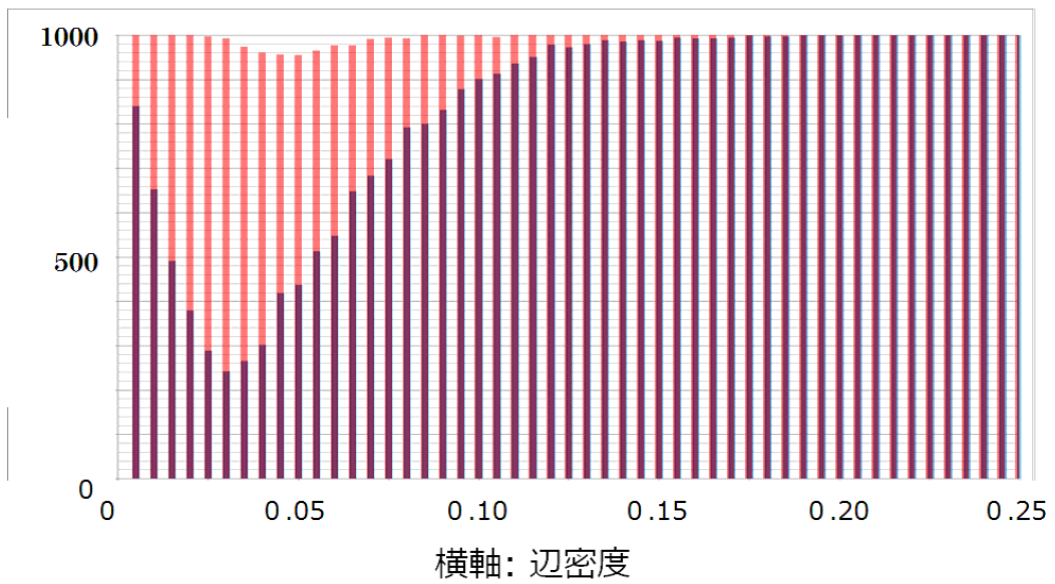


(1) 最大流問題を解いた結果



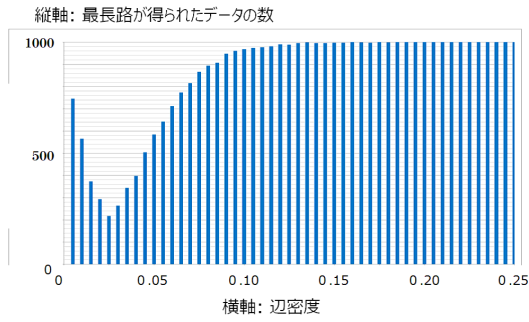
(2) 提案アルゴリズムを適用した場合

縦軸: 最長路が得られたデータの数

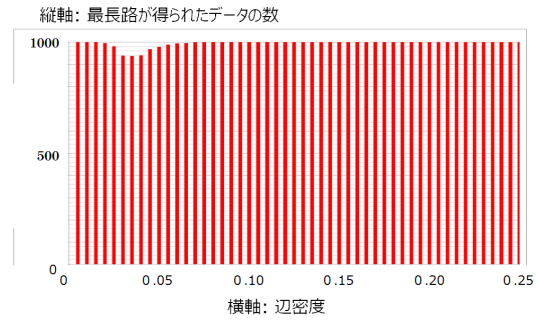


(3) 図 37.1 と 37.2 を重ねた図

図 37: アルゴリズム適用の有無による実験結果の比較 (頂点数 40)

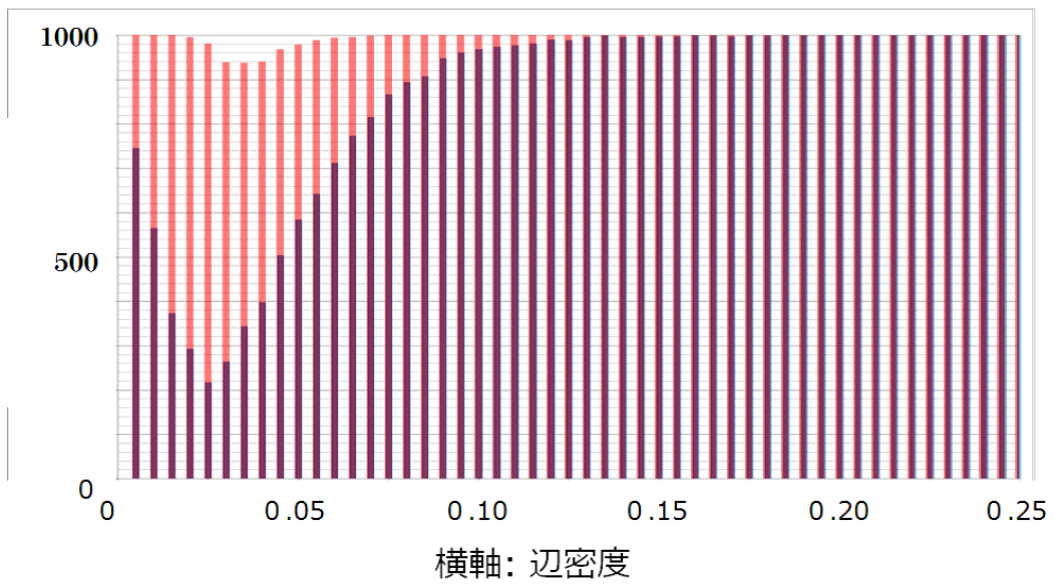


(1) 最大流問題を解いた結果



(2) 提案アルゴリズムを適用した場合

縦軸: 最長路が得られたデータの数



(3) 図 38.1 と 38.2 を重ねた図

図 38: アルゴリズム適用の有無による実験結果の比較 (頂点数 50)

5.4.2 考察

図 35, 36, 37, 38 より、頂点数が異なる場合であっても改善されることが示唆された。また、各頂点数の実験において次の値を求め、表 4 にまとめた。また、表 4 の各頂点数に対する辺密度と最長路が得られたデータの数の値をプロットした結果は図 39 のようになった。

- (1). 最も最長路が得られなかったときの辺密度
- (2). (1) の辺密度において最長路が得られなかったデータの数
- (3). (1) の辺密度において最長路が得られなかった割合

表 4: アルゴリズムを適用した結果

頂点数	(1)	(2)	(3)
10	0.18	14	1.4%
20	0.09	27	2.7%
30	0.060	42	4.2%
40	0.050	44	4.4%
50	0.035	62	6.2%

表 4 および図 39 より、アルゴリズムを適用した場合に最長路が得られなかったデータの数は頂点数が大きくなるにつれ増える傾向にあると考えられる。しかし、最長路が得られなくなっている辺密度を観察したところ、頂点数が大きくなるにつれその辺密度は小さくなっていく傾向も見られる。

そこで、辺密度がいくつ以上のときに最長路が得られた割合が 99%以上となっているかを観察してみたところ、表 5 のような結果が得られた。

表 5: 最長路が得られた割合が 99%以上となる辺密度

頂点数	辺密度
10	0.18
20	0.12
30	0.085
40	0.070
50	0.055

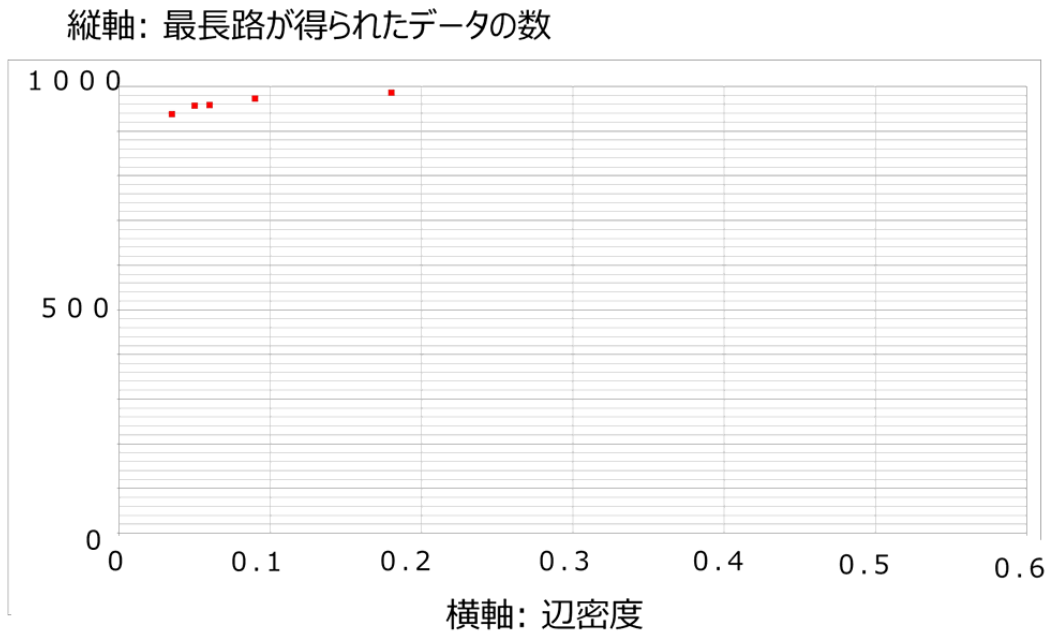


図 39: 表 4 をプロットした結果

表 5 より、頂点数が高くなるにつれより低い辺密度でも最長路が得られた割合が 99%以上となっていることがわかる。また、さらに辺密度が高くなると最長路が得られた割合はいずれの頂点数においても 100%になる。よって、頂点数が大きくなれば最長路が得られる確率は減少するが、それと同時に最長路が得られない辺密度の範囲も小さくなると考えられるため、提案アルゴリズムは高い確率で最長路問題を解くことができると考えられる。

6 まとめ

実験結果より、本研究で提案する発見的アルゴリズムは高確率で最長路問題の解を出力し、またアルゴリズム計算量は頂点数に対する多項式時間である。

また、頂点数が大きくなるにつれて最長路が得られる確率が 1 に収束する辺密度が小さくなる傾向にあることが示唆されている。そのため、頂点数が大きい場合にも最長路が問題なく最長路が得られると考えられる。

ゆえに、提案アルゴリズムはランダムグラフ上においては最長路問題を解くのに有用であると考えられる。

さらに今回の発見的アルゴリズムでは最長路が得られないグラフの特徴に対し、最大流問題を解くときにこの特徴を取り除けるようなアルゴリズムを実装できれば、そのアルゴリズムは確実に最長路が得られると考えられる。

参考文献

- [1] 高木 雄大. “確率的に構成したグラフにおける最長路問題の性質” 電気通信大学 情報理工学部 卒業論文 (2014).
- [2] 乾 伸雄、品野 勇治、鴻池 祐輔、小谷 善行 “最長しりとり問題の解法” 情報処理学会論文誌 数理モデル化と応用 Vol.46 No. SIG 2(TOM 11) Jan. 2005 pp. 105–117
- [3] Alexander Schrijver 著, “Theory of Linear and Integer Programming”, John Wiley and Sons 出版, 1986 年
- [4] Leizhen Cai, Boting Yang. “Parameterized Complexity of Even/Odd Subgraph Problems.” *Journal of Discrete Algorithms* 9(3) (2011): pp. 231–240.
- [5] Marek Cygan, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, Ildikó Schlotter “Parameterized complexity of Eulerian deletion problems.” *Algorithmica* 68(1) (2014): pp. 41–61.
- [6] Tomasz Łuczak “The Phase Transition in the Evolution of Random Digraphs.” *Journal of Graph Theory* 14 (1990): pp. 217–223
- [7] J. Kleinberg · E. Tardos 著, 浅野 孝夫, 浅野 泰仁, 小野 孝男, 平田 富夫 訳, “アルゴリズムデザイン”、共立出版, 2008 年