

最大クリーク問題の多項式時間的可解性
に関する研究

中西 裕陽

電気通信大学

2016年12月

最大クリーク問題の多項式時間的可解性
に関する研究

中西 裕陽

電気通信大学大学院電気通信学研究科
博士（工学）学位申請論文

2016年12月

最大クリーク問題の多項式時間的可解性 に関する研究

博士論文審査委員会

主査 西野哲郎 教授

委員 高橋治久 教授

委員 伊藤大雄 教授

委員 庄野逸 教授

委員 垂井淳 准教授

委員 富田悦次 名誉教授

著作權所有者

中西 裕陽

2016

A Study on Polynomial-Time Solvability of the Maximum Clique Problem

Hiroaki Nakanishi

Abstract

The Maximum Clique Problem is well-known to be a typical NP-complete problem, and hence it is believed to be impossible to solve it in polynomial-time. Therefore, it is important to establish a reasonable sufficient condition under which the maximum clique problem can be proved to be polynomial-time solvable. It is known that the maximum clique problem is polynomial-time solvable on some special graphs such as; Planar Graph, Chordal Graph, and so on. There was, however, no quantitative reasonable sufficient condition on general graphs for this polynomial time solvability.

In this study, first we propose a basic depth-first-search algorithm for solving the maximum clique problem based on CLIQUES (E. Tomita, A. Tanaka, H. Takahashi: Theoretical Computer Science, 2006) that enumerates all maximal cliques. Following the result, we improve the condition of polynomial-time solvability of this basic algorithm by adding stronger bounding methods step by step. Finally we prove that: given a graph $G = (V, E)$ of $|V| = n$ and whose maximum degree is Δ , if Δ is less than or equal to $3.177d \lg n$ ($d \geq 0$: a constant), then the maximum clique problem is solvable in the polynomial time of $O(n^{1+d})$.

In addition, we prove the following extended result: given a graph $G = (V, E)$ of $|V| = n$ and $n_0 \geq 2$, if $\deg(v)$ is less than or equal to $3.177d \lg n$ ($d \geq 0$: a constant) for the minimum degree vertex v in any connected subgraphs $G(C)$ ($C \subseteq V$) of $|C| = n_0$, then the maximum clique problem is solvable in the polynomial time of $O(n^{\max\{2, 1+d\}})$. This greatly relaxes the condition in the sense that if C is a connected subgraph of $|C| = n_0$, then any vertex $v \in C$ except the minimum degree vertices in C is free from any condition. In conclusion, this thesis proposes a new paradigm for

the polynomial-time solvability of the maximum clique problem.

最大クリーク問題の多項式時間的可解性に関する研究

中西 裕陽

概要

いわゆる“最大クリーク問題”は典型的な NP 完全問題であり、多項式時間的に本問題を解くことはほぼ不可能であると強く予測されている。従って、少なくともどのような条件下ならばこの NP 完全問題を多項式時間的に解くことが出来るかを明らかにすることは重要な課題である。これに対し、平面グラフ、コーダルグラフ等いくつかの特殊グラフに対しては多項式時間的可解性が成立することが示されている。しかし一般グラフにおいては、最大クリーク問題が多項式時間的可解となる条件について、これまでにおいて有意義な定量的結果は発表されていなかった。そこで本研究では、先ず極大クリーク全列挙アルゴリズム CLIQUES (E. Tomita, A. Tanaka, H. Takahashi: Theoretical Computer Science, 2006) を基にして、基本的な最大クリーク抽出の深さ優先探索アルゴリズムを確立した。この基本的アルゴリズムに対して探索領域限定操作をより強化し、対応したより詳細な場合分けを伴った解析を行うことにより、アルゴリズムが多項式時間的に終端する条件を逐次緩和し、次の定量的な多項式時間的可解性条件を与えた。即ち、先ず一般グラフにおいてグラフの最大次数 Δ のみを条件とした、最大クリーク問題に対する以下の多項式時間的可解性の成立を示した。「節点数 n のグラフ $G = (V, E)$ の最大次数 Δ が、 $\Delta \leq 3.177d \lg n$ ($d \geq 0$:定数) なる条件を満たすとき、最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である。」さらに本研究においては、全節点に対する前記条件をより緩和した、次の拡張結果も与えた。「サイズ $n_0 \geq 2$ なる任意の連結な誘導部分グラフ $G(C)$ ($C \subseteq V$) に対して、 C 中の最小次数節点 v が、 $\deg(v) \leq 3.177d \lg n$ ($d \geq 0$:定数) を満たすとき、最大クリーク問題は $O(n^{\max\{2, 1+d\}})$ の多項式時間で可解である。」これは、サイズ n_0 である連結な誘導部分グラフのうち、次数最小の節点を除き全く無条件としたもので、制限条件の大きい緩和である。以上本論文では、最大クリーク問題の多項式時間的可解性について、新しい枠組みを与えた。

目次

第 1 章	序章	1
1.1	背景と本研究結果	1
1.2	本論文の概要	13
第 2 章	事前準備	15
2.1	本論文の各章共通の定義と表記法	15
2.2	本論文各章共通の基本的補題	17
第 3 章	アルゴリズム MCP_0	22
3.1	概要	22
3.2	アルゴリズム MCP_0	22
3.2.1	基本手続き EXPAND	23
3.3	部分森の同一化	25
3.4	部分問題の単純省略	26
3.5	例題	27
3.6	最大時間計算量評価	34
第 4 章	アルゴリズム MCP_1	42
4.1	概要	42
4.2	アルゴリズム MCP_1	42
4.2.1	部分問題の統合による探索領域削減	43
4.2.2	例題	47
4.3	最大時間計算量評価	51

第 5 章	アルゴリズム MCP'_1	57
5.1	概要	57
5.2	アルゴリズム MCP'_1	57
5.3	部分問題の統合による探索領域削減の詳細化	58
5.3.1	例題	62
5.4	最大時間計算量評価	66
第 6 章	アルゴリズム MCP_2	73
6.1	概要	73
6.2	アルゴリズム	73
6.2.1	部分問題の統合による探索領域削減の統一	75
6.2.2	部分問題サイズの減少	76
6.2.3	少数部分問題の分割	78
6.2.4	例題	85
6.3	最大時間計算量評価	89
第 7 章	アルゴリズム MCP_3	102
7.1	概要	102
7.2	アルゴリズム	102
7.2.1	部分問題の統合による探索領域削減の詳細化 (拡張版)	105
7.2.2	部分問題サイズの減少 (拡張版)	109
7.2.3	少数部分問題の分割 (拡張版)	111
7.2.4	節点の隣接関係による限定	116
7.2.5	例題	118
7.3	最大時間計算量評価	121
第 8 章	多項式時間的可解性の拡張	138
8.1	概要	138
8.2	拡張結果 (1)	139
8.3	拡張結果 (2)	142

8.4	拡張結果 (3)	145
第 9 章	まとめと今後の展望	149
謝辞		154
参考文献		155
関連論文の印刷公表方法と時期		167

第 1 章

序章

1.1 背景と本研究結果

NP 完全問題, あるいは NP 困難問題は数多くの重要な問題を含み, 膨大な研究が積み重ねられてきている [42], [111]. この結果として, NP 完全問題, NP 困難問題は多項式時間的には解決不可能であることが強く予測され, もし NP 完全問題のうちの 1 つでも多項式時間的に解決可能であると示されるならば, 全ての NP 完全問題が多項式時間的解決可能となる, 即ち, $P=NP$ ($P \neq NP$ が否定される) ということとなる. しかし, この $P \neq NP$ 問題は未だ解決の糸口さえ見出されていない. この様な状況において, ある NP 完全問題に対し, その問題が多項式時間的に可解となる適切な条件について, 総合的文献 [42] 中を始めとして, これまでに数多くの研究が発表されてきている.

与えられたグラフ $G = (V, E)$ に対して, G の節点集合の部分集合 Q において, Q 中の任意の 2 節点が互いに隣接するとき, Q はクリークであるという. このとき, Q の要素数 $|Q|$ を, クリーク Q のサイズという. また, G 中のあるクリーク Q が他の任意のクリークの部分集合でないとき, Q は極大クリークであるという. 与えられたグラフ G , 及び定数 k に対して, G 中の最大クリークで, サイズが k 以上であるものが存在するか否かを判定する問題を, 最大クリーク問題という. 最大クリーク問題 [42], またはこれと双対な最大独立節点集合問題 [42] は, Karp によって最初に NP 完全性を示された 21 の代表的な NP 完全問題の 1 つである [59]. 判定問題である最大クリーク問題に対して, 与えられたグラフ G 中の, サイズが最大であるクリークを抽出する問題を, 最大クリーク抽出問題といい,

これは代表的な NP 困難問題の 1 つである [11]. また, グラフ中の極大クリークを全て列挙する問題を, 極大クリーク全列挙問題という [19], [64], [51], [106].

最大クリーク抽出問題, あるいは, 極大クリーク全列挙問題は, バイオインフォマティクス及びケモインフォマティクス [4], [48], [1], [61], [73], [5], [96], [6], [21], [23], [47], [121], [91], [65], [18], [66], [30], [70], [40], [75], 経済学 [10], 試験時間割作成 [24], [25], 金融ネットワーク [10], ロケーション [20], スケジューリング [31], [110], ソーシャルネットワーク解析 [7], [90], [88], [119], [120], ワイヤレスネットワーク [28], [55], データマイニング [118], [69], [108], [86], [109], [87], 画像処理 [85], [49], 量子回路の最適設計 [84] など多くの重要な応用面を持ち [77], [112], これらの分野でそれぞれ有益な成果が得られている.

上記の通り, NP 完全問題, NP 困難問題は多項式時間的には解決不可能であることが強く予測されているため, 最大クリーク問題に対してもこれを一般的に高速に (即ち, 多項式時間で) 解くアルゴリズムは存在しないことがほぼ確実視されている. 更に, 最大クリークについては, 近似解を得ることすら困難であることが示されている [122].

与えられたグラフ $G = (V, E)$ に対して, その最大クリークを抽出する最も単純な方法は, G の節点集合 V の全ての部分集合を抽出し, それらの集合がクリークであるか否かを確かめることによる方法である. $|V| = n$ とすれば, この方法による計算量上界は $O^*(2^n)$ -時間^{*1} という, 節点数の指数関数となる.

1977 年, Tarjan-Trojanowski は最大クリーク抽出問題に関して, $O(2^{n/3})$ -時間 [97] という結果を示した. ここで用いられたアルゴリズムは, 分枝限定 (Branch & Bound) 法, あるいは分枝縮小 (Branch & Reduce) 法と呼ばれる手法である. これは, 「分枝規則 (Branching rule)」, 「限定規則 (Bounding rule)」, 及び 「縮小規則 (Reduction rule)」という, それぞれ有限個の規則により構成されたアルゴリズムである. これら 2 種の規則を次の様に適用し, アルゴリズムは問題を解決する.

1 $O^()$: 一般に, 任意の関数 $f(n), g(n)$, 及び多項式 $P(n)$ に対して,

$$f(n) = O(P(n) \cdot g(n))$$

であるとき, 多項式 $P(n)$ を省略した場合,

$$f(n) = O^*(g(n))$$

と表記する.

・分枝規則により, 与えられた入力 (以下, これを問題という) を, 一般に複数の, より小さな入力 (以下, 部分問題) に分割する.

・問題に対して, 限定 (縮小) 規則により, 問題に対する実行を限定する, 又は問題のサイズを減少させることが可能であるか否かを判定し, 可能である場合には実行を限定, 又は問題のサイズを減少させる.

問題に対して, 実行の限定を重視するアルゴリズムは分枝限定アルゴリズム, 問題サイズの減少を重視するアルゴリズムは分枝縮小アルゴリズムと呼ばれる. 実際にはこの両手法は併用されることも多いが, いずれも分枝規則により, 与えられた問題をより小さい問題へと分割するという枠組においては共通である. 以下, 本論文ではこの枠組みによるアルゴリズムを, 分岐アルゴリズム (Branching Algorithm [39]) と呼ぶ. 本論文のアルゴリズムにおいては, 限定規則と縮小規則の双方を合わせて適用している.

分岐アルゴリズムの最大時間計算量評価は, 次の枠組による. 即ち, いま評価を行うアルゴリズムに対して, サイズ n の入力のうち最も計算時間を要する入力に対する時間計算量を $T(n)$ とする. サイズ n の入力が, 分枝規則によって r 個の部分問題に分割されたとき, 次の不等式が成立する.

$$T(n) \leq T(n_1) + T(n_2) + \cdots + T(n_r) + P(n) \quad (n_i < n, 1 \leq i \leq r)$$

r は分枝規則, 及び限定 (縮小) 規則によって決定される. (例えば, ある分枝規則における条件が成立する / しない 場合により問題を 2 種類に分割する, など.)

n_1, n_2, \dots, n_r はそれぞれ分割された部分問題のサイズであり, 分割前の問題サイズ n より真に小さい. 限定 (縮小) 規則は, このような問題分割時に適用され, 部分問題の数 r , 及びサイズ n_i をより小さくする操作を試みる.

式末尾の $P(n)$ は n の多項式であり, 問題分割に要する時間計算量の上界を表す.

このような基本的不等式を用いて, アルゴリズムの実行過程に伴う場合分けを行うことにより, アルゴリズムの計算量上界を求める. 指針としては単純なものである.

分岐アルゴリズムは分枝規則と限定 (縮小) 規則により構成される. 従ってその改良は, アルゴリズムにより有効な規則を導入することにより達成できる. しかしなが

ら、これらの規則の追加は、結果的にアルゴリズムを非常に複雑なものとする。実際に Tarjan-Trojanowski のアルゴリズムは、アルゴリズム中に多くの条件式を含む長大なものであった。

Tarjan-Trojanowski の後、分岐アルゴリズムを用いた研究が続けられ、最大クリーク問題の理論的計算量上界は着実に改善されてきた。

1986 年, Jian は $O(2^{0.304n})$ -時間 [54] なる結果を示した。

同じく 1986 年, Robson は $O(2^{0.296n})$ -時間 [93] なる結果を示した。この結果はアルゴリズムの計算領域が、入力の多項式サイズとなる場合のものである。Robson は同文献 [93] において、計算領域が入力の指数関数となる場合において $O(2^{0.276n})$ -時間なる結果も示した。

1999 年, Beigel は $O(2^{0.290n})$ -時間 [8] なる結果を示した。

2001 年, Robson は指数領域アルゴリズムの評価として、 $O(2^{n/4})$ -時間 [94] (Technical Report) なる結果を示した。

これらの研究におけるアルゴリズムの計算量上界は、いずれも上記の不等式における問題数 r , 及び各部分問題 n_1, n_2, \dots, n_r のサイズにより決定される。即ち、計算量評価は与えられた問題のサイズ n を唯一の基準とする。

2006 年, Fomin らは文献 [37] において、最大独立節点集合問題に対して Measure & Conquer 法という新たな解析手法を適用した。即ち、従来手法による計算量上界が $O^*(2^{0.406n})$ -時間である単純な最大独立節点集合抽出アルゴリズムに Measure & Conquer 法による解析を適用することにより、全く同一のアルゴリズムに対して $O^*(2^{0.298n})$ -時間なる改善した解析結果を得た [37]。同じく, Fomin らは上記結果を更に改善し、 $O^*(2^{0.288n})$ -時間なる結果を示した [37]。この結果は、純粋に解析手法のみの改善による評価結果である。

Measure & Conquer 法 [37], [38], [39] は、以下の様な観点に基づく解析手法である。

例えば、グラフ $G = (V, E)$ に対する最大クリーク問題において、与えられた問題 $W \subseteq V$ を、問題中のある節点 v を含む / 含まない という規則に従い分割する。このとき問題は $W - \{v\}$ と、 W 中の節点のうち v に隣接する節点の集合 $W' \subseteq W$ とに分割される。この場合、上記不等式は、

$$T(|W|) \leq T(|W - \{v\}|) + T(|W'|) + P(|W|)$$

となる (ここで, $|W - \{v\}| \leq |W| - 1$, $|W'| \leq |W| - 1$). 節点数 n を計算量評価の基準とする場合, この不等式における節点 選択 / 非選択の操作による影響は, 部分問題サイズの減少のみである.

しかしながら, この節点選択は, 問題のサイズを減少させる以外にも問題に影響を与えている. 例えば, 節点 v を問題から取り除くことにより, v に隣接する節点の次数はそれぞれ 1 だけ減少する. アルゴリズムに節点の次数を基準とした限定 (縮小) 規則が導入されている場合には, この次数減少により新たに限定 (縮小) 規則が適用可能な節点が発生し得る. 従って, 節点選択は問題サイズの直接的な減少以外にも, 問題に対する 計算量上界 に影響を与える.

このような観点から, Measure & Conquer 法においては, 測度 (measure) という新たな概念を導入する. 測度は, 分岐アルゴリズムにおける, 各規則の適用による影響を複数の観点から評価するためのパラメーターである. この評価手法においては, より効果の大きい各規則の導入に加えて, 各操作による影響を適切に反映した測度を設定することも, 評価において重要である.

Measure & Conquer 法により, 従来手法により計算量上界が示されたアルゴリズムと比較して単純なアルゴリズムであっても, 解析手法によってはより良い解析上の計算量上界が得られる可能性があることが示された. 近年, 同手法を導入したアルゴリズムの評価結果がいくつか示されている.

2009 年, Kneis らはグラフの最大次数が低い場合には計算機を用いた解析を適用し, それ以外の場合には従来手法による詳細な解析を行うという手法を用いて, $O^*(1.2132^n) = O^*(2^{0.2789n})$ -時間なる結果を示した [62].

2010 年, Bourgeois らは Measure & Conquer 法による解析過程において, Bottom-up 法と呼ぶ, 解析における計算量上界の値を改善する手法を導入し, これにより $O^*(1.2127^n) = O^*(2^{0.2783n})$ -時間なる結果を示した [16].

2012 年, この結果は同じく Bourgeois らにより, $O^*(1.2114^n) = O^*(2^{0.2767n})$ -時間へと更に改良された [17].

2013 年, Xiao-Nagamochi はグラフの最大次数, 及び最小次数の値によって適用する

アルゴリズムを使い分けるという手法に加えて、新たな分枝規則を導入し、これにより $O^*(1.2002^n) = O^*(2^{0.2633n})$ -時間なる結果を示した [116]. これは、一般グラフにおける最大クリーク抽出問題に対する計算量上界としては、現在最良の査読済み結果である. (この結果は、その後更に $O^*(1.1996^n) = O^*(2^{0.2626n})$ -時間なる結果へと改良された [117]*2)

上記, Measure & Conquer 法の導入以降における各結果は、導入以前の解析手法 (以下, 従来手法) により評価されたアルゴリズムと比較して、格段に単純なアルゴリズムにより達成されている. これは、単純なアルゴリズムの理論計算量上界を、従来手法による場合よりも正確に評価することが可能となった結果であると言える. しかしながら、全てのアルゴリズムに対して、Measure & Conquer 法による評価により、従来手法における計算量上界を改善可能であるかどうかは、必ずしも明確ではない. 実際に、ある NP 完全問題 (Edge Dominating Set Problem) に関するごく最近の定量的改善結果において、従来手法の解析によって最良の評価を出し、Measure & Conquer 法の適用可能性は今後の検討課題としているものもある [52], [53]. Measure & Conquer 法の適用可能性は問題あるいはアルゴリズムに依存するものである.

1990 年, Shindo-Tomita は文献 [95] において、文献 [98] の極大クリーク全列挙アルゴリズム CLIQUES を基に、単純な最大クリーク抽出アルゴリズム MAXCLIQUE を提唱し、その時間計算量上界が $O(2^{n/2.863}) = O(2^{0.3493n})$ -時間であることを示した. CLIQUES [98] は、節点数 n のグラフに対して、 $O(3^{n/3})$ -時間でグラフ中の極大クリークを全列挙する. 一般に節点数 n のグラフに対して、グラフ中の極大クリークの最大数は $3^{n/3}$ 個であることが示されている [74]. 従って、CLIQUES は最大時間計算量評価においては理論的に最適なアルゴリズムである. MAXCLIQUE は CLIQUES にごく単純な限定規則のみを導入して構成されており、非常に単純な記述により表現可能であった. しかしながらその評価過程は、非常に複雑なものとなっていた. また、理論的計算量上界に関しても Tarjan-Trojanowski の結果と比較してより良いものではなかった. しかしながら、この研究においては Tarjan-Trojanowski のアルゴリズム、及び MAXCLIQUE のそれぞれを実働化した上、現実的に検証可能な範囲において計算機実験評価を行っている. 実験の結果、MAXCLIQUE は計算機実験を行った範囲において、Tarjan-Trojanowski のアル

*2 2016 年 11 月現在, arXiv における結果.

表 1.1 時間計算量 $O()$, O^* () 評価の直接的な主要結果

発表年	著者 参考文献	計算量上界	備考
1977	Tarjan, Trojanowski SIAM J. Computing [97]	$O(2^{n/3})$	
1986	Jian IEEE Computers [54]	$O(2^{0.304n})$	
同上	Robson J. Algorithms [93]	$O(2^{0.296n}), O(2^{0.276n})$	$O(2^{0.276n})$ は指数領域
1990	Shindo, Tomita Syst. & Comput. Japan [95]	$O(2^{n/2.863}) = O(2^{0.3493n})$	アルゴリズムを実働評価
1999	Beigel SODA 1999 [8]	$O(2^{0.290n})$	
2001	Robson Tech. Rep. Université Bordeaux [94]	$O(2^{n/4})$	指数領域 (Technical Report)
2006	Fomin ら SODA 2006 [37]	$O^*(2^{0.288n})$	Measure & Conquer 法を適用
2009	Fomin ら J. ACM [38]	$O^*(2^{0.287n})$	
2009	Kneis ら FSTTCS 2009 [62]	$O^*(1.2132^n) = O^*(2^{0.2789n})$	Measure & Conquer 法
2010	Bourgeois ら SWAT 2010 [16]	$O^*(1.2127^n) = O^*(2^{0.2783n})$	Measure & Conquer 法
2012	Bourgeois ら Algorithmica [17]	$O^*(1.2114^n) = O^*(2^{0.2767n})$	Measure & Conquer 法
2013	Xiao, Nagamochi ISAAC 2013 [116]	$O^*(1.2002^n) = O^*(2^{0.2633n})$	Measure & Conquer 法 * 現在, 査読済みの最良結果
同上	Xiao, Nagamochi arXiv [117]	$O^*(1.1996^n) = O^*(2^{0.2626n})$	Measure & Conquer 法, (arXiv)

ゴリズムより大幅に高速に動作することが示されている。このように、理論的手法によるアルゴリズムの評価の優劣は、必ずしもアルゴリズムの実働上の優劣を示すものではない。本論文におけるアルゴリズムは、MAXCLIQUE と同じく CLIQUES を基にして構成されており、この意味において文献 [95] の内容を継承、発展させたものである。(以上、理論的手法における時間計算量 $O()$, $O^*()$ 評価の、指数値改善の主要な結果を表 1.1 にまとめる。)

最大クリーク問題、あるいは極大クリーク全列挙問題に対するアルゴリズムの性能を評価するもう 1 つの手法は、実験的手法である。即ち、与えられた同一のグラフに対して、実際にプログラム化した複数のアルゴリズムを用いて計算機上で実験を行い、抽出に要する実行時間の大小によりアルゴリズムの性能を比較する手法である。

NP 完全問題、NP 困難問題は問題サイズの増加に伴い、その計算量が爆発的に増大するという性質を持っている。従ってあるサイズ以上の問題に関しては、妥当な時間内で最大クリーク抽出を実行することは不可能となる。故に、この手法において比較できるアルゴリズムの性能評価は、現在存在する計算機により妥当な時間で実行可能な問題に対してのものとなる。

この手法における、最大クリーク問題、あるいは極大クリーク全列挙問題に関する研究は、主としてグラフの節点間にある確率により枝を生成したランダムグラフ、及び文献 [56] において提示された、最大クリーク問題に対する DIMACS ベンチマークグラフを対象として計算機実験が行われ、着実な進歩が達成されてきた ([11], [89], [100], [67], [102], [103], [33], [34], [35], [104], [68], [105], [112], [107] など)。

分岐アルゴリズムにおける分枝、あるいは限定 (縮小) 規則には、理論的計算量評価において非常に多くの場合分けを発生させるものがあり、このような規則を導入したアルゴリズムは妥当な計算量上界を求めることが極めて困難となる。従って、理論的手法においてアルゴリズムに導入する各規則は、評価可能なものであることが求められる。

これに対して、実験的手法において導入される各規則は、アルゴリズムの正当性が保証される範囲内であれば自由に導入することが可能である。また、結果は実際の計算時間により評価されるため、この手法により優れていると判定されたアルゴリズムは、上記の応用分野における問題に対してそのままより有効な手法となり、各分野においてそれぞれ有

益な結果が得られている。

以上の様に、最大クリーク問題に対するアルゴリズムは、理論、実験の両面から着実な進歩が達成されてきている。

ところで、ある NP 完全問題に対して、扱う入力に制約条件を課すことにより、一般の入力に対する計算量上界とは異なる上界を得ることが可能である。

上記の研究と平行して、最大クリーク問題 (或いは、最大独立節点集合問題) においても、このような制約条件、及び条件に伴う計算量上界に関しては多くの研究がなされている。

計算量上界が入力サイズの指数関数となる場合においては、特に最大独立節点集合問題に関して、グラフの次数に対して制約条件を課す研究が広く行われている [57], [26], [27], [41], [92], [12], [13], [14], [15], [113], [114], [17], [50], [115].

また、グラフに対して更に強い制約条件を加えることにより、計算量上界を入力サイズの多項式とすることが可能となる場合も知られており、このような制約条件に関する研究は古くから行われてきた。

先ず、制約条件として対象を特殊グラフに限定した結果が多く示されており、最大クリーク問題は次の様な特殊グラフにおいては多項式時間で可解であるとの結果が得られている：平面グラフ [42], [29], コーダルグラフ [43], コンパラビリティグラフ [36], サークルグラフ [44], サーキュラーアークグラフ [9], [45], および円筒上長方形交グラフ [60].

また、最大独立節点集合問題についても、次の様な特殊グラフにおいては多項式時間的に可解であるとの結果が得られている： コーダルグラフ [43], サークルグラフ [44], サーキュラーアークグラフ [45], コンパラビリティグラフ [46], クローフリーグラフ [72], ダイアモンドフリーグラフ [76], およびフォークフリーグラフ [2].

一方で、一般グラフに対する制約条件に関しては、以下の結果が示されていた。

先ず、一般グラフの最大次数 Δ が定数である場合には、最大クリーク問題は多項式時間的に可解であること (の結果のみ) が文献 [42] に示されている。具体的には、節点数 n のグラフに対して $O(n \log n)$ -時間で可解との結果が発表されている [71]. なお、グラフの最大次数が定数である場合には、最大クリーク問題は $O(n)$ -時間で可解であることを示すことができる。

また、 $k = 3$ なる最大クリーク決定問題は $O(m^{1.41})$ -時間で実行可能である [3]. ここで

m はグラフの枝の数であり, 決定問題とはサイズ k 以上の最大クリークが存在するか否かを決定する問題のことである. 最大次数 Δ が高々 2 である場合には, 最大独立節点集合問題は極く単純に多項式時間的に可解である [42].

2010 年, 筆者らは関連論文 (1) (文献 [78]) において, 文献 [102] の極大クリーク全列挙アルゴリズム CLIQUES を基にした, ごく単純な最大クリーク抽出アルゴリズムを提唱した. このアルゴリズムを用いて, 関連論文 (1) においては, 最大クリーク問題の多項式時間的に可解性に関する次の結果を示した.

「節点数 n , 最大次数 $\Delta \geq 1$ なるグラフに対して, Δ が $\Delta \leq 2.495d \lg n$ ($d \geq 0$:定数) を満たすならば, 最大クリーク問題は $O(n^{2+d})$ -時間で可解である.」

即ち, 与えられたグラフの最大次数が, グラフの節点数の対数オーダーである場合, 最大クリーク問題は多項式時間的に可解である. ここで, 定数 d の値は標準的には $d = 1$ を想定している. 従って, この結果における多項式のオーダーは標準的には $O(n^3)$ である. このような, 一般グラフに対するより自然な制約条件を定量的に明確に示したのは, 同文献が初めてである.

同文献におけるアルゴリズムの, 次数制限のない一般グラフに対する理論計算量評価結果は $O(2^{0.40079n})$ -時間 であり, Measure & Conquer 法を用いない解析における, Fomin らのアルゴリズム [37] の $O^*(2^{0.406n})$ -時間なる結果とほぼ同一である.

筆者らはその後, 関連論文 (1) のアルゴリズムに対して全く新規な独自の手法による改良を行い, 関連論文 (2)-(5) において上記の制約条件を順次緩和してきた. 具体的には以下の通りである.

2011 年, 関連論文 (2) (文献 [80]) においては, 上記結果における定数値 2.495 を, 2.613 に改善した. また, 多項式オーダーについても, $O(n^{2+d})$ を $O(n^{1+d})$, 即ち標準設定において $O(n^2)$ へと改善した. これは, 関連論文 (1) においてグラフを隣接行列により表現していたものを, 隣接リストに改めたことによる.

同文献におけるアルゴリズムの, 次数制限のない一般グラフに対する理論計算量評価結果は $O(2^{0.3827n})$ -時間 である.

同じく 2011 年, 関連論文 (3) (文献 [81]) においては, 関連論文 (1) におけるアルゴリズムに, 関連論文 (2) の手法を拡張した改良を行い, 関連論文 (2) における定数値 2.613 を, 2.773 に改善した.

同文献におけるアルゴリズムの、次数制限のない一般グラフに対する理論計算量評価結果は $O(2^{0.3606n})$ -時間 である。

2012年、関連論文(4) (文献[82])においては、更に新規の手法による改良を行い、関連論文(3)における定数値 2.773 を、2.994 に改善した。

同文献における改良アルゴリズムの、次数制限のない一般グラフに対する理論計算量評価結果は $O(2^{0.334n})$ -時間 である。

2014年、関連論文(5) (文献[83])においては、関連論文(4)におけるアルゴリズムに更なる改良を行い、次の結果を得ている。

「節点数 n 、最大次数 $\Delta \geq 1$ なるグラフに対して、 Δ が $\Delta \leq 3.177d \lg n$ ($d \geq 0$:定数) を満たすならば、最大クリーク問題は $O(n^{1+d})$ -時間で可解である。」

同文献におけるアルゴリズムの、次数制限のない一般グラフに対する理論計算量評価結果は、 $O(2^{0.3147n})$ -時間である。

ここで、各関連論文における計算量評価は、いずれも Measure & Conquer 法によらず、従来手法によって行っている。従って本論文におけるアルゴリズムの改良は、アルゴリズムの解析上の改善ではなく、アルゴリズム自身の改良によるグラフに対する探索領域の削減により達成されている。

ここで一般に、節点数が n の一般グラフにおいて、理論計算量評価結果が、

$$O^*(2^{cn})\text{-時間} \quad (0 < c < 1 : \text{定数})$$

であるアルゴリズムをサブルーチンとして用いれば、グラフ中の最大クリークは

$$\Delta \leq 1/(c + \alpha) \cdot d \lg n \quad (\alpha > 0: \text{定数})$$

なる条件のもとに、 $O(n^{1+d})$ -時間で抽出可能である。(ここで、正定数 α は、 $O^*(2^{cn})$ なる上界を、 $O()$ 表記に変形するときに生じる定数である。 α の具体的な決定方法は、第2章、2.2節において述べる。)

従って、関連論文(5)よりも計算量評価結果の優れた(即ち、 c の値がより小さい)アルゴリズムをサブルーチンとして用いることにより、より大きな定数値 $1/(c + \alpha)$ を得ることが可能である。即ち、現在知られている査読済みで最も大きな定数値の値は、文献[116]のアルゴリズムを用いた場合における、

$$1/(0.2633 + \alpha)$$

である。(具体的な適用例を、第3章、3.6節において述べる)

関連論文 (1)-(5) における, 定数値 c の値の改善は, いずれもグラフの最大次数を節点数の対数オーダーとする制約条件の下での結果であり, これは文献 [116] 等のアルゴリズムをサブルーチンとして用いた場合においても同様である.

これに対して, グラフの節点数 n が十分大きい場合においては, 定数 c の値の大小よりもグラフの次数に関する制限対象範囲を緩和することが, 相対的により重要となる.

また, 最大クリーク問題の実際の適用例においては, グラフ中の節点の次数は必ずしも均一ではなく, 次数の大きい節点と小さい節点が混在し得る. 従って, グラフの次数に関する制限対象範囲の緩和は, 実用的観点からも重要である.

これらの観点から, 関連論文 (5) に関しては更に, 上記の条件を緩和した 次の「拡張結果」を得ている.

「一般グラフ $G = (V, E)$ に対して, $(v, w) \in E$ なる任意の 2 節点 $v, w \in V$ の次数 $deg(v)$ 及び $deg(w)$ が次の条件を満たすとする.

$$\min\{deg(v), deg(w)\} \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

このとき, G に対する最大クリーク問題は $O(n^{2+\max\{1,d\}})$ なる多項式時間的に可解である。」

これは, 隣接 2 節点に対する前提条件について, 次数の大きい方の節点に関しては全く無条件でよいとしたものである. この新たな制限のもとでは, 次数の非常に大きい節点の存在がある程度許される. 従って, 前提条件の緩和としては, 3.177 等の定数の定量的増大以上に, 定性的に強力な新しい枠組を与えるものである. (この枠組自体は, 2011 年, 文献 [79] において筆者らにより初めて示された. 同文献においては上記結果の定数値は 2.994 であった.)

この「拡張結果」に対しても, 同様にして優れた理論計算量上界を持つアルゴリズムをサブルーチンとして組み合わせることにより, 定数 3.177 部分を大きくした, より一層条件緩和された拡張結果を得ることが出来る. 従って, 本論文における主要な成果は, このような多項式時間的可解性の新規な枠組を提示したことにある. また, 本論文において提示した, アルゴリズムの理論計算量上界を改善する手法は本論文独自の新規なものであり, 今後この手法の拡張により更なる理論計算量上界の改善が期待できるものともなっている.

このように上記結果の意義は, 最大クリーク問題の多項式時間的可解性における新たな

枠組を提示したことにある。ここで、上記結果における多項式オーダーは、 $0 \leq d \leq 1$ においては $O(n^3)$ 、 $d > 1$ においては $O(n^{2+d})$ である。従って、グラフ全体の最大次数に対して制限を設けた場合よりも 1 増加している。本論文においては、この結果における多項式オーダーを改善した、次の改良結果も得た。

「一般グラフ $G = (V, E)$ に対して、 $(v, w) \in E$ なる任意の 2 節点 $v, w \in V$ の次数 $\deg(v)$ 及び $\deg(w)$ が次の条件を満たすとする。

$$\min\{\deg(v), \deg(w)\} \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

このとき、 G に対する最大クリーク問題は $O(n^{\max\{2, 1+d\}})$ なる多項式時間的に可解である。」

この結果においては、 $0 \leq d \leq 1$ である場合に $O(n^2)$ 、即ちグラフ全体の最大次数に制限を設けた場合と同じ多項式オーダーとなる。

本論文においてはこの結果を基に、上記枠組の更なる拡張となる、次の結果も与える。

「サイズ $n_0 \geq 2$ なる任意の連結な誘導部分グラフ $G(C)$ ($C \subseteq V$) に対して、 C 中の最小次数節点 v が、

$$\deg(v) \leq 3.177d \lg n \quad (d \geq 0 : \text{定数})$$

を満たすとき、最大クリーク問題は $O(n^{\max\{2, 1+d\}})$ の多項式時間で可解である。」

例えば、上記隣接 2 節点における次数の制限を、サイズ 3 の連結な誘導部分グラフ中の最小次数節点に対する制限とする。このとき、グラフ中には次数が非常に大きい次数のみで構成された隣接節点对が存在するような場合が許される。従って、このような $n_0 = 3, 4, \dots$ との拡張は、次数の制限を更に明確に拡張したものとなる。

1.2 本論文の概要

本論文は、8 つの章で構成されている。第 1 章は序章であり、研究の背景、位置づけ、目的、従来の研究状況、本研究の基礎的概念、および、本論文の章構成について述べている。

第 2 章においては、本論文で共通的に使用する、用語の定義、および記法について述べる。また本論文を通じて最大時間計算量評価の過程で用いられる基本的補題についても、ここで示す。

第 3 章においては、極大クリーク全列挙アルゴリズム CLIQUES [98], [99], [101], [102]

を基にした, 基本的な最大クリーク抽出アルゴリズム MCP_0 を提唱し, このアルゴリズムの最大時間計算量評価を行い, その結果を用いて, 最大クリーク問題における多項式時間的可解性の定量的条件を示す. 以降, 第 4-7 章においては, この単純な基本アルゴリズムに順次新たな探索領域削減手法を導入し, これに伴うより改善された多項式時間的可解性の条件を与えていく.

第 4 章においては, 第 3 章のアルゴリズム MCP_0 に対して, “部分問題の統合による探索領域削減” という新たな手法を導入した改良アルゴリズム MCP_1 を提唱し, 改善された多項式時間的可解性の結果を示す.

第 5 章においては, 第 4 章のアルゴリズム MCP_1 において導入した, “部分問題の統合による探索領域削減” を更に詳細に検討し, これを反映した手法を導入した改良アルゴリズム MCP'_1 を提唱し, これを用いてより改善された多項式時間的可解性の結果を示す.

第 6 章においては, 第 3 章のアルゴリズム MCP_0 に対して, 第 4 章において導入, 第 5 章において更に詳細化した

“部分問題の統合による探索領域削減” に加えて, “部分問題サイズの減少”, 及び “少数部分問題の分割” という新たな 2 種の探索領域削減手法を導入した改良アルゴリズム MCP_2 を提唱し, これを用いてさらに改善された多項式時間的可解性の結果を示す.

第 7 章においては, 第 6 章のアルゴリズム MCP_2 に対して, 第 6 章までにおいて提唱した手法をそれぞれ拡張した結果を与える. これに加えて, “節点の隣接関係による限定” という全く新しい手法を与えることにより, 本論文の最終的な結論の基礎となる多項式時間的可解性の結果を示す.

第 8 章においては, 第 7 章において示した定量的結果を用いて, 本論文の最終的結論となる頭書の結果を定理として示す.

最後の第 9 章においては, 本論文の結果を総括し, 合わせて今後の展望を述べている.

第 2 章

事前準備

2.1 本論文の各章共通の定義と表記法

定義 2.1.1 本論文においては, 自己閉路のない単純無向グラフ $G = (V, E)$ を対象とする. ここで, V は節点集合, $E \subseteq V \times V$ は枝集合である.

$G = (V, E)$, 及び $V' \subseteq V$ に対して, グラフ,

$$G' = (V', E \cap (V' \times V'))$$

を, G の V' による誘導部分グラフという. この場合, 単に $G(V')$ とも書く.

なお, 本論文を通じて与えられたグラフの表現 (データ構造) は 隣接リスト を用いて行うものとする.

以下, 定義 2.1.2 - 2.1.8 を通じて, V はグラフ $G = (V, E)$ の節点集合 V を表す.

定義 2.1.2 $W \subseteq V$ なる順序付き節点集合 W が,

$$W = \{w_1, w_2, \dots, w_{|W|}\}$$

と表されるとき, $1 \leq i \leq j \leq |W|$ に対して,

$$W[i, j] = \{w_i, w_{i+1}, \dots, w_j\}$$

と定義する.

定義 2.1.3 $v \in V$ なる節点 v に対して,

$$\Gamma(v) = \{u \in V \mid (u, v) \in E\} \quad (\ni v)$$

と定義する. 即ち $\Gamma(v)$ は, $G = (V, E)$ において v に隣接する節点全ての集合である.

また,

$$\deg(v) = |\Gamma(v)|$$

を, 節点 v の G における次数, 又は単に v の次数という.

G の誘導部分グラフ $G(V')$ ($V' \subseteq V$) に対して,

$$|\Gamma(v) \cap V'|$$

を, 節点 v の $G(V')$ における次数という.

定義 2.1.4 $W \subseteq V$ なる順序付き集合 $W = \{w_1, w_2, \dots, w_{|W|}\}$ に対して,

$$\hat{\Gamma}(W) = \Gamma(w_1) \cap \Gamma(w_2) \cap \dots \cap \Gamma(w_{|W|})$$

と定義する. 即ち, $\hat{\Gamma}(W)$ は W に含まれる全ての節点に 共通して隣接 する節点の集合である.

定義 2.1.5 $V' \subseteq V$ なる順序付き節点集合を,

$$\{v'_1, v'_2, \dots, v'_{|V'|}\}$$

とする. V' において,

$$(v'_i, v'_{i+1}) \in E \quad (1 \leq i \leq |V'|)$$

が成立するとき, $G(V')$ を節点 v'_1 から節点 $v'_{|V'|}$ への道という. この場合, 単に V' は道であるともいう.

定義 2.1.6 節点集合 $C \subseteq V$ に対する誘導部分グラフ $G(C)$ において, $v, w \in C$ なる任意の2節点に対して, 節点 v から節点 w への道が存在するとき, $G(C)$ は連結であるという.

定義 2.1.7 節点の集合 $Q \subseteq V$ に対する誘導部分グラフ $G(Q)$ において,

$$\text{任意の } v, w \in Q \ (v \neq w) \text{ に対して } (v, w) \in E$$

が成立するとき, $G(Q)$ はクリークであるという. この場合, 単に Q はクリークであるともいう. あるクリークがグラフ中の全てのクリークに対して, 自分以外のどのクリークの部分グラフでもないとき, Q は極大クリークであるという. 節点数が最大のクリークを最大クリークという.

節点の部分集合 $W \subseteq V$ による誘導部分グラフ $G(W)$ 中の最大クリークのサイズ (節点数) を $\omega(W)$ で表す.

定義 2.1.8 本論文における最大クリーク問題は, 与えられたグラフ G および正整数 k に対して, G が節点数 k 以上の最大クリークを含むか否かを決定する決定問題として定義される.

2.2 本論文各章共通の基本的補題

以下に示す各補題は, 第3-7章の各章を通してアルゴリズムの最大時間計算量評価で用いられる基本的な補題である.

補題 2.2.1 α を $0 < \alpha < 0.00001$ なる正定数, 定数 m を正整数とする. また, 正定数 $C' > 0$ を,

$$C' \geq \frac{m!}{(\ln 2^\alpha)^m}$$

を満たす定数として定める. このとき, 任意の正整数 η において,

$$\eta^m < C' 2^{\alpha \eta}.$$

(証明) いま, 実数 x 上の関数 $f(x)$ を

$$f(x) = C' 2^{\alpha x} - x^m$$

と定義するとき, $1 \leq i \leq m$ に対して,

$$(2^{\alpha x})' = \ln 2^\alpha 2^{\alpha x},$$

及び,

$$(x^m)^{(i)} = m(m-1)(m-2) \cdots (m-i+1)x^{m-i},$$

により,

$$f^{(i)}(x) = C' (\ln 2^\alpha)^i 2^{\alpha x} - {}_m P_i x^{m-i},$$

$$f^{(i)}(0) = C' (\ln 2^\alpha)^i > 0.$$

ここで定数 C' の設定により,

$$\begin{aligned} f^{(m)}(x) \\ = C' (\ln 2^\alpha)^m 2^{\alpha x} - m! x^{m-m} \end{aligned}$$

$$\begin{aligned} &\geq C'(\ln 2^\alpha)^m - m! \\ &\geq 0. \end{aligned}$$

従って, $x \geq 0$ において $f^{(m)}(x) \geq 0$ であり, $f^{(m-1)}(x)$ は $x \geq 0$ において単調非減少となる. 以下同様に,

$$f^{(i)}(0) > 0, \quad (1 \leq i \leq m),$$

かつ,

$$f^{(m-2)}(x), \dots, f^{(1)}(x)$$

は全て単調非減少により, $f(x)$ は $x \geq 0$ において単調非減少である.

更に, $f(0) > 0$ であることより, $x \geq 0$ において $f(x) > 0$ である. 従って, $\eta \geq 1$ において,

$$\eta^m < C'2^{\alpha\eta}. \quad \square$$

補題 2.2.1 は, 本論文第 3-7 章において提示する各アルゴリズムに対して,

$$O(2^{c\eta}\eta^m) \quad (\eta \geq 1)$$

なる計算量上界を示したとき, これを,

$$O(2^{(c+\alpha)\eta})$$

なる, $O()$ 内の関数が指数関数のみである形に変形するとき用いる.

以降, 本論文において提示する各アルゴリズムに対して, 補題 2.2.1 を適用するとき,

$$\alpha = 0.0003, \quad m = 3, \quad \text{及び},$$

$$C' = 6.7 \cdot 10^{11} \left(> \frac{3!}{(\ln 2^{0.0003})^3} \right)$$

としている.

補題 2.2.2 任意の $n \geq 1$ において,

$$(\lg n)^2 < (\lg e)n.$$

(証明) いま実数 x 上の関数 $f(x)$ を,

$$f(x) = (\lg e)x - (\lg x)^2$$

と定義する. まず, $0 < f(1) < f(2) < f(3) < f(4)$ である.

$$f'(x) = \lg e \left(1 - 2 \cdot \frac{\lg x}{x} \right),$$

$$\begin{aligned}
f''(x) &= 2\lg e \cdot \frac{\lg x - \frac{\lg e}{x} \cdot x}{x^2} \\
&= 2\lg e \cdot \frac{\lg x - \lg e}{x^2} \\
&= 2\lg e \cdot \frac{\lg e \cdot \ln x - \lg e}{x^2} \\
&= 2(\lg e)^2 \cdot \frac{\ln x - 1}{x^2}.
\end{aligned}$$

ここで、 $f''(x) > 0$ が $x \geq 4$ において成立するから、 $f'(x)$ は $x \geq 4$ において単調非減少である。更に $f'(4) = 0$ であるから、 $x \geq 4$ において $f'(x) \geq 0$ が成立する。従って $f(x)$ は $x \geq 4$ において単調非減少である。即ち $n \geq 5$ に対して題意は成立する。

以上から、任意の $n \geq 1$ に対する題意の成立は示された。□

次の補題は、本論文のアルゴリズムにおける探索領域削減操作のうち、最も単純なもの
の基礎である。

補題 2.2.3 グラフ $G = (V, E)$ に対して、 $W', W \subseteq V$ として、

$W' \subseteq W$ であるならば、

$$\omega(W') \leq \omega(W).$$

(証明) 最大クリークの定義から明らか。□

補題 2.2.3 の拡張として次の補題を得る。

補題 2.2.4 グラフ $G = (V, E)$ に対して、 $W, W' \subseteq V$ として、

$$R := W' - W$$

とおく。このとき以下が成立する。

$$\omega(W') > \omega(W)$$

であるならば、

W' 中の最大クリークは R 中の節点を少なくとも 1 個含む。

(証明) 対偶、即ち、 W' 中の最大クリークが R 中の節点を含まないならば、

$$\omega(W') \leq \omega(W).$$

を示す。

いま, Q を W' 中の任意の最大クリークとして,

$$Q \cap R = \emptyset$$

であるとする.

R は W に含まれない W' 中の節点の集合であるから,

$$Q \subseteq W.$$

従って, 補題 2.2.3 により,

$$\omega(W') = |Q| \leq \omega(W). \quad \square$$

更に, 次の補題が成立する.

補題 2.2.5 一般に, グラフ $G = (V, E)$ における 2 つの節点部分集合 $SG, SG' \subseteq V$ に対して,

$$R := SG - SG'$$

とおく. このとき以下が成立する.

(1) $|R| \leq 1$ であるならば,

$$\omega(SG) \leq \omega(SG') + 1.$$

(2) $|R| \geq 2$ であり,

$$\omega(SG) > \omega(SG') + 1$$

が成立する場合, SG 中の任意の最大クリークは, R 中の節点を少なくとも 2 個含む.

(証明) (1) いま $R \subseteq SG$ であるから,

$$\begin{aligned} & \omega(SG) \\ & \leq \omega(SG - R) + \omega(R) \\ & \leq \omega(SG - R) + 1 \leq \omega(SG') + 1. \quad (\text{補題 2.2.3 による.}) \end{aligned}$$

(2) 対偶を証明する. 即ち, SG 中の任意の最大クリーク Q に対して,

$$|Q \cap R| \leq 1$$

であるならば,

$$\omega(SG) \leq \omega(SG') + 1$$

である事を示す.

$$\omega(SG)$$

$$\begin{aligned} &= \omega(SG - R) + \omega(Q \cap R) \\ &\leq \omega(SG - R) + 1 \\ &\leq \omega(SG') + 1. \quad \square \end{aligned}$$

第 3 章

アルゴリズム MCP_0

3.1 概要

本章の構成は、以下の通りである。

まず、3.2 節において、基本的な最大クリーク抽出アルゴリズムを提示し、このアルゴリズムの構成、及び、基本的な探索手法について述べる。

続く 3.3 節、及び 3.4 節においては、アルゴリズムに導入するごく基本的な探索領域削減手法について述べる。

3.5 節においては、基本アルゴリズムを実際の例題に適用し、その具体的な動作内容を示す。

最後に、3.6 節において基本アルゴリズムの最大時間計算量評価を行い、その結果を用いて、本章の結論となる多項式時間的可解性の結果を示す。

以上の構成は、本章に続く第 4 章以降の各章において共通であり、計算量評価の手法も同様である。従って本章の内容は、単純な最大クリーク抽出アルゴリズムを用いて、その内容の説明、及び解析の基本的な構成を示したものである。

3.2 アルゴリズム MCP_0

本章において説明及び解析を行う、最大クリーク抽出アルゴリズム MCP_0 をアルゴリズム 1 に示す。

アルゴリズム MCP_0 は、グラフ $G = (V, E)$ を入力とし、 G 中の最大クリークのサイズ

アルゴリズム 1 MCP_0

```

0000: procedure  $MCP_0(G)$ 
0100: global  $Q\_size, Qmax\_size;$ 
0200: begin
0300:  $Q\_size := 0, Qmax\_size := 0;$ 
0400: EXPAND( $V$ )
0500: return  $Qmax\_size$ 
0600: end {of  $MCP_0$ }

0700: procedure EXPAND( $SUBG$ )
0800: begin
0900: if  $SUBG \neq \emptyset$  then
1000:    $u :=$  a vertex in  $SUBG$  that
           maximizes  $|\Gamma(u) \cap SUBG|;$ 
1100:    $SUBG_u := \Gamma(u) \cap SUBG;$ 
1200:    $EXT_u := SUBG - \{u\} - SUBG_u;$ 
1300:    $k := |EXT_u|;$ 
1400:   let  $EXT_u = \{v_1, v_2, \dots, v_k\};$ 
           1500:    $Q\_size := Q\_size + |\{u\}|;$ 
           1600:   EXPAND( $SUBG_u$ )
           1700:    $Q\_size := Q\_size - |\{u\}|;$ 
           1800:   for  $i = 1$  to  $k - 1$  do
           1900:      $v_i :=$  the first vertex in  $EXT_u;$ 
           2000:      $SUBG_{v_i}$ 
             :=  $\Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u);$ 
           2100:      $Q\_size := Q\_size + |\{v_i\}|;$ 
           2200:     EXPAND( $SUBG_{v_i}$ )
           2300:      $Q\_size := Q\_size - |\{v_i\}|$ 
           2400:   od
           2500:   else {i.e.,  $SUBG = \emptyset$ }
           2600:     if  $Q\_size > Qmax\_size$  then
           2700:        $Qmax\_size := Q\_size$  fi fi
           2800:   end {of EXPAND}

```

を出力とする。

MCP_0 は、大きく分けて本体部分 (0000-0600 行) と、手続き **EXPAND** (0700-2800 行) により構成される。このうち本体部分の内容は、使用変数の定義等を行った後に **EXPAND** を呼び出し、結果を出力する操作である。第 4-7 章における各アルゴリズムについて、本体部分の内容は同一である。

具体的なクリークの探索は、手続き **EXPAND** により実行される。そこで以下では、**EXPAND** の探索手法について詳述する。

3.2.1 基本手続き **EXPAND**

基本手続き **EXPAND** は、 $G = (V, E)$ に対して節点集合 $SUBG \subseteq V$ を入力とする。 MCP_0 本体部分から、**EXPAND** が呼び出されるとき (0400 行)、 $SUBG = V$ である。

EXPAND は、以下の操作により入力 $SUBG$ から最大クリークを抽出する。

いま、 Q を現在探索中のクリークとする。また、 Q_{\max} をこれまでの探索の過程で得られた最大クリークとする。 $SUBG = V$ であるとき、 $Q = Q_{\max} = \emptyset$ である。実際のアルゴリズム上では、

$$Q_size := |Q|, Qmax_size := |Q_{max}|$$

と、それぞれのクリークのサイズのみを大域変数として定義する (0100 行). これは、最大クリーク問題において最大クリークはそのサイズのみ決定できれば十分であるという理由による.

EXPAND は、与えられた入力 $SUBG$ に対して、 $G(SUBG)$ における最大次数節点 u を $SUBG$ から選択し、

$$SUBG_u := \Gamma(u) \cap SUBG$$

を求める (1000-1100 行). 即ち $SUBG_u$ は、 $SUBG$ 中の節点 u に隣接する節点の集合である.

$SUBG_u$ 構成後、節点 u を Q に追加する ($Q := Q \cup \{u\}$). その後、 $SUBG_u$ を新たな入力として EXPAND を再度実行する (1500-1600 行).

この手続きを再帰的に $SUBG_u = \emptyset$ となるまで繰り返す. この繰り返し動作においては、

$SUBG_u \subsetneq SUBG$ が成立する. 従って、 $G(SUBG)$ 中の最大次数節点を u ,

$$\Delta = |SUBG_u|$$

とおくと、EXPAND の繰り返し動作は最大 Δ 回で終了する.

$SUBG = \emptyset$ となった時点において、 Q が $|Q| = Q_size > Qmax_size$ を満たすならば、 $Qmax_size$ を Q_size によって置き換える. その後バックトラックして、 u を Q から削除 (1700 行)、新たに節点 v を $SUBG$ から選択して同様の操作を $SUBG = \emptyset$ となるまで繰り返す. この基本アルゴリズムは最大クリーク抽出においてはよく知られた手法である. 一般に、探索中のクリークが $Q = \{p_0, p_1, \dots, p_d\}$ ($0 \leq d \leq |V|$) と表されるとき、探索候補節点集合は

$$SUBG = V \cap \Gamma(p_0) \cap \Gamma(p_1) \cap \dots \cap \Gamma(p_d)$$

となる. 探索中のクリークが

$$\{p_0, p_1, \dots, p_{d-1}\}$$

であるとき、EXPAND の次の再帰における候補節点集合 $SUBG$ は、次に選択される節点 p_d によって定まる. そこで、以下本論文を通じて、この $SUBG$ に対して手続き EXPAND を実行することを、節点 p_d からの探索 を行うと言う.

3.3 部分森の同一化

手続き EXPAND においては, 与えられた入力 $SUBG \subseteq V$ に対して, まず以下のような節点の並べ替え処理を行う.

節点 u を $G(SUBG)$ 中の最大次数節点として,

$$SUBG_u := \Gamma(u) \cap SUBG$$

とする. また,

$$EXT_u := SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|)$$

とする. 即ち, EXT_u は $SUBG$ 中の節点のうち, (u 自身を除く) u に非隣接な各節点の集合である.

これらの定義のもと, $SUBG$ を,

$$\{u\} \cup EXT_u \cup SUBG_u$$

と並べ替える. この並べ替え操作後, EXPAND は節点 u からの探索, v_1 からの探索, v_2 からの探索, ... と, 左から順に 各節点からの探索を実行する.

u からの探索, 即ち,

$$SUBG_u = \Gamma(u) \cap SUBG$$

に対する EXPAND の実行により得られる全てのクリークは, EXT_u 中の各節点を含まない. そこで, u からの探索終了後は,

$$EXT_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|)$$

中の各節点からの探索を, 左から順に行う. いま,

$$SUBG_{v_1} = \Gamma(v_1) \cap SUBG$$

として, v_1 からの探索を $SUBG_{v_1}$ に対する EXPAND の実行とする.

v_1 からの探索終了後, 節点 v_1 を含む最大クリークは既に抽出済である. 従って, 節点 v_2 からの探索は,

$$SUBG_{v_2} = \Gamma(v_2) \cap (SUBG - \{v_1\})$$

に対する EXPAND の実行とする. 以下同様にして, $v_i \in EXT_u$ ($1 \leq i \leq k$) なる節点 v_i からの探索においては, 既に探索された節点 v_1, v_2, \dots, v_{i-1} は探索の候補から除外する.

即ち, $1 \leq i \leq k$ において, 節点 v_i からの探索を,

$$\begin{aligned}
& SUBG_{v_i} \\
&= \Gamma(v_i) \cap (SUBG - \{v_1, v_2, \dots, v_{i-1}\}) \\
&= \Gamma(v_i) \cap ((\{u\} \cup EXT_u \cup SUBG_u) - \{v_1, v_2, \dots, v_{i-1}\}) \\
&= \Gamma(v_i) \cap ((\{u\} \cup \{v_1, v_2, \dots, v_k\} \cup SUBG_u) - \{v_1, v_2, \dots, v_{i-1}\}) \\
&= \Gamma(v_i) \cap (\{u\} \cup \{v_i, v_{i+1}, \dots, v_k\} \cup SUBG_u) \\
&= \Gamma(v_i) \cap (\{u\} \cup EXT_u[i, k] \cup SUBG_u) \\
&= \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u). \quad ((v_i, u) \notin E)
\end{aligned}$$

に対する EXPAND の実行とする. *¹

また, u からの探索, 及び EXT_u 中各節点からの探索が実行された後, $SUBG_u$ 中の各節点からの探索で得られる全てのクリーク Q に対して, より大きいクリーク $Q \cup \{u\}$ が存在することがわかる. 何故ならば, $Q \subseteq SUBG_u = \Gamma(u) \cap SUBG$ の全ての節点は u に隣接しているからである. 従って, 節点 u からの探索, 即ち $SUBG_u$ 中の最大クリーク抽出が正しく行われた後であれば, $SUBG$ 最後部である $SUBG_u$ 中のどの節点からも新たに探索を行う必要はない. そこで, $SUBG_u$ 中各節点からの探索は, これを常に省略する.

この手法は [19] を始めとして, [58], [63] などでも用いられている. 本論文においてはこの手法を 部分森の同一化 と呼ぶ.

3.4 部分問題の単純省略

一般に, $W' \subseteq W$ である 2 節点集合 $W, W' \subseteq V$ に対して,

$$\omega(W') \leq \omega(W). \quad (\text{第2章, 補題 2.2.3})$$

$SUBG$ に対する上記並べ替えのもと, この事実に基づくごく基本的な探索領域削減操作を加える.

EXT_u 中の最後尾節点 v_k からの探索は, 上記定義により,

$$SUBG_{v_k} = \Gamma(v_k) \cap (EXT_u[k, k] \cup SUBG_u)$$

に対する EXPAND の実行である. しかしながら,

$$\begin{aligned}
& SUBG_{v_k} \\
&= \Gamma(v_k) \cap (EXT_u[k, k] \cup SUBG_u)
\end{aligned}$$

*¹ 関連論文 (4), (5) においては, $EXT_u[i, k]$ を $EXT_u[i]$ と表記していた.

$$\begin{aligned}
&= \Gamma(v_k) \cap (\{v_k\} \cup SUBG_u) \\
&= \Gamma(v_k) \cap SUBG_u \quad ((v_k, v_k) \notin E) \\
&\subseteq SUBG_u
\end{aligned}$$

である. そこで, 上記事実に基づき EXT_u の最後尾節点 v_k からの探索は省略する.

いま,

$$|SUBG| = \eta \quad *2$$

とすると,

$$\begin{aligned}
&|SUBG_u| \\
&= |SUBG| - |\{u\}| - |EXT_u| \\
&= \eta - k - 1
\end{aligned}$$

である. 以上の基本的な並べ替え, 及び探索の省略のみを導入した場合, EXPAND の実行過程においては, u, v_1, \dots, v_{k-1} (計 k 個) からの探索が実行される. 即ち, $SUBG$ は k 個の部分問題に分割され, それぞれの部分問題のサイズは高々 $\eta - k - 1$ である.

MCP_0 による最大クリーク抽出の過程は, 図 3.2 に示すように探索木の集まりである探索森として表すことができる.

3.5 例題

図 3.1 のグラフ $G = (V, E)$ に対するアルゴリズム MCP_0 の実行過程を, 図 3.2 の探索森として表す.

この探索森の根は, 常にグラフ G の節点全ての集合,

$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

である. ここで,

$$V = SUBG^0$$

と書く.

探索森において, 中括弧により示した各節点集合は, その集合の親を分割した候補節

*2 関連論文 (1)~(5) においては, $|SUBG|$ を記号 Δ を用いて表記していたが, 記号 Δ は同時にグラフの最大次数を表す記号としても用いられていた. 本論文において, 記号 Δ はグラフの最大次数を表す記号としてのみ使用し, $|SUBG|$ については記号 η を用いる.

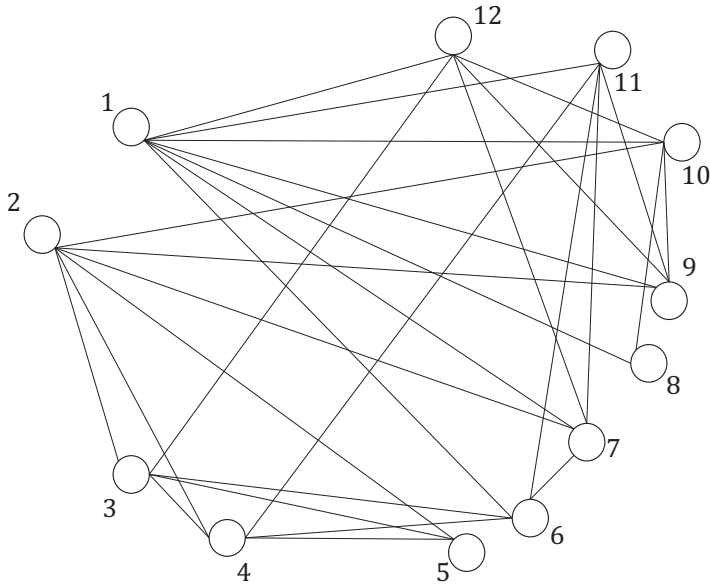


図 3.1 例題

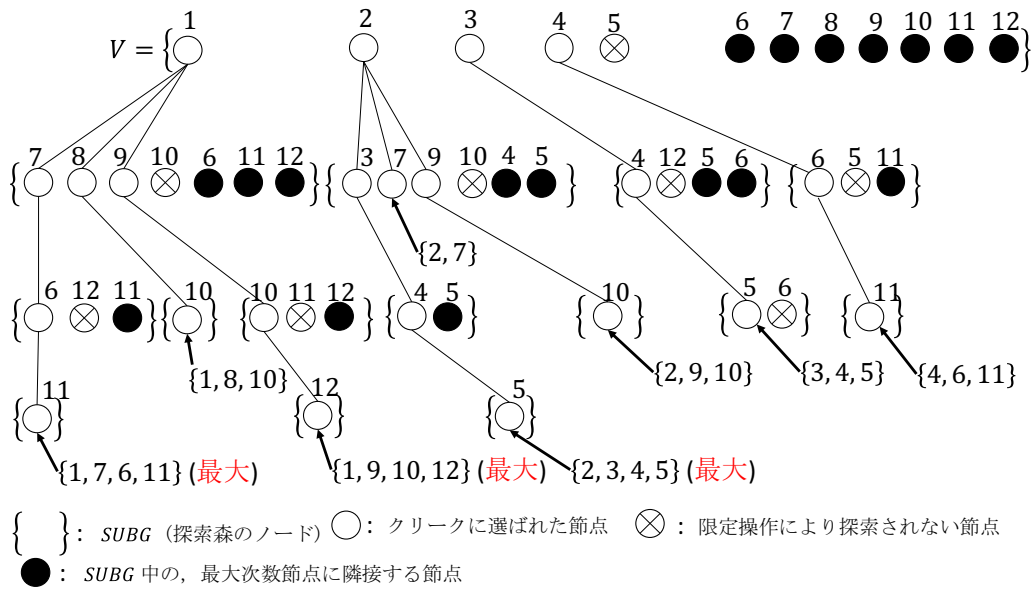


図 3.2 MCP_0 によるクリーク探索森

点集合 $SUBG$ であり、これが探索森における各ノードである。ここで、各候補節点集合 $SUBG$ に対して、 $SUBG$ の位置における探索森の深さ (depth) を d ($0 \leq d \leq |V|$) とする。このとき $SUBG$ を、

$$SUBG^d$$

とも書く.

図 3.2 中, 白丸によって示した各節点は, $SUBG$ 中の節点のうち, その節点からの探索が実行される節点である. これらの節点は, 抽出されたグラフ中のクリークのうち少なくとも 1 個に含まれる. $SUBG^0 (= V)$ 中の任意の白丸によって示された節点を p_0 とするとき, 探索森の根 $SUBG^0$ の子ノードは,

$$\Gamma(p_0) \cap SUBG^0$$

である. 以下, この節点集合を,

$$SUBG_{\{p_0\}}^1$$

と表記する. ここで, 節点 p_0 はグラフ G 中のあるクリーク Q に含まれる. 3.2.1 節に述べた通り, 一般に, 探索中のクリークが $Q = \{p_0, p_1, \dots, p_d\}$ ($0 \leq d \leq |V|$) と表されるととき, 探索候補節点集合は

$$SUBG = V \cap \Gamma(p_0) \cap \Gamma(p_1) \cap \dots \cap \Gamma(p_d)$$

となる.

以下の実行例においては, この $SUBG$ を

$$SUBG_{\{p_0, p_1, \dots, p_d\}}^{d+1}$$

と書く. 即ち, $d \geq 1$ において,

$$\begin{aligned} & SUBG_{\{p_0, p_1, \dots, p_d\}}^{d+1} \\ &= \Gamma(p_d) \cap SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d. \end{aligned}$$

図中においては, 探索中のクリークに含まれる節点 p_0 は p_1 と, 各節点 p_i ($1 \leq i \leq d-1$) はそれぞれ節点 p_{i-1} 及び p_{i+1} と, p_d は p_{d-1} と実線で結んで表記する.

ある d_0 ($0 \leq d_0 \leq |V|$) において,

$$SUBG_{\{p_0, p_1, \dots, p_{d_0}\}}^{d_0+1} = \emptyset$$

となったとき, 節点集合 $\{p_0, p_1, \dots, p_{d_0}\}$ はクリークである.

上の表記を用いて, $(V =)SUBG^0$ 中の $G(SUBG^0)$ における最大次数節点を u_0 ,

$$SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d \text{ 中の}$$

$$G(SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d) \text{ における最大次数節点を } u_d \text{ とおき,}$$

$$\begin{aligned} & SUBG_{\{p_0, p_1, \dots, p_{d-1}, u_d\}}^{d+1} \\ &= \Gamma(u_d) \cap SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d \\ &= SUBG_{u_d}^{d+1} \end{aligned}$$

とおく. このとき,

$$\begin{aligned} EXT_{u_0}^1 &= SUBG^0 - \{u_0\} - SUBG_{\{u_0\}}^1, \\ EXT_{u_d}^{d+1} &= SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d - \{u_d\} - SUBG_{u_d}^{d+1} \end{aligned}$$

と書く. また, $0 \leq d \leq |V|$ に対して,

$$EXT_{u_d}^{d+1} = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_{u_d}^{d+1}|),$$

及び,

$$EXT_{u_d}^{d+1}[i, k] = \{v_i, v_{i+1}, \dots, v_k\} \quad (1 \leq i \leq k)$$

とするとき,

$$\begin{aligned} &SUBG_{\{p_0, p_1, \dots, p_{d-1}, v_i\}}^{d+1} \\ &= \Gamma(v_i) \cap (EXT_{u_d}^{d+1}[i, k] \cup SUBG_{u_d}^{d+1}) \\ &= SUBG_{v_i}^{d+1} \end{aligned}$$

と書く. 前節に述べた通り, MCP_0 において, $EXT_{u_d}^{d+1}$ の最後尾節点 v_k からの探索は常に省略する. このように, 探索領域削減操作によってその節点からの探索を省略する節点は, これを白丸に×印をつけて示す. 本実行例においては, EXT_u の最後尾節点がこの節点となる.

黒丸によって示した各節点は, $SUBG$ 中の節点のうち部分森の同一化により探索を省略する節点を表す.

以上の表記を用いて, 具体的な実行過程を示す.

まず,

$$(V =)$$

$$SUBG^0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

から, グラフ中の最大次数節点 1 が選択され,

$$SUBG_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\}$$

を得る. 従って,

$$EXT_1^1 = SUBG^0 - \{1\} - \{6, 7, 8, 9, 10, 11, 12\} = \{2, 3, 4, 5\}.$$

ここで, 部分森の同一化により,

$$SUBG'^0 = \{1\} \cup \{2, 3, 4, 5\} \cup \{6, 7, 8, 9, 10, 11, 12\}$$

なる順序付き節点集合を得る。以下、候補節点集合 $SUBG$ に対して、部分森の同一化による並べ替えを実行した集合を $SUBG'$ と書く。

以下、表記法に従いクリーク探索の実行例を示す。

$SUBG'^0$ において、節点 1 からの探索を実行、

$$SUBG'_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG'_{\{1,7\}}^2 = \{6, 11, 12\},$$

$$EXT_7^2 = SUBG'_{\{1\}}^1 - \{7\} - \{6, 11, 12\} = \{8, 9, 10\},$$

$$SUBG'_{\{1\}}^1 = \{7\} \cup \{8, 9, 10\} \cup \{6, 11, 12\},$$

節点 7 からの探索を実行、

$$SUBG'_{\{1,7\}}^2 = \{6, 11, 12\},$$

$$SUBG'_{\{1,7,6\}}^3 = \{11\},$$

$$EXT_6^3 = SUBG'_{\{1,7\}}^2 - \{6\} - \{11\} = \{12\},$$

$$SUBG'_{\{1,7\}}^2 = \{6\} \cup \{12\} \cup \{11\},$$

節点 6 からの探索を実行、

$$SUBG'_{\{1,7,6\}}^3 = \{11\},$$

$$SUBG'_{\{1,7,6,11\}}^4 = \emptyset,$$

$$EXT_{11}^4 = SUBG'_{\{1,7,6\}}^3 - \{11\} - \emptyset = \emptyset,$$

$$SUBG'_{\{1,7,6\}}^3 = \{11\} \cup \emptyset \cup \emptyset,$$

11 からの探索を実行、

$$SUBG'_{\{1,7,6,11\}}^4 = \emptyset,$$

により、クリーク $\{1, 7, 6, 11\}$ を得る。ここで、 $Q_{max_size} := 4$ とする。

ここから 1 段階バックトラックすると、

$$EXT_6^3 = \{12\},$$

であるため、 EXT_6^3 の最後尾節点 12 からの探索は省略する。

1 段階バックトラック、

$$EXT_7^2 = \{8, 9, 10\},$$

により、次に 8 からの探索を実行する。

$$SUBG'_{\{1,8\}}^2 = \{10\},$$

$$SUBG'_{\{1,8,10\}}^3 = \emptyset,$$

$$EXT_9^3 = SUBG_{\{1,8\}}^2 - \{10\} - \emptyset = \emptyset,$$

$$SUBG_{\{1,8\}}'^2 = \{10\} \cup \emptyset \cup \emptyset,$$

10 からの探索を実行,

$$SUBG_{\{1,8,10\}}^3 = \emptyset,$$

により, クリーク $\{1, 8, 10\}$ を得る. ただし, $|\{1, 8, 10\}| = 3 < 4$ であるため, Q_{max_size} の更新は行わない. 1 段階バックトラック,

$$SUBG_{\{1,9\}}^2 = \{10, 11, 12\},$$

$$SUBG_{\{1,9,10\}}^3 = \{12\},$$

$$EXT_{10}^3 = SUBG_{\{1,9\}}^2 - \{10\} - \{12\} = \{11\},$$

$$SUBG_{\{1,9\}}'^2 = \{10\} \cup \{11\} \cup \{12\},$$

$$SUBG_{\{1,9,10\}}^3 = \{12\},$$

$$SUBG_{\{1,9,10,12\}}^4 = \emptyset,$$

$$EXT_{12}^4 = SUBG_{\{1,9,10\}}^3 - \{12\} - \emptyset = \emptyset,$$

$$SUBG_{\{1,9,10\}}'^3 = \{12\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{1,9,10,12\}}^4 = \emptyset, \text{ (クリーク : } \{1, 9, 10, 12\}, \text{ 更新なし)}$$

1 段階バックトラック, 11 からの探索を省略,

1 段階バックトラック, 10 からの探索を省略,

$$EXT_1^1 = \{2, 3, 4, 5\},$$

$$SUBG_{\{2\}}^1 = \{3, 4, 5, 7, 9, 10\},$$

$$SUBG_{\{2,3\}}^2 = \{4, 5\},$$

$$EXT_3^2 = SUBG_{\{2\}}^1 - \{3\} - \{4, 5\} = \{7, 9, 10\},$$

$$SUBG_{\{2\}}'^1 = \{3\} \cup \{7, 9, 10\} \cup \{4, 5\},$$

$$SUBG_{\{2,3\}}^2 = \{4, 5\},$$

$$SUBG_{\{2,3,4\}}^3 = \{5\},$$

$$EXT_4^3 = SUBG_{\{2,3\}}^2 - \{4\} - \{5\} = \emptyset,$$

$$SUBG_{\{2,3\}}'^2 = \{4\} \cup \emptyset \cup \{5\},$$

$$SUBG_{\{2,3,4\}}^3 = \{5\},$$

$$SUBG_{\{2,3,4,5\}}^4 = \emptyset,$$

$$EXT_5^4 = SUBG_{\{2,3,4\}}^3 - \{5\} - \emptyset = \emptyset,$$

$$SUBG'_{\{2,3,4\}}^3 = \{5\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{2,3,4,5\}}^4 = \emptyset, \text{ (更新なし)},$$

2 段階バックトラック,

$$EXT_3^2 = \{7, 9, 10\},$$

$$SUBG_{\{2,7\}}^2 = \emptyset, \text{ (更新なし)}$$

$$SUBG_{\{2,9\}}^2 = \{10\},$$

$$SUBG_{\{2,9,10\}}^3 = \emptyset,$$

$$EXT_{10}^3 = \emptyset,$$

$$SUBG'_{\{2,9\}}^2 = \{10\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{2,9,10\}}^3 = \emptyset, \text{ (更新なし)}$$

1 段階バックトラック, 10 からの探索を省略,

1 段階バックトラック,

$$SUBG_{\{3\}}^1 = \{4, 5, 6, 12\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$EXT_4^2 = \{12\},$$

$$SUBG'_{\{3\}}^1 = \{4\} \cup \{12\} \cup \{5, 6\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset,$$

$$EXT_5^3 = \{6\},$$

$$SUBG'_{\{3,4\}}^2 = \{5\} \cup \{6\} \cup \emptyset,$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset, \text{ (更新なし)}$$

2 段階バックトラック,

$$SUBG_{\{4\}}^1 = \{6, 5, 11\},$$

$$SUBG_{\{4,6\}}^2 = \{11\},$$

$$EXT_6^2 = \{5\},$$

$$SUBG'_{\{4\}}^1 = \{6\} \cup \{5\} \cup \{11\},$$

$$SUBG_{\{4,6\}}^2 = \{11\},$$

$$SUBG_{\{4,6,11\}}^3 = \emptyset,$$

$$EXT_{11}^3 = \emptyset,$$

$$SUBG_{\{4,6\}}^2 = \{11\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{4,6,11\}}^3 = \emptyset, \text{ (更新なし)}$$

2段階バックトラック, 5は $EXT_1^1 = \{2, 3, 4, 5\}$ の最後尾節点であるから, 探索を省略して実行終了する.

結果として, MCP_0 は例題のグラフ G 中の最大クリークサイズ 4 を得る.

本章に続く第 4-7 章においては, 基本アルゴリズム MCP_0 の探索領域削減手法を順次拡充していく. 各章においては, 本節と同一の例題に対する実行過程を示すことにより, 拡充の具体的な効果を確認する.

3.6 最大時間計算量評価

与えられたグラフ $G = (V, E)$ について, MCP_0 の最大時間計算量評価を行う. これは, オーダー評価においては $EXPAND(V)$ の最大時間計算量評価と等価である.

いま, $T(|SUBG|)$ を $SUBG \subseteq V$ に対する $EXPAND(SUBG)$ の最大時間計算量とする. ここで,

$$T(|\emptyset|) = T(0) = 0$$

とする. (本章以降の各章においても同様.)

$EXPAND()$ の最大時間計算量評価の基礎として, まず以下の様な非再帰手続き $EXPAND_0(SUBG)$ を考え, その最大時間計算量評価を行う.

EXPAND₀(SUBG): アルゴリズム 1 の手続き $EXPAND(SUBG)$ において, 1600 行の $EXPAND(SUBG_u)$ 及び 2200 行の $EXPAND(SUBG_{v_i})$ をそれぞれ $EXPAND(\emptyset)$ で置き換えた非再帰手続き.

$EXPAND_0(SUBG)$ の実行時間については, 次の補題が成立する.

補題 3.6.1 隣接リスト表現されたグラフ $G = (V, E)$ において, $SUBG \subseteq V$, 節点 u を $G(SUBG)$ における最大次数節点とし, 以下の様にそれぞれ定義する.

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$EXT_u[i, k] = \{v_i, v_{i+1}, \dots, v_k\} \quad (1 \leq i \leq k), \text{ 及び,}$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u).$$

このとき, $EXPAND_0(SUBG)$ の最大時間計算量は $O(|SUBG| \cdot |SUBG_u|^2)$ である.

(証明) 先ず, グラフの最大次数を Δ として, 場合分けを行う.

$\Delta \geq 1$ である場合:

1000 行における最大次数節点の選択は, $SUBG$ 中の各節点の次数を計算することにより実行可能であるから, グラフの隣接リスト表現により $O(|SUBG| \cdot |SUBG_u|)$ -時間で実行可能である.

1100-1700 行における各操作は, いずれも $O(|SUBG|)$ -時間で実行可能である. 1900-2300 行における **for** 文中の各操作は, いずれも

$O(|SUBG_{v_i}|^2) \leq O(|SUBG_u|^2)$ -時間で実行可能であるから, 1800-2400 行における **for** 文の繰り返し操作全体は,

$$O(|SUBG_u|^2 \cdot |EXT_u|) = O(|SUBG| \cdot |SUBG_u|^2)$$

-時間で実行可能である. 2500-2700 各行の操作に関しても $O(|SUBG|)$ -時間で実行可能である. 以上から, $\Delta \geq 1$ である場合, $EXPAND_0(SUBG)$ 全体は,

$$O(|SUBG| \cdot |SUBG_u|^2)$$

-時間で実行可能である.

$\Delta = 0$ である場合:

グラフの隣接リスト表現により, 上記各操作は, $O(|SUBG|)$ -時間で実行可能である. 従って, この場合における $EXPAND_0(SUBG)$ の全実行時間は, $O(|SUBG|)$ である.

(この結果は, 第 4-7 章における各アルゴリズムに対して, 全て同様に成立する.) □

補題 3.6.1 の結果から, $EXPAND_0(SUBG)$ の実行時間の上限を, ある定数 C を用いて $C|SUBG| \cdot |SUBG_u|^2$ と表す.

これにより, 次の補題が成り立つ.

補題 3.6.2 節点集合 $SUBG$ による誘導部分グラフにおける, $EXPAND(SUBG)$ の最大時間計算量 $T(|SUBG|)$ について, 節点 u を $G(SUBG)$ における最大次数節点とし, 以下の様にそれぞれ定義する.

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$EXT_u[i, k] = \{v_i, v_{i+1}, \dots, v_k\} \quad (1 \leq i \leq k) \text{ および,}$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u).$$

また, $|SUBG_u| = \Delta$ とおく.

このとき, 以下が成り立つ.

$\Delta = 0$ である場合:

$$T(|SUBG|) \leq C|SUBG|.$$

$\Delta \geq 1$ である場合:

$$T(|SUBG|) \leq T(|SUBG_u|) + \sum_{i=1}^{k-1} T(|SUBG_{v_i}|) + C|SUBG| \cdot |SUBG_u|^2. \quad (3.6.1)$$

(証明) 手続き $EXPAND(SUBG)$ の定義, および上記の定数 C の定義から明らか. \square

上記準備のもとに結論となる定理を導くため, まず誘導部分グラフ $G(SUBG)$ に関する次の主要補題を示す.

補題 3.6.3 節点集合 $SUBG \subseteq V$ に対して, $|SUBG| = \eta$, $C' = 10^4 C$ とおく. このとき, $EXPAND(SUBG)$ の最大時間計算量 $T(|SUBG|)$ は以下を満たす.

$$T(|SUBG|) \leq C' 2^{0.4008\eta} \eta^3.$$

(証明) 証明は $\eta \geq 1$ についての数学的帰納法による.

まず, $\eta = 1$ である場合, 題意は明らかに成立する.

次に, ある $\eta \geq 1$ 以下の全ての正整数に関して, 題意が成立することを仮定する. 即ち,

$$T(|SUBG|) \leq C' 2^{0.4008\eta} \eta^3.$$

この仮定のもとで, $|SUBG| = \eta + 1$ である場合を考える. いま, u を $G(SUBG)$ における最大次数節点として,

$$SUBG_u = \Gamma(u) \cap SUBG$$

とする. このとき $|SUBG_u| \leq \eta$ である.

補題 3.6.2 により, $|SUBG_u| = 0$ である場合題意は明らかに成立する. そこで, 以下では,

$$|SUBG_u| = \eta - k \quad (0 \leq k < \eta)$$

とおく (第4章~7章における計算量評価においても同様). このとき,

$$\begin{aligned} |EXT_u| &= |SUBG - \{u\} - SUBG_u| \\ &= |SUBG| - |\{u\}| - |SUBG_u| \\ &= (\eta + 1) - 1 - (\eta - k) \\ &= k. \end{aligned}$$

ここで, 補題 3.6.2 および帰納法の仮定から,

$$\begin{aligned} T(|SUBG|) &\leq T(|SUBG_u|) + \sum_{i=1}^{k-1} T(|SUBG_{v_i}|) + C|SUBG| \cdot |SUBG_u|^2 \\ &< T(|SUBG_u|) + \sum_{i=1}^{k-1} T(|SUBG_{v_i}|) + C|SUBG|^3 \\ &\leq ((\eta + 1) - (\eta - k) - 1) \cdot T(|SUBG_u|) + C|SUBG|^3 \\ &\leq ((\eta + 1) - (\eta - k) - 1) \cdot C'2^{0.4008(\eta-k)}(\eta-k)^3 + C(\eta + 1)^3 \\ &< kC'2^{0.4008(\eta-k)}(\eta + 1)^3 + C(\eta + 1)^3 \\ &= C' \left(\frac{k}{2^{0.4008k}} + \frac{1}{10^4 \cdot 2^{0.4008\eta}} \right) 2^{0.4008\eta} (\eta + 1)^3 \\ &< C' \left(\frac{k}{2^{0.4008k}} + 10^{-4} \right) 2^{0.4008\eta} (\eta + 1)^3. \end{aligned}$$

いま, 任意の非負整数 k に関して,

$$\frac{k}{2^{0.4008k}} < 1.3166 (< 2^{0.4008})$$

が成立するならば, 数学的帰納法により任意の $\eta \geq 1$ において題意は成立する. そこで, このことを示すため, 実数 x 上の関数

$$f(x) = \frac{x}{2^{0.4008x}}$$

を考える.

まず, $f(0) < f(1) < f(2) < f(3) < f(4) < 1.3166$ である.

次に,

$$\begin{aligned}
& f'(x) \\
&= \frac{2^{0.4008x} - x \cdot \ln 2^{0.4008} 2^{0.4008x}}{(2^{0.4008x})^2} \\
&= \frac{1 - \ln 2^{0.4008} x}{2^{0.4008x}} \\
&< \frac{1 - 0.27x}{2^{0.4008x}}
\end{aligned}$$

より, $x > 4$ において $f'(x) < 0$ であり, したがって $f(x)$ は $x > 4$ において単調減少である. 故に, $\frac{k}{2^{0.4008k}} < 1.3166$ は任意の整数 $k \geq 0$ に対して成立する.

これより, 以下が証明できる.

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&< (1.3166 + 0.0001)C'2^{0.4008\eta}(\eta + 1)^2 \\
&= 1.3167C'2^{0.4008\eta}(\eta + 1)^2 \\
&< 2^{0.4008}C'2^{0.4008\eta}(\eta + 1)^2 \quad (\text{ここで, } 2^{0.4008} = 1.3203\dots) \\
&= C'2^{0.4008(\eta+1)}(\eta + 1)^2.
\end{aligned}$$

以上より, 題意の不等式は $|SUBG| = \eta + 1$ の場合にも成立することが示せた. 従って, 帰納法による帰結として, $\eta \geq 1$ なるすべての整数 η に関して題意は成立する. \square

以上を用いて, 本章の結論となる次の定理を示す.

定理 3.6.1 節点数 $|V| = n$, 最大次数 $\Delta \geq 1$ なるグラフに対して, いま定数 $C' = 10^4 C$, $C'' = 6.7 \cdot 10^{11}$, とする. このとき Δ が,

$$\Delta \leq 2.493d \lg n \quad (d \geq 0: \text{定数})$$

を満たすならば, アルゴリズム MCP₀ は与えられたグラフの最大クリークサイズを $O(n^{1+d})$ -時間で抽出可能である.

(証明) 式 (3.6.1) により,

$$\begin{aligned}
& T(|V|) \\
&= T(n) \\
&\leq T(|SUBG_u|) + T(|SUBG_{v_1}|) + T(|SUBG_{v_2}|) + \dots + T(|SUBG_{v_{k-1}}|) + Cn\Delta^2.
\end{aligned}$$

ここで, 補題 3.6.3 により,

$$\begin{aligned} T(n) & \\ & \leq nC'2^{0.4008\Delta}\Delta^3 + Cn\Delta^2. \end{aligned}$$

ここで定数 C'' の定義, および補題 2.2.1 から,

$$\begin{aligned} T(n) & \\ & = n \cdot C'2^{0.4008\Delta}\Delta^3 + Cn\Delta^2 \\ & < n \cdot C'2^{0.4008\Delta} \cdot C''2^{0.0003\Delta} + Cn\Delta^2 \\ & = nC'C''2^{0.4011\Delta} + Cn\Delta^2. \end{aligned}$$

ここで, $\Delta \leq 2.493d \lg n$ であるとする,

$$\begin{aligned} T(n) & \\ & < nC'C''2^{0.4011\Delta} + Cn\Delta^2 \\ & \leq nC'C''2^{0.4011 \cdot 2.493d \lg n} + Cn(2.493d \lg n)^2 \\ & < nC'C''n^d + 6.216Cn(\lg n^d)^2. \end{aligned}$$

ここで, 補題 2.2.2 により, $(\lg n)^2 < (\lg e)n$ が成立するから, $((\lg n)^2)^d < ((\lg e)n)^d$ が成立する. 従って,

$$\begin{aligned} T(n) & \\ & < C'C''n^{1+d} + 6.216Cn((\lg e)n)^d \\ & < C'C''n^{1+d} + 6.216(\lg e)^d Cn^{1+d} \\ & = (C'C'' + 6.216(\lg e)^d C)n^{1+d}. \end{aligned}$$

故に, $T(n) = O(n^{1+d})$ である. \square

ここで, 定理 3.6.1 に対して, $O^*(2^{0.288n})$ -時間で最大独立節点集合問題を解くことが出来るとの解析評価を得ている文献 [37] の結果を適用すると, グラフの最大次数 $\Delta \geq 1$ に対して, 定理 3.6.1 の結果数値 2.493 を 3.468 とすることが, 下記の様にして可能である.

まず, 文献 [37] のアルゴリズムの理論的計算量上界が $O^*(2^{0.288n})$ であるとの結果により,

$$\begin{aligned} T(\Delta) & \text{ に対して, 正定数 } D \text{ 及び正整数 } m \text{ を用いて,} \\ T(\Delta) & \leq D2^{0.288\Delta}\Delta^m \end{aligned}$$

が成立する. ここで, $\alpha = 0.0003$ なる正定数 α に対して,

$$D' = \frac{m!}{(\ln 2^\alpha)^m}$$

とすれば, 補題 2.2.1 により, $\Delta \geq 1$ において,

$$\Delta^m < D' 2^{\alpha \cdot \Delta}.$$

従って,

$$\begin{aligned} T(\Delta) &\leq D 2^{0.288\Delta} \Delta^m \\ &< D D' 2^{0.288\Delta} \cdot 2^{\alpha \Delta} \\ &= D D' 2^{(0.288+\alpha)\Delta}. \\ &= D D' 2^{0.2883\Delta}. \end{aligned}$$

これに基づき, 下記命題 A の結果を得る. なお, この命題における定数 3.468 は, 下記関係式による.

$$1/0.2883 = 3.468... > 3.468.$$

[命題 A] 節点数 $n \geq 1$ のグラフにおいて, グラフの最大次数 $\Delta \geq 1$ に対して,

$$\Delta \leq 3.468 \cdot d \lg n \quad (d \geq 0: \text{定数})$$

であるならば, 最大クリーク問題は $O(n^{1+d})$ -時間で実行可能である.

(証明) 式 (3.6.1) において $SUBG = V$ として,

$$\begin{aligned} SUBG_u &= \Gamma(u) \cap V, \\ EXT_u &= V - \{u\} - SUBG_u \\ &= \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|), \end{aligned}$$

とそれぞれ定義する. このとき式 (3.6.1) により,

$$\begin{aligned} T(|V|) &= T(n) \\ &\leq T(|SUBG_u|) + \sum_{i=1}^{k-1} T(|SUBG_{v_i}|) + C \Delta^2 \cdot n. \end{aligned}$$

いま, $|SUBG_{v_i}| \leq |SUBG_u| \leq \Delta$ である.

ここで, 各部分問題 $SUBG_u, SUBG_{v_i}$ ($1 \leq i \leq k-1$) に対して, 文献 [37] のアルゴリズムを適用する. 上記の通り, このとき,

$$T(\Delta) < D D' 2^{0.2883\Delta}$$

であるから,

$$\begin{aligned}
& T(n) \\
& < (n-1)DD'2^{0.2883\Delta} + C\Delta^2 \cdot n \\
& < nDD'2^{0.2883\Delta} + C\Delta^2 \cdot n.
\end{aligned}$$

ここで, $\Delta \leq 3.468d \lg n$ とすれば,

$$\begin{aligned}
& T(n) \\
& \leq nDD'2^{0.2883 \cdot 3.468d \lg n} + C \cdot (3.468d \lg n)^2 \cdot n \\
& < nDD'n^d + C \cdot (3.468 \lg n^d)^2 \cdot n \\
& < nDD'n^d + 12.028C(\lg n^d)^2 \cdot n.
\end{aligned}$$

いま, $n \geq 1$ において $n^d \geq 1$ であり, 補題 2.2.2 により,

$$(\lg n^d)^2 < (\lg e)n^d.$$

従って,

$$\begin{aligned}
& T(n) \\
& < DD'n^{1+d} + 12.028(\lg e)Cn^{1+d} \\
& < (DD' + 12.028 \cdot 1.45C)n^{1+d} \\
& < (DD' + 17.45C)n^{1+d}. \quad \square
\end{aligned}$$

第 4 章

アルゴリズム MCP_1

4.1 概要

本章では, 前章における基本アルゴリズム MCP_0 に対し, “部分問題の統合による探索領域削減” という新たな限定操作を導入し, 以下の改良結果を与える.

「節点数 n の一般グラフにおいて, 最大次数 Δ が $\Delta \leq 2.610d \lg n$ ($d \geq 0$: 定数) なる条件を満たしている時, このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である. 」

4.2 アルゴリズム MCP_1

前章のアルゴリズム MCP_0 を拡張して, 本論文において新たに提唱するアルゴリズム MCP_1 をアルゴリズム 2 に示す. MCP_0 に対し, 本章における新たな追加, 及び変更部分についてはこれを赤字により表す.

MCP_1 における, MCP_0 に対する追加部分は,

1401 - 1409 行,

及び,

2401 - 2407 行

の各部分である. これらの部分は, 本章において導入する, “部分問題の統合による探索領域削減” を実現するための各操作である. MCP_1 における MCP_0 からの変更部分は, 1800 行における

for $i := 1$ to $k - 1$

を

for $i := 1$ to $k - 2$

とした点のみである.

アルゴリズム 2 アルゴリズム MCP₁

```

0000: procedure MCP1(G)
0100: global Q_size, Q_max_size;
0200: begin
0300: Q_size := 0, Q_max_size := 0;
0400: EXPAND(V)
0500: return Q_max_size
0600: end {of MCP1}

0700: procedure EXPAND(SUBG)
0800: begin
0900: if SUBG ≠ ∅ then
1000:   u := a vertex in SUBG that
         maximizes |Γ(u) ∩ SUBG|;
1100:   SUBGu := Γ(u) ∩ SUBG;
1200:   EXTu := SUBG - {u} - SUBGu;
1300:   k := |EXTu|;
1400:   let EXTu = {v1, v2, ..., vk};

1401:   for i = 1 to k
1402:   do
1403:     Ui := Γ(vi) ∩ EXTu;
1404:     if |Ui| < k - 1 then
1405:       swap vi with vk;
1406:       swap (EXTu - Ui)[1] with vk-1;
1407:       i := k + 1
1408:     fi
1409:   od

1500:   Q_size := Q_size + |{u}|;
1600:   EXPAND(SUBGu)
1700:   Q_size := Q_size - |{u}|;
1800:   for i = 1 to k - 2 do
1900:     vi := the first vertex in EXTu;
2000:     SUBGvi
           := Γ(vi) ∩ (EXTu[i, k] ∪ SUBGu);
2100:     Q_size := Q_size + |{vi}|;
2200:     EXPAND(SUBGvi)
2300:     Q_size := Q_size - |{vi}|
2400:   od

2401:   if (vk-1, vk) ∈ E then
2402:     SUBGTAIL :=  $\hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u$ ;
2403:     ROOT := {vk-1, vk};
2404:     Q_size := Q_size + |{ROOT}|;
2405:     EXPAND(SUBGTAIL)
2406:     Q_size := Q_size - |{ROOT}|
2407:   fi

2500: else {i.e., SUBG = ∅}
2600:   if Q_size > Q_max_size then
2700:     Q_max_size := Q_size fi fi
2800:   end {of EXPAND}

```

4.2.1 部分問題の統合による探索領域削減

MCP₀において,

$$EXT_u$$

$$= SUBG - \{u\} - SUBG_u$$

$$= \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|)$$

に対し, EXT_u の最後尾節点 v_k からの探索は, これを常に省略した. この手法は最大ク
リーク抽出における探索領域削減手法としてはごく基本的なものであり, 本章のアルゴリ

ズム MCP_1 以降の各アルゴリズムにおいてもこれを採用する.

本章においては, 上記の基本的手法を拡張し, 新たな探索領域削減手法を導入するため, $k = |EXT_u| \geq 2$ である場合において, EXT_u 中の最後尾から 2 番目の節点 v_{k-1} からの探索について考察する.

いま, 前章の定義により,

$$\begin{aligned} SUBG_{v_{k-1}} &= \Gamma(v_{k-1}) \cap (EXT_u[k-1, k] \cup SUBG_u) \\ &= \Gamma(v_{k-1}) \cap (\{v_k\} \cup SUBG_u) \end{aligned}$$

である. 従って,

$$R = SUBG_{v_{k-1}} - SUBG_u$$

とおくと,

$(v_{k-1}, v_k) \notin E$ である場合,

$$R = \emptyset,$$

$$\begin{aligned} SUBG_{v_{k-1}} &= \Gamma(v_{k-1}) \cap SUBG_u \\ &\subseteq SUBG_u \end{aligned}$$

である. そこで, この場合には節点 v_{k-1} からの探索は省略する.

次に, $(v_{k-1}, v_k) \in E$ である場合,

$$R = \{v_k\}$$

である. ここで, 第2章, 補題 2.2.4 において,

$$W = SUBG_u,$$

$$W' = SUBG_{v_{k-1}}$$

とそれぞれおくと, 次が成立する.

補題 4.2.1 $\omega(SUBG_{v_{k-1}}) > \omega(SUBG_u)$

なる不等式が成立するのは, $SUBG_{v_{k-1}}$ 中の最大クリークが節点 v_k を含む場合に限られる.

(証明) 補題 2.2.4 による. \square

この結果により、節点 v_{k-1} からの探索においては節点 v_k を確実に含むクリークのみを抽出すれば十分であることがわかる。

いま、 $(v_{k-1}, v_k) \in E$ である場合、

$$SUBG_{TAIL} := \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u$$

とおく。このとき、節点集合 $\{v_{k-1}, v_k\} \cup SUBG_{TAIL}$ 中の最大クリークは v_k を必ず含む。そこで、節点 v_{k-1} からの探索は、この $SUBG_{TAIL}$ に対する EXPAND の実行とする。ここで、

$$\begin{aligned} & SUBG_{TAIL} \\ &= \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u \\ &= \Gamma(v_k) \cap (\Gamma(v_{k-1}) \cap SUBG_u) \\ &= \Gamma(v_k) \cap (\Gamma(v_{k-1}) \cap (\{v_k\} \cup SUBG_u)) \\ &= \Gamma(v_k) \cap SUBG_{v_{k-1}} \end{aligned}$$

である。即ち、 $SUBG_{TAIL}$ は $SUBG_{v_{k-1}}$ のうち、節点 v_k に隣接する節点の集合である。また、明らかに、

$$|SUBG_{TAIL}| \leq |SUBG_{v_{k-1}}| - 1$$

である。以上から、節点 v_{k-1} からの探索においては、どのような場合においても、探索領域削減が可能である。即ち、

- ・ $(v_{k-1}, v_k) \notin E$ である場合、 v_{k-1} からの探索を省略する。
- ・ $(v_{k-1}, v_k) \in E$ である場合、 v_{k-1} からの探索を、 $SUBG_{TAIL}$ ($|SUBG_{TAIL}| < |SUBG_{v_{k-1}}|$) に対する EXPAND の実行とする。

アルゴリズム上では、以下の手順によって上記の探索領域削減操作を実現する。

集合 $EXT_u = \{v_1, v_2, \dots, v_k\}$ ($k = |EXT_u|$) 中の各節点 v_i ($1 \leq i \leq k$) について、

$$U_i = \Gamma(v_i) \cap EXT_u,$$

即ち、節点 v_i と隣接する EXT_u 中の節点の集合を考える。

(1) ある $1 \leq i \leq k$ において $|U_i| < k - 1$ が成立するならば、 EXT_u 中には節点 v_i に隣接しない節点が少なくとも1個存在する。そこで、このような v_i が存在する場合には、節点 v_i を EXT_u 中最後尾の節点 v_k と交換 (swap) し (1405 行)、更に節点 $(EXT_u - U_i)[1]$,

即ち v_i と隣接しない節点と v_{k-1} を交換する. (1406 行)

節点交換操作が実行された場合, 交換後の EXT_u において,

$$(v_{k-1}, v_k) \notin E$$

である. 従ってこの場合には, EXT_u の最後尾 2 節点 v_{k-1} 及び v_k からの探索は単純に省略する.

(2) 任意の $1 \leq i \leq k$ において $|U_i| = k - 1$ である場合, (1) の節点交換操作は実行されず, 1800-2400 行における **for** 文の実行が終了した時点において,

$$(v_{k-1}, v_k) \in E$$

となる (この場合, $G(EXT_u)$ 中に非隣接節点对は 1 個も存在しない. 即ち $G(EXT_u)$ は クリークである). この場合には, $SUBG_{TAIL}$ を上記の定義に従って構成し, この集合に対し **EXPAND** を実行する. 実行に当たっては,

$$ROOT := \{v_{k-1}, v_k\}$$

として, 集合 $ROOT$ を現在探索中のクリークに追加する (2404 行).

これらの操作追加に合わせて, アルゴリズム 2, 1800 行における **for** 文の終了値は $k-2$ とする. 即ち, $(v_{k-1}, v_k) \notin E$ である場合には, 2402-2407 行の操作は実行されない. これは, 実質的には $SUBG_{TAIL} = \emptyset$ とすることと同じである.

以上の探索領域削減手法を, 以下“部分問題の統合による探索領域削減”と呼ぶ.

前章同様に,

$$|SUBG| = \eta,$$

$$k = |EXT_u|$$

とそれぞれおく. アルゴリズム MCP₀ において, $SUBG$ は k 個の部分問題に分割され, それぞれの部分問題のサイズは高々 $\eta - k - 1$ であった. これに対して, MCP₁ においては $u, v_1, v_2, \dots, v_{k-2}$ ($k - 1$ 個) からの探索が確実に実行され, これらの探索における部分問題のサイズは高々 $\eta - k - 1$ である.

v_{k-1} からの探索は省略されるか, 探索が実行される場合においても部分問題のサイズは高々 $\eta - k - 2$ である. よって, “部分問題の統合による探索領域削減” の導入により, どのような場合においてもアルゴリズムの探索領域削減は達成されている.

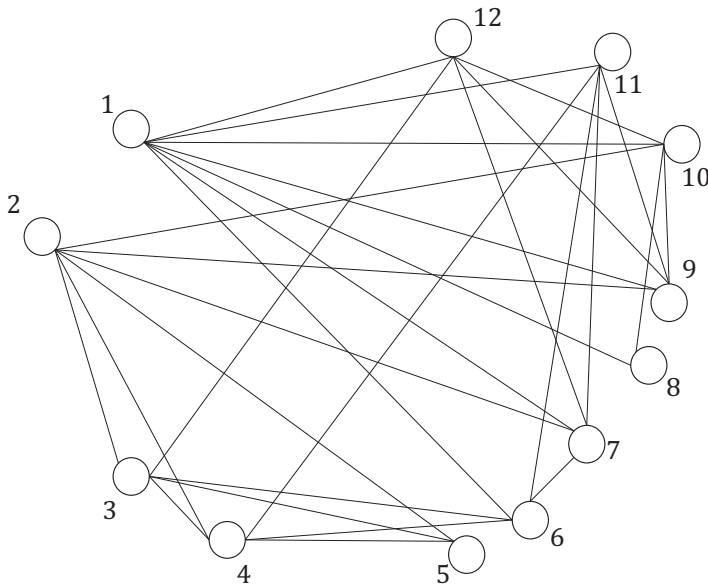
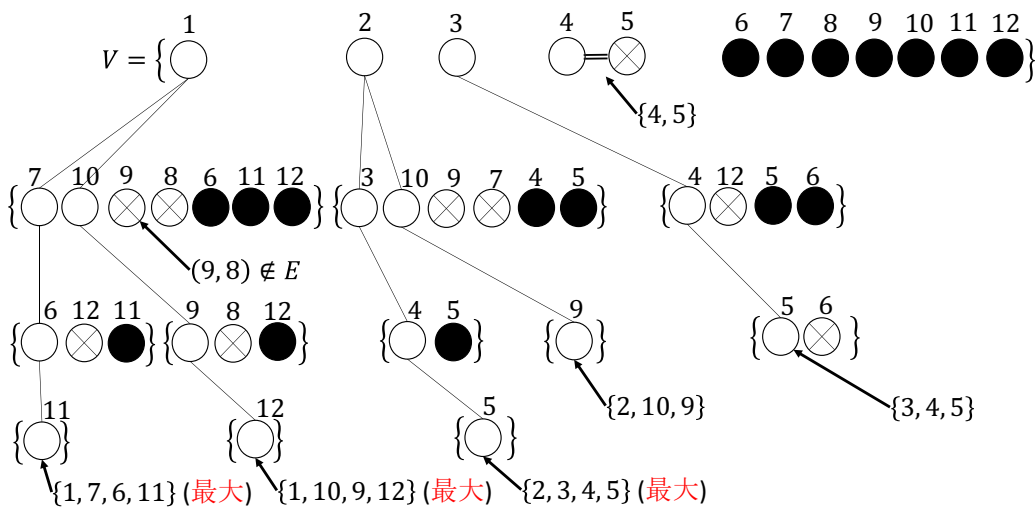


図 4.1 例題



※二重線(=)はEXPANDによらない節点の選択を表す.

図 4.2 MCP_1 によるクリーク探索森

4.2.2 例題

図 4.1 のグラフ $G = (V, E)$ (第 3 章において用いたグラフと、同一のグラフ) に対するアルゴリズム MCP_1 の実行過程を、図 4.2 の探索森として表す.

以下の実行過程における集合等の表記, 及び図 4.2 中における表記法は, 前章例題におけるものを踏襲する.

本章において新たに導入した, “部分問題の統合による探索領域削減” については, 新たに実行過程中, 及び図中における表記法を定義する.

前章例題中の記述に従って,

$$\begin{aligned}
& SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d \\
& SUBG_{u_d}^{d+1} \\
& = SUBG_{\{p_0, p_1, \dots, p_{d-1}, u_d\}}^{d+1} \\
& EXT_{u_0}^1 = SUBG^0 - \{u_0\} - SUBG_{\{u_0\}}^1, \text{ 及び,} \\
& EXT_{u_d}^{d+1} \\
& = SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d - \{u_d\} - SUBG_{u_d}^{d+1} \\
& = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_{u_d}^{d+1}|)
\end{aligned}$$

とするとき, 節点集合,

$$\hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG^0$$

を,

$$SUBG_{\{v_{k-1}, v_k\}}^1,$$

又は,

$$\begin{aligned}
& SUBG_{TAIL:\{v_{k-1}, v_k\}}^1, \text{ また,} \\
& \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_{\{p_0, p_1, \dots, p_{d-1}\}}^d
\end{aligned}$$

を,

$$SUBG_{\{p_0, p_1, \dots, p_{d-1}, v_{k-1}, v_k\}}^{d+1},$$

又は,

$$SUBG_{TAIL:\{p_0, p_1, \dots, p_{d-1}, v_{k-1}, v_k\}}^{d+1}$$

とそれぞれ表記する, ここで, v_{k-1}, v_k は $SUBG_{TAIL}$ に対する探索を実行するとき, 探索中のクリークに追加される集合 $ROOT$ 中の 2 節点である. 図中においては, この集合 $ROOT$ 中の節点を, 2 重線で結ぶことにより表記する.

以上の表記を用いて, 前章同様に具体的な実行過程を示す.

$$(V =)$$

$$SUBG^0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\},$$

$$EXT_1^1 = \{2, 3, 4, 5\},$$

$$SUBG'^0 = \{1\} \cup \{2, 3, 4, 5\} \cup \{6, 7, 8, 9, 10, 11, 12\},$$

ここで,

$$|U_1| = |\Gamma(2) \cap EXT_1^1| = |\{3, 4, 5\}| = 4 - 1,$$

$$|U_2| = |\Gamma(3) \cap EXT_1^1| = |\{2, 4, 5\}| = 4 - 1,$$

$$|U_3| = |\Gamma(4) \cap EXT_1^1| = |\{2, 3, 5\}| = 4 - 1,$$

$$|U_4| = |\Gamma(5) \cap EXT_1^1| = |\{2, 3, 4\}| = 4 - 1,$$

であるため, 前節, (1) の交換操作は実行されない.

$$SUBG'^0 = \{1\} \cup \{2, 3, 4, 5\} \cup \{6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG_{\{1,7\}}^2 = \{6, 11, 12\},$$

$$EXT_7^2 = SUBG_{\{1\}}^1 - \{7\} - \{6, 11, 12\} = \{8, 9, 10\},$$

$$SUBG'_{\{1\}}^1 = \{7\} \cup \{8, 9, 10\} \cup \{6, 11, 12\},$$

ここで,

$$|U_1| = |\Gamma(8) \cap EXT_7^2| = |\{10\}| = 1 < 3 - 1$$

であるため, 節点 8 を 10 と, 節点 9 を 9 自身とそれぞれ交換する.

$$SUBG'_{\{1\}}^1 = \{7\} \cup \{10, 9, 8\} \cup \{6, 11, 12\},$$

$$SUBG_{\{1,7\}}^2 = \{6, 11, 12\},$$

$$SUBG_{\{1,7,6\}}^3 = \{11\},$$

$$EXT_6^3 = SUBG_{\{1,7\}}^2 - \{6\} - \{11\} = \{12\},$$

$$SUBG'_{\{1,7\}}^2 = \{6\} \cup \{12\} \cup \{11\},$$

交換操作は実行されない.

$$SUBG_{\{1,7,6\}}^3 = \{11\},$$

$$SUBG_{\{1,7,6,11\}}^4 = \emptyset,$$

$$EXT_{11}^4 = SUBG_{\{1,7,6\}}^3 - \{11\} - \emptyset = \emptyset,$$

$$SUBG'_{\{1,7,6\}}^3 = \{11\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{1,7,6,11\}}^4 = \emptyset, \text{ (更新: } Qmax_size := 4)$$

2 段階バックトラック,

$$\begin{aligned}
EXT_7^2 &= \{10, 9, 8\}, \\
SUBG_{\{1,10\}}^2 &= \{8, 9, 12\}, \\
SUBG_{\{1,10,9\}}^3 &= \{12\}, \\
EXT_9^3 &= SUBG_{\{1,10\}}^2 - \{10\} - \{12\} = \{8\}, \\
SUBG'_{\{1,10\}}^2 &= \{9\} \cup \{8\} \cup \{12\}, \\
SUBG_{\{1,10,9\}}^3 &= \{12\}, \\
SUBG_{\{1,10,9,12\}}^4 &= \emptyset, \\
EXT_{12}^4 &= SUBG_{\{1,10,9\}}^3 - \{12\} - \emptyset = \emptyset, \\
SUBG'_{\{1,10,9,12\}}^3 &= \{12\} \cup \emptyset \cup \emptyset, \\
SUBG_{\{1,10,9,12\}}^4 &= \emptyset, \text{ (更新なし)}
\end{aligned}$$

2段階バックトラック, $(9, 8) \notin E$ であるため, 節点9及び8からの探索はいずれも省略する.

1段階バックトラック,

$$\begin{aligned}
EXT_1^1 &= \{2, 3, 4, 5\}, \\
SUBG_{\{2\}}^1 &= \{3, 4, 5, 7, 9, 10\}, \\
SUBG_{\{2,3\}}^2 &= \{4, 5\}, \\
EXT_3^2 &= SUBG_{\{2\}}^1 - \{3\} - \{4, 5\} = \{7, 9, 10\}, \\
SUBG_{\{2\}}^1 &= \{3\} \cup \{7, 9, 10\} \cup \{4, 5\}, \\
|U_1| &= |\Gamma(7) \cap EXT_3^2| = |\emptyset| = 0 < 3 - 1,
\end{aligned}$$

交換操作を実行,

$$\begin{aligned}
SUBG'_{\{2\}}^1 &= \{3\} \cup \{10, 9, 7\} \cup \{4, 5\}, \\
SUBG_{\{2,3\}}^2 &= \{4, 5\}, \\
SUBG_{\{2,3,4\}}^3 &= \{5\}, \\
EXT_4^3 &= SUBG_{\{2,3\}}^2 - \{4\} - \{5\} = \emptyset, \\
SUBG'_{\{2,3\}}^2 &= \{4\} \cup \emptyset \cup \{5\}, \\
SUBG_{\{2,3,4\}}^3 &= \{5\}, \\
SUBG_{\{2,3,4,5\}}^4 &= \emptyset, \\
EXT_5^4 &= SUBG_{\{2,3,4\}}^3 - \{5\} - \emptyset = \emptyset, \\
SUBG'_{\{2,3,4\}}^3 &= \{5\} \cup \emptyset \cup \emptyset,
\end{aligned}$$

$$SUBG_{\{2,3,4,5\}}^4 = \emptyset, \text{ (更新なし)}$$

2 段階バックトラック,

$$SUBG_{\{2,10\}}^2 = \{9\},$$

$$SUBG_{\{2,10,9\}}^3 = \emptyset,$$

$$EXT_9^3 = SUBG_{\{2,10\}}^2 - \{9\} - \emptyset = \emptyset,$$

$$SUBG_{\{2,10\}}'^2 = \{9\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{2,10,9\}}^3 = \emptyset, \text{ (更新なし)}$$

1 段階バックトラック, $(9, 7) \notin E$ であるため, 節点 9 及び 7 からの探索は省略する.

1 段階バックトラック,

$$SUBG_{\{3\}}^1 = \{4, 5, 6, 12\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$EXT_4^2 = SUBG_{\{3\}}^1 - \{4\} - \{5, 6\} = \{12\},$$

$$SUBG_{\{3\}}'^1 = \{4\} \cup \{12\} \cup \{5, 6\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset,$$

$$EXT_5^3 = SUBG_{\{3,4\}}^2 - \{5\} - \emptyset = \{6\},$$

$$SUBG_{\{3,4\}}'^2 = \{5\} \cup \{6\} \cup \emptyset,$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset, \text{ (更新なし)}$$

2 段階バックトラック, $(4, 5) \in E$ であるため,

$$SUBG_{TAIL:\{4,5\}}^1 = \emptyset$$

に対し探索を実行, (更新なし) 実行終了.

結果として, MCP_1 は例題のグラフ G 中の最大クリークサイズ 4 を得る.

4.3 最大時間計算量評価

$SUBG \subseteq V$ に対する $EXPAND(SUBG)$ の最大時間計算量を $T(|SUBG|)$ として, 先ず, 第 3 章における最大時間計算量評価と同様に, 以下の様な非再帰手続き $EXPAND_0(SUBG)$ を考え, その最大時間計算量評価を行う.

EXPAND₀(SUBG): アルゴリズム 2 の手続き EXPAND(SUBG) において, 1600 行の EXPAND(SUBG_u), 2200 行の EXPAND(SUBG_{v_i}) 及び 2405 行の EXPAND(SUBG_{TAIL}) をそれぞれ EXPAND(∅) で置き換えた非再帰手続き.

EXPAND₀(SUBG) の実行時間については, 次の補題が成立する.

補題 4.3.1 隣接リスト表現されたグラフ $G = (V, E)$ において, $SUBG \subseteq V$, 節点 u を $G(SUBG)$ における最大次数節点とし, 以下の様にそれぞれ定義する.

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$EXT_u[i, k] = \{v_i, v_{i+1}, \dots, v_k\} \quad (1 \leq i \leq k) \text{ および,}$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u).$$

このとき, EXPAND₀(SUBG) の最大時間計算量は $O(|SUBG| \cdot |SUBG_u|^2)$ である.

(証明) 第3章のアルゴリズム MCP₀ に対して, 本章において追加した 1401-1409 行, 及び 2401-2407 行以外の部分については, 前章補題 3.6.1 において題意の成立を示した. 従ってここでは上記の追加部分に対して題意が成立することを示す.

1403-1407 各行における操作は, いずれも $O(|SUBG_u|)$ -時間で実行可能である. よって, 1401-1409 行における for 文全体は, $O(|EXT_u| \cdot |SUBG_u|) = O(|SUBG| \cdot |SUBG_u|)$ -時間で実行可能である.

2401-2407 各行における操作についても, $O(|SUBG| \cdot |SUBG_u|)$ -時間で実行可能であるから, EXPAND₀(SUBG) の最大時間計算量は $O(|SUBG| \cdot |SUBG_u|^2)$ である. □

補題 4.3.1 の結果から, EXPAND₀(SUBG) の実行時間の上限を, ある定数 C を用いて $C|SUBG| \cdot |SUBG_u|^2$ と表す.

第3章, 補題 3.6.2 と同様に, $T(|SUBG|)$ に対して以下が成立する. 集合 $SUBG \subseteq V$ に対して, 節点 u を $G(SUBG)$ における最大次数節点とし, 以下の様にそれぞれ定義する.

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

また, $(v_{k-1}, v_k) \notin E$ である場合,

$$SUBG_{TAIL} = \emptyset,$$

$(v_{k-1}, v_k) \in E$ である場合,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u.$$

これらの定義のもと, 次の各関係式が成立する.

まず, $(v_{k-1}, v_k) \notin E$ である場合,

$$T(|SUBG|) \leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + C|SUBG| \cdot |SUBG_u|^2. \quad (4.3.1)$$

この場合, $G(EXT_u)$ 中には少なくとも1組非隣接節点对が存在する. 即ち $G(EXT_u)$ は非クリークである.

これに対して, $(v_{k-1}, v_k) \in E$ である場合, $G(EXT_u)$ はクリークである. このとき,

$$\begin{aligned} & T(|SUBG|) \\ & \leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2. \end{aligned} \quad (4.3.2)$$

この場合の上界は, $G(EXT_u)$ が非クリークである場合以上であるため, 以降の解析においてはこの上界を採用する.

前章の MCP₀ に対する解析過程においては, 解析中特に場合分けを設けなかった. これに対して, MCP₁ において新たに“部分問題の統合による探索領域削減”を導入したことにより, $G(EXT_u)$ がクリークである場合における計算量上界は, 非クリークである場合におけるもの以上となるという結果を得ている.

前章同様, 主要定理証明に先立ち, 先ず次の補題を示す.

補題 4.3.2 節点集合 $SUBG \subseteq V$ に対して, $|SUBG| = \eta$, $C' = 10^4 C$ とおく. このとき, $EXPAND(SUBG)$ の最大時間計算量 $T(|SUBG|)$ は以下を満たす.

$$T(|SUBG|) < C' 2^{0.3828\eta} \eta^3.$$

(証明) 証明は $\eta \geq 1$ に関する数学的帰納法による.

先ず, $\eta = 1$ の場合, 題意は明らかに成立する.

ある $\eta \geq 1$ 以下の全ての正整数において題意が成立すると仮定する. この仮定のもとで, $|SUBG| = \eta + 1$ である場合を考える.

前章と同様に, u を $G(SUBG)$ における最大次数節点として,

$$SUBG_u = \Gamma(u) \cap SUBG$$

として,

$$|SUBG_u| = \eta - k \quad (0 \leq k < \eta)$$

とする. このとき,

$$\begin{aligned} |EXT_u| &= |SUBG - \{u\} - SUBG_u| \\ &= |SUBG| - |\{u\}| - |SUBG_u| \\ &= (\eta + 1) - 1 - (\eta - k) \\ &= k. \end{aligned}$$

ここで, $|EXT_u|$ のサイズにより場合分けを行う.

(a) $|EXT_u| = k \geq 2$ である場合:

式 (4.3.2) より,

$$\begin{aligned} T(|SUBG|) &= T(\eta + 1) \\ &\leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2 \\ &< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\ &\leq T(\eta - k) + (k - 2)T(\eta - k) + T(|SUBG_{TAIL}|) + C(\eta + 1)^3. \end{aligned}$$

ここで,

$$\begin{aligned} & |SUBG_{TAIL}| \\ & \leq |SUBG_{v_{k-1}}| - 1 \\ & \leq |SUBG_u| - 1 \\ & = \eta - k - 1 \end{aligned}$$

である. 従って,

$$\begin{aligned} & T(\eta + 1) \\ & < (k - 1)T(\eta - k) + T(\eta - k - 1) + C(\eta + 1)^3. \end{aligned}$$

帰納法の仮定により,

$$\begin{aligned} & T(\eta + 1) \\ & < (k - 1)C'2^{0.3828(\eta-k)}(\eta - k)^3 + C'2^{0.3828(\eta-k-1)}(\eta - k - 1)^3 + C(\eta + 1)^3 \\ & < (k - 1)C'2^{0.3828(\eta-k)}(\eta + 1)^3 + C'2^{0.3828(\eta-k-1)}(\eta + 1)^3 + C(\eta + 1)^3 \\ & < \left(\frac{k-1+2^{-0.3828}}{2^{0.3828k}} + 10^{-4}\right)C'2^{0.3828\eta}(\eta + 1)^3. \end{aligned}$$

ここで, 第3章補題 3.6.3, 証明と同様にして, $k \geq 0$ において

$$\frac{k-1+2^{-0.3828}}{2^{0.3828k}} < 1.3034$$

であることを証明できる (左辺は $k = 4$ において最大値 1.3033... をとる). よって,

$$\begin{aligned} & T(\eta + 1) \\ & < (1.3034 + 0.0001)C'2^{0.3828\eta}(\eta + 1)^3 \\ & = 1.3035C'2^{0.3828\eta}(\eta + 1)^3 \quad (\text{ここで, } 2^{0.3828} = 1.3038\dots) \\ & < 2^{0.3828} \cdot C'2^{0.3828\eta}(\eta + 1)^3 \\ & = C'2^{0.3828(\eta+1)}(\eta + 1)^3. \end{aligned}$$

以上により, $k \geq 2$ である場合における題意の成立は示された.

(b) $|EXT_u| \leq 1$ である場合:

$SUBG$ 中の節点のうち, u からの探索のみが実行される. 従って,

$$\begin{aligned} & T(|SUBG|) \\ & = T(\eta + 1) \\ & < T(|SUBG_u|) + C|SUBG|^3 \\ & < (1 + 0.0001)C'2^{0.3828(\eta-k)}(\eta + 1)^3 \end{aligned}$$

$$\begin{aligned} &< 2^{0.3828} \cdot C' 2^{0.3828\eta} (\eta + 1)^3 \\ &= C' 2^{0.3828(\eta+1)} (\eta + 1)^3. \end{aligned}$$

即ち、この場合にも題意は成立する。

以上、および帰納法による帰結により、任意の $\eta \geq 1$ において、題意は成立する。 □

補題 4.3.2 により、本章の結論として次の定理を得る。

定理 4.3.1 節点数 $|V| = n$, 最大次数 $\Delta \geq 1$ なるグラフに対して、いま定数 $C' = 10^4 C$, $C'' = 6.7 \cdot 10^{11}$, とする。このとき Δ が $\Delta \leq 2.610d \lg n$ ($d \geq 0$:定数) を満たすならば、アルゴリズム MCP₁ は与えられたグラフの最大クリークサイズを $O(n^{1+d})$ -時間で抽出可能である。

(証明) 第3章, 定理 3.6.1 と同様。 □

第 5 章

アルゴリズム MCP'_1

5.1 概要

第 3 章から第 4 章における定量的改善は, “部分問題の統合による探索領域削減” という新たな手法の導入によっている. 本章では, この部分問題の統合による探索領域削減をより詳細に再定義・解析することにより, 更に探索領域の削減を進め, 新たに次の改良結果を与える.

「節点数 n の一般グラフにおいて, 最大次数 Δ が $\Delta \leq 2.773d \lg n$ ($d \geq 0$: 定数) なる条件を満たしているとき, このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である .」

5.2 アルゴリズム MCP'_1

アルゴリズム 3 にアルゴリズム MCP'_1 を示す. このアルゴリズムは前章のアルゴリズム MCP_1 における新規限定操作, “部分問題の統合による探索領域削減” の手法をより詳細化して導入したものであり, 記述されたアルゴリズムとしては第 3 章のアルゴリズム MCP_0 の直接的改良である. 前章同様, MCP_0 に対して, MCP'_1 における追加, 及び変更部分はこれを赤字により表す. また, 前章までのアルゴリズムと記述が同一である部分については, 基本的に記述を省略する. 但し, 本章における追加部分との関連が深い操作については, これをコメントとして記載する. (以降の各章においても, 同様の記述方法をとる.)

アルゴリズム 3 アルゴリズム MCP'_1

0000: **procedure** $MCP'_1(G)$

/* 0100 - 0500 行 (略) */

0600: **end** {of MCP'_1 }0700: **procedure** **EXPAND**($SUBG$)

/* 0800 - 1400 行 (略) */

/* u は $SUBG$ 中の最大次数節点, $SUBG_u := \Gamma(u) \cap SUBG$ */1401: $R_u := \emptyset$;1402: **if** $k \geq 2$ **and** $(v_{k-1}, v_k) \in E$ **then**1403: $SUBG_{TAIL}$ $:= \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u$;1404: $R := SUBG_u - SUBG_{TAIL}$;1405: **if** $|R| = 1$ **then** /* Case(a) */1406: $SUBG_u := \emptyset$ 1407: **else**1408: **if** $|R| = 2$ **then** /* Case(b) */1409: **if** $(R[1], R[2]) \in E$ **then**1410: $SUBG_u := \hat{\Gamma}(R) \cap SUBG_u$;1411: $R_u := R$ 1412: **else** $SUBG_u := \emptyset$ **fi fi**1413: **fi** /* Case(c) */1414: **fi**

1500: $Q_size := Q_size + |\{u\}| + |R_u|$;1600: **EXPAND**($SUBG_u$)1700: $Q_size := Q_size - |\{u\}| - |R_u|$;

/* 1800 - 2400 行 (略) */

2401: **if** $(v_{k-1}, v_k) \in E$ **then**2402: $Q_size := Q_size + |\{v_{k-1}, v_k\}|$;2403: **EXPAND**($SUBG_{TAIL}$)2404: $Q_size := Q_size - |\{v_{k-1}, v_k\}|$;2405: **fi**

/* 2500 - 2700 行 (略) */

2800: **end** {of **EXPAND**}

5.3 部分問題の統合による探索領域削減の詳細化

前章において、 $|EXT_u| = k \geq 2$ である場合に対して、 EXT_u 中の最後尾から 2 番目の節点 v_{k-1} の隣接部分に関する考察を行い、新たな限定操作として“部分問題の統合による探索領域削減”を提唱した。この操作の概要は以下の通り。

$(v_{k-1}, v_k) \notin E$ である場合:

$$\omega(\{v_{k-1}\} \cup SUBG_{v_{k-1}}) \leq \omega(\{u\} \cup SUBG_u)$$

により、節点 v_{k-1} からの探索は省略する。

$(v_{k-1}, v_k) \in E$ である場合:

$$\omega(\{v_{k-1}\} \cup SUBG_{v_{k-1}}) > \omega(\{u\} \cup SUBG_u)$$

が成立するのは, $\{v_{k-1}\} \cup SUBG_{v_{k-1}}$ 中の最大クリークが節点 v_k を含んでいる場合に限られる (前章, 補題 4.2.1).

そこで, 節点 v_{k-1} からの探索においては,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u \quad (\subsetneq SUBG_{v_{k-1}})$$

に対し探索を行う.

ここで, 更なる探索領域の削減を行うため, $SUBG_{TAIL}$ のサイズに関して場合分けを行い, 次の補題を示す. 本章における探索領域の更なる削減は, この補題の結果を用いた手法により達成されている.

いま, $SG = SUBG_u$, $SG' = SUBG_{TAIL}$, とそれぞれおくと,

$$R = SG - SG' \geq 1$$

である. このとき, 第2章, 補題 2.2.5 から次が成立する.

補題 5.3.1 グラフ $G = (V, E)$ において, $G(SUBG)$ ($SUBG \subseteq V$) における最大次数節点を u ,

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|)$$

とそれぞれおく. ここで,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u, \text{ 及び,}$$

$$R = SUBG_u - SUBG_{TAIL}$$

とおくと, 以下が成立する.

(1) $|R| = 1$ である場合,

$$\omega(\{v_{k-1}, v_k\} \cup SUBG_{TAIL}) \geq \omega(\{u\} \cup SUBG_u).$$

(2) $|R| = 2$ である場合,

$$\omega(\{v_{k-1}, v_k\} \cup SUBG_{TAIL}) < \omega(\{u\} \cup SUBG_u)$$

が成立するのは, $SUBG_u$ 中の任意の最大クリークが集合 $SUBG_u - SUBG_{TAIL}$ 中の全ての節点 (2 個) を含む場合だけである.

(証明) (1) 補題 2.2.5 より,

$$\omega(SUBG_u) \leq \omega(SUBG_{TAIL}) + 1$$

である. よって,

$$\omega(SUBG_u) + 1 \leq \omega(SUBG_{TAIL}) + 2$$

から,

$$\omega(\{u\} \cup SUBG_u) \leq \omega(\{v_{k-1}, v_k\} \cup SUBG_{TAIL}).$$

(2) 補題 2.2.5 より,

$$\omega(SUBG_u) > \omega(SUBG_{TAIL}) + 1,$$

$$\omega(SUBG_u) + 1 > \omega(SUBG_{TAIL}) + 2,$$

即ち,

$$\omega(\{u\} \cup SUBG_u) > \omega(\{v_{k-1}, v_k\} \cup SUBG_{TAIL})$$

が成立するのは, $SUBG_u$ が R 中の節点を 2 個とも含む場合だけである. \square

補題 5.3.1 の結果を利用して, アルゴリズムに以下の新たな限定操作を導入する. この操作は,

$$R = SUBG_u - SUBG_{TAIL}$$

としたとき, 集合 R のサイズによる場合分けに基づき実行される.

Case(a) $|R| = 1$ である場合:

$$\omega(\{v_{k-1}, v_k\} \cup SUBG_{TAIL}) \geq \omega(\{u\} \cup SUBG_u)$$

であるから, $SUBG_u := \emptyset$ と設定して, 節点 u からの探索は省略する. (1406 行)

Case(b) $|R| = 2$ である場合:

$$R = \{r_1, r_2\} \text{ とする.}$$

$$\omega(\{v_{k-1}, v_k\} \cup SUBG_{TAIL}) < \omega(\{u\} \cup SUBG_u)$$

であるならば, $\{u\} \cup SUBG_u$ 中の最大クリークは R 中の 2 節点を 必ず含む. そこで, $SUBG_u$ を,

$$\hat{\Gamma}(R) \cap SUBG_u,$$

で置き換える. (1410 行) 即ち, $SUBG_u$ の部分集合のうち R の 2 節点に必ず隣接する集合に対して EXPAND を実行する. このとき節点集合,

$$R \cup \{u\} \cup (\hat{\Gamma}(R) \cap SUBG_u)$$

中の最大クリークは R 中の 2 節点を必ず含む.

但し, $(r_1, r_2) \notin E$ である場合,

$$R \cup \{u\} \cup (\hat{\Gamma}(R) \cap SUBG_u)$$

中の最大クリークが R 中の 2 節点をともに含むことはない. 従ってこの場合には Case(a) と同様に,

$$SUBG_u := \emptyset$$

とする. (1412 行)

Case(b) における置き換え操作,

$$SUBG_u := \hat{\Gamma}(R) \cap SUBG_u$$

が実行された場合には, 節点集合 R_u を

$$R_u := R$$

と設定する. (1411 行) この設定のもと, 1500 行, 即ち 1600 行における $SUBG_u$ に対する EXPAND の実行前において, 探索中のクリークに節点 u とともに集合 R_u を加える. また, $SUBG_u$ に対する EXPAND の実行後において探索中のクリークから節点 u とともに集合 R_u を除く. (1700 行)

Case(b) 以外の場合においては,

$$R_u := \emptyset \text{ と設定される. (1401 行)}$$

ここで, 次が成立する.

補題 5.3.2 $R = SUBG_u - SUBG_{TAIL} = \{r_1, r_2\}$ とするとき,

$$|\hat{\Gamma}(R) \cap SUBG_u| \leq |SUBG_u| - 2.$$

(証明) 集合 $\hat{\Gamma}(R)$ は $r_1, r_2 \in SUBG$ をともに含まないことによる. □

Case(c) $|R| \geq 3$ である場合:

この場合は, $SUBG_u$ に対して特別な操作は行わず, そのまま探索を行う.

以上の操作全体を合わせて、改めて拡張した意味において“部分問題の統合による探索領域削減”と呼ぶ。

前章において、“部分問題の統合による探索領域削減”を導入したとき、あわせて次の節点並び替え操作も導入した。即ち、 $G(EXT_u)$ が非クリークである場合、 $G(EXT_u)$ 中の非隣接節点对のうち1組を、 EXT_u の最後尾2節点 v_{k-1} 及び v_k として配置する。

$G(EXT_u)$ が非クリークである場合、この操作により v_{k-1} 、及び v_k からの探索はともに省略した。この操作は本章のアルゴリズム MCP'_1 においても導入可能であり、これにより更なる探索領域削減が可能である。しかしながら、 MCP'_1 に対する計算量評価においては、この操作が実行された場合における計算量上界は結果的に小さなものとなる。従って、本章の最大時間計算量評価においては、この操作の有無は結果に影響しない。

以上の理由から、 MCP'_1 においては上記の節点入れ替え操作は導入していない。

5.3.1 例題

図 5.1 のグラフ $G = (V, E)$ (第3章、及び第4章において用いたグラフと、同一のグラフ) に対するアルゴリズム MCP'_1 の実行過程を、図 5.2 の探索森として表す。

以下の実行過程における集合等の表記、及び図 5.2 中における表記法は、前章例題におけるものを踏襲する。

本章において新たに拡張した、“部分問題の統合による探索領域削減”について、図中における表記法を定義する。

探索森の深さ d ($0 \leq d \leq |V|$) において、前章表記に従い、Case(b) の置き換え操作が実行された場合は、

$$R = SUBG_{u_d}^{d+1} - SUBG_{TAIL:\{p_0, p_1, \dots, p_{d-1}, v_{k-1}, v_k\}}^{d+1}$$

のもとに、

$$SUBG_{u_d}^{d+1} := \hat{\Gamma}(R) \cap SUBG_{u_d}^{d+1}$$

とする。このとき、 $R = \{r_1, r_2\}$ として、

$$\begin{aligned} & \hat{\Gamma}(R) \cap SUBG_{u_d}^{d+1} \\ &= SUBG_{\{p_0, p_1, \dots, p_{d-1}, u_d, r_1, r_2\}}^{d+1}. \end{aligned}$$

この場合、図中においては各節点 u_d 、 r_1 、及び r_2 を、2重線で結ぶことにより表記する。

以下は具体的実行過程となる。

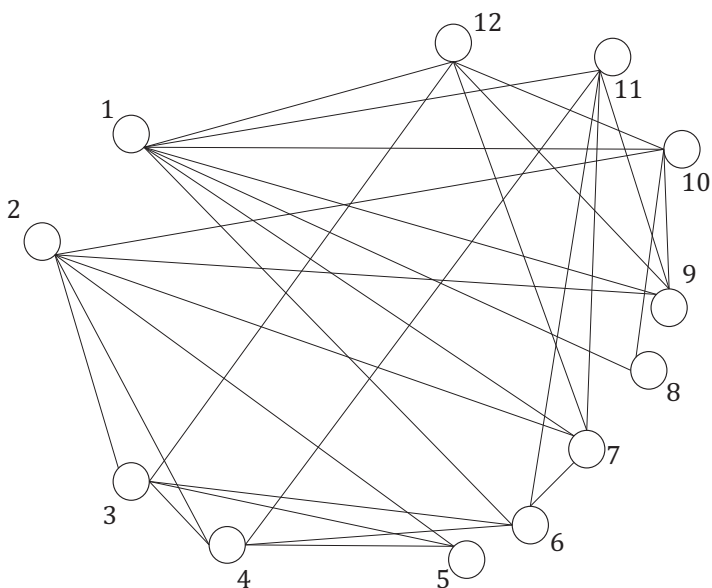


図 5.1 例題

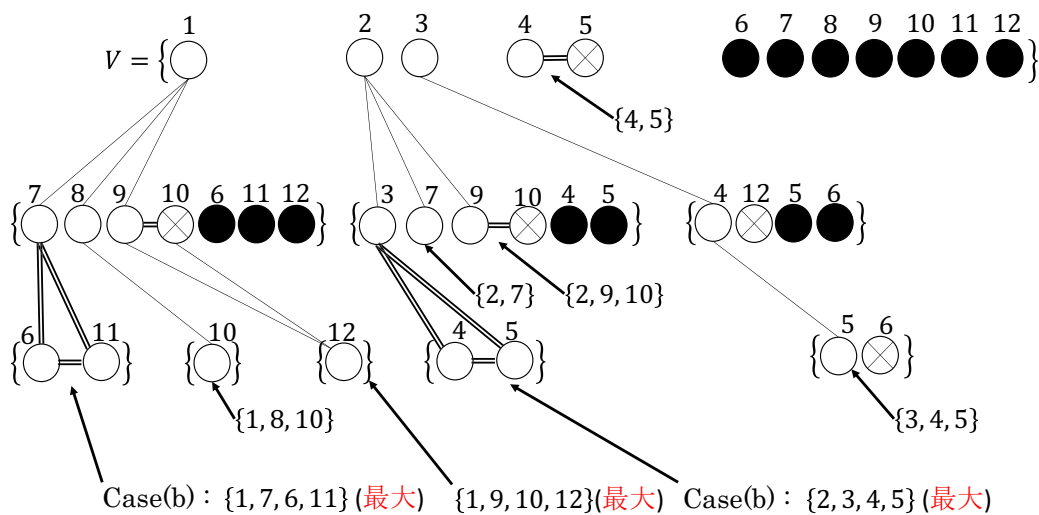


図 5.2 MCP'_1 によるクリーク探索森

$(V =)$

$SUBG^0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\},$

$SUBG^1_{\{1\}} = \{6, 7, 8, 9, 10, 11, 12\},$

$EXT^1 = \{2, 3, 4, 5\},$

$$SUBG'^0 = \{1\} \cup \{2, 3, 4, 5\} \cup \{6, 7, 8, 9, 10, 11, 12\},$$

$(4, 5) \in E$ であるため,

$$SUBG^1_{TAIL:\{4,5\}} = \emptyset$$

を構成する. ここで,

$$|R| = |SUBG^1_{\{1\}} - SUBG^1_{TAIL:\{4,5\}}| = |\{6, 7, 8, 9, 10, 11, 12\}| = 7.$$

即ち Case(c): $|R| \geq 3$ であるから, $SUBG^1_{\{1\}}$ に対して特別な操作は行わず, そのまま EXPAND を実行する.

$$SUBG^1_{\{1\}} = \{6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG^2_{\{1,7\}} = \{6, 11, 12\},$$

$$EXT^2_7 = \{8, 9, 10\},$$

$$SUBG'^1_{\{1\}} = \{7\} \cup \{8, 9, 10\} \cup \{6, 11, 12\},$$

$(9, 10) \in E$ であるため,

$$SUBG^2_{TAIL:\{1,9,10\}} = \{12\}$$

を構成する. ここで,

$$|R| = |SUBG^2_{\{1,7\}} - SUBG^2_{TAIL:\{1,9,10\}}| = |\{6, 11\}| = 2.$$

即ち Case(b): $|R| = 2$, かつ $(6, 11) \in E$ であるから,

$$SUBG^2_{\{1,7\}}$$

$$:= \hat{\Gamma}(\{6, 11\}) \cap SUBG^2_{\{1,7\}}$$

$$= SUBG^2_{\{1,7,6,11\}} = \emptyset$$

として, $SUBG^2_{\{1,7,6,11\}}$ に対して EXPAND を実行する.

$$SUBG^2_{\{1,7,6,11\}} = \emptyset, \text{ (更新: } Qmax_size := 4)$$

1 段階バックトラック,

$$EXT^2_7 = \{8, 9, 10\},$$

$$SUBG^2_{\{1,8\}} = \{10\},$$

$$SUBG^3_{\{1,8,10\}} = \emptyset,$$

$$EXT^3_9 = \emptyset,$$

$$SUBG'^2_{\{1,8\}} = \{10\} \cup \emptyset \cup \emptyset,$$

$$SUBG^3_{\{1,8,10\}} = \emptyset, \text{ (更新なし)}$$

1 段階バックトラック,

$$\begin{aligned}
& SUBG_{TAIL:\{1,9,10\}}^2 \\
&= SUBG_{\{1,9,10\}}^2 = \{12\}, \\
& SUBG_{\{1,9,10,12\}}^3 = \emptyset, \\
& EXT_{12}^3 = \emptyset, \\
& SUBG'_{\{1,9,10\}}^2 = \{12\} \cup \emptyset \cup \emptyset, \\
& SUBG_{\{1,9,10,12\}}^3 = \emptyset, \text{ (更新なし)}
\end{aligned}$$

2 段階バックトラック,

$$\begin{aligned}
& EXT_1^1 = \{2, 3, 4, 5\}, \\
& SUBG_{\{2\}}^1 = \{3, 4, 5, 7, 9, 10\}, \\
& SUBG_{\{2,3\}}^2 = \{4, 5\}, \\
& EXT_3^2 = \{7, 9, 10\}, \\
& SUBG'_{\{2\}}^1 = \{3\} \cup \{7, 9, 10\} \cup \{4, 5\}, \\
& (9, 10) \in E \text{ であるため,} \\
& SUBG_{TAIL:\{2,9,10\}}^2 = \emptyset
\end{aligned}$$

を構成する. ここで,

$$|R| = |SUBG_{\{2,3\}}^2 - SUBG_{TAIL:\{2,9,10\}}^2| = |\{4, 5\}| = 2.$$

即ち Case(b): $|R| = 2$, かつ $(4, 5) \in E$ であるから,

$$\begin{aligned}
& SUBG_{\{2,3\}}^2 \\
&:= \hat{\Gamma}(\{4, 5\}) \cap SUBG_{\{2,3\}}^2 \\
&= SUBG_{\{2,3,4,5\}}^2 = \emptyset
\end{aligned}$$

として, $SUBG_{\{2,3,4,5\}}^2$ に対して EXPAND を実行する.

$$SUBG_{\{2,3,4,5\}}^2 = \emptyset, \text{ (更新なし)}$$

1 段階バックトラック,

$$\begin{aligned}
& EXT_3^2 = \{7, 9, 10\}, \\
& SUBG_{\{2,7\}}^2 = \emptyset, \text{ (更新なし)}
\end{aligned}$$

1 段階バックトラック,

$$SUBG_{TAIL:\{2,9,10\}}^2 = \emptyset, \text{ (更新なし)}$$

1 段階バックトラック,

$$SUBG_{\{3\}}^1 = \{4, 5, 6, 12\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$EXT_4^2 = \{12\},$$

$$SUBG_{\{3\}}^1 = \{4\} \cup \{12\} \cup \{5, 6\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset,$$

$$EXT_5^3 = \{6\},$$

$$SUBG_{\{3,4\}}^2 = \{5\} \cup \{6\} \cup \emptyset,$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset, \text{ (更新なし)}$$

2段階バックトラック,

$$SUBG_{TAIL:\{4,5\}}^1 = \emptyset, \text{ (更新なし)}$$

実行終了.

結果として, MCP'_1 は例題のグラフ G 中の最大クリークサイズ 4 を得る.

5.4 最大時間計算量評価

前章までと同様に, 先ず次の非再帰手続き $EXPAND_0(SUBG)$ の時間計算量を考える.

EXPAND₀(SUBG): アルゴリズム 3 の $EXPAND(SUBG)$ において, 再帰呼び出し部分 1600, 2200, 及び 2403 行の $EXPAND(SUBG_u)$, $EXPAND(SUBG_{v_i})$, 及び $EXPAND(SUBG_{TAIL})$ を, それぞれ $EXPAND(\emptyset)$ に置き換えた非再帰手続き.

補題 5.4.1 $EXPAND_0(SUBG)$ の最大時間計算量は, $O(|SUBG| \cdot |SUBG_u|^2)$ である.

(証明) 第3章のアルゴリズム MCP_0 に対して, 本章において追加した 1401-1414 行及び 2401-2405 行以外の部分については, 補題 3.6.1 において題意の成立を示した. 従ってここでは上記の追加部分に対して題意が成立することを示す. 1402-1412 行における操作は, いずれも $O(|SUBG_u|)$ -時間で実行可能である. 2401-2404 行における操作についても, $O(|SUBG_u|)$ -時間で実行可能であるから, $EXPAND_0(SUBG)$ の最大時間計算量は $O(|SUBG| \cdot |SUBG_u|^2)$ である. \square

補題 5.4.1 の結果から, $\text{EXPAND}_0(\text{SUBG})$ の実行時間の上限を, ある定数 C を用いて $C|\text{SUBG}| \cdot |\text{SUBG}_u|^2$ と表す.

第3章, 補題 3.6.2 と同様に, $T(|\text{SUBG}|)$ に対して以下が成立する. 集合 $\text{SUBG} \subseteq V$ に対して, 節点 u を $G(\text{SUBG})$ における最大次数節点とし, 以下の様にそれぞれ定義する.

$$\text{SUBG}_u = \Gamma(u) \cap \text{SUBG},$$

$$\text{EXT}_u = \text{SUBG} - \{u\} - \text{SUBG}_u = \{v_1, v_2, \dots, v_k\} \quad (k = |\text{EXT}_u|),$$

$$\text{SUBG}_{\text{TAIL}} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap \text{SUBG}_u.$$

このとき,

$$\begin{aligned} & T(|\text{SUBG}|) \\ & \leq T(|\text{SUBG}_u|) + \sum_{i=1}^{k-2} T(|\text{SUBG}_{v_i}|) + T(|\text{SUBG}_{\text{TAIL}}|) + C|\text{SUBG}| \cdot |\text{SUBG}_u|^2. \end{aligned} \tag{5.4.1}$$

(前章, 式 (4.3.2) と同一.)

前章までと同様に, 先ず $\text{EXPAND}(\text{SUBG})$ の時間計算量評価結果を以下に与える.

補題 5.4.2 節点集合 $\text{SUBG} \subseteq V$ に対して, $|\text{SUBG}| = \eta$, $C' = 10^4 C$ とおく. このとき, $\text{EXPAND}(\text{SUBG})$ の最大時間計算量 $T(|\text{SUBG}|)$ は以下を満たす.

$$T(|\text{SUBG}|) \leq C' 2^{0.3603\eta} \eta^3.$$

(証明) 証明は, $\eta \geq 1$ に関する数学的帰納法による.

$\eta = 1$ である場合, 題意は明らかに成立する.

次に, ある $\eta \geq 1$ 以下の全ての正整数において題意が成立すると仮定する. この仮定のもとで, $|\text{SUBG}| = \eta + 1$ である場合を考える.

前章と同様に, u を $G(\text{SUBG})$ における最大次数節点として,

$$\text{SUBG}_u = \Gamma(u) \cap \text{SUBG}$$

として,

$$|\text{SUBG}_u| = \eta - k \quad (0 \leq k < \eta)$$

とする. このとき,

$$\begin{aligned}
& |EXT_u| \\
&= |SUBG - \{u\} - SUBG_u| \\
&= |SUBG| - |\{u\}| - |SUBG_u| \\
&= (\eta + 1) - 1 - (\eta - k) \\
&= k.
\end{aligned}$$

まず, $k \leq 1$ である場合には, 第4章, 補題 4.3.2 証明, 場合分けの (b) と同様にして示せる.

そこで, 以下では $k \geq 2$ である場合のみを考える.

ここで, 更に場合分けを行う.

(1) $(v_{k-1}, v_k) \in E$ である場合:

$$R = SUBG_u - SUBG_{TAIL}$$

として, R のサイズにより更に以下の場合分けを行う.

Case(c) $|R| \geq 3$ である場合 :

式 (5.4.1) より,

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&\leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2 \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3.
\end{aligned}$$

ここで,

$$|SUBG_{v_i}| \leq |SUBG_u| = \eta - k,$$

及び,

$$|SUBG_{TAIL}| \leq |SUBG_u| - 3 = \eta - k - 3$$

と帰納法の仮定により,

$$\begin{aligned}
& T(\eta + 1) \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\
&\leq (k-1)T(\eta - k) + T(\eta - k - 3) + C(\eta + 1)^3
\end{aligned}$$

$$\begin{aligned}
&\leq (k-1)C'2^{0.3603(\eta-k)}(\eta-k)^3 + C'2^{0.3603(\eta-k-3)}(\eta-k-3)^3 + C(\eta+1)^3 \\
&< (k-1)C'2^{0.3603(\eta-k)}(\eta+1)^3 + C'2^{0.3603(\eta-k-3)}(\eta+1)^3 + C(\eta+1)^3 \\
&< \left(\frac{k-1+2^{-0.3603 \cdot 3}}{2^{0.3603k}} + 10^{-4}\right)C'2^{0.3603\eta}(\eta+1)^3.
\end{aligned}$$

ここで, 第3章, 補題 3.6.3 証明と同様にして,

$$\frac{k-1+2^{-0.3603 \cdot 3}}{2^{0.3603k}} < 1.2832$$

(左辺は $k=5$ において最大値 1.28312... をとる.) が成立する. 従って,

$$\begin{aligned}
&T(|SUBG|) \\
&= T(\eta+1) \\
&< \left(\frac{k-1+2^{-0.3603 \cdot 3}}{2^{0.3603k}} + 10^{-4}\right)C'2^{0.3603\eta}(\eta+1)^3 \\
&< 1.2833 \cdot C'2^{0.3603\eta}(\eta+1)^3 \\
&< 2^{0.3603} \cdot C'2^{0.3603\eta}(\eta+1)^3 \quad (\text{ここで, } 2^{0.3603} = 1.2836\dots) \\
&= C'2^{0.3603(\eta+1)}(\eta+1)^3.
\end{aligned}$$

Case(b) $|R|=2$ である場合 ;

式 (5.4.1) より,

$$\begin{aligned}
&T(|SUBG|) \\
&= T(\eta+1) \\
&\leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2 \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3.
\end{aligned}$$

Case(b) の操作により, このとき,

$$SUBG_u := \hat{\Gamma}(R) \cap SUBG_u \quad (R = SUBG_u - SUBG_{TAIL}),$$

又は,

$$SUBG_u := \emptyset$$

との置き換えがなされている. 従って,

$$|SUBG_u| \leq \eta - k - 2,$$

及び,

$$|SUBG_{TAIL}| = \eta - k - 2$$

と帰納法の仮定により,

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\
&\leq T(\eta - k - 2) + (k - 2)T(\eta - k) + T(\eta - k - 2) + C(\eta + 1)^3 \\
&= 2T(\eta - k - 2) + (k - 2)T(\eta - k) + C(\eta + 1)^3 \\
&\leq 2C'2^{0.3603(\eta-k-2)}(\eta - k - 2)^3 + (k - 2)C'2^{0.3603(\eta-k)}(\eta - k)^3 + C(\eta + 1)^3 \\
&< 2C'2^{0.3603(\eta-k-2)}(\eta + 1)^3 + (k - 2)C'2^{0.3603(\eta-k)}(\eta + 1)^3 + C(\eta + 1)^3 \\
&< \left(\frac{k-2+2 \cdot 2^{-0.3603 \cdot 2}}{2^{0.3603k}} + 10^{-4}\right)C'2^{0.3603\eta}(\eta + 1)^3.
\end{aligned}$$

ここで Case(c) と同様に,

$$\frac{k-2+2 \cdot 2^{-0.3603 \cdot 2}}{2^{0.3603k}} < 1.2089$$

(左辺は $k = 5$ において最大値 1.20880... をとる.) が成立する. 従って,

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&< \left(\frac{k-2+2 \cdot 2^{-0.3603 \cdot 2}}{2^{0.3603k}} + 10^{-4}\right)C'2^{0.3603\eta}(\eta + 1)^3 \\
&< 1.209 \cdot C'2^{0.3603\eta}(\eta + 1)^3 \\
&< 2^{0.3603} \cdot C'2^{0.3603\eta}(\eta + 1)^3 \\
&= C'2^{0.3603(\eta+1)}(\eta + 1)^3.
\end{aligned}$$

Case(a) $|R| = 1$ である場合 ;

式 (5.4.1) より,

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&\leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2 \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3.
\end{aligned}$$

Case(a) の操作により, このとき,

$$SUBG_u := \emptyset$$

との代入がなされている. 従って,

$$|SUBG_{TAIL}| = \eta - k - 1$$

と帰納法の仮定により,

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\
&\leq (k-2)T(\eta - k) + T(\eta - k - 1) + C(\eta + 1)^3 \\
&\leq (k-2)C'2^{0.3603(\eta-k)}(\eta - k)^3 + C'2^{0.3603(\eta-k-1)}(\eta - k - 1)^3 + C(\eta + 1)^3 \\
&< (k-2)C'2^{0.3603(\eta-k)}(\eta + 1)^3 + C'2^{0.3603(\eta-k-1)}(\eta + 1)^3 + C(\eta + 1)^3 \\
&< \left(\frac{k-2+2^{-0.3603}}{2^{0.3603k}} + 10^{-4}\right)C'2^{0.3603\eta}(\eta + 1)^3.
\end{aligned}$$

ここで Case(c) と同様に,

$$\frac{k-2+2^{-0.3603}}{2^{0.3603k}} < 1.0842$$

(左辺は $k = 5$ において最大値 1.08410... をとる.) が成立する. 従って,

$$\begin{aligned}
& T(|SUBG|) \\
&= T(\eta + 1) \\
&< \left(\frac{k-2+2^{-0.3603}}{2^{0.3603k}} + 10^{-4}\right)C'2^{0.3603\eta}(\eta + 1)^3 \\
&< 1.0843 \cdot C'2^{0.3603\eta}(\eta + 1)^3 \\
&< 2^{0.3603} \cdot C'2^{0.3603\eta}(\eta + 1)^3 \\
&= C'2^{0.3603(\eta+1)}(\eta + 1)^3.
\end{aligned}$$

(2) $(v_{k-1}, v_k) \notin E$ である場合:

上記 (1) において, $SUBG_u$ のサイズが最大となるのは Case(c) の場合である. 同場合において, $SUBG_{TAIL} = \emptyset$ とすることにより, 容易に題意の成立が示せる.

以上, および数学的帰納法により, 題意の成立は示された. \square

補題 5.4.2 より, 本章の結論である次の定理が示される.

定理 5.4.1 節点数 n , 最大次数 $\Delta \geq 1$ なるグラフにおいて, 定数 C'' を $C'' = 6.7 \cdot 10^{11}$ とおく. このとき, Δ が

$$\Delta \leq 2.773d \lg n \quad (d \geq 0: \text{定数})$$

を満たすならば, アルゴリズム MCP'_1 は与えられたグラフの最大クリークサイズを $O(n^{1+d})$ -時間で抽出可能である.

(証明) 第3章, 定理 3.6.1 と同様. \square

第 6 章

アルゴリズム MCP_2

6.1 概要

本章においては, アルゴリズム MCP_0 に, 第 4 章, 及び 5 章における探索領域削減手法を加え, 更に,

- ・ 部分問題サイズの減少
- ・ 少数部分問題の分割

という 2 種の限定操作を新規に追加する. これにより, 次の改良結果を与える.

「節点数 n の一般グラフにおいて, 最大次数 $\Delta \leq 2.994d \lg n$ ($d \geq 0$: 定数) なる条件が満たされているとき, このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である.」

6.2 アルゴリズム

本章において最大時間計算量評価を行うアルゴリズム MCP_2 をアルゴリズム 4 に示す*¹. MCP_2 の構成は次の通りである.

このアルゴリズムは第 3 章のアルゴリズム MCP_0 に対して, 先ず第 4 章, 及び 第 5 章においてそれぞれ導入した各操作をともに追加する. これらの部分については, 追加した各行を青字により表す. (1401-1411 行, 及び 1412-1423 行)

*¹ 関連論文 (4) におけるアルゴリズム MCP_2 に対して, アルゴリズムの論理構造を整理し, 記述を改めた.

上記追加部分に加えて、本章において導入した新たな探索領域削減操作を追加したものが、アルゴリズム MCP₂ である。この更なる新規追加部分に関しては、追加した各行を赤字により表す。(2001-2012行)

アルゴリズム 4 アルゴリズム MCP₂

0000: **procedure** MCP₂(G)

/* 0100 - 0500 行 (略) */

0600: **end** {of MCP₂}0700: **procedure** EXPAND($SUBG$)

/* 0800 - 1400 行 (略) */

1401: $R_u := \emptyset$;1402: **if** $k \geq 2$ **then**1403: **for** $i = 1$ **to** k **do**1404: $U_i := \Gamma(v_i) \cap EXT_u$;1405: **if** $|U_i| < k - 1$ **then**1406: $v := (EXT_u - U_i)[1]$;1407: **swap** v_i **with** v_k ;1408: **swap** v **with** v_{k-1} ;1409: **goto** 15001410: **fi**1411: **od**1412: $SUBG_{TAIL}$:= $\hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u$;1413: $R := SUBG_u - SUBG_{TAIL}$;1414: **if** $|R| = 1$ **then** /* Case(a) */1415: $SUBG_u := \emptyset$ 1416: **else**1417: **if** $|R| = 2$ **then** /* Case(b) */1418: **if** $(R[1], R[2]) \in E$ **then**1419: $SUBG_u := \hat{\Gamma}(R) \cap SUBG_u$;1420: $R_u := R$ 1421: **else** $SUBG_u := \emptyset$ **fi**1422: **fi** /* Case(c) */1423: **fi**1500: $Q_size := Q_size + |\{u\}| + |R_u$;1600: **EXPAND**($SUBG_u$)1700: $Q_size := Q_size - |\{u\}| - |R_u$;1800: **for** $i = 1$ **to** $k - 2$ **do**1900: $v_i := EXT_u[i]$;2000: $SUBG_{v_i}$:= $\Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$;2001: **if** $(v_{k-1}, v_k) \in E$ **and** $k = 4$ **then**2002: $u_i :=$ a vertex with the maximum
degree in $G(SUBG_{v_i})$;2003: **if** $|\Gamma(u_i) \cap SUBG_{v_i}|$ ≤ $|SUBG_{v_i}| - 4$ **then**2004: **for** $j := i + 1$ **to** 4 **do**2005: $SUBG_{(v_i, v_j)} := \Gamma(v_j) \cap SUBG_{v_i}$;2006: $Q_size := Q_size + |\{v_i, v_j\}|$;2007: **EXPAND**($SUBG_{(v_i, v_j)}$)2008: $Q_size := Q_size - |\{v_i, v_j\}|$;2009: $SUBG_{v_i} := SUBG_{v_i} - \{v_j\}$ 2010: **od**2011: **goto** 2400 **fi**2012: **fi**2100: $Q_size := Q_size + |\{v_i\}|$;2200: **EXPAND**($SUBG_{v_i}$)2300: $Q_size := Q_size - |\{v_i\}|$ 2400: **od**2401: **if** $(v_{k-1}, v_k) \in E$ **then**2402: $Q_size := Q_size + |\{v_{k-1}, v_k\}|$;2403: **EXPAND**($SUBG_{TAIL}$)2404: $Q_size := Q_size - |\{v_{k-1}, v_k\}|$ 2405: **fi**

/* 2500 - 2700 行 (略) */

2900: **end** {of EXPAND}1700: $Q_size := Q_size - |\{u\}| - |R_u$;

6.2.1 部分問題の統合による探索領域削減の統一

第4章, 及び第5章に続き, ここでも

$$EXT_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|)$$

に対して, $k \geq 2$ である場合, EXT_u 中の最後尾から2番目の節点 v_{k-1} からの探索について考察する.

第4章においては, 以下の場合分け (1), (2) において, それぞれ探索領域削減操作を実行した.

(1) $(v_{k-1}, v_k) \notin E$ である場合:

v_{k-1} からの探索は省略する.

(2) $(v_{k-1}, v_k) \in E$ である場合:

節点 v_{k-1} からの探索は,

$$\begin{aligned} & SUBG_{TAIL} \\ & = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u \end{aligned}$$

に対する EXPAND の実行とする.

場合分け (1) において, $G(EXT_u)$ は非クリークである. 第4章のアルゴリズム MCP₁ においては, この場合 EXT_u 中の非隣接節点2個を最後尾2節点として配置し, これにより v_{k-1} からの探索を省略した. 第5章のアルゴリズム MCP'₁ においては, この操作を導入しなかったが, MCP₂ においてはこの操作を再び導入する. (1403-1411行).

場合分け (2) において, $G(EXT_u)$ はクリークである. 第4章のアルゴリズム MCP₁ においてはこの場合, v_{k-1} からの探索を, 上記 $SUBG_{TAIL}$ に対する EXPAND の実行とした. $SUBG_{TAIL}$ に対しては,

$$|SUBG_{TAIL}| \leq |SUBG_{v_{k-1}}| - 1$$

が成立するため, これにより第4章における計算量上界の改善が得られた. 第5章のアルゴリズム MCP'₁ においては, 更にこの $|SUBG_{TAIL}|$ のサイズに応じた場合分けを伴う操作によって, 第4章からの改良結果を得ていた. これらの操作については本章のアルゴリズム MCP₂ においても引き続き用いる. (1412-1422行)

一般的に, (1) における操作のように, ある節点からの探索を単純に省略できる場合, その探索領域削減の効果は非常に大きい. これに対して, (2) における操作では探索される部

分問題のサイズの上限をより小さくできるものの、探索領域削減の効果は相対的に小さくなる。

この観点から、本章においては (2), 即ち $G(EXT_u)$ がクリークである場合において、より有効な新規探索領域削減操作を導入する。

6.2.2 部分問題サイズの減少

手続き EXPAND(*SUBG*) において、各節点 u, v_1, v_2, \dots からの探索が実行された後、これらの節点は *SUBG* から順次削除される。

u からの探索においては、探索終了によって *SUBG* から削除された節点は存在しない。これに対して、節点 $v_i \in EXT_u$ ($1 \leq i \leq k$) からの探索においては、各節点 v_1, v_2, \dots, v_{i-1} が *SUBG* から削除される。前章までにおけるアルゴリズムの計算量評価において、この操作が探索領域に与える影響は考慮されていなかった。本章においては、この操作の影響を考慮した次の補題を与える。

補題 6.2.1 節点集合 *SUBG* に対して、 u を $G(SUBG)$ 中の最大次数節点として、

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \quad (1 \leq i \leq k)$$

とする。このとき、もし $G(EXT_u)$ がクリークであるならば、

$$|SUBG_{v_i}| \leq |SUBG_u| - i + 1$$

である。

(証明)

$$SUBG_{v_i}$$

$$= \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$$

$$= \Gamma(v_i) \cap (\{v_i, v_{i+1}, \dots, v_k\} \cup SUBG_u)$$

$$= \Gamma(v_i) \cap (EXT_u[i+1, k] \cup SUBG_u).$$

定義により、 $EXT_u[i+1, k] \cap SUBG_u = \emptyset$ である。従って、

$$|SUBG_{v_i}|$$

$$= |(\Gamma(v_i) \cap EXT_u[i+1, k]) \cup (\Gamma(v_i) \cap SUBG_u)|$$

$$= |\Gamma(v_i) \cap EXT_u[i+1, k]| + |\Gamma(v_i) \cap SUBG_u|.$$

いま $G(EXT_u)$ はクリークであるから, v_i は $EXT_u[i+1, k]$ 中の全ての節点と隣接している. 従って,

$$|SUBG_{v_i}| = (k - i) + |\Gamma(v_i) \cap SUBG_u|.$$

ここで, 節点 u は $G(SUBG)$ 中の最大次数節点であるから,

$$|\Gamma(v_i) \cap SUBG_u| \leq |SUBG_u| - (k - 1)$$

でなければならない. 何故ならば,

$$|\Gamma(v_i) \cap SUBG_u| > |SUBG_u| - (k - 1)$$

であるとすると,

$$\begin{aligned} & |\Gamma(v_i) \cap SUBG| \\ &= |\Gamma(v_i) \cap (\{u\} \cup EXT_u \cup SUBG_u)| \\ &= |\Gamma(v_i) \cap \{u\}| + |\Gamma(v_i) \cap EXT_u| + |\Gamma(v_i) \cap SUBG_u| \\ &> 0 + (k - 1) + |SUBG_u| - (k - 1) \\ &= |SUBG_u|. \end{aligned}$$

すなわち, v_i は $G(SUBG)$ における次数が u よりも大となる. これは u が $G(SUBG)$ 中の最大次数節点であるとしたことに反する. 以上により,

$$\begin{aligned} & |SUBG_{v_i}| \\ &= (k - i) + |\Gamma(v_i) \cap SUBG_u| \\ &\leq k - i + (|SUBG_u| - (k - 1)) \\ &= |SUBG_u| - i + 1. \quad \square \end{aligned}$$

手続き EXPAND による $SUBG$ の分割においては, $k = |EXT_u|$ の値が大きくなるほど各部分問題のサイズ上限は小さく, 部分問題の数は大きくなる.

補題 6.2.1 の結果は, $G(EXT_u)$ がクリークである場合, 節点 v_i からの探索において, 探索領域は i の値が大きいくほど確実に小さくなることを示す. このサイズ減少は,

$$\begin{aligned} & SUBG_{v_i} \\ &= \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \end{aligned}$$

なる定義によるものであり, この定義自体は第3章のアルゴリズム MCP₀ において既に行われていた. 補題 6.2.1 の結果は, この定義による探索領域への影響をより適切に評価

したものである.

$G(EXT_u)$ がクリークであり, かつ $k = |EXT_u|$ の値が十分大きい場合には, サイズが小さい各部分問題のうち, 探索順が後のものほど更にサイズが減少する. これにより, 上記場合においては計算量上界が小さくなることを容易に示せる.

本章においては,

$k \geq 5$, かつ $G(EXT_u)$ がクリークである場合

が, 「 k の値が十分大きい」 場合に当たる.

また, 評価上 $k \leq 3$ である場合にも, 計算量上界が小さくなることは容易に示せる (詳細は, 6.3 節にて後述する).

$k = 4$, かつ $G(EXT_u)$ がクリークである場合には, ここに述べた手法のみでは本章冒頭の結果を得ることができない. そこで, 更に以下の手法を用いて探索領域を削減する.

6.2.3 少数部分問題の分割

$k = 4$, かつ $G(EXT_u)$ がクリークである場合, EXPAND($SUBG$) においては, 節点

u, v_1, v_2, v_3

からの探索がそれぞれ実行される. ここで, 節点 v_3 からの探索においては,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_3, v_4\}) \cap SUBG_u$$

に対して EXPAND が実行される. また, $SUBG_{TAIL}$ に関しては, そのサイズに応じて更なる探索領域削減操作が実行される. この状況のもと, ここでは $k = 4$ における新たな探索領域削減手法として,

節点 v_1 , および v_2 からの探索における領域削減手法

を考える.

いま $W = SUBG_u$, $W' = SUBG_{v_i}$ ($i = 1, 2$) とすると, $G(EXT_u)$ がクリークである場合,

$$R = SUBG_{v_i} - SUBG_u = EXT_u[i, k]$$

である. 従って, 第2章, 補題 2.2.4 により, 次が成立する.

補題 6.2.2 節点集合 $SUBG \subseteq V$ に対して, 節点 u を $G(SUBG)$ における最大次数節点,

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \quad (1 \leq i \leq k)$$

とする.

上記定義のもとに, 以下が成り立つ.

$$\omega(SUBG_u) < \omega(SUBG_{v_i})$$

であるならば, $SUBG_{v_i}$ 中の最大クリークは,

$$EXT_u[i+1, k] = \{v_{i+1}, v_{i+2}, \dots, v_k\}$$

中の節点を少なくとも 1 個含む.

(証明) 補題 2.2.4 による. \square

補題 6.2.2 において, $i = k - 1$ とすれば, 第 4 章, 補題 4.2.1 の結果を得る. 従って, この結果は補題 4.2.1 の拡張である.

$k = 4$ である場合, 節点 v_3 からの探索において, 最大となりうるクリークは,

v_4 を含むもの

ただ 1 個であった. そこで, 節点 v_3 からの探索においては,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_3, v_4\}) \cap SUBG_u$$

に対して EXPAND を実行した. (この定義のもと, $\{v_3, v_4\} \cup SUBG_{TAIL}$ は節点 v_4 を必ず含む.) $SUBG_{TAIL} \subsetneq SUBG_{v_3}$ であるため, この操作は探索領域を確実に削減する.

この手法と同様に, 節点 v_1 からの探索を考える. 同節点からの探索において最大となりうるクリークは,

v_2, v_3, v_4 のうち, 少なくとも 1 個を含むもの

となる. このとき, v_1 からの探索を $SUBG_{TAIL}$ のような部分問題に対する EXPAND の実行とすると, 部分問題の候補は一般に 2 個以上存在する. 即ち, 通常の v_1 からの探索における $SUBG_{v_1}$ に対して, $SUBG_{v_1}$ を 2 個以上の部分問題 に分割することになる.

ここで, 問題の分割を適切に行うため, 分割方法を規定する次の補題を示す.

補題 6.2.3 いま $G(EXT_u)$ はクリークであるとする.

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$$

とすると, 以下が成立する.

$SUBG_{v_i}$ から $\omega(SUBG_u)$ よりも大きい最大クリークを抽出するためには,

$$\begin{aligned} & SUBG_{(v_i, v_j)} \\ &= \Gamma(v_j) \cap (SUBG_{v_i} - \{v_{i+1}, v_{i+2}, \dots, v_{j-1}\}) \quad (i+1 \leq j \leq k) \end{aligned}$$

なる $k-i$ 個の各節点集合に対して EXPAND を実行すれば十分である.

更に, $SUBG_{v_i}$ 中の, $G(SUBG_{v_i})$ における最大次数節点を u_i とするとき, 以下が成立する.

$$|SUBG_{(v_i, v_j)}| \leq |\Gamma(u_i) \cap SUBG_{v_i}| - (j-i-1).$$

(証明) いま, $SUBG_{(v_i, v_j)}$ 中の最大クリークを Q とすると,

$$\{v_i, v_j\} \cup Q$$

は節点 v_j ($i+1 \leq j \leq k$) を必ず含む. よって, 補題 6.2.2 により補題前半は成立する.

次に, 後半部分

$$|SUBG_{(v_i, v_j)}| \leq |\Gamma(u_i) \cap SUBG_{v_i}| - (j-i-1).$$

を示す.

条件より, $G(EXT_u)$ はクリークであるから,

$$\{v_{i+1}, v_{i+2}, \dots, v_{j-1}\} \subseteq SUBG_{v_i},$$

及び,

$$\{v_{i+1}, v_{i+2}, \dots, v_{j-1}\} \subseteq \Gamma(v_j).$$

従って,

$$\begin{aligned} & |SUBG_{(v_i, v_j)}| \\ &= |\Gamma(v_j) \cap (SUBG_{v_i} - \{v_{i+1}, v_{i+2}, \dots, v_{j-1}\})| \\ &= |\Gamma(v_j) \cap SUBG_{v_i}| - |\Gamma(v_j) \cap \{v_{i+1}, v_{i+2}, \dots, v_{j-1}\}| \\ &= |\Gamma(v_j) \cap SUBG_{v_i}| - (j-i-1) \\ &\leq |\Gamma(u_i) \cap SUBG_{v_i}| - (j-i-1). \quad \square \end{aligned}$$

$SUBG_{(v_i, v_j)}$ は, $SUBG_{v_i}$ の真部分集合のうち,

「節点 v_j に隣接する節点の集合から, 各節点 v_{i+1}, \dots, v_{j-1} を除いたもの」

である. この定義に従い, $i = 1, 2$ において, v_i からの探索における問題分割は, 次の様に行う.

v_1 からの探索:

$SUBG_{(v_1, v_2)}$ (v_1, v_2 に隣接する節点の集合) ,

$SUBG_{(v_1, v_3)}$ (v_1, v_3 に隣接する節点の集合から, v_2 を除いたもの) ,

及び,

$SUBG_{(v_1, v_4)}$ (v_1, v_4 に隣接する節点の集合から, v_2, v_3 を除いたもの) .

$\{v_2\} \cup SUBG_{(v_1, v_2)}$ 中に, v_2, v_3 の2個を含むクリーク, v_2, v_3, v_4 を全て含むクリークが存在する場合, それらは抽出可能である. 従って, $SUBG_{(v_1, v_3)}$ は v_2 を含まないように定義する.

v_2 からの探索:

$SUBG_{(v_2, v_3)}$ (v_2, v_3 に隣接する節点の集合) ,

及び,

$SUBG_{(v_2, v_4)}$ (v_2, v_4 に隣接する節点の集合から, v_3 を除いたもの) ,

v_1, v_2 のいずれに対しても, 分割される部分問題は2個以上である. 問題の単純な置き換えと異なり, この様な分割を適切な管理なしに実行してしまうと, 解析上逆に計算量上界は大きくなる. そこで, 以下 $SUBG_{(v_i, v_j)}$ のサイズ上限の大小により, この問題分割を実行するか否かを判定する.

以上を踏まえて, MCP₂ においては $k = |EXT_u|$ の値, 及び $G(EXT_u)$ がクリークであるか否かにより場合分けを行い, それぞれ以下の操作を実行する.

(A) $k \neq 4$, 又は $G(EXT_u)$ が非クリークである場合:

$v_i \in EXT_u$ ($1 \leq i \leq k-1$) からの探索を, 前章までと同様に行う. 但し, 第4, 5章における探索領域削減操作はともに実行される.

(B) $k = 4$, かつ $G(EXT_u)$ がクリークである場合:

更に, 次のような場合分けを行う.

いま, $EXT_u = \{v_1, v_2, v_3, v_4\}$, $SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$ ($i = 1, 2$) とする. また, $SUBG_{v_i}$ 中の, $G(SUBG_{v_i})$ における最大次数節点を u_i とする.

ここで,

$$SUBG_{u_i} = \Gamma(u_i) \cap SUBG_{v_i},$$

$$EXT_{u_i} = SUBG_{v_i} - \{u_i\} - SUBG_{u_i} \quad *2$$

とする. 以下, $SUBG_{v_1}$ および $SUBG_{v_2}$ の各々に対し, $SUBG_{u_i}$ のサイズに基づく場合分けにより, それぞれ以下の操作を行う.

(B-0) $|SUBG_{u_i}| \geq |SUBG_{v_i}| - 2$ である場合;

v_i からの探索において, u_i からの探索のみを行う. 即ち, $SUBG_{u_i}$ に対してのみ EXPAND を実行する.

(B-1) $|SUBG_{u_i}| = |SUBG_{v_i}| - 3$ である場合:

$|EXT_{u_i}| = 2$ のもとに, $EXT_{u_i} = \{v'_1, v'_2\}$ とおく. v_i からの探索において, u_i からの探索, 及び v'_1 からの探索を行う.

u_i からの探索は, $SUBG_{u_i}$ に対する EXPAND の実行,

v'_1 からの探索は, $(v'_1, v'_2) \notin E$ である場合省略, $(v'_1, v'_2) \in E$ である場合には,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v'_1, v'_2\}) \cap SUBG_{u_i} \quad *3$$

に対する EXPAND の実行である. ここで, $SUBG_{TAIL}$ 構成時には第5章において導入した Case(a) ~ Case(c) の場合分けに基づく探索領域削減操作も実行する. (図 6.1)

(B-2) $|SUBG_{u_i}| \leq |SUBG_{v_i}| - 4$ である場合:

この場合には, 上記の問題分割を実行する. 即ち, 節点 v_i ($i = 1, 2$) からの探索を,

$$SUBG_{(v_i, v_j)} = \Gamma(v_j) \cap (SUBG_{v_i} - \{v_{i+1}, v_{i+2}, \dots, v_{j-1}\}) \quad (i + 1 \leq j \leq 4)$$

なる $4 - i$ 個の部分問題に対する EXPAND の実行とする. (図 6.2) アルゴリズム MCP₂

*2 文献 [82] においては, $SUBG_{u_i}$ を $SUBG_{v_{i u_i}}$, EXT_{u_i} を $EXT_{v_{i u_i}}$ とそれぞれ表記した.

*3 文献 [82] においては, この $SUBG_{TAIL}$ を $SUBG_{v_{i TAIL}}$ と表記した.

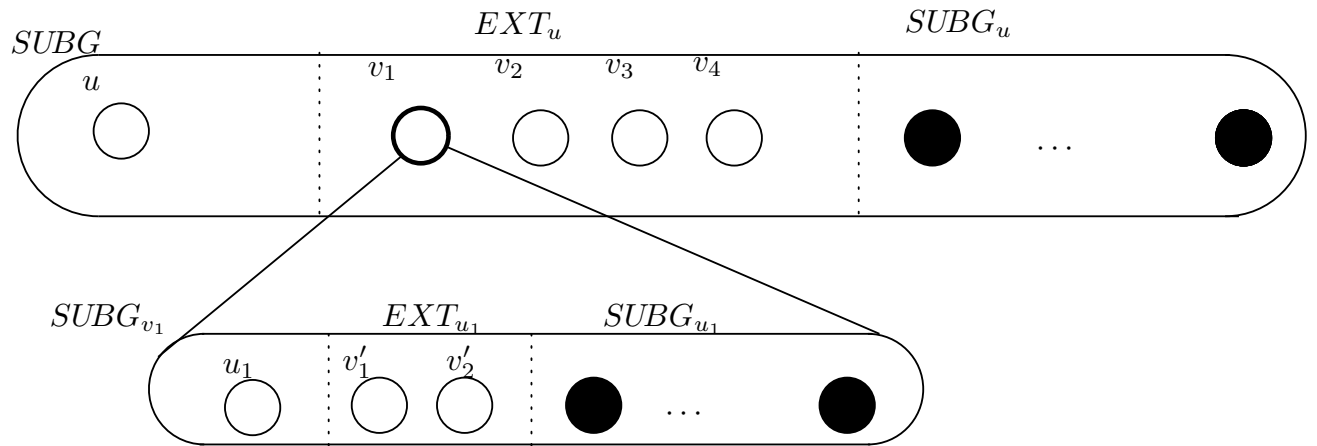
における新規追加部分 (2001- 2012 行) は, 全て (B-2) の場合における操作を実現するための部分である.

以下, (B) の操作全体をまとめて, “特別処理” と呼ぶ.

“特別処理” のうち, (B-0) あるいは (B-1) である場合における各操作は, $SUBG_{v_i}$ に対する通常の EXPAND の実行と同様である. 従って, “特別処理” における新規追加操作は, (B-2) である場合の操作である. 以下, この操作を, “少数部分問題の分割” と呼ぶ.

$k \neq 4$, かつ $G(EXT_u)$ がクリークである場合, MCP_2 は第5章のアルゴリズム MCP'_1 と全く同様に動作する. 但し, 後述の解析過程においては, 補題 6.2.1 の影響を考慮するため, 異なる評価結果を得る.

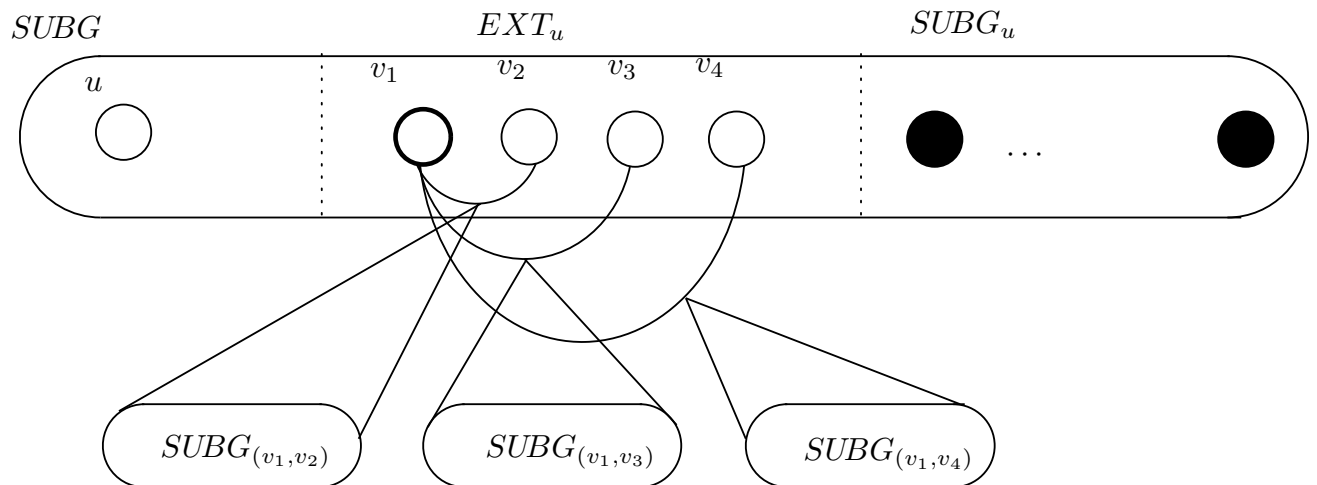
k の値に関わらず, $G(EXT_u)$ が非クリークである場合においては, MCP_2 は第4章のアルゴリズム MCP_1 と全く同様に動作する.



$$SUBG_{v_1} = \Gamma(v_1) \cap (\{v_1, v_2, v_3, v_4\} \cup SUBG_u)$$

$$SUBG_{u_1} = \Gamma(u_1) \cap SUBG_{v_1}$$

図 6.1 (B-1) $|SUBG_{u_1}| = |SUBG_{v_1}| - 3$ である場合



$$SUBG_{(v_1, v_2)} = \Gamma(v_2) \cap SUBG_{v_1}$$

$$SUBG_{(v_1, v_3)} = \Gamma(v_3) \cap (SUBG_{v_1} - \{v_2\})$$

$$SUBG_{(v_1, v_4)} = \Gamma(v_4) \cap (SUBG_{v_1} - \{v_2, v_3\})$$

図 6.2 (B-2) $|SUBG_{u_1}| \leq |SUBG_{v_1}| - 4$ である場合

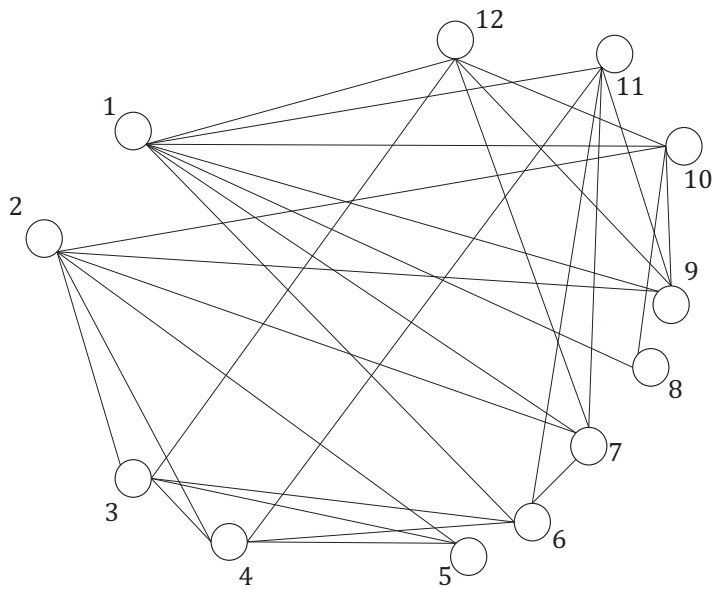


図 6.3 例題

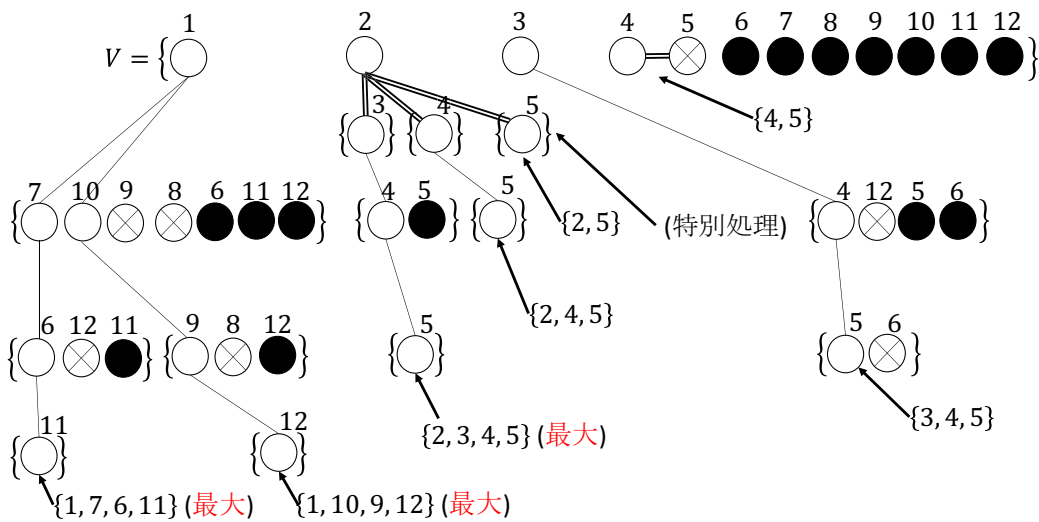


図 6.4 MCP_2 によるクリーク探索森

6.2.4 例題

図 6.3 のグラフ $G = (V, E)$ (第 3-5 章において用いたグラフと、同一のグラフ) に対するアルゴリズム MCP_2 の実行過程を、図 6.4 の探索森として表す。

以下の実行過程における集合等の表記, 及び図 6.4 中における表記法は, 前章までの例題におけるものを踏襲する.

本章において新たに導入した, “特別処理” 操作のうち, (B-2) である場合の操作が実行されたとき, 構成される各部分問題については, 次のように表記する.

前章までと同様の表記法において,

$$EXT_{u_d}^{d+1} = \{v_1, v_2, v_3, v_4\} \quad (0 \leq d \leq |V|), \text{ 及び,}$$

$$SUBG_{\{v_i\}}^1$$

$$= \Gamma(v_i) \cap (EXT_{u_d}^1[i, k] \cup SUBG_{\{u_d\}}^1),$$

又は,

$$SUBG_{\{p_0, p_1, \dots, p_{d-1}, v_i\}}^{d+1}$$

$$= \Gamma(v_i) \cap (EXT_{u_d}^{d+1}[i, k] \cup SUBG_{\{p_0, p_1, \dots, p_{d-1}, u_d\}}^{d+1}) \quad (1 \leq i \leq k)$$

として, (B-2) である場合の操作が実行されたとき,

$$SUBG_{\{v_i\}}^1 \quad (i = 1, 2),$$

又は,

$$SUBG_{\{p_0, p_1, \dots, p_{d-1}, v_i\}}^{d+1} \quad (i = 1, 2)$$

を分割した各部分問題を, $i + 1 \leq j \leq 4$ に対して,

$$SUBG_{\{p_0, p_1, \dots, p_{d-1}, v_i, v_j\}}^{d+1}$$

$$= \Gamma(v_j) \cap (SUBG_{\{p_0, p_1, \dots, p_{d-1}, v_i\}}^{d+1} - \{v_{i+1}, \dots, v_{j-1}\})$$

とする. 例題図中においては, これを v_i と v_j を二重線で結ぶことにより表現する.

以下は具体的実行過程となる.

$$(V =)$$

$$SUBG^0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\},$$

$$EXT_1^1 = \{2, 3, 4, 5\},$$

$$SUBG^0 = \{1\} \cup \{2, 3, 4, 5\} \cup \{6, 7, 8, 9, 10, 11, 12\},$$

ここで, $G(\{2, 3, 4, 5\})$ はクリークであるため, 交換操作は実行されない.

$$SUBG_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG_{\{1,7\}}^2 = \{6, 11, 12\},$$

$$EXT_7^2 = SUBG_{\{1\}}^1 - \{7\} - \{6, 11, 12\} = \{8, 9, 10\},$$

$$SUBG_{\{1\}}^1 = \{7\} \cup \{8, 9, 10\} \cup \{6, 11, 12\},$$

ここで,

$$|U_1| = |\Gamma(8) \cap EXT_7^2| = |\{10\}| = 1 < 3 - 1$$

により, 交換操作実行.

$$SUBG_{\{1\}}^1 = \{7\} \cup \{10, 9, 8\} \cup \{6, 11, 12\},$$

$$SUBG_{\{1,7\}}^2 = \{6, 11, 12\},$$

$$SUBG_{\{1,7,6\}}^3 = \{11\},$$

$$EXT_6^3 = SUBG_{\{1,7\}}^2 - \{6\} - \{11\} = \{12\},$$

$$SUBG_{\{1,7\}}^2 = \{6\} \cup \{12\} \cup \{11\},$$

$$SUBG_{\{1,7,6\}}^3 = \{11\},$$

$$SUBG_{\{1,7,6,11\}}^4 = \emptyset,$$

$$EXT_{11}^4 = SUBG_{\{1,7,6\}}^3 - \{11\} - \emptyset = \emptyset,$$

$$SUBG_{\{1,7,6\}}^3 = \{11\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{1,7,6,11\}}^4 = \emptyset, \text{ (更新: } Qmax_size := 4)$$

2段階バックトラック,

$$EXT_7^2 = \{10, 9, 8\},$$

$$SUBG_{\{1,10\}}^2 = \{8, 9, 12\},$$

$$SUBG_{\{1,10,9\}}^3 = \{12\},$$

$$EXT_9^3 = SUBG_{\{1,10\}}^2 - \{9\} - \{12\} = \{8\},$$

$$SUBG_{\{1,10\}}^2 = \{9\} \cup \{8\} \cup \{12\},$$

$$SUBG_{\{1,10,9\}}^3 = \{12\},$$

$$SUBG_{\{1,10,9,12\}}^4 = \emptyset,$$

$$EXT_{12}^4 = SUBG_{\{1,10,9\}}^3 - \{12\} - \emptyset = \emptyset,$$

$$SUBG_{\{1,10,9\}}^3 = \{12\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{1,10,9,12\}}^4 = \emptyset, \text{ (更新なし)}$$

3段階バックトラック,

$$|EXT_1^1| = |\{2, 3, 4, 5\}| = 4,$$

かつ $(4, 5) \in E$, 即ち $G(EXT_1^1)$ はクリークである.

$$|SUBG_{\{2\}}^1| = |\{3, 4, 5, 7, 9, 10\}| = 6,$$

$$|\Gamma(3) \cap SUBG_{\{2\}}^1| = |\{4, 5\}| = 2 = |SUBG_{\{2\}}^1| - 4,$$

$SUBG_{\{2\}}^1$ に対して (B-2) である場合の操作を実行,

$$SUBG_{\{2,3\}}^1 = \Gamma(3) \cap SUBG_{\{2\}}^1 = \{4, 5\},$$

$$SUBG_{\{2,3,4\}}^2 = \{5\},$$

$$EXT_4^2 = \emptyset,$$

$$SUBG_{\{2,3\}}^1 = \{4\} \cup \emptyset \cup \{5\},$$

$$SUBG_{\{2,3,4\}}^2 = \{5\},$$

$$SUBG_{\{2,3,4,5\}}^3 = \emptyset,$$

$$EXT_5^3 = \emptyset,$$

$$SUBG_{\{2,3,4\}}^2 = \{5\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{2,3,4,5\}}^3 = \emptyset, \text{ (更新なし)}$$

2 段階バックトラック,

$$SUBG_{\{2,4\}}^1 = \Gamma(4) \cap (SUBG_{\{2\}}^1 - \{3\}) = \{5\},$$

$$SUBG_{\{2,4,5\}}^2 = \emptyset$$

$$EXT_5^2 = \emptyset,$$

$$SUBG_{\{2,4\}}^1 = \{5\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{2,4,5\}}^2 = \emptyset, \text{ (更新なし)}$$

1 段階バックトラック,

$$SUBG_{\{2,5\}}^1 = \Gamma(5) \cap (SUBG_{\{2\}}^1 - \{3, 4\}) = \emptyset, \text{ (更新なし)}$$

$$|SUBG_{\{3\}}^1| = |\{4, 5, 6, 12\}| = 4,$$

$$|\Gamma(4) \cap SUBG_{\{3\}}^1| = |\{5, 6\}| = 2 = |SUBG_{\{3\}}^1| - 2,$$

$SUBG_{\{3\}}^1$ に対して (B-1) である場合の操作を実行,

$$SUBG_{\{3\}}^1 = \{4, 5, 6, 12\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$EXT_4^2 = \{12\},$$

$$SUBG_{\{3\}}^1 = \{4\} \cup \{12\} \cup \{5, 6\},$$

$$SUBG_{\{3,4\}}^2 = \{5, 6\},$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset,$$

$$EXT_5^3 = \{6\},$$

$$SUBG_{\{3,4\}}^2 = \{5\} \cup \{6\} \cup \emptyset,$$

$$SUBG_{\{3,4,5\}}^3 = \emptyset, \text{ (更新なし)}$$

2段階バックトラック,

$$SUBG_{TAIL:\{4,5\}}^1 = \emptyset, \text{ (更新なし)}$$

実行終了.

結果として, MCP₂ は例題のグラフ G 中の最大クリークサイズ 4 を得る.

6.3 最大時間計算量評価

前章までと同様に, 次の非再帰手続き EXPAND₀(SUBG) の時間計算量を考える.

EXPAND₀(SUBG): アルゴリズム 4 における EXPAND(SUBG) 中の 1600 行, 2007 行, 2200 行及び 2403 行における EXPAND(SUBG_{*u*}), EXPAND(SUBG_{*v_i*}), EXPAND(SUBG_{*(v_i, v_j)*}), および EXPAND(SUBG_{TAIL}) をそれぞれ EXPAND(∅) とした非再帰手続き.

補題 6.3.1 EXPAND₀(SUBG) の最大時間計算量は,

$$O(|SUBG| \cdot |SUBG_u|^2)$$

である.

(証明) まず, 本章追加部分であるアルゴリズム 4 における 2001 -2012 行を除く部分については, 第 3 章, 補題 3.6.1, 第 4 章, 補題 4.2.1 及び第 5 章, 補題 5.4.2 により題意が成立することを示した. 従ってここでは, 同追加部分について題意が成立することを示す.

2004 -2010 行における **for** 文中の, 2005 行における集合 SUBG_{*(v_i, v_j)*} の構成は, O(|SUBG_{*u*}|)-時間で可能である. 2006 -2009 行における各操作は, いずれも O(|SUBG_{*u*}|)-時間で可能であるから, 2004 -2010 行における **for** 文 (繰り返し回数 3) 全体の操作は, O(3 · |SUBG_{*u*}|) = O(|SUBG_{*u*}|)-時間で実行可能である.

2001 -2012 行における, 上記以外の各行の操作は, いずれも O(|SUBG| · |SUBG_{*u*}|)-時間で実行可能である. 以上より, 題意は成立する. □

前章までと同様に, MCP₂ に対しても $T(|SUBG|)$ に対して以下の基礎的關係式が成立する.

集合 $SUBG \subseteq V$ に対して, 節点 u を $G(SUBG)$ における最大次数節点とし,

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{v_i} = \hat{\Gamma}(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \quad (1 \leq i \leq k),$$

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u$$

とする. このとき, 次の關係式が成立する.

$$\begin{aligned} & T(|SUBG|) \\ & \leq T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2 \end{aligned} \quad (6.3.1)$$

(第4章, 式(4.3.2) と同一.)

ここで,

$$|SUBG_u| < |SUBG|$$

により, 次が成立する.

$$T(|SUBG|) < T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \quad (6.3.2)$$

$G(EXT_u)$ がクリーク, かつ $k = 4$ である場合には, $SUBG_{v_i}$ ($i = 1, 2$) に対して“特別処理”が実行される.

節点 $u_i \in SUBG_{v_i}$ を $G(SUBG_{v_i})$ における最大次数節点として,

$$SUBG_{u_i} = \Gamma(u_i) \cap SUBG_{v_i},$$

$$EXT_{u_i} = SUBG_{v_i} - \{u_i\} - SUBG_{u_i}$$

とそれぞれおく. このとき, 節点集合 $SUBG_{u_i}$ のサイズによる場合分けに基づき, 次の各操作が実行される.

(B-0) $|SUBG_{u_i}| \geq |SUBG_{v_i}| - 2$ である場合:

$$T(|SUBG_{v_i}|) \leq T(|SUBG_{u_i}|) + C|SUBG_{v_i}| \cdot |SUBG_{u_i}|^2.$$

即ち,

$$T(|SUBG_{v_i}|) < T(|SUBG_{u_i}|) + C|SUBG_{v_i}|^3 \quad (6.3.3)$$

(B-1) $|SUBG_{u_i}| = |SUBG_{v_i}| - 3$ である場合:

$|EXT_{u_i}| = 2$ のもとに, $EXT_{u_i} = \{v'_1, v'_2\}$ とする.

$(v'_1, v'_2) \in E$ である場合,

$$SUBG_{TAIL} := \hat{\Gamma}(\{v'_1, v'_2\}) \cap SUBG_{u_i},$$

$(v'_1, v'_2) \notin E$ である場合,

$$SUBG_{TAIL} := \emptyset$$

とする. このとき,

$$T(|SUBG_{v_i}|) \leq T(|SUBG_{u_i}|) + T(|SUBG_{TAIL}|) + C|SUBG_{v_i}| \cdot |SUBG_{u_i}|^2.$$

即ち,

$$T(|SUBG_{v_i}|) < T(|SUBG_{u_i}|) + T(|SUBG_{TAIL}|) + C|SUBG_{v_i}|^3 \quad (6.3.4)$$

(B-2) $|SUBG_{u_i}| \leq |SUBG_{v_i}| - 4$ である場合:

$$SUBG_{(v_i, v_j)} = \Gamma(v_j) \cap (SUBG_{v_i} - \{v_{i+1}, \dots, v_{j-1}\}) \quad (1 \leq i \leq 2 < j \leq 4)$$

とおくと,

$$T(|SUBG_{v_i}|) \leq \sum_{j=i+1}^4 T(|SUBG_{(v_i, v_j)}|) + C|SUBG_{v_i}| \cdot |SUBG_{u_i}|^2.$$

即ち,

$$T(|SUBG_{v_i}|) < \sum_{j=i+1}^4 T(|SUBG_{(v_i, v_j)}|) + C|SUBG_{v_i}|^3 \quad (6.3.5)$$

主要定理証明に先立ち, まず, EXPAND($SUBG$) に対する時間計算量評価結果を以下に与える.

補題 6.3.2 節点集合 $SUBG \subseteq V$ に対して, $|SUBG| = \eta$, $C' = 10^4 C$ とおく. このとき, $EXPAND(SUBG)$ の最大時間計算量 $T(|SUBG|)$ は以下を満たす.

$$T(|SUBG|) \leq C' 2^{0.3337\eta} \eta^3.$$

(証明) 証明は, $\eta \geq 1$ に関する数学的帰納法による.

まず, $\eta = 1$ の場合, 題意は明らかに成立する.

次に, $\eta \geq 1$ 以下である全ての正整数について, 題意が成立すると仮定する. この仮定のもとで, $|SUBG| = \eta + 1$ である場合を考える.

前章と同様に, u を $G(SUBG)$ における最大次数節点として,

$$SUBG_u = \Gamma(u) \cap SUBG$$

として,

$$|SUBG_u| = \eta - k \quad (0 \leq k < \eta)$$

とする. このとき,

$$\begin{aligned} |EXT_u| &= |SUBG - \{u\} - SUBG_u| \\ &= |SUBG| - |\{u\}| - |SUBG_u| \\ &= (\eta + 1) - 1 - (\eta - k) \\ &= k. \end{aligned}$$

先ず,

(0) $k \leq 1$ である場合:

第4章, 補題 4.3.2 証明, 場合分け (b) と同様にして示せる.

従って, 以下では全て $k \geq 2$ である場合 を考える.

ここで, 本章新規追加操作である, “特別処理” が実行されるか否かにより場合分けを行う.

【1】 $k \neq 4$, 又は $G(EXT_u)$ が非クリークである場合 :

更に, $G(EXT_u)$ がクリークであるか否かにより場合分けを行う.

(1-2) $G(EXT_u)$ が非クリークである場合;

このとき, $SUBG_{TAIL} = \emptyset$ である. 従って式 (6.3.2) により,

$$\begin{aligned} & T(|SUBG|) \\ & < T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + C|SUBG|^3 \\ & \leq T(\eta - k) + (k - 2)T(\eta - k) + C(\eta + 1)^3. \end{aligned}$$

帰納法の仮定により,

$$\begin{aligned} & T(|SUBG|) \\ & < (k - 1)T(\eta - k) + C(\eta + 1)^3 \\ & \leq (k - 1)C'2^{0.3337 \cdot (\eta - k)}(\eta - k)^3 + C(\eta + 1)^3 \\ & < (k - 1)C'2^{0.3337 \cdot (\eta - k)}(\eta + 1)^3 + C(\eta + 1)^3 \\ & < \left(\frac{k-1}{2^{0.3337k}} + 10^{-4}\right)C'2^{0.3337\eta}(\eta + 1)^3. \end{aligned}$$

ここで, 第3章, 補題 3.6.3 と同様にして, $k \geq 0$ において,

$$\frac{k-1}{2^{0.3337k}} < 1.2584 \text{ が成立することが証明できる. (左辺は } k = 5 \text{ において最大値}$$

1.25832... をとる.) 従って,

$$\begin{aligned} & T(|SUBG|) \\ & < (1.2584 + 10^{-4})C'2^{0.3337\eta}(\eta + 1)^3 \\ & = 1.2585 \cdot C'2^{0.3337\eta}(\eta + 1)^3 \quad (\text{ここで, } 2^{0.3337} = 1.2602\dots) \\ & < 2^{0.3337} \cdot C'2^{0.3337\eta}(\eta + 1)^3 \\ & = C'2^{0.3337(\eta+1)}(\eta + 1)^3. \end{aligned}$$

MCP₂ の最大時間計算量評価において, 計算量上界が最大となるのはこの場合であり, これが定数値 0.3337 の根拠である.

(1-3) $G(EXT_u)$ がクリークである場合;

式 (6.3.2) により,

$$\begin{aligned} & T(|SUBG|) \\ & < T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3. \end{aligned}$$

いま, $G(EXT_u)$ はクリークであるから, 補題 6.2.1 により,

$$|SUBG_{v_i}| \leq |SUBG_u| - i + 1.$$

ここで, 先ず $SUBG_{TAIL}$ のサイズに対応した各場合の操作に対して, その時間計算量評価を与える.

補題 6.3.3 $SUBG \subseteq V$, 定数 $p \geq 1$, $C' > 0$, $0.138 < c < 0.34$ に対して,

$|SUBG| = \eta \geq 1$ 以下の全ての整数について,

$$T(|SUBG|) \leq C' 2^{c\eta} \eta^p$$

が成立するとき,

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u$$

とそれぞれおく. このとき, “部分問題の統合による探索領域削減” の操作 Case(a), Case(b) および Case(c) 各場合における操作のうち, 計算量上界が最大となるのは Case(c) の場合である.

(証明) いま $SUBG$ の部分問題として, $SUBG_u$ 及び $SUBG_{TAIL} \neq \emptyset$ が構成されたとする.

このとき, Case(a)~Case(c) 各場合において以下の操作が実行される.

Case(a) $|SUBG_{TAIL}| = |SUBG_u| - 1$ である場合:

$$SUBG_u = \emptyset$$

との置き換えがなされる. 従って,

$$|SUBG_u| = 0, |SUBG_{TAIL}| = |SUBG_u| - 1.$$

Case(b) $|SUBG_{TAIL}| = |SUBG_u| - 2$ である場合:

$$R = SUBG_u - SUBG_{TAIL}$$

のもと,

$$SUBG_u = \hat{\Gamma}(R) \cap SUBG_u$$

との置き換えがなされる. 従って,

$$|SUBG_u| \leq |SUBG_u| - 2, |SUBG_{TAIL}| = |SUBG_u| - 2.$$

Case(c) $|SUBG_{TAIL}| \leq |SUBG_u| - 3$ である場合:

$SUBG_u$ に対し特別な操作は行わない. 従って,

$$|SUBG_{TAIL}| \leq |SUBG_u| - 3.$$

ここで, $|SUBG_u| = \eta'$ ($\eta' < \eta$) とおき, 部分問題 $SUBG_u$ 及び $SUBG_{TAIL}$ の計算量上界を, Case(a)~Case(c) の各場合につき計算する.

Case(a) :

$$\begin{aligned} & T(|SUBG_u|) + T(|SUBG_{TAIL}|) \\ & \leq 0 + C'2^{c(\eta'-1)}(\eta' - 1)^p \\ & < 2^{-c}C'2^{c\eta'}\eta^p. \end{aligned}$$

Case(b) :

$$\begin{aligned} & T(|SUBG_u|) + T(|SUBG_{TAIL}|) \\ & \leq C'2^{c(\eta'-2)}(\eta' - 2)^p + C'2^{c(\eta'-2)}(\eta' - 2)^p \\ & < 2 \cdot 2^{-2c}C'2^{c\eta'}\eta^p. \end{aligned}$$

Case(c) :

$$\begin{aligned} & T(|SUBG_u|) + T(|SUBG_{TAIL}|) \\ & \leq C'2^{c\eta'}\eta^p + C'2^{c(\eta'-3)}(\eta' - 3)^p \\ & < (1 + 2^{-3c})C'2^{c\eta'}\eta^p. \end{aligned}$$

ここで, Case(b) の上界から Case(a) の上界を引くと,

$$\begin{aligned} & 2 \cdot 2^{-2c}C'2^{c\eta'}\eta^p - 2^{-c}C'2^{c\eta'}\eta^p \\ & = (2 \cdot 2^{-c} - 1)2^{-c} \cdot C'2^{c\eta'}\eta^p. \end{aligned}$$

$0 < c < 1$ より, $2^{-c} > 2^{-1}$ である. 従って,

$$\begin{aligned} & (2 \cdot 2^{-c} - 1)2^{-c} \cdot C'2^{c\eta'}\eta^p \\ & > (2 \cdot 2^{-1} - 1)2^{-c} \cdot C'2^{c\eta'}\eta^p \end{aligned}$$

$$= 0.$$

即ち, (Case(b) の上界) > (Case(a) の上界) である.

また, Case(c) の上界から Case(b) の上界を引くと,

$$\begin{aligned} & (1 + 2^{-3c})C'2^{cn'}\eta^p - 2 \cdot 2^{-2c}C'2^{cn'}\eta^p \\ &= (1 + 2^{-3c} - 2 \cdot 2^{-2c})C'2^{cn'}\eta^p \\ &= (1 + (2^{-c} - 2)2^{-2c})C'2^{cn'}\eta^p \\ &> (1 + (2^{0.34} - 2)2^{-2c})C'2^{cn'}\eta^p \\ &> (1 - 1.21 \cdot 2^{-2c})C'2^{cn'}\eta^p \\ &> (1 - 1.21 \cdot 2^{-0.276})C'2^{cn'}\eta^p \\ &> 0. \end{aligned}$$

即ち, (Case(c) の上界) > (Case(b) の上界) である. 以上から, 題意は成立する. \square

補題 6.3.3 により, 本章における帰納法の仮定のもと, $SUBG_{TAIL}$ が構成された場合は常に Case(c) において計算量上界最大となる. そこで以下では,

$$\begin{aligned} |SUBG_u| &= \eta - k, \\ |SUBG_{TAIL}| &\leq |SUBG_u| - 3 = \eta - k - 3 \end{aligned}$$

とする.

$$\begin{aligned} & T(|SUBG|) \\ &= T(\eta + 1) \\ &< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\ &\leq T(\eta - k) + \sum_{i=1}^{k-2} T(\eta - k - i + 1) + T(\eta - k - 3) + C(\eta + 1)^3. \end{aligned}$$

帰納法の仮定により,

$$\begin{aligned} & T(\eta + 1) \\ &< C'2^{0.3337(\eta-k)}(\eta - k)^3 + \sum_{i=1}^{k-2} C'2^{0.3337(\eta-k-i+1)}(\eta - k - i + 1)^3 \\ &\quad + C'2^{0.3337(\eta-k-3)}(\eta - k - 3)^3 + C(\eta + 1)^3 \\ &< C'2^{0.3337(\eta-k)}(\eta + 1)^3 + \sum_{i=1}^{k-2} C'2^{0.3337(\eta-k-i+1)}(\eta + 1)^3 \\ &\quad + C'2^{0.3337(\eta-k-3)}(\eta + 1)^3 + C(\eta + 1)^3 \end{aligned}$$

$$\begin{aligned}
&= C'2^{0.3337(\eta-k)}(\eta+1)^3 + \sum_{i=1}^{k-2} 2^{-0.3337(i-1)} \cdot C'2^{0.3337(\eta-k)}(\eta+1)^3 \\
&\quad + C'2^{0.3337(\eta-k-3)}(\eta+1)^3 + C(\eta+1)^3 \\
&< \left(\left(1 + \frac{1-2^{-0.3337(k-2)}}{1-2^{-0.3337}} + \frac{1}{2^{0.3337 \cdot 3}} \right) \frac{1}{2^{0.3337k}} + 10^{-4} \right) C'2^{0.3337\eta}(\eta+1)^3.
\end{aligned}$$

いま第3章, 補題 3.6.3 と同様にして, 整数 $2 \leq k \leq 3$ 及び $k \geq 5$ において,

$$\left(1 + \frac{1-2^{-0.3337(k-2)}}{1-2^{-0.3337}} + \frac{1}{2^{0.3337 \cdot 3}}\right) \frac{1}{2^{0.3337k}} < 1.2489$$

が成立することを証明できる. (左辺は $2 \leq k \leq 3$ 及び $k \geq 5$ のもと, $k = 3$ において最大値 1.24885... をとる.) 従って,

$$\begin{aligned}
&T(|SUBG|) \\
&\leq (1.2489 + 10^{-4})C'2^{0.3337\eta}(\eta+1)^3 \\
&= 1.249C'2^{0.3337\eta}(\eta+1)^3 \\
&< 2^{0.3337}C'2^{0.3337\eta}(\eta+1)^3 \\
&= C'2^{0.3337(\eta+1)}(\eta+1)^3.
\end{aligned}$$

以上, “特別処理” が実行されない各場合において, 題意は成立する.

最後に, “特別処理” が実行される場合における題意の成立を示す.

[2] $k = 4$, かつ $G(EXT_u)$ がクリークである場合:

$k = 4$ のもと, 式 (6.3.2) により,

$$\begin{aligned}
&T(|SUBG|) \\
&< T(|SUBG_u|) + \sum_{i=1}^2 T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3.
\end{aligned}$$

EXT_u においては, v_1, v_2, v_3 からの探索がそれぞれ実行される.

v_1 , 及び v_2 からの探索においては, “特別処理” が実行される.

v_3 からの探索においては,

$$\begin{aligned}
&SUBG_{TAIL} \\
&= \hat{\Gamma}(\{v_3, v_4\}) \cap SUBG_u
\end{aligned}$$

に対して EXPAND が実行される. そこで, 以下これらの各探索における計算量上界を考える.

まず, 補題 6.3.3 により, Case(c) である場合に $T(|SUBG|)$ の計算量上界は最大となる.

従って,

$$\begin{aligned} & |SUBG_{TAIL}| \\ & \leq |SUBG_u| - 3 \\ & = \eta - k - 3. \end{aligned}$$

v_i ($i = 1, 2$) からの各探索においては, “特別処理” が実行される. そこで, (B-0)~(B-2) 各場合における操作が実行されたときの, 計算量上界をそれぞれ求める.

(B-0) $|SUBG_{u_i}| \geq |SUBG_{v_i}| - 2$ である場合;

式 (6.3.3) より,

$$\begin{aligned} & T(|SUBG_{v_i}|) \\ & < T(|SUBG_{u_i}|) + C|SUBG_{v_i}|^3 \\ & \leq T(|SUBG_{v_i}| - 1) + C|SUBG_{v_i}|^3 \end{aligned}$$

いま $G(EXT_u)$ はクリークであるから,

$$|SUBG_{v_i}| \leq |SUBG_u| - i + 1.$$

従って,

$$\begin{aligned} & T(|SUBG_{v_i}|) \\ & < T(|SUBG_u| - i + 1) + C(\eta + 1)^3 \\ & = T(|SUBG_u| - i) + C(\eta + 1)^3 \\ & = T(\eta - k - i) + C(\eta + 1)^3 \\ & \leq C' 2^{0.3337(\eta - k - i)} (\eta - k - i)^3 + C(\eta + 1)^3 \\ & < 2^{-0.3337i} C' 2^{0.3337(\eta - k)} (\eta + 1)^3 + C(\eta + 1)^3. \end{aligned}$$

ここで,

$$C_0(i) = 2^{-0.3337i}$$

とおく.

(B-1) $|SUBG_{u_i}| = |SUBG_{v_i}| - 3$ である場合;

式 (6.3.4) より,

$$T(|SUBG_{v_i}|) < T(|SUBG_{u_i}|) + T(|SUBG_{TAIL}|) + C|SUBG_{v_i}|^3.$$

補題 6.3.3 により, Case(c), 即ち

$$|SUBG_{TAIL}| \leq |SUBG_{u_i}| - 3$$

である場合計算量上界は最大となる。従って,

$$\begin{aligned} & T(|SUBG_{v_i}|) \\ & < T(|SUBG_{u_i}|) + T(|SUBG_{TAIL}|) + C|SUBG_{v_i}|^3 \\ & \leq T(|SUBG_{v_i}| - 3) + T(|SUBG_{v_i}| - 3 - 3) + C|SUBG_{v_i}|^3 \\ & < T(|SUBG_{v_i}| - 3) + T(|SUBG_{v_i}| - 6) + C(\eta + 1)^3 \\ & \leq T(|SUBG_u| - i + 1 - 3) + T(|SUBG_u| - i + 1 - 6) + C(\eta + 1)^3 \\ & = T(|SUBG_u| - i - 2) + T(|SUBG_u| - i - 5) + C(\eta + 1)^3 \\ & = T(\eta - k - (i + 2)) + T(\eta - k - (i + 5)) + C(\eta + 1)^3 \\ & < C'2^{0.3337(\eta - k - (i + 2))}(\eta + 1)^3 + C'2^{0.3337(\eta - k - (i + 5))}(\eta + 1)^3 + C(\eta + 1)^3 \\ & = (2^{-0.3337 \cdot 2} + 2^{-0.3337 \cdot 5})2^{-0.3337i}C'2^{0.3337(\eta - k)}(\eta + 1)^3 + C(\eta + 1)^3. \end{aligned}$$

ここで,

$$C_1(i) = (2^{-0.3337 \cdot 2} + 2^{-0.3337 \cdot 5})2^{-0.3337i}$$

とおく.

(B-2) $|SUBG_{u_i}| \leq |SUBG_{v_i}| - 4$ である場合;

式 (6.3.5) より,

$$\begin{aligned} & T(|SUBG_{v_i}|) \\ & < \sum_{j=i+1}^4 T(|SUBG_{(v_i, v_j)}|) + C|SUBG_{v_i}|^3 \\ & < \sum_{j=i+1}^4 T(|SUBG_{(v_i, v_j)}|) + C(\eta + 1)^3. \end{aligned}$$

ここで, 補題 6.2.3 より,

$$\begin{aligned} & |SUBG_{(v_i, v_j)}| \\ & \leq |\Gamma(u_i) \cap SUBG_{v_i}| - (j - i - 1) \\ & \leq (|SUBG_{v_i}| - 4) - (j - i - 1) \\ & = |SUBG_{v_i}| - 3 - j + i \\ & \leq (|SUBG_u| - i + 1) - 3 - j + i \\ & = |SUBG_u| - j - 2. \end{aligned}$$

従って,

$$T(|SUBG_{v_i}|)$$

$$\begin{aligned}
&< \sum_{j=i+1}^4 T(|SUBG_u| - j - 2) + C(\eta + 1)^3 \\
&= \sum_{j=i+1}^4 T((\eta - k) - j - 2) + C(\eta + 1)^3 \\
&< \sum_{j=i+1}^4 2^{-0.3337(j+2)} C' 2^{0.3337(\eta-k)} (\eta + 1)^3 + C(\eta + 1)^3.
\end{aligned}$$

ここで,

$$C_2(i) = \sum_{j=i+1}^4 2^{-0.3337(j+2)}$$

とおく.

いま, $i = 1$ において,

$$C_0(1) = 2^{-0.3337} = 0.7934\dots,$$

$$C_1(1) = (2^{-0.3337 \cdot 2} + 2^{-0.3337 \cdot 5}) 2^{-0.3337 \cdot 1} = 2^{-0.3337 \cdot 3} + 2^{-0.3337 \cdot 6} = 0.7492\dots,$$

$$C_2(1) = \sum_{j=1+1}^4 2^{-0.3337(j+2)} = 2^{-0.3337 \cdot 4} + 2^{-0.3337 \cdot 5} + 2^{-0.3337 \cdot 6} = 0.9606\dots$$

従って, $T(|SUBG_{v_1}|)$ の計算量上界は (B-2) である場合に最大となる. 即ち,

$$\begin{aligned}
&T(|SUBG_{v_1}|) \\
&< C_2(1) C' 2^{0.3337(\eta-k)} (\eta + 1)^3 + C(\eta + 1)^3.
\end{aligned}$$

また, $i = 2$ において,

$$C_0(2) = 2^{-0.3337 \cdot 2} = 0.6296\dots,$$

$$C_1(2) = (2^{-0.3337 \cdot 2} + 2^{-0.3337 \cdot 5}) 2^{-0.3337 \cdot 2} = 2^{-0.3337 \cdot 4} + 2^{-0.3337 \cdot 7} = 0.5945\dots,$$

$$C_2(2) = \sum_{j=2+1}^4 2^{-0.3337(j+2)} = 2^{-0.3337 \cdot 5} + 2^{-0.3337 \cdot 6} = 0.5641\dots$$

従って, $T(|SUBG_{v_2}|)$ の計算量上界は (B-0) である場合に最大となる. 即ち,

$$\begin{aligned}
&T(|SUBG_{v_2}|) \\
&< C_0(2) C' 2^{0.3337(\eta-k)} (\eta + 1)^3 + C(\eta + 1)^3.
\end{aligned}$$

以上から,

$$\begin{aligned}
&T(|SUBG|) \\
&< T(|SUBG_u|) + \sum_{i=1}^2 T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\
&< (1 + C_2(1) + C_0(2) + 2^{-0.3337 \cdot 3} + 3 \cdot 10^{-4}) C' 2^{0.3337(\eta-4)} (\eta + 1)^3 \\
&< (1.225 + 0.0003) C' 2^{0.3337\eta} (\eta + 1)^3 \\
&< 2^{0.3337} \cdot C' 2^{0.3337\eta} (\eta + 1)^3 \\
&= C' 2^{0.3337(\eta+1)} (\eta + 1)^3.
\end{aligned}$$

従って, 【2】においても題意は成立する.

以上より, いずれの場合においても,

$$T(|SUBG|) \leq C'2^{0.3337(\eta+1)}(\eta+1)^3$$

は成立する. 帰納法による帰結として, 任意の $\eta \geq 1$ に関して題意は成立する. \square

これより, 前章までと同様に次が成立する.

定理 6.3.1 節点数 n , 最大次数 $\Delta \geq 1$ なるグラフにおいて, 定数 C'' を $C'' = 6.7 \cdot 10^{11}$ とおく. このとき, Δ が

$$\Delta \leq 2.994d \lg n \quad (d \geq 0: \text{定数})$$

を満たすならば, アルゴリズム MCP_2 は与えられたグラフの最大クリークサイズを $O(n^{1+d})$ -時間で抽出可能である.

(証明) 第3章, 定理 3.6.1 証明と同様. \square

第 7 章

アルゴリズム MCP_3

7.1 概要

本章においては, 前章までの結果を更に改良した次の結果を与える.

「節点数 n の一般グラフにおいて, 最大次数 $\Delta \leq 3.177d \lg n$ ($d \geq 0$: 定数) なる条件が満たされているとき, このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である。」

7.2 アルゴリズム

本章において解析を行う, 拡張アルゴリズム MCP_3 をアルゴリズム 5 に示す *1. このアルゴリズムは第 6 章のアルゴリズム MCP_2 に対し, 本章における新規限定操作である,

- ・部分問題の統合による探索領域の削減 (拡張版)
- ・少数部分問題の分割 (拡張版)

及び,

- ・節点の隣接関係による限定

をそれぞれ追加したものである. MCP_3 において, MCP_2 に対する MCP_3 での新規追加部分に関しては, 前章までと同様に赤字により表す.

また, MCP_2 と記述内容が同一であり, 行番号のみ異なる部分 (1435- 1444 行) につ

*1 前章におけるアルゴリズム MCP_2 と同様に, 関連論文 (5) における同名のアルゴリズムの論理構造を整理し, 改訂を行った.

アルゴリズム 5 アルゴリズム MCP₃

```

/* 1500 - 1700 行 (略) */
0000: procedure MCP3(G)
/* 0100 - 0500 行 (略) */
0600: end {of MCP3}

0700: procedure EXPAND(SUBG)
/* 0800 - 1400 行 (略) */

1401: Ru := ∅;
1402: if k ≥ 3 then
1403:   for i = 1 to k do
1404:     if |Γ(vi) ∩ EXTu| = k - 2 then
1405:       v := (EXTu - Γ(vi))[1];
1406:       if (vk-1, vk) ∈ E then
1407:         swap vk-1 with vi;
1408:         swap vk with v fi
1409:       else if |Γ(vi) ∩ EXTu| ≤ k - 3 then
1410:         v := (EXTu - Γ(vi))[1];
1411:         w := (EXTu - Γ(vi))[2];
1412:         swap vk with vi;
1413:         swap vk-1 with v;
1414:         swap vk-2 with w;
1415:         i := k + 1 fi fi od fi

1416:   if k ≥ 3 and (vk-1, vk) ∈ E then
1417:     SUBGvk-1 := Γ(vk-1) ∩ SUBG;
1418:     SUBGvk := Γ(vk) ∩ SUBG;
1419:     if |SUBGvk - SUBGvk-1| ≤ 1 then
1420:       swap vk-1 with v1;
1421:       swap vk with v2 fi fi

1422:   SUBGTAIL := ∅;
1423:   t := 2;
1424:   ROOTt := ∅;
1425:   if k ≥ 2 and (vk-1, vk) ∈ E then
1426:     ROOTt := {vk-1, vk};
1427:     SUBGTAIL :=  $\hat{\Gamma}$ (ROOTt) ∩ SUBGu
1428:   else {i.e., k ≤ 1 or (vk-1, vk) ∉ E}
1429:     if k ≥ 3
1430:       and |Γ(vk) ∩ EXTu| ≤ k - 3
1431:       and (vk-2, vk-1) ∈ E then
1432:         ROOTt := {vk-2, vk-1};
1433:         SUBGTAIL :=  $\hat{\Gamma}$ (ROOTt) ∩ ({vk} ∪ SUBGu);
1434:         t := 3 fi
1435:     fi /* ROOTt ≠ ∅ iff SUBGTAIL ≠ ∅ */
1436:     if ROOTt ≠ ∅ then
1437:       /* 1435-1444 行 : MCP2
1438:
1439:         1413-1422 行と同一 */
1440:     fi

1441:   fi

1701:   if k ≤ 2 then goto 2401 fi

1800:   for i = 1 to k - t do
1900:     vi := EXTu[i];
2000:     SUBGvi := Γ(vi) ∩ (EXTu[i, k] ∪ SUBGu);
2001:     L := ∅;
2002:     if |SUBGv2 - SUBGv1| ≤ 1 then
2003:       SUBGv2 := ∅ fi
2004:     if i = k - 2 and
2005:       ( (vk-1, vk) ∈ E and 3 ≤ k ≤ 5 )
2006:       or
2007:       ( (vk-1, vk) ∉ E and 4 ≤ k ≤ 5
2008:         and t = 2 )
2009:     then
2010:       if SUBGvi ≠ ∅ then
2011:         ui := a vertex with the maximum
2012:           degree in G(SUBGvi);
2013:         if |Γ(ui) ∩ SUBGvi| ≤ |SUBGvi| - 4
2014:         then /*Case(B-2)*/
2015:           A := Γ(vi) ∩ (EXTu[i + 1, k] - L);
2016:           /* A は EXTu[i + 1, k] における
2017:             順序を保存 */
2018:           if A = ∅ then goto 2100 fi
2019:           for j := 1 to |A| do
2020:             SUBG(vi, A[j]) := Γ(A[j])
2021:               ∩ (SUBGvi - A[1, j - 1]);
2022:             Qsize := Qsize + |{vi, A[j]}| + |L|;
2023:             EXPAND(SUBG(vi, A[j]))
2024:             Qsize := Qsize - |{vi, A[j]}| - |L|;
2025:           od
2026:           goto 2400
2027:         else
2028:           if |Γ(ui) ∩ SUBGvi| ≥ |SUBGvi| - 2
2029:           then /*Case(B-0)*/
2030:             SUBGvi := Γ(ui) ∩ SUBGvi;
2031:             L := L ∪ {ui};
2032:             if |L| ≤ 1 then goto 2006
2033:             else goto 2100 fi fi
2034:         fi
2035:       fi

2100:     Qsize := Qsize + |{vi}| + |L|;
2200:     EXPAND(SUBGvi)
2300:     Qsize := Qsize - |{vi}| - |L|
2400:   od

2401:   if (vk-t+1, vk-t+2) ∈ E then
2402:     Qsize := Qsize + |ROOTt|;
2403:     EXPAND(SUBGTAIL)
2404:     Qsize := Qsize - |ROOTt|
2405:   fi

/* 2500 - 2700 行 (略) */
2800: end {of EXPAND}

```

いては、それぞれ対応する行番号 (MCP_2 における, 1413- 1422 行) をコメントとして記述の上, 省略している.

前章までの結果を踏まえた, 追加操作の概要は以下の通りである.

第 4, 及び 6 章において提唱した各アルゴリズムには, いずれも次の探索領域削減手法を採用した.

節点集合 $SUBG$ 中の, $G(SUBG)$ における最大次数節点を u として,

$$SUBG_u = \Gamma(u) \cap SUBG,$$

及び,

$$EXT_u = SUBG - \{u\} - SUBG_u$$

として, EXT_u を,

$$EXT_u := \{v_1, v_2, \dots, v_k\} \text{ なる順序付き節点集合とする.}$$

この EXT_u 中に非隣接節点对が存在する, 即ち $G(EXT_u)$ が非クリークである場合には, EXT_u 中の最後尾 2 節点 v_{k-1}, v_k ($k = |EXT_u|$) が非隣接となるように節点の順序を入れ替える. この入れ替えの判定操作を実行の上, $(v_{k-1}, v_k) \notin E$ なる場合, これら 2 節点からの探索を単純に省略した.

$G(EXT_u)$ がクリークである場合には, “部分問題の統合による探索領域削減” を導入し, 節点 v_{k-1} からの探索における領域削減を達成した.

$G(EXT_u)$ が非クリークである場合における部分問題の単純削減は, 解析上非常に大きな効果を発揮する. このため, 第 4 章の解析過程において, $G(EXT_u)$ が非クリークである場合における計算量上界は, $G(EXT_u)$ がクリークである場合よりも小さなものとなった.

第 5 章においては, “部分問題の統合による探索領域削減” についてより詳細な場合分けに基づく探索領域削減操作を導入した. この操作は $G(EXT_u)$ がクリークである場合に適用される. 一方で, $G(EXT_u)$ が非クリークである場合においては第 3 章の基本アルゴリズムと全く同様の操作のみを適用した.

第 6 章においては, まず補題 6.2.1 により, $G(EXT_u)$ がクリークである場合, EXT_u 中各節点からの探索は, 探索順が後の節点程探索領域が小さくなることを示した. これに加えて, $G(EXT_u)$ がクリーク, かつ $k = |EXT_u| = 4$ である場合に更なる探索領域削減操作を導入し, これにより計算量上界の改善を達成した.

第5章, 及び第6章における新規探索領域削減操作は, いずれも $G(EXT_u)$ がクリークである場合に探索領域の削減を達成する操作である. この結果として, 第6章における解析上, $G(EXT_u)$ がクリークである場合における計算量上界は小さくなり, 計算量上界最大の場合ではなくなっていた.

一方で, $G(EXT_u)$ が非クリークである場合における探索領域削減操作は, 第4章で導入したものだけであった. この結果として, 第6章において計算量上界が最大となる場合は, $G(EXT_u)$ が非クリークである場合となっている.

以上から, 第6章における計算量上界を改善するためには, 先ず $G(EXT_u)$ が非クリークである場合に, 更なる探索領域削減を達成する必要がある. 本章における追加操作のうち,

- ・部分問題の統合による探索領域の削減 (拡張版)
- ・少数部分問題の分割 (拡張版)

は, いずれも $G(EXT_u)$ が非クリークである場合における探索領域削減を達成するための手法である. 各手法の詳細な内容については, それぞれ第7.2.2節, 及び7.2.3節において後述する.

上記2手法の導入により, アルゴリズム MCP_3 の解析過程において, $G(EXT_u)$ が非クリークである場合の計算量上界はより小さくなる. この結果として, 本章の解析過程において計算量上界が最大となる場合は, $G(EXT_u)$ がクリークである場合となる. 新規追加操作の残り1つである,

- ・節点間の隣接関係による限定

は, $G(EXT_u)$ がクリークである場合において有効な新規探索領域削減手法である. この手法の詳細については, 第7.2.4節において後述する.

7.2.1 部分問題の統合による探索領域削減の詳細化 (拡張版)

前節に述べた通り, 本章の改善結果を得るに当たっては, $G(EXT_u)$ が非クリークである場合における探索領域削減操作を追加する必要がある.

前章までの各アルゴリズムにおいて, “部分問題の統合による探索領域削減” は, $G(EXT_u)$ がクリークである場合にのみ実行される操作であった. 本節においては, この手法を拡張し, $G(EXT_u)$ が非クリークである場合にも適用可能な手法とする.

拡張を考えるため、まず前章までにおける“部分問題の統合による探索領域削減”の概要について改めて考察する。

EXPAND(*SUBG*) の実行過程において、各節点 $u, v_1, v_2, \dots, v_{k-2} \in SUBG$ からの探索終了後、

$SUBG_{v_{k-1}} = \Gamma(v_{k-1}) \cap (EXT_u[k-1, k] \cup SUBG_u)$ とする。このとき以下が成立した (第4章, 補題 4.2.1)。

$$\omega(SUBG_{v_{k-1}}) > \omega(SUBG_u)$$

なる不等式が成立するのは、 $SUBG_{v_{k-1}}$ 中の最大クリークが節点 v_k を含む場合に限られる。

この補題に基づき、

$(v_{k-1}, v_k) \in E$ である場合、

$$SUBG_{TAIL} := \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u,$$

$(v_{k-1}, v_k) \notin E$ である場合、

$$SUBG_{TAIL} := \emptyset,$$

として、節点 v_{k-1} からの探索は、 $SUBG_{TAIL}$ 、即ち節点 v_k に必ず隣接するような $SUBG_{v_{k-1}}$ 中の節点の集合に対する EXPAND の実行とした。

第6章においては、補題 4.2.1 を拡張した、次の事実 (補題 6.2.2) を示した。

節点集合 $SUBG \subseteq V$ に対して、節点 u を $G(SUBG)$ における最大次数節点、

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \quad (1 \leq i \leq k)$$

とする。

上記定義のもとに、以下が成り立つ。

$$\omega(\{u\} \cup SUBG_u) < \omega(\{v_i\} \cup SUBG_{v_i})$$

であるならば、サイズ $\omega(\{v_i\} \cup SUBG_{v_i})$ のクリークは、節点 v_i の他に

$$EXT_u[i+1, k] = \{v_{i+1}, v_{i+2}, \dots, v_k\}$$

中の節点を少なくとも1個含む。

この結果は、 EXT_u 中各節点 v_i ($1 \leq i \leq k$) からの探索においては、自身より探索順が後の各節点 v_{i+1}, \dots, v_k のうち少なくとも1個を含むような部分問題に対して EXPAND

を実行すれば十分であることを示す. そこで, 第6章においては各節点 v_{i+1}, \dots, v_k のうち少なくとも1個に必ず隣接するような $SUBG_{v_i}$ 中の節点の集合を考え, 節点 v_i からの探索をこれらの集合に対する EXPAND の実行とする手法を導入した.

この結果から考えると, $SUBG_{v_{k-1}}$ は, 上記の探索必要な部分集合が高々1個となる集合である. 従って, $SUBG_{v_{k-1}}$ は1個の真部分集合によって置き換え可能であり, 常に探索領域削減が達成される.

上記の手法において, $G(EXT_u)$ がクリークである場合, $SUBG_{v_i}$ に対して探索必要な部分集合は,

$$k - (i + 1) + 1 = k - i$$

個だけ存在する. これに対して, $G(EXT_u)$ が非クリークである場合には, EXT_u 中に非隣接節点对が存在するため, 部分集合数は 高々 $k - i$ 個となる.

以上を考慮すれば, “部分問題の統合による探索領域削減” は, $G(EXT_u)$ が非クリークである場合においても適用可能である.

いま, 節点 $v_{k-t} \in EXT_u$ ($1 \leq t \leq k - 1$) を,

$$|\Gamma(v_{k-t}) \cap EXT_u[k - t, k]| = 1,$$

である節点とする. 即ち v_{k-t} は, 自身に隣接し, かつ探索順が後の節点が1個しか存在しない節点である. そのような節点, 即ち $\Gamma(v_{k-t}) \cap EXT_u[k - t, k]$ 中の唯一の節点を v とする. このとき, 節点集合 $SUBG_{TAIL}$ を以下で定義する.

$(v_{k-t}, v) \notin E$ である場合,

$$SUBG_{TAIL} = \emptyset,$$

$(v_{k-t}, v) \in E$ である場合,

$$\begin{aligned} & SUBG_{TAIL} \\ &= \Gamma(v) \cap SUBG_{v_{k-t}}, \end{aligned}$$

即ち,

$$\begin{aligned} & SUBG_{TAIL} \\ &= \Gamma(v) \cap \Gamma(v_{k-t}) \cap (EXT_u[k - t, k] \cup SUBG_u) \\ &= \Gamma(v) \cap \Gamma(v_{k-t}) \cap SUBG_u. \end{aligned}$$

第5章において導入した, $SUBG_{TAIL}$ のサイズによる場合分けに基づく探索領域削減

操作 (即ち, アルゴリズム MCP₂ における, 1413- 1422 行の操作) は, この新たな定義においても全く同様に適用可能である. (アルゴリズム MCP₃ においては, 1435- 1444 行にこの操作を記述する.)

アルゴリズム MCP₃ においては, この定義に基づき $G(EXT_u)$ が非クリークである場合にも次の新たな探索領域削減操作を導入する.

EXT_u 中に,

$|Γ(v_i) ∩ EXT_u| ≤ k - 3$ ($1 ≤ i ≤ k$) をみたす節点 $v_i ∈ EXT_u$, 即ち自身に隣接しない節点を 2 個以上持つ節点 v_i が存在する場合,

v_i を v_k と入れ替える. (1412 行)

更に, v_i に隣接しない各節点のうち 2 個を,

v_{k-1} , 及び v_{k-2} とそれぞれ入れ替える.

(1413, 1414 行)

この入れ替え操作が実行された場合,

$(v_{k-2}, v_k) ∉ E$, かつ $(v_{k-1}, v_k) ∉ E$

である. 従って, 2 節点 v_{k-1} , 及び v_k からの探索はこれを省略する.

節点 v_{k-2} からの探索については,

$(v_{k-2}, v_{k-1}) ∈ E$ である場合,

$$SUBG_{TAIL} = \hat{Γ}(\{v_{k-2}, v_{k-1}\}) ∩ SUBG_u,$$

$(v_{k-2}, v_{k-1}) ∉ E$ である場合,

$$SUBG_{TAIL} = ∅$$

とした上, $SUBG_{TAIL}$ に対する EXPAND の実行とする.

アルゴリズムにおいては, 1800 行の **for** 文の終了位置を $k - t$ としてこの操作を実現する. 即ち, 変数 t を,

上記入れ替え操作が実行された場合のみ $t = 3$ (1432 行),

それ以外の場合には $t = 2$ と設定される (1423 行)

変数とする.

この操作が実行された場合, EXT_u 中の各節点のうち $v_1, v_2, …, v_{k-3}$ からの探索は基本的に省略なしに実行される. このとき, $|SUBG| = η$, $|EXT_u| = k$ とすると,

$$|SUBG_{v_i}| ≤ |SUBG_u| = η - k - 1 \quad (1 ≤ i ≤ k - 3)$$

である. 更に, 節点 v_{k-2} からの探索において,

$$|SUBG_{TAIL}| \leq \eta - k - 2$$

である. 従って, 前章の計算量評価における, $G(EXT_u)$ が非クリークである場合より, 探索領域は確実に削減される.

上記操作が実行されない場合, EXT_u 中の任意の節点 v_i ($1 \leq i \leq k$) に対して,

$$|\Gamma(v_i) \cap EXT_u| \geq k - 2$$

である. 即ち, 各節点 v_i に隣接しない EXT_u 中の節点は高々 1 個である. この場合, 上記の,

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-2}, v_{k-1}\}) \cap SUBG_u$$

とする探索領域削減は適用できない. しかしながら, 次節において述べる新規操作により, この場合においても更なる探索領域の削減が可能である.

7.2.2 部分問題サイズの減少 (拡張版)

前章, 補題 6.2.1 において, $G(EXT_u)$ がクリークである場合,

EXT_u 中各節点からの探索は, 探索順が後の節点ほど探索領域が小さくなる

ことを示した. この結果は, 以下によるものである.

$$EXT_u = \{v_1, v_2, \dots, v_k\}$$

とするとき,

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \quad (1 \leq i \leq k)$$

とする. 即ち $SUBG_{v_i}$ は, $SUBG$ 中の節点 v_i に隣接する各節点の集合から, 既に探索済の各節点の集合,

$$u, v_1, v_2, \dots, v_{i-1}$$

を除いたものである.

$G(EXT_u)$ がクリークである場合, 節点 v_i は除去済の各節点 v_1, v_2, \dots, v_{i-1} 全てと隣接する. 従って節点除去の影響は最大となり, 補題 6.2.1 の結果となるが, v_i と除去済の各節点 v_1, v_2, \dots, v_{i-1} との関係をより詳細に考慮すれば, $G(EXT_u)$ が非クリークである場合にも適用可能な次の拡張結果が得られる.

補題 7.2.1 節点集合 $SUBG$ に対して, u を $G(SUBG)$ 中の最大次数節点として,

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \quad (1 \leq i \leq k)$$

とそれぞれおく. このとき,

$$\begin{aligned} & |SUBG_{v_i}| \\ & \leq |SUBG_u| - |\Gamma(v_i) \cap EXT_u[1, i-1]|. \end{aligned}$$

(証明) 定義より,

$$\begin{aligned} & SUBG_{v_i} \\ & = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u) \\ & = (\Gamma(v_i) \cap EXT_u[i+1, k]) \cup (\Gamma(v_i) \cap SUBG_u). \end{aligned}$$

ここで, $EXT_u[i+1, k] \cap SUBG_u = \emptyset$ である. 従って,

$$\begin{aligned} & |SUBG_{v_i}| \\ & = |\Gamma(v_i) \cap EXT_u[i+1, k]| + |\Gamma(v_i) \cap SUBG_u|. \end{aligned}$$

ここで, 節点 u は $G(SUBG)$ 中の最大次数節点であるから,

$$|\Gamma(v_i) \cap SUBG_u| \leq |SUBG_u| - |\Gamma(v_i) \cap EXT_u|$$

でなければならない. 何故ならば, 仮に

$$|\Gamma(v_i) \cap SUBG_u| > |SUBG_u| - |\Gamma(v_i) \cap EXT_u|$$

であるとすると,

$$\begin{aligned} & |\Gamma(v_i) \cap SUBG| \\ & = |\Gamma(v_i) \cap (\{u\} \cup EXT_u \cup SUBG_u)| \\ & = |\Gamma(v_i) \cap \{u\}| + |\Gamma(v_i) \cap EXT_u| + |\Gamma(v_i) \cap SUBG_u| \\ & > 0 + |\Gamma(v_i) \cap EXT_u| + |SUBG_u| - |\Gamma(v_i) \cap EXT_u| \\ & = |SUBG_u|. \end{aligned}$$

すなわち, v_i は $G(SUBG)$ における次数が u よりも大となる. これは u が $G(SUBG)$ 中の最大次数節点であるとしたことに反する. 故に,

$$\begin{aligned} & |SUBG_{v_i}| \\ & = |\Gamma(v_i) \cap EXT_u[i+1, k]| + |\Gamma(v_i) \cap SUBG_u| \end{aligned}$$

$$\begin{aligned}
&= |\Gamma(v_i) \cap EXT_u| - |\Gamma(v_i) \cap EXT_u[1, i]| + |\Gamma(v_i) \cap SUBG_u| \\
&\leq |\Gamma(v_i) \cap EXT_u| - |\Gamma(v_i) \cap EXT_u[1, i]| + (|SUBG_u| - |\Gamma(v_i) \cap EXT_u|) \\
&= |SUBG_u| - |\Gamma(v_i) \cap EXT_u[1, i - 1]|. \quad \square
\end{aligned}$$

前節に述べた通り、部分問題の統合による探索領域削減の詳細化 (拡張版) が実行されない場合、 EXT_u 中の任意の節点 v_i ($1 \leq i \leq k$) に対して、

$$|\Gamma(v_i) \cap EXT_u| \geq k - 2$$

である。従って、 $i \geq 2$ においては、

$$\begin{aligned}
&|SUBG_{v_i}| \\
&\leq |SUBG_u| - |\Gamma(v_i) \cap EXT_u[1, i - 1]| \\
&\leq |SUBG_u| - i + 2.
\end{aligned}$$

即ち、 $SUBG_{v_3}$ 以降はサイズ上限が確実に 1 ずつ小さくなる。

補題 6.2.1 により、 $G(EXT_u)$ がクリーク、かつ k が十分大きい場合には、特に複雑な探索領域削減手法を用いずとも、計算量上界が十分小さくなることを示すことが可能であった。補題 7.2.1 は、 $G(EXT_u)$ が非クリーク、かつ k が十分大きい場合においても同様の結果を得られることを示す。これにより、前章における、

- ・ k が十分大きい、あるいは小さい場合には、単純な探索領域削減操作のみを適用する
- ・ k がある 定数範囲内 である場合にのみ、より複雑な探索領域削減操作を追加適用する

という、アルゴリズムの基本的な方針をより一般化して確立している。

7.2.3 少数部分問題の分割 (拡張版)

前節において述べた通り、 k がある定数範囲内である場合においては、解析上より複雑な探索領域削減操作を導入する必要がある。具体的に第 6 章においては、 $k = 4$ である場合に追加の探索領域削減操作を適用し、この操作を“特別処理”と呼んだ。本章においても、 k がある定数範囲内において適用する操作を、以下“特別処理”と呼ぶ。

特別処理が適用される k の範囲は、最大時間計算量評価における指数の値によって決定される。具体的に MCP₃ においては、

(A) $G(EXT_u)$ がクリーク ($(v_{k-1}, v_k) \in E$), かつ $k \neq 3, 4, 5$ である場合:

又は、

$G(EXT_u)$ が非クリーク ($(v_{k-1}, v_k) \notin E$), かつ $k \neq 4, 5$ である場合:

においては、特別処理を適用しない。以下、この場合における処理を特別処理に対して“通常処理”と呼ぶ。

(B) $G(EXT_u)$ がクリーク ($(v_{k-1}, v_k) \in E$), かつ $3 \leq k \leq 5$ である場合:

又は、

$G(EXT_u)$ が非クリーク ($(v_{k-1}, v_k) \notin E$), かつ $4 \leq k \leq 5$

かつ、

$t = 2$ である場合:

においては、特別処理を適用する。

$G(EXT_u)$ が非クリークである場合において、 $t = 2$ を更に条件として加えている。これは、 $t = 3$ である場合、7.2.1 節の手法により十分大きな探索領域削減の効果が得られるため、更なる削減手法の適用が解析上必要ないことによる。

以下、特別処理の内容について述べる。

(B) 特別処理

いま、 $EXT_u = \{v_1, v_2, \dots, v_k\}$,

$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$ ($1 \leq i \leq k - 2$)

として、 $G(SUBG_{v_i})$ ($1 \leq i \leq k - 2$) 中の最大次数節点を u_i とする (2007 行)。また、

$SUBG_{u_i} = \Gamma(u_i) \cap SUBG_{v_i}$,

$EXT_{u_i} = SUBG_{v_i} - \{u_i\} - SUBG_{u_i}$

とする。

以下, 各節点 $v_i \in EXT_u$ ($1 \leq i \leq k-2$) の各々に対し, 次の (B-0), (B-1), あるいは (B-2) の場合に応じて処理の切り替えを行う.

(B-0) $|SUBG_{u_i}| \geq |SUBG_{v_i}| - 2$ である場合:

$SUBG_{v_i}$ に対して EXPAND を実行する. このとき, $SUBG_{v_i}$ は結果的に $SUBG_{u_i}$ だけ 1 個に置き換えられる.

この操作により, $SUBG_{v_i}$ のサイズは少なくとも 1 減少する. 探索領域を単純に削減可能な操作であるため, この操作に関しては回数制限 (後述) のもと, 繰り返し実行する. 即ち,

$$SUBG_{v_i} := SUBG_{u_i}$$

との設定を行い, 集合 L に u_i を追加する. この設定のもと, よりサイズが小さくなった $SUBG_{v_i}$ に対して (B-0), あるいは下記 (B-1) あるいは (B-2) の処理を再び実行する. (2021 -2023 行)

この操作が 1 回実行されるとき, $SUBG_{v_i}$ のサイズは単純に 1 以上減少する. 従って, この操作は繰り返し回数が増えるほど有効である. 但し, 計算量解析を単純化するため, この操作の繰り返し回数は上限 2 とする. このため, $|L| \geq 2$, 即ち既に 2 回この操作が実行された場合には, 繰り返しを中止し, $SUBG_{v_i}$ に対して EXPAND を実行する. (2024 行) (B-0) の操作が 1 回以上実行された後に $SUBG_{v_i}$ に対して EXPAND を実行する場合には, 集合 L 中の節点を v_i とともに探索中のクリーク Q に加える. (2013 行, 及び 2100 行)

(B-1) $|SUBG_{u_i}| = |SUBG_{v_i}| - 3$ である場合:

$SUBG_{v_i}$ に対して, 通常通り EXPAND を実行する. 実行により, 以下の部分問題に対してそれぞれ EXPAND が実行される. (ここで, $|EXT_{u_i}| = 2$ であるから, $EXT_{u_i} = \{v'_1, v'_2\}$ とする.)

まず,

$$SUBG_{u_i}$$

に対し EXPAND が実行される. 更に,

$$(v'_1, v'_2) \in E \text{ である場合には,}$$

$$SUBG_{TAIL} = \hat{\Gamma}(\{v'_1, v'_2\}) \cap SUBG_{u_i},$$

$(v'_1, v'_2) \notin E$ である場合には,

$$SUBG_{TAIL} = \emptyset$$

として, $SUBG_{TAIL}$ に対しても EXPAND が実行される.

(B-2) $|SUBG_{u_i}| \leq |SUBG_{v_i}| - 4$ である場合:

第6章, 6.2.3節において, (B) の操作は,

$$k = 4, \text{ かつ } G(EXT_u) \text{ がクリーク}$$

である場合においてのみ実行された. これに対して, 本章においては $G(EXT_u)$ が非クリークである場合にも前章同様の操作を実行する.

これを保証するため, 先ず第6章, 補題 6.2.3 を単純に拡張し, 次の結果を得る.

補題 7.2.2 $SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$ ($1 \leq i \leq k - 2$)

とすると, 以下が成立する.

$SUBG_{v_i}$ から $\omega(SUBG_u)$ よりも大きい最大クリークを抽出するためには,

$1 \leq i < j \leq k$ として,

$$\begin{aligned} & SUBG_{(v_i, v_j)} \quad ((v_i, v_j) \in E) \\ &= \Gamma(v_j) \cap (SUBG_{v_i} - \{v_{i+1}, v_{i+2}, \dots, v_{j-1}\}) \\ &= \Gamma(v_j) \cap (SUBG_{v_i} - EXT_u[i+1, k]) \quad (i+1 \leq j \leq k) \end{aligned}$$

なる 高々 $k - i$ 個の各節点集合に対して EXPAND を実行すれば十分である.

更に, $SUBG_{v_i}$ 中の, $G(SUBG_{v_i})$ における最大次数節点を u_i とするとき, 以下が成立する.

$G(SUBG_{v_i})$ ($1 \leq i \leq k - 2$) 中の最大次数節点を u_i とするとき,

$$|SUBG_{(v_i, v_j)}| \leq |\Gamma(u_i) \cap SUBG_{v_i}| - |\Gamma(v_i) \cap EXT_u[i+1, j-1]|.$$

(補題中, 下線を引いた部分は補題 6.2.3 からの拡張部分.)

(証明) 補題 6.2.3 と同様. \square

この (B-2) における操作のように, v_i からの探索を複数の部分問題 に対する EXPAND の実行とする, 即ち, 問題を複数の部分問題に分割する場合には, 分割によって解析上探索

領域削減が達成出来るかどうかを適切に管理する必要がある。このため、あらかじめ通常処理による場合と特別処理による場合それぞれにおける計算量上界を計算して、適切な特別処理対象を決める。その結果として、アルゴリズム MCP_3 においては、

$$i = k - 2,$$

即ち v_{k-2} に対してのみ特別処理対象を限定する (2004 行)。

上記に基づき、 $i = k - 2$ として、 $SUBG_{v_{k-2}}$ に対する (B-2) の操作の具体的内容について述べる。

いま、節点集合 A を、

$$A = \Gamma(v_{k-2}) \cap \{v_{i+1}, \dots, v_k\} = \Gamma(v_{k-2}) \cap EXT_u[i+1, k],$$

即ち、節点 v_i より探索順が後の節点のうち、 v_i に隣接する節点の集合とする。(ここで、 A は節点集合 $EXT_u[i+1, k]$ における順序を保存しているものとする。)

上記 (B-0) の操作において、 u_i として A 中の節点は選択され得る。このため、(B-2) の操作実行時には、

$$A = \Gamma(v_i) \cap (EXT_u[i+1, k] - L)$$

とする。(2009 行) この A により、 $SUBG_{v_{k-2}}$ を、

$$SUBG_{(v_{k-2}, A[j])} := \Gamma(A[j]) \cap (SUBG_{v_{k-2}} - A[1, j-1]) \quad (1 \leq j \leq |A|)$$

なる高々 $k - (k - 2) = 2$ 個の部分問題に分割し、それぞれに対して EXPAND を実行する。(2012- 2015 行)

$i = k - 2$ かつ $G(EXT_u)$ がクリークである場合、

$$A = \{v_{k-1}, v_k\},$$

即ち、部分問題数はちょうど 2 個である。

$i = k - 2$ かつ $G(EXT_u)$ が非クリークである場合、

$$A \subseteq \{v_{k-1}, v_k\}$$

である。従って、(B-0) の操作が 1 回実行された後、(B-2) の操作を実行すると、 $G(EXT_u)$ が非クリークである場合には、 $A = \emptyset$ となり得る。この場合には、 $SUBG_{v_{k-2}}$ に対して通常の EXPAND を実行する。(2010 行)

7.2.4 節点の隣接関係による限定

以上の各限定操作に加えて, $G(EXT_u)$ 中の各節点の隣接関係によって更に探索領域を削減できる. これを保証する次の補題を与える.

補題 7.2.3 $SUBG_{v_i} = \Gamma(v_i) \cap (EXT_u[i, k] \cup SUBG_u)$,

$(v_i, v_{i+1}) \in E$ とする. ここで,

$$|SUBG_{v_{i+1}} - SUBG_{v_i}| \leq 1$$

であるならば,

$$\omega(SUBG_{v_i}) \geq \omega(SUBG_{v_{i+1}}).$$

(証明) いま,

$$R := SUBG_{v_{i+1}} - SUBG_{v_i}$$

$$|R| \leq 1$$

とそれぞれおく.

$|R| = 0$ である場合,

$$SUBG_{v_{i+1}} \subseteq SUBG_{v_i}.$$

従って, 補題 2.2.3 により題意は成立する.

$|R| = 1$ である場合,

いま, $SUBG_{v_{i+1}}$ 中の任意の最大クリークを Q とおくと, Q は節点 v_{i+1} を含まない. ここで,

$$\omega(SUBG_{v_{i+1}}) > \omega(SUBG_{v_i})$$

であると仮定すると, 補題 2.2.4 により, Q は R 中の節点ただ 1 個を必ず含む. 更に,

$(v_i, v_{i+1}) \in E$ より,

$$(Q - R) \cup \{v_{i+1}\} \subseteq SUBG_{v_i}.$$

従って,

$$\begin{aligned} & \omega(SUBG_{v_i}) \\ & \geq |(Q - R) \cup \{v_{i+1}\}| \\ & = |Q| \\ & = \omega(SUBG_{v_{i+1}}). \end{aligned}$$

これは仮定に反する. □

MCP₃ においては, この結果を用いて $G(EXT_u)$ がクリークである場合に以下のように探索領域を削減する.

EXT_u に対して, 1403-1415 行の処理が実行された後, もし $(v_{k-1}, v_k) \in E$ であるならば, EXT_u 中には非隣接節点对が存在しない. 即ち, $G(EXT_u)$ はクリークである. ここで,

$$|(\Gamma(v_k) - \Gamma(v_{k-1})) \cap SUBG| \leq 1$$

ならば, $SUBG_{TAIL}$ を構成する前に, v_{k-1}, v_k をそれぞれ v_1, v_2 と入れ替える (1419-1421 行).

この入れ替え操作後は

$$|SUBG_{v_2} - SUBG_{v_1}| \leq 1$$

となるため, 補題 7.3.1 により,

$$\omega(\{v_1\} \cup SUBG_{v_1}) \geq \omega(\{v_2\} \cup SUBG_{v_2}).$$

従って, 節点 v_2 からの探索は省略する.

また,

$$R = (\Gamma(v_k) - \Gamma(v_{k-1})) \cap SUBG$$

とおくと, この入れ替え操作が実行されない場合,

$$|R| \geq 2$$

である. 即ち, v_k に隣接し, v_{k-1} に隣接しない節点は 2 個以上存在する. 更に, いま $G(EXT_u)$ はクリークであるから,

$$R \subseteq SUBG_u.$$

従って,

$$\begin{aligned} & |SUBG_{TAIL}| \\ &= |\hat{\Gamma}(\{v_k, v_{k-1}\}) \cap SUBG_u| \\ &\leq |SUBG_{v_{k-1}}| - \{v_k\} - |R| \\ &= |SUBG_{v_{k-1}}| - 3 \\ &\leq (|SUBG_u| - (k-1) + 1) - 3 \\ &= |SUBG_u| - k - 1. \end{aligned}$$

即ち, $SUBG_{TAIL}$ のサイズは小さくなる.

$(v_{k-1}, v_k) \notin E$ である場合には $SUBG_{TAIL} := \emptyset$ となるため, このサイズ限定を行わなくとも, 本章冒頭の結果を得られる.

7.2.5 例題

図 7.1 のグラフ $G = (V, E)$ (第 3-6 章において用いたグラフと, 同一のグラフ) に対するアルゴリズム MCP₃ の実行過程を, 図 7.2 の探索森として表す.

以下の実行過程における集合等の表記, 及び図 7.2 中における表記法は, 前章までの例題におけるものを踏襲する.

以下は具体的実行過程となる.

$$(V =)$$

$$SUBG^0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\},$$

$$SUBG_{\{1\}}^1 = \{6, 7, 8, 9, 10, 11, 12\},$$

$$EXT_1^1 = \{2, 3, 4, 5\},$$

$$SUBG'^0 = \{1\} \cup \{2, 3, 4, 5\} \cup \{6, 7, 8, 9, 10, 11, 12\},$$

ここで, $G(\{2, 3, 4, 5\})$ はクリークであるため, 1405-1414 行の交換操作は実行されない. 一方,

$$|(\Gamma(5) - \Gamma(4)) \cap SUBG^0| = |\emptyset| = 0$$

であるため, 1419-1421 行の交換操作が実行され, 節点 4 を 2 と, 5 を 3 とそれぞれ交換する.

$$SUBG'^0 = \{1\} \cup \{4, 5, 2, 3\} \cup \{6, 7, 8, 9, 10, 11, 12\},$$

$(2, 3) \in E$ であるため,

$$SUBG_{TAIL:\{2,3\}}^1 = \emptyset$$

を構成する. ここで,

$$|R| = |SUBG_{\{1\}}^1 - SUBG_{TAIL:\{2,3\}}^1| = |\{6, 7, 8, 9, 10, 11, 12\}| = 7.$$

即ち Case(c): $|R| \geq 3$ であるから, $SUBG_{\{1\}}^1$ に対して特別な操作は行わず, そのまま EXPAND を実行する.

以降, 節点 1 からの探索が終了するまでの過程は, 前章, 6.2.4 節と同一であり,

$$Q_{max_size} = 4$$

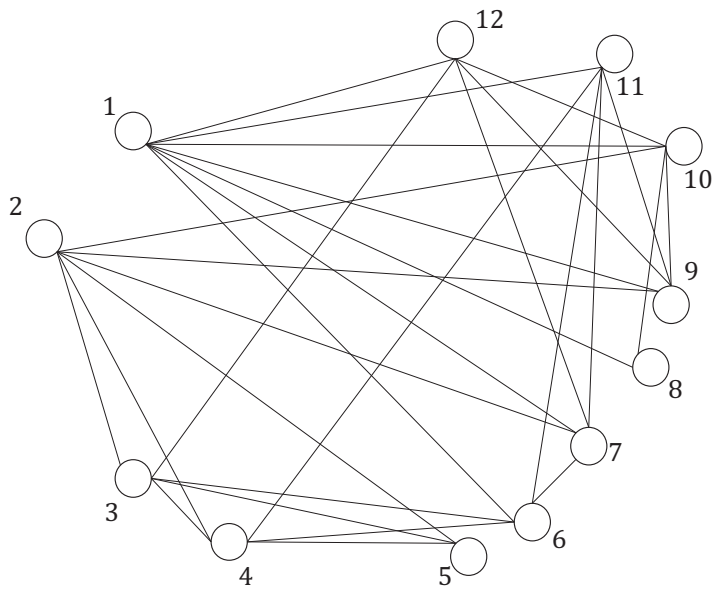


図 7.1 例題

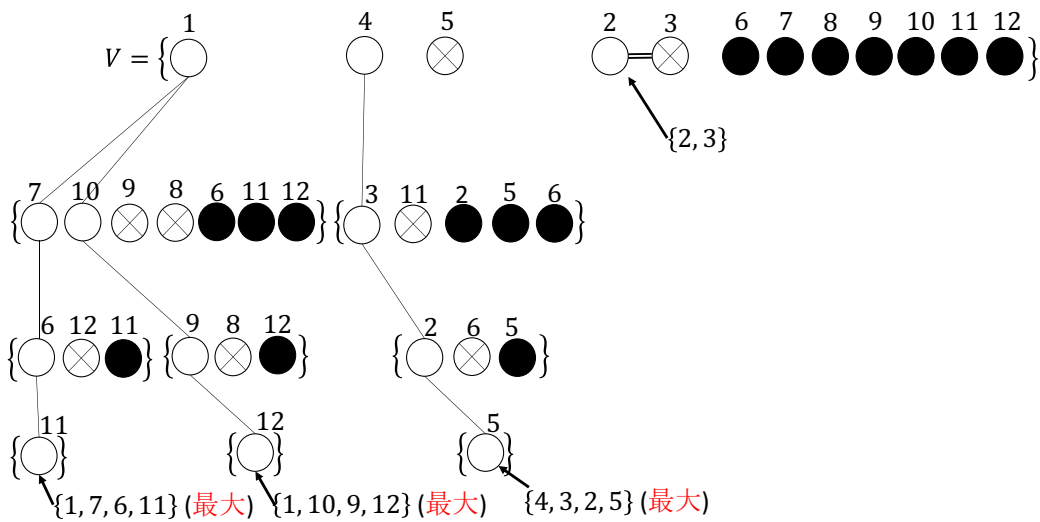


図 7.2 MCP_3 によるクリーク探索森

を得る.

$$|EXT_1^1| = |\{4, 5, 2, 3\}| = 4$$

かつ, EXT_1^1 はクリークであるが, いま,

$$k - 2 = |EXT_1^1| - 2 = 2$$

である. 従って, $SUBG_{\{4\}}^1 (= SUBG_{\{v_1\}}^1)$ に対しては通常の EXPAND を実行する.

$$SUBG_{\{4\}}^1 = \{2, 3, 5, 6, 11\},$$

$$SUBG_{\{4,3\}}^2 = \{2, 5, 6\},$$

$$EXT_3^2 = \{11\},$$

$$SUBG_{\{4\}}'^1 = \{3\} \cup \{11\} \cup \{2, 5, 6\},$$

$$SUBG_{\{4,3\}}^2 = \{2, 5, 6\},$$

$$SUBG_{\{4,3,2\}}^3 = \{5\},$$

$$EXT_2^3 = \{6\},$$

$$SUBG_{\{4,3\}}'^2 = \{2\} \cup \{6\} \cup \{5\},$$

$$SUBG_{\{4,3,2\}}^3 = \{5\},$$

$$SUBG_{\{4,3,2,5\}}^4 = \emptyset,$$

$$EXT_5^4 = \emptyset,$$

$$SUBG_{\{4,3,2\}}^3 = \{5\} \cup \emptyset \cup \emptyset,$$

$$SUBG_{\{4,3,2,5\}}^4 = \emptyset, \text{ (更新なし)}$$

3 段階バックトラック,

$$EXT_1^1 = \{4, 5, 2, 3\},$$

ここで,

$$SUBG_{\{4\}}^1 = \{2, 3, 5, 6, 11\},$$

$$SUBG_{\{5\}}^1 = \{2, 3\},$$

$$|SUBG_{\{5\}}^1 - SUBG_{\{4\}}^1| = |\emptyset| = 0 < 1,$$

により, 節点 5 からの探索は省略する.

$$SUBG_{TAIL:\{2,3\}}^1 = \emptyset, \text{ (更新なし)}$$

実行終了.

結果として, MCP₃ は例題のグラフ G 中の最大クリークサイズ 4 を得る.

7.3 最大時間計算量評価

アルゴリズム MCP_3 の最大時間計算量評価の基本として、まず、次の様な非再帰手続き $EXPAND_0(SUBG)$ の時間計算量を考える。

EXPAND₀(SUBG): アルゴリズム 5 の procedure $EXPAND(SUBG)$ において、各再帰呼び出し部分 $EXPAND(SUBG_*)$ をそれぞれ $EXPAND(\emptyset)$ に置き換えた非再帰手続き。

補題 7.3.1 $EXPAND_0(SUBG)$ の最大時間計算量は、
 $O(|SUBG| \cdot |SUBG_u|^2)$ である。

(証明) 本章追加部分であるアルゴリズム 5 における 1409-1434 行, 1701 行, 及び 2001-2026 行を除く部分については、前章までにおいて題意の成立を示した。またこの追加部分のうち、2019-2024 行部分を除く各操作については、前章までと同様にして題意の成立を示せる。

同部分における各行の操作は、 $O(|SUBG_u|)$ -時間で実行可能であり、2023 行による行移動の結果、2019 行の条件が再び成立した場合には、2019-2024 行の操作が再び実行される。この繰り返し操作は最大で 2 回実行される。従って、2001-2026 行における操作は、 $O(|SUBG_u|^2)$ -時間で実行可能である。以上から、アルゴリズム 5 に関して、題意は成立する。□

前章までと同様に、 MCP_3 に関しても $T(|SUBG|)$ に対して以下の基礎的關係式が成立する。

集合 $SUBG \subseteq V$ に対して、節点 u を $G(SUBG)$ における最大次数節点として、

$$SUBG_u = \Gamma(u) \cap SUBG,$$

$$EXT_u = SUBG - \{u\} - SUBG_u = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_u|),$$

$$SUBG_{TAIL} = \hat{\Gamma}(\{v_{k-1}, v_k\}) \cap SUBG_u.$$

このとき、次の關係式が成立する。

$$\begin{aligned}
& T(|SUBG|) \\
& \leq T(|SUBG_u|) + \sum_{i=1}^{k-t} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG| \cdot |SUBG_u|^2
\end{aligned} \tag{7.3.1}$$

また,

$$T(|SUBG|) < T(|SUBG_u|) + \sum_{i=1}^{k-t} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \tag{7.3.2}$$

$(v_{k-1}, v_k) \in E$ かつ $3 \leq k \leq 5$ である場合, および $(v_{k-1}, v_k) \notin E$, かつ $4 \leq k \leq 5$ である場合には, $SUBG_{v_{k-2}}$ に対して特別処理が実行される. (下記 補題 7.3.2(証明), 場合分けの【2】, 及び【3】において詳述する.)

主要定理証明に先立ち, 先ず, EXPAND($SUBG$) に対する時間計算量評価結果を以下に与える.

補題 7.3.2 節点集合 $SUBG \subseteq V$ に対して, $|SUBG| = \eta$, $C' = 10^4 C$ とおく. このとき, EXPAND($SUBG$) の最大時間計算量 $T(|SUBG|)$ は以下を満たす.

$$T(|SUBG|) \leq C' 2^{0.3144\eta} \eta^3.$$

(証明) $\eta \geq 1$ に関する数学的帰納法による.

先ず, $\eta = 1$ の場合, 題意は明らかに成立する.

次に, $|SUBG|$ がある正整数 η 以下である場合において, 題意が成立すると仮定する.

この仮定のもとで, $|SUBG| = \eta + 1$ である場合を考える.

前章と同様に, u を $G(SUBG)$ における最大次数節点として,

$$SUBG_u = \Gamma(u) \cap SUBG$$

として,

$$|SUBG_u| = \eta - k \quad (0 \leq k < \eta)$$

とする.

このとき,

$$\begin{aligned}
 & |EXT_u| \\
 &= |SUBG - \{u\} - SUBG_u| \\
 &= |SUBG| - |\{u\}| - |SUBG_u| \\
 &= (\eta + 1) - 1 - (\eta - k) \\
 &= k.
 \end{aligned}$$

以下, $G(EXT_u)$ に関しての場合分けをした上で, 解析を進める.

まず, 次の簡単な各場合に対し, 題意の成立を示す.

[1] k が十分小さい, 又は大きい場合:

(1-0) $k \leq 1$ である場合;

第4章, 補題 4.3.2 証明, 場合分けの (b) と同様にして示せる.

(1-1) $k = 2$ である場合;

$EXT_u = \{v_1, v_2\}$ とするとき, もし v_1 と v_2 が隣接しない場合, $SUBG$ の子問題において探索されるのは $SUBG_u$ のみである. 従って, (1-1) と同様にして題意は成立する.

v_1 と v_2 が隣接する場合, 式 (7.3.2) により,

$$\begin{aligned}
 & T(|SUBG|) \\
 &= T(\eta + 1) \\
 &< T(|SUBG_u|) + T(|SUBG_{TAIL}|) + C(\eta + 1)^3 \\
 &\leq T(\eta - 2) + T((\eta - 2) - 1) + C(\eta + 1)^3 \\
 &< (2^{-0.3144 \cdot 2} + 2^{-0.3144 \cdot 3} + 10^{-4}) C' 2^{0.3144\eta} (\eta + 1)^3 \\
 &< 1.1669 \cdot C' 2^{0.3144\eta} (\eta + 1)^3 \\
 &< 2^{0.3144} C' 2^{0.3144\eta} (\eta + 1)^3 \quad (\text{ここで, } 2^{0.3144} = 1.2434\dots) \\
 &< C' 2^{0.3144(\eta+1)} (\eta + 1)^3.
 \end{aligned}$$

よって, この場合も題意は成立する.

以上, $k \leq 2$ である場合には題意が成立する事が示された.

従って、以下では、全て $k \geq 3$ である場合 における題意の成立を示す。

(1-2) $G(EXT_u)$ がクリーク ($(v_{k-1}, v_k) \in E$), かつ $k \geq 6$ である場合:

式 (7.3.2) ($t = 2$) より,

$$T(|SUBG|)$$

$$< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3.$$

帰納法の仮定により,

$$T(|SUBG|)$$

$$= T(\eta + 1)$$

$$< C'2^{0.3144 \cdot (\eta - k)}(\eta - k)^3 + \sum_{i=1}^{k-2} C'2^{0.3144 \cdot ((\eta - k) - i + 1)}((\eta - k) - i + 1)^3 \\ + T(|SUBG_{TAIL}|) + C(\eta + 1)^3.$$

ここで,

$$|SUBG_{TAIL}|$$

$$\leq |SUBG_{v_{k-1}}| - 1$$

$$= (\eta - k) - (k - 1) + 1 - 1 \quad (\text{補題 7.2.1 より})$$

$$= (\eta - k) - (k - 1).$$

従って,

$$T(|SUBG|)$$

$$= T(\eta + 1)$$

$$< (1 + \sum_{i=1}^{k-2} 2^{-0.3144(i-1)})C'2^{0.3144 \cdot (\eta - k)}(\eta + 1)^3 + C'2^{0.3144((\eta - k) - (k - 1))}(\eta + 1)^3 \\ + C(\eta + 1)^3$$

$$= (1 + \sum_{i=1}^{k-2} 2^{-0.3144(i-1)} + 2^{-0.3144(k-1)})C'2^{0.3144(\eta - k)}(\eta + 1)^3 + C(\eta + 1)^3$$

$$< (1 + \frac{1 - 2^{-0.3144(k-2)}}{1 - 2^{-0.3144}} + 2^{-0.3144(k-1)} + 10^{-4})2^{-0.3144k}C'2^{0.3144\eta}(\eta + 1)^3.$$

ここで, 第3章, 補題 3.6.3 と同様にして,

$$(1 + \frac{1 - 2^{-0.3144(k-2)}}{1 - 2^{-0.3144}} + 2^{-0.3144(k-1)} + 10^{-4})2^{-0.3144k} < 1.1652$$

が成立することを証明できる (左辺は $k \geq 6$ のもと, $k = 6$ で最大値 1.16507... をとる).

従って,

$$T(|SUBG|) = T(\eta + 1)$$

$$< 1.1652 \cdot C'2^{0.3144\eta}(\eta + 1)^3$$

$$\begin{aligned} &< 2^{0.3144} \cdot C' 2^{0.3144\eta} (\eta + 1)^3 \\ &= C' 2^{0.3144(\eta+1)} (\eta + 1)^3. \end{aligned}$$

(1-3) $G(EXT_u)$ が非クリーク, かつ $k = 3$ あるいは $k \geq 6$ である場合:

先ず, EXT_u において,

$$|\Gamma(v_i) \cap EXT_u| \leq k - 3$$

である節点 (即ち, EXT_u 内で隣接しない節点を 2 個以上持つ節点) v_i が存在する場合を考える. アルゴリズム 5 における 1412-1414 行の操作により, そのような節点 v_i は EXT_u 中の最後尾節点 v_k と, また v_i と隣接しない節点のうち 2 個がそれぞれ v_{k-2}, v_{k-1} と入れ換えられる. 従って, 式 (7.3.2) ($t = 3$) より,

$$\begin{aligned} &T(|SUBG|) \\ &< T(|SUBG_u|) + \sum_{i=1}^{k-3} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3. \end{aligned}$$

補題 6.3.3 により, $SUBG_{TAIL} \neq \emptyset$ であるならば Case(c) の場合に $T(|SUBG|)$ の上界は最大となる. 従って帰納法の仮定により,

$$\begin{aligned} &T(|SUBG|) \\ &= T(\eta + 1) \\ &< (k - 2)C' 2^{0.3144(\eta-k)} (\eta + 1)^3 + C' 2^{0.3144(\eta-k-3)} \cdot (\eta + 1)^3 + C(\eta + 1)^3 \\ &< \left(\frac{k-2+2^{-0.3144 \cdot 3}}{2^{0.3144k}} + 10^{-4} \right) C' 2^{0.3144\eta} (\eta + 1)^3. \end{aligned}$$

ここで, $k \geq 3$ において,

$$\frac{k-2+2^{-0.3144 \cdot 3}}{2^{0.3144k}} + 10^{-4} < 1.2227 < 2^{0.3144}$$

である事が示せる. (左辺は $k \geq 3$ において, $k = 6$ で最大値 1.22269... をとる.) 従って,

$$\begin{aligned} &T(|SUBG|) \\ &= T(\eta + 1) \\ &< C' 2^{0.3144(\eta+1)} \cdot (\eta + 1)^3. \end{aligned}$$

$SUBG_{TAIL} = \emptyset$ である場合に題意が成立することは, 前記計算過程から明らか.

従って, 以降における 補題 7.3.2 の証明の全てにおいて, $G(EXT_u)$ が非クリークである場合には,

任意の節点 $v_i \in EXT_u$ ($1 \leq i \leq k$) に対して,

$$\underline{|\Gamma(v_i) \cap EXT_u| \geq k - 2}$$

である場合のみを考える.

このとき, 補題 7.2.1 より,

$$\begin{aligned} & |SUBG_{v_i}| \\ & \leq |SUBG_u| - |\Gamma(v_i) \cap EXT_u[1, i-1]| \end{aligned}$$

であるから, $2 \leq i \leq k-2$ において,

$$\begin{aligned} & |SUBG_{v_i}| \\ & \leq |SUBG_u| - i + 2 \\ & = \eta - k - i + 2. \end{aligned}$$

従って, 式 (7.3.2) ($t = 2, SUBG_{TAIL} = \emptyset$) より,

$$\begin{aligned} & T(|SUBG|) \\ & = T(\eta + 1) \\ & < T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + C|SUBG|^3 \\ & \leq T(\eta - k) + T(\eta - k) + \sum_{i=2}^{k-2} T(\eta - k - i + 2) + C(\eta + 1)^3. \end{aligned}$$

帰納法の仮定により,

$$\begin{aligned} & T(|SUBG|) \\ & = T(\eta + 1) \\ & < (1 + 1 + 2^{0.3144} \sum_{i=2}^{k-2} 2^{-0.3144(i-1)} + 10^{-4}) 2^{-0.3144k} C' 2^{0.3144\eta} (\eta + 1)^3. \\ & = (1 + 1 + 2^{0.3144} \cdot (\frac{1-2^{-0.3144(k-2)}}{1-2^{-0.3144}} - 1) + 10^{-4}) 2^{-0.3144k} C' 2^{0.3144\eta} (\eta + 1)^3. \end{aligned}$$

前章, 補題 6.2.1 は, $G(EXT_u)$ がクリークである場合に対してのみ問題サイズの削減に利用できた. 本章において新たに補題 7.2.1 として拡張した事により, $G(EXT_u)$ がクリークでない場合にも, 同様の問題サイズ削減が可能となった.

いま $k = 3$ あるいは $k \geq 6$ において,

$$(1 + 1 + 2^{0.3144} \cdot (\frac{1-2^{-0.3144(k-2)}}{1-2^{-0.3144}} - 1) + 10^{-4}) 2^{-0.3144k} < 1.205 < 2^{0.3144}$$

が成立する. (左辺は $k = 3$ あるいは $k \geq 6$ のもと, $k = 6$ で最大値 1.2039... をとる.)

従って, (1-2) と同様に,

$$\begin{aligned} & T(|SUBG|) \\ & = T(\eta + 1) \\ & \leq C' 2^{0.3144(\eta+1)} (\eta + 1)^3. \end{aligned}$$

以上のように, $k = |EXT_u|$ が十分小さい, あるいは大きい場合には, いずれもアルゴリズムにおける探索領域削減操作のうち, 単純なものだけを用いて題意の成立を示すことが可能である. ここで, 本証明において題意の成立がまだ示されていない場合は, 次の2通りである.

$G(EXT_u)$ がクリーク, かつ $3 \leq k \leq 5$ である場合
及び,

$G(EXT_u)$ が非クリーク, かつ $4 \leq k \leq 5$,

かつ,

$t = 2$, 即ち, 上記 (1-3) の結果から,

任意の節点 $v_i \in EXT_u$ ($1 \leq i \leq k - 2$) に対して,

$$|\Gamma(v_i) \cap EXT_u| \geq k - 2$$

である場合

これらの2通りに関しては, 【1】において用いた基本的な探索領域削減操作に加えて, 更に複雑な探索領域削減操作を適用する. ただし, いずれの場合も k は高々定数範囲内であり, 各操作によって発生する場合分けも, 有限個の範囲内となる.

【2】 $G(EXT_u)$ がクリーク, かつ $3 \leq k \leq 5$ である場合:

式 (7.3.2) ($t = 2$) より,

$$T(|SUBG|)$$

$$< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3.$$

先ず,

(2-1) $|(\Gamma(v_k) - \Gamma(v_{k-1})) \cap SUBG| \leq 1$ である場合;

1417-1421 行の操作により, この場合,

$$|SUBG_{v_2} - SUBG_{v_1}| \leq 1.$$

このとき, 2002, 2003 行により, $SUBG_{v_2} := \emptyset$ との代入がなされる. そこでこの場合は, (1-2) の場合において

$$|SUBG_{v_2}| = 0,$$

$$|SUBG_{TAIL}| = \eta - k - 1$$

とすれば, $3 \leq k \leq 5$ における題意の成立を証明できる.

(2-2) $|SUBG_{v_k} - SUBG_{v_{k-1}}| \geq 2$ である場合;

まず, 7.2.4 節に述べた通り, この場合,

$$\begin{aligned} & |SUBG_{TAIL}| \\ &= |SUBG_u| - k - 1. \end{aligned}$$

即ち,

$$\begin{aligned} & T(|SUBG_{TAIL}|) \\ & \leq T(|SUBG_u| - k - 1) \\ & < C'2^{-0.3144(k+1)}C'2^{0.3144(\eta-k)}(\eta+1)^3. \end{aligned}$$

次に, 節点 v_{k-2} に対しては特別処理が実行される. そこで,

v_{k-2} からの探索において, 特別処理 (B-0) ~ (B-2) がそれぞれ実行された場合における, 計算量上界について示す.

まず, (B-0) の操作が実行されなかった各場合について考える.

(B-1) $|SUBG_{u_{k-2}}| = |SUBG_{v_{k-2}}| - 3$ である場合:

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < T(|SUBG_{u_{k-2}}|) + T(|SUBG_{TAIL}|) + C|SUBG_{v_{k-2}}|^3 \\ & < T(|SUBG_{v_{k-2}}| - 3) + T(|SUBG_{v_{k-2}}| - 6) + C(\eta+1)^3. \quad (\text{Case(c) による}) \end{aligned}$$

いま, $G(EXT_u)$ はクリークであるから,

$$\begin{aligned} & |SUBG_{v_{k-2}}| \\ & \leq |SUBG_u| - (k-2) + 1 \\ & = |SUBG_u| - k + 3. \end{aligned}$$

従って,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < T(|SUBG_u| - k + 3 - 3) + T(|SUBG_u| - k + 3 - 6) + C(\eta+1)^3 \\ & = T(|SUBG_u| - k) + T(|SUBG_u| - k - 3) + C(\eta+1)^3 \end{aligned}$$

$$\begin{aligned}
&= T((\eta - k) - k) + T((\eta - k) - k - 3) + C(\eta + 1)^3 \\
&< (2^{-0.3144k} + 2^{-0.3144(k+3)})C'2^{0.3144(\eta-k)}(\eta + 1)^3 + C(\eta + 1)^3 \\
&< 2^{-0.3144k}(1 + 2^{-0.3144 \cdot 3} + 10^{-4})C'2^{0.3144(\eta-k)}(\eta + 1)^3.
\end{aligned}$$

ここで, $1 + 2^{-0.3144 \cdot 3} + 10^{-4} = 1.5201\dots$ より, (B-1)において,

$$\begin{aligned}
&T(|SUBG_{v_{k-2}}|) \\
&< 1.5202 \cdot 2^{-0.3144k}C'2^{0.3144(\eta-k)}(\eta + 1)^3.
\end{aligned}$$

(B-2) $|SUBG_{u_{k-2}}| \leq |SUBG_{v_{k-2}}| - 4$ である場合:

いま $G(EXT_u)$ はクリークであるから, 7.2.3 節に述べた通り,

$$A := \Gamma(v_{k-2}) \cap \{v_{k-1}, v_k\} = \{v_{k-1}, v_k\}.$$

従って,

$$\begin{aligned}
&T(|SUBG_{v_{k-2}}|) \\
&< \sum_{j=k-1}^k T(|SUBG_{(v_{k-2}, v_j)}|) + C|SUBG_{v_{k-2}}|^3 \\
&= T(|SUBG_{(v_{k-2}, v_{k-1})}|) + T(|SUBG_{(v_{k-2}, v_k)}|) + C|SUBG_{v_{k-2}}|^3 \\
&\leq T(|SUBG_{v_{k-2}}| - 4) + T(|SUBG_{v_{k-2}}| - 5) + C|SUBG_{v_{k-2}}|^3 \\
&\leq T(|SUBG_u| - k + 3 - 4) + T(|SUBG_u| - k + 3 - 5) + C|SUBG_{v_{k-2}}|^3 \\
&= T(|SUBG_u| - k - 1) + T(|SUBG_u| - k - 2) + C|SUBG_{v_{k-2}}|^3 \\
&< (2^{-0.3144(k+1)} + 2^{-0.3144(k+2)})C'2^{0.3144(\eta-k)}(\eta + 1)^3 + C(\eta + 1)^3 \\
&< 2^{-0.3144k}(2^{-0.3144} + 2^{-0.3144 \cdot 2} + 10^{-4})C'2^{0.3144(\eta-k)}(\eta + 1)^3.
\end{aligned}$$

ここで, $2^{-0.3144} + 2^{-0.3144 \cdot 2} + 10^{-4} = 1.4509\dots$ より, (B-2)において,

$$\begin{aligned}
&T(|SUBG_{v_{k-2}}|) \\
&< 1.451 \cdot 2^{-0.3144k}C'2^{0.3144(\eta-k)}(\eta + 1)^3.
\end{aligned}$$

(B-0) $|SUBG_{u_{k-2}}| \geq |SUBG_{v_{k-2}}| - 2$ である場合:

(B-0)における置き換え操作,

$SUBG_{v_{k-2}} := \Gamma(u_{k-2}) \cap SUBG_{v_{k-2}}$ が2回実行された場合, 置き換え後の $SUBG_{v_{k-2}}$ に対して EXPAND が実行される. ここで, 置き換え操作2回実行後の節点集合を $SUBG''_{v_{k-2}}$ とおくと, 操作1回につき集合のサイズは少なくとも1減少する. 従って,

$$T(|SUBG_{v_{k-2}}|)$$

$$\begin{aligned}
&< T(|SUBG''_{v_{k-2}}|) + 2C(\eta + 1)^3 \\
&\leq T(|SUBG_{v_{k-2}}| - 2) + 2C(\eta + 1)^3 \\
&\leq T(|SUBG_u| - k + 3 - 2) + 2C(\eta + 1)^3 \\
&= T(|SUBG_u| - k + 1) + 2C(\eta + 1)^3 \\
&< 2^{-0.3144(k-1)} C' 2^{0.3144(\eta-k)} (\eta + 1)^3 + 2C(\eta + 1)^3 \\
&< 2^{-0.3144k} (2^{0.3144} + 2 \cdot 10^{-4}) C' 2^{0.3144(\eta-k)} (\eta + 1)^3.
\end{aligned}$$

ここで, $2^{0.3144} + 2 \cdot 10^{-4} = 1.2436\dots$ より,

$$\begin{aligned}
&T(|SUBG_{v_{k-2}}|) \\
&< 1.2437 \cdot 2^{-0.3144k} C' 2^{0.3144(\eta-k)} (\eta + 1)^3.
\end{aligned}$$

次に, (B-0) の操作が 1 回実行された場合, 置き換え操作 1 回実行後の節点集合を $SUBG'_{v_{k-2}}$ とおくと,

$$|SUBG'_{v_{k-2}}| \leq |SUBG_{v_{k-2}}| - 1.$$

上記より, $SUBG'_{v_{k-2}}$ に対して,

(B-1) の操作が実行された場合,

$$\begin{aligned}
&T(|SUBG_{v_{k-2}}|) \\
&< T(|SUBG'_{v_{k-2}}|) + C(\eta + 1)^3 \\
&\leq T(|SUBG_{v_{k-2}}| - 1) + C(\eta + 1)^3 \\
&< T(|SUBG_{v_{k-2}}| - 1 - 3) + T(|SUBG_{v_{k-2}}| - 1 - 6) + 2C(\eta + 1)^3 \\
&< 2^{-0.3144(k+1)} (1 + 2^{-0.3144 \cdot 3} + 2 \cdot 10^{-4}) C' 2^{0.3144(\eta-k)} (\eta + 1)^3.
\end{aligned}$$

(B-2) の操作が実行された場合,

$$\begin{aligned}
&T(|SUBG_{v_{k-2}}|) \\
&< T(|SUBG'_{v_{k-2}}|) + C(\eta + 1)^3 \\
&\leq T(|SUBG_{v_{k-2}}| - 1) + C(\eta + 1)^3 \\
&< T(|SUBG_{v_{k-2}}| - 1 - 4) + T(|SUBG_{v_{k-2}}| - 1 - 5) + 2C(\eta + 1)^3 \\
&< 2^{-0.3144(k+1)} (2^{-0.3144} + 2^{-0.3144 \cdot 2} + 2 \cdot 10^{-4}) C' 2^{0.3144(\eta-k)} (\eta + 1)^3.
\end{aligned}$$

いずれの場合にも, (B-0) の置き換え操作が実行されなかった場合より明らかに計算量上

界は小さくなる. 従って, v_{k-2} からの探索において計算量上界が最大となるのは (B-1) の操作が実行された場合となる. そこで,

$$C_{k-2} = 2^{-0.3144k}(1 + 2^{-0.3144 \cdot 3} + 10^{-4})$$

とおく. 即ち,

$$T(|SUBG_{v_{k-2}}|) < C_{k-2}C'2^{0.3144(\eta-k)}(\eta+1)^3.$$

$1 \leq i \leq k-3$ において, 節点 v_i に対しては通常処理が実行される. このとき,

$$\begin{aligned} & T(|SUBG_{v_i}|) \\ & \leq T(|SUBG_u| - i + 1) \\ & < 2^{-0.3144(i-1)}C'2^{0.3144(\eta-k)}(\eta+1)^3. \end{aligned}$$

以上から,

$$\begin{aligned} & T(|SUBG|) \\ & < T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\ & < C'2^{0.3144(\eta-k)}(\eta+1)^3 \\ & \quad + \sum_{i=1}^{k-3} 2^{-0.3144(i-1)}C'2^{0.3144(\eta-k)}(\eta+1)^3 \\ & \quad + C_{k-2}C'2^{0.3144(\eta-k)}(\eta+1)^3 \\ & \quad + 2^{-0.3144(k+1)}C'2^{0.3144(\eta-k)}(\eta+1)^3 \\ & \quad + C(\eta+1)^3. \end{aligned}$$

いま $3 \leq k \leq 5$ である. よって,

$k = 3$ である場合,

$$\begin{aligned} & T(|SUBG|) \\ & < T(|SUBG_u|) + T(|SUBG_{v_1}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\ & < C'2^{0.3144(\eta-3)}(\eta+1)^3 \\ & \quad + C_1C'2^{0.3144(\eta-3)}(\eta+1)^3 \\ & \quad + 2^{-0.3144(3+1)}C'2^{0.3144(\eta-3)}(\eta+1)^3. \\ & \quad + C(\eta+1)^3 \\ & = (1 + C_1 + 2^{-0.3144 \cdot 4} + 10^{-4})2^{-0.3144 \cdot 3}C'2^{0.3144\eta}(\eta+1)^3 \\ & < 1.1489C'2^{0.3144\eta}(\eta+1)^3 \end{aligned}$$

$$\begin{aligned}
&< 2^{0.3144} C' 2^{0.3144\eta} (\eta + 1)^3 \\
&= C' 2^{0.3144(\eta+1)} (\eta + 1)^3.
\end{aligned}$$

$k = 4$ である場合,

$$\begin{aligned}
&T(|SUBG|) \\
&< T(|SUBG_u|) + \sum_{i=1}^2 T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\
&< (1 + 1 + C_2 + 2^{-0.3144 \cdot 5} + 10^{-4}) 2^{-0.3144 \cdot 4} C' 2^{0.3144\eta} (\eta + 1)^3 \\
&< \underline{1.2432} C' 2^{0.3144\eta} (\eta + 1)^3 \\
&< 1.2434 C' 2^{0.3144\eta} (\eta + 1)^3 \quad (\text{ここで, } 2^{0.3144} = 1.24349\dots) \\
&< 2^{0.3144} C' 2^{0.3144\eta} (\eta + 1)^3 \\
&= C' 2^{0.3144(\eta+1)} (\eta + 1)^3.
\end{aligned}$$

$k = 5$ である場合,

$$\begin{aligned}
&T(|SUBG|) \\
&< T(|SUBG_u|) + \sum_{i=1}^3 T(|SUBG_{v_i}|) + T(|SUBG_{TAIL}|) + C|SUBG|^3 \\
&< (1 + 1 + 2^{-0.3144} + C_3 + 2^{-0.3144 \cdot 6} + 10^{-4}) 2^{-0.3144 \cdot 5} C' 2^{0.3144\eta} (\eta + 1)^3 \\
&< 1.2062 C' 2^{0.3144\eta} (\eta + 1)^3 \\
&< 2^{0.3144} C' 2^{0.3144\eta} (\eta + 1)^3 \\
&= C' 2^{0.3144(\eta+1)} (\eta + 1)^3.
\end{aligned}$$

以上, いずれの場合においても題意は成立する. 本章における MCP₃ の計算量評価において, 計算量上界が最も大きくなるのは上記のうち $k = 4$ である場合であり, これが定数値 0.3144 の根拠となる.

[3] $G(EXT_u)$ が非クリーク, かつ $4 \leq k \leq 5$ である場合:

$$\begin{aligned}
&\text{式 (7.3.2) } (t = 2, SUBG_{TAIL} = \emptyset) \text{ より,} \\
&T(|SUBG|) \\
&= T(\eta + 1) \\
&< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + C|SUBG|^3.
\end{aligned}$$

いま $t = 2$, 及び (1-3) の結果から,

任意の節点 $v_i \in EXT_u$ ($1 \leq i \leq k$) に対して,

$$|\Gamma(v_i) \cap EXT_u| \geq k - 2$$

である. 即ち, v_i に隣接しない EXT_u 中の節点は高々 1 個である. 従って,

$$|SUBG_{v_i}| \leq |SUBG_u| - i + 2.$$

【2】 (2-2) と同様に, 節点 v_{k-2} に対しては特別処理が実行される. そこで, v_{k-2} からの探索において, 特別処理 (B-0) ~ (B-2) がそれぞれ実行された場合における, 計算量上界について示す.

まず, (B-0) の操作が実行されなかった各場合について考える.

(B-1) $|SUBG_{u_{k-2}}| = |SUBG_{v_{k-2}}| - 3$ である場合:

【2】 (2-2) における (B-1) と同様に,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < T(|SUBG_{u_{k-2}}|) + T(|SUBG_{TAIL}|) + C|SUBG_{v_{k-2}}|^3 \\ & \leq T(|SUBG_{v_{k-2}}| - 3) + T(|SUBG_{v_{k-2}}| - 6) + C(\eta + 1)^3. \quad (\text{Case(c) による}) \end{aligned}$$

但し,

$$\begin{aligned} & |SUBG_{v_{k-2}}| \\ & \leq |SUBG_u| - (k - 2) + 2 \\ & = |SUBG_u| - k + 4. \end{aligned}$$

従って,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < (2^{-0.3144(k-1)} + 2^{-0.3144(k+2)})C'2^{0.3144(\eta-k)}(\eta + 1)^3 + C(\eta + 1)^3 \\ & < 2^{-0.3144k}(2^{0.3144} + 2^{-0.3144 \cdot 2} + 10^{-4})C'2^{0.3144(\eta-k)}(\eta + 1)^3. \end{aligned}$$

ここで, $2^{0.3144} + 2^{-0.3144 \cdot 2} + 10^{-4} = 1.8903\dots$ より, (B-1) において,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < 1.8904 \cdot 2^{-0.3144k}C'2^{0.3144(\eta-k)}(\eta + 1)^3. \end{aligned}$$

(B-2) $|SUBG_{u_{k-2}}| \leq |SUBG_{v_{k-2}}| - 4$ である場合:

いま $(v_{k-1}, v_k) \notin E$ であるから, 前提により v_{k-1}, v_k はともに v_{k-2} に隣接する. よって,

$$A := \Gamma(v_{k-2}) \cap \{v_{k-1}, v_k\} = \{v_{k-1}, v_k\}.$$

従って,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < \sum_{j=k-1}^k T(|SUBG_{(v_{k-2}, v_j)}|) + C|SUBG_{v_{k-2}}|^3 \\ & = T(|SUBG_{(v_{k-2}, v_{k-1})}|) + T(|SUBG_{(v_{k-2}, v_k)}|) + C|SUBG_{v_{k-2}}|^3 \\ & \leq T(|SUBG_{v_{k-2}}| - 4) + T(|SUBG_{v_{k-2}}| - 4) + C|SUBG_{v_{k-2}}|^3 \\ & \leq T(|SUBG_u| - k + 4 - 4) + T(|SUBG_u| - k + 4 - 4) + C|SUBG_{v_{k-2}}|^3 \\ & = T(|SUBG_u| - k) + T(|SUBG_u| - k) + C|SUBG_{v_{k-2}}|^3 \\ & < 2^{-0.3144k} (2 + 10^{-4}) C' 2^{0.3144(\eta-k)} (\eta + 1)^3. \end{aligned}$$

従って, (B-2) において,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < 2.0001 \cdot 2^{-0.3144k} C' 2^{0.3144(\eta-k)} (\eta + 1)^3. \end{aligned}$$

(B-0) $|SUBG_{u_{k-2}}| \geq |SUBG_{v_{k-2}}| - 2$ である場合:

(B-0) における置き換え操作,

$SUBG_{v_{k-2}} := \Gamma(u_{k-2}) \cap SUBG_{v_{k-2}}$ が 2 回実行された場合, (2-2) と同様に,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < T(|SUBG''_{v_{k-2}}|) + 2C(\eta + 1)^3 \\ & \leq T(|SUBG_{v_{k-2}}| - 2) + 2C(\eta + 1)^3 \\ & \leq T(|SUBG_u| - k + 4 - 2) + 2C(\eta + 1)^3 \\ & = T(|SUBG_u| - k + 2) + 2C(\eta + 1)^3 \\ & < 2^{-0.3144(k-2)} C' 2^{0.3144(\eta-k)} (\eta + 1)^3 + 2C(\eta + 1)^3 \\ & < 2^{-0.3144k} (2^{0.3144 \cdot 2} + 2 \cdot 10^{-4}) C' 2^{0.3144(\eta-k)} (\eta + 1)^3. \end{aligned}$$

ここで, $2^{0.3144 \cdot 2} + 2 \cdot 10^{-4} = 1.5464\dots$ より,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < 1.5465 \cdot 2^{-0.3144k} C' 2^{0.3144(\eta-k)} (\eta + 1)^3. \end{aligned}$$

次に, (B-0) の操作が 1 回実行された場合, 置き換え操作 1 回実行後の節点集合を $SUBG'_{v_{k-2}}$ とおくと,

$$|SUBG'_{v_{k-2}}| \leq |SUBG_{v_{k-2}}| - 1.$$

上記より, $SUBG'_{v_{k-2}}$ に対して,

(B-1) である場合,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < T(|SUBG'_{v_{k-2}}|) + C(\eta + 1)^3 \\ & \leq T(|SUBG_{v_{k-2}}| - 1) + C(\eta + 1)^3 \\ & < T(|SUBG_{v_{k-2}}| - 1 - 3) + T(|SUBG_{v_{k-2}}| - 1 - 6) + 2C(\eta + 1)^3 \\ & < 2^{-0.3144(k+1)}(2^{0.3144} + 2^{-0.3144 \cdot 2} + 2 \cdot 10^{-4})C'2^{0.3144(\eta-k)}(\eta + 1)^3. \end{aligned}$$

(B-2) である場合,

$A \neq \emptyset$ であるならば, 【2】 と同様に,

$$\begin{aligned} & T(|SUBG_{v_{k-2}}|) \\ & < T(|SUBG'_{v_{k-2}}|) + C(\eta + 1)^3 \\ & \leq T(|SUBG_{v_{k-2}}| - 1) + C(\eta + 1)^3 \\ & < T(|SUBG_{v_{k-2}}| - 1 - 4) + T(|SUBG_{v_{k-2}}| - 1 - 4) + 2C(\eta + 1)^3 \\ & < 2^{-0.3144(k+1)}(2 + 2 \cdot 10^{-4})C'2^{0.3144(\eta-k)}(\eta + 1)^3. \end{aligned}$$

$A = \emptyset$ となる場合は (B-0) 1 回目の実行において, u_{k-2} として v_{k-1} , 又は v_k が選択された場合である. いま, $(v_{k-1}, v_k) \notin E$ であるから,

$$SUBG'_{v_{k-2}} = \Gamma(u_{k-2}) \cap SUBG_{v_{k-2}}$$

は, $v_{k-1}, v_k \in SUBG_{v_{k-2}}$ をともに含まない. 従って,

$$|SUBG'_{v_{k-2}}| \leq |SUBG_{v_{k-2}}| - 2.$$

$A = \emptyset$ により, $SUBG'_{v_{k-2}}$ に対して通常の EXPAND が実行される. よって, 少なくとも,

$$\begin{aligned} & T(|SUBG'_{v_{k-2}}|) \\ & < T(|SUBG_{v_{k-2}}| - 2) + C(\eta + 1)^3 \\ & < 2^{-0.3144k}(2^{0.3144 \cdot 2} + 10^{-4})C'2^{0.3144(\eta-k)}(\eta + 1)^3. \end{aligned}$$

いずれの場合にも, (B-0) の置き換え操作が実行されなかった場合以下の計算量上界を得る. 従って, v_{k-2} からの探索において計算量上界が最大となるのは (B-2) の操作が実行さ

れた場合となる. そこで,

$$C'_{k-2} = 2.0001 \cdot 2^{-0.3144k} + 10^{-4}$$

とおく. 即ち,

$$T(|SUBG_{v_{k-2}}|) < C'_{k-2} C' 2^{0.3144(\eta-k)} (\eta+1)^3.$$

$1 \leq i \leq k-3$ において, 節点 v_i に対しては通常処理が実行される. このとき,

$$T(|SUBG_{v_1}|) < C' 2^{0.3144(\eta-k)} (\eta+1)^3,$$

$$T(|SUBG_{v_i}|)$$

$$\leq T(|SUBG_u| - i + 2)$$

$$< 2^{-0.3144(i-2)} C' 2^{0.3144(\eta-k)} (\eta+1)^3. \quad (2 \leq i \leq k-3)$$

以上から,

$$T(|SUBG|)$$

$$< T(|SUBG_u|) + \sum_{i=1}^{k-2} T(|SUBG_{v_i}|) + C|SUBG|^3$$

$$< C' 2^{0.3144(\eta-k)} (\eta+1)^3$$

$$+ C' 2^{0.3144(\eta-k)} (\eta+1)^3$$

$$+ \sum_{i=2}^{k-3} 2^{-0.3144(i-2)} C' 2^{0.3144(\eta-k)} (\eta+1)^3$$

$$+ C'_{k-2} C' 2^{0.3144(\eta-k)} (\eta+1)^3$$

$$+ C(\eta+1)^3.$$

いま $4 \leq k \leq 5$ である. よって,

$k = 4$ である場合,

$$T(|SUBG|)$$

$$< T(|SUBG_u|) + \sum_{i=1}^2 T(|SUBG_{v_i}|) + C|SUBG|^3$$

$$< (1 + 1 + C'_2 + 10^{-4}) 2^{-0.3144 \cdot 4} C' 2^{0.3144\eta} (\eta+1)^3$$

$$< 1.1865 C' 2^{0.3144\eta} (\eta+1)^3$$

$$< 1.2434 C' 2^{0.3144\eta} (\eta+1)^3$$

$$< 2^{0.3144} C' 2^{0.3144\eta} (\eta+1)^3$$

$$= C' 2^{0.3144(\eta+1)} (\eta+1)^3.$$

$k = 5$ である場合,

$$\begin{aligned}
& T(|SUBG|) \\
& < T(|SUBG_u|) + \sum_{i=1}^3 T(|SUBG_{v_i}|) + C|SUBG|^3 \\
& < (1 + 1 + 1 + C'_3 + 10^{-4})2^{-0.3144 \cdot 5} C' 2^{0.3144\eta} (\eta + 1)^3 \\
& < 1.2354 C' 2^{0.3144\eta} (\eta + 1)^3 \\
& < 2^{0.3144} C' 2^{0.3144\eta} (\eta + 1)^3 \\
& = C' 2^{0.3144(\eta+1)} (\eta + 1)^3.
\end{aligned}$$

以上より, いずれの場合においても,

$$T(|SUBG|) \leq C' 2^{0.3144(\eta+1)} (\eta + 1)^3$$

は成立する.

従って, 帰納法による帰結として, 任意の $|SUBG|$ に関して題意は成立する. \square

これより, 次の主要定理が成立する. なお, この主要定理における定数 3.177 は,

$$1/(0.3144 + 0.0003) = 1/0.3147 = 3.1776... > 3.177$$

による.

定理 7.3.1 節点数 $n \geq 1$ のグラフにおいて, グラフの最大次数 Δ が

$$\Delta \leq 3.177d \lg n \quad (d \geq 0: \text{定数})$$

を満たすならば, アルゴリズム MCP₃ は与えられたグラフの最大クリークサイズを $O(n^{1+d})$ -時間で抽出可能である.

(証明) 第3章定理 3.6.1 証明と同様. \square

NP 困難な最大クリーク抽出問題に対する上記結果から, NP 完全な最大クリーク問題に対する次の結果を得る.

一般グラフ $G = (V, E)$ に対して, 最大次数 Δ が

$$\Delta \leq 3.177d \lg n \quad (d \geq 0: \text{定数}).$$

を満たすとき, G に対する最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可解である.

第 8 章

多項式時間的可解性の拡張

8.1 概要

前章までにおいて、アルゴリズム MCP_3 を用いて、最大クリーク問題の多項式時間的
可解性に関する次の結果を得た。

「節点数 n の一般グラフにおいて、最大次数 $\Delta \leq 3.177d \lg n$ ($d \geq 0$: 定数), なる条件
が満たされているとき, このグラフの最大クリーク問題は $O(n^{1+d})$ なる多項式時間で可
解である。」

この結果は、グラフ全体の最大次数に制約条件を与えた結果である。グラフの次数に関
するこの制限対象範囲を緩和することは、多項式時間的可解性の観点から定数 c の値の大
小よりも相対的により重要となる。

本章においては、この観点から先ずグラフの次数に関する制約条件の全く新規な枠組
を考える。この枠組に対してアルゴリズム MCP_3 を適用し、実際の定量的結果を明確に
示す。

更に、アルゴリズム MCP_3 に対して若干の追加改良を行うことにより、上記結果の多項
式オーダーを明確に改善できることを示す。

最後に、この多項式オーダーの改善結果を基にして、アルゴリズムに更なる追加改良を
行う。このアルゴリズムを用いて、制約条件の枠組を更に拡張した次の最終的結果を示す。

「一般グラフ $G = (V, E)$ ($|V| = n$), 及びある定数 n_0 ($2 \leq n_0 \leq n/2$) に対して,
 $G(C)$ ($C \subseteq V$) を, $|C| = n_0$ なる任意の連結な誘導部分グラフとして,

$C = \{c_1, c_2, \dots, c_{n_0}\}$ とする. また,

$\min d(C) = \min\{\deg(c_1), \deg(c_2), \dots, \deg(c_{n_0})\}$ とする.

これらの定義のもと, 以下が成立する.

$$\min d(C) \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

であるならば, G に対する最大クリーク問題は,

$O(n^{\max\{2, 1+d\}})$ なる多項式時間的で可解である.]

8.2 拡張結果 (1)

前章のアルゴリズム MCP_3 の評価結果を用いて, 次の拡張結果を示す.

定理 8.2.1 一般グラフ $G = (V, E)$ に対して, $(v, w) \in E$ なる任意の 2 節点 $v, w \in V$ の次数 $\deg(v)$ 及び $\deg(w)$ が次の条件を満たすとする.

$$\min\{\deg(v), \deg(w)\} \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

このとき, アルゴリズム MCP_3 は与えられたグラフの最大クリークサイズを $O(n^{2+\max\{1, d\}})$ -時間で抽出可能である.

(証明) いま, $|V| = n$, $v \in V$ を G の最大次数節点,

$$SUBG_v = \Gamma(v) \cap V,$$

$$EXT_v = V - \{v\} - SUBG_v = \{v_1, v_2, \dots, v_k\} \quad (k = |EXT_v|),$$

$$SUBG_{v_i} = \Gamma(v_i) \cap (EXT_v[i, k] \cup SUBG_v) \quad (1 \leq i \leq k)$$

とそれぞれおく.

このとき, 式 (7.3.1) により,

$$T(|V|)$$

$$= T(n)$$

$$< T(|SUBG_v|) + \sum_{i=1}^k T(|SUBG_{v_i}|) + Cn|SUBG_v|^2$$

$$\leq nT(|SUBG_v|) + Cn|SUBG_v|^2.$$

ここで, $G(SUBG_v)$ 中の最大次数節点を w ,

$$SUBG_w = \Gamma(w) \cap SUBG_v$$

とおく. 条件から, $|SUBG_w| \leq 3.177d \lg n$ であるから, 定理 7.3.1 により, $T(|SUBG_v|)$

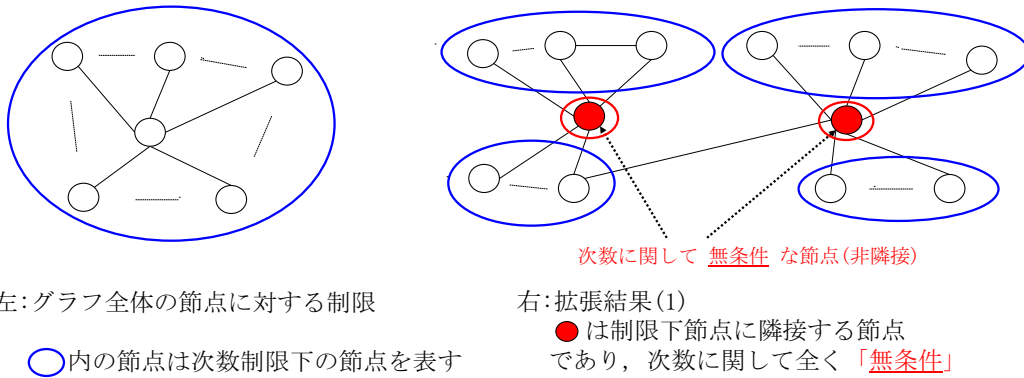


図 8.1 拡張結果 (1)

は $O(|SUBG_v|^{1+d})$ である. よって, 定数 D を用いて $T(|SUBG_v|) = D|SUBG_v|^{1+d}$ とおくと,

$$T(n) < nD|SUBG_v|^{1+d} + Cn|SUBG_v|^2 \tag{8.2.1}$$

$|SUBG_v| < n$ であるから,

$$\begin{aligned} T(n) &< nD|SUBG_v|^{1+d} + Cn^2|SUBG_v| \\ &< Dn^{2+d} + Cn^2|SUBG_v| \\ &< Dn^{2+d} + Cn^{2+1} \\ &\leq CDn^{2+d} + CDn^{2+1}. \end{aligned}$$

従って,

$$\begin{aligned} T(|V|) &= T(n) \\ &< CDn^{2+\max\{1,d\}}. \quad \square \end{aligned}$$

定理 8.2.1 の制約条件においては, グラフ $G = (V, E)$ の節点集合 V 中の任意の隣接 2 節点のうち, 次数の大きい方の節点の次数に関しては全くの「無条件」とする. この結果, グラフ中には (互いに隣接しない) 次数の非常に大きい節点の存在が許される (図 8.1).

従って、定理 8.2.1 の制約条件は、グラフ 全体 の最大次数に対する制約条件を明確に緩和したものであり、定数 3.177 部分の増大とは全く異なった観点からのものである。

更に、この結果は以下の様に容易に拡張可能である。

一般にグラフ G の節点集合 V において、 $3 \leq n_0$ なる定数 n_0 に対して、 $G(C)$ ($C \subseteq V$) を、 $|C| = n_0$ なる任意の連結な誘導部分グラフとして、

$$C = \{c_1, c_2, \dots, c_{n_0}\}$$

とする。また、

$$\min d(C) = \min\{\deg(c_1), \deg(c_2), \dots, \deg(c_{n_0})\}$$

として、

$$\min d(C) \leq 3.177d \lg n$$

とする。このとき、定理 8.2.1 の証明と同様に、式 (8.2.1) の関係を 2 回、3 回、... と繰り返し適用すると、多項式のオーダーを $O(n^{3+\max\{1,d\}})$, $O(n^{4+\max\{1,d\}})$, ... と増加させることにより、それぞれ多項式時間的可解性を示すことが可能である。

例えば、定理 8.2.1 における次数の制限を、サイズ 3 の連結な誘導部分グラフ中における、最小次数節点に対する制限とする。このとき、グラフ中には次数が非常に大きい節点で、かつ隣接節点集合中にも次数の非常に大きい節点が存在するような場合が許される。従って、 $n_0 = 3, 4, \dots$ との拡張は、定理 8.2.1 における次数の制限を、更に明確に拡張したものとなる。

定理 8.2.1 の結果は、

$$O(n^{2+\max\{1,d\}}),$$

即ち、最小で $O(n^3)$ と、グラフ全体の最大次数に制限を与えた場合より 1 増加している。 $n_0 = 3, 4, \dots$ との拡張においても、定理 8.2.1 と同様に式 (8.2.1) の関係を用いる手法を採ると、定理 8.2.1 の多項式オーダーは 1 ずつ増加していく。

この結果に対しては、アルゴリズム MCP_3 に次のような若干の追加変更を行うことにより、多項式オーダーの増加を抑えることが可能である。

8.3 拡張結果 (2)

アルゴリズム MCP_3 に簡単な追加改良を行うことにより, 上記結果の多項式オーダーを改良できることを示す, 次の結果を与える.

[一般グラフ $G = (V, E)$ に対して, $(v, w) \in E$ なる任意の 2 節点 $v, w \in V$ の次数 $deg(v)$ 及び $deg(w)$ が次の条件を満たすとする.

$$\min\{deg(v), deg(w)\} \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

このとき, G に対する最大クリーク問題は,

$$O(n^{\max\{2, 1+d\}}) \text{ なる多項式時間的で可解である.]$$

この結果では, 定数 $0 \leq d \leq 1$ における多項式オーダーは $O(n^2)$, $d > 1$ においては $O(n^{1+d})$ となる. 従って, この結果における多項式オーダーは, グラフ 全体 の最大次数に制限を設けた場合と基本的に同じである.

アルゴリズム 6 アルゴリズム MCP_{ex}

```

0000: procedure  $MCP_{ex}(G)$ 
0100: global  $Q\_size, Qmax\_size$ ;
0200: begin
0300:  $Q\_size := 0, Qmax\_size := 0$ ;

0301: sort vertices in  $V$  in ascending order with respect to their degrees in  $G$ ;
0302: let  $V := \{v_1, v_2, \dots, v_{|V|}\}$ ; /* an ordered set of vertices */
0303: for  $i = 1$  to  $|V|$  do
0304:    $Q\_size := Q\_size + |\{v_i\}|$ ;
0305:   EXPAND( $\Gamma(v_i) \cap (V - \{v_1, v_2, \dots, v_{i-1}\})$ )
0306:    $Q\_size := Q\_size - |\{v_i\}|$ 
0307: od

0400: return  $Qmax\_size$ 
0500: end of  $MCP_{ex}$ 

```

定理 8.2.1 の多項式オーダーを改善可能である, アルゴリズム MCP_{ex} をアルゴリズム 6 に示す.

MCP_3 からの変更点は以下である.

与えられた入力 G に対して, G の節点集合 V 中の各節点を次数の 昇順 にソートする (0301 行).

ソートされた V 中の各節点からの探索を, 左から順に行う. このとき, 探索済の節点は

V から削除する. (0305 行)

MCP_{ex} においては, V に対する節点の探索順は, 前章までの $SUBG \subsetneq V$ に対する部分森の同一化による探索順とは異なる. V の真部分集合である $SUBG$ に対しては, MCP_{ex} は MCP_3 と全く同一に動作する. よって, $|SUBG| = \eta$ なる $SUBG$ に対して, MCP_{ex} の計算量上界は $O(2^{0.3147\eta})$ である.

アルゴリズム MCP_{ex} の計算量上界を用いて, 最大クリーク抽出問題の多項式時間的可解性に関する拡張結果である, 次の定理を与える.

定理 8.3.1 一般グラフ $G = (V, E)$ に対して, $(v, w) \in E$ なる任意の 2 節点 $v, w \in V$ の次数 $deg(v)$ 及び $deg(w)$ が次の条件を満たすとする.

$$\min\{deg(v), deg(w)\} \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

このとき, アルゴリズム MCP_{ex} は与えられたグラフの最大クリークサイズを $O(n^{\max\{2, 1+d\}})$ -時間で抽出可能である.

(証明) いま,

$$\begin{aligned} L &:= \{v \in V \mid |\Gamma(v) \cap V| \leq 3.177d \lg n\} \\ &= \{p_1, p_2, \dots, p_{|L|}\} \quad (\text{次数 昇順}) \end{aligned}$$

とする. また,

$$V - L = \{v_1, v_2, \dots, v_{|V-L|}\}$$

とする. V 中の節点は次数の小さい節点から順に探索され, 探索済の各節点, および節点に接続する枝は V から削除される. この探索順に基づき, $p_i \in L$ ($1 \leq i \leq |L|$) に対して,

$$SUBG_{p_i} := \Gamma(p_i) \cap (V - L[1, i - 1])$$

とする. また,

$$\begin{aligned} v_j \in V - L \quad (1 \leq j \leq |V - L|) \text{ に対して,} \\ SUBG_{v_j} := \Gamma(v_j) \cap (V - L - ((V - L)[1, j - 1])) \end{aligned}$$

とする. このとき V に対する計算量上界 $T(|V|) = T(n)$ に対して次が成立する.

$$\begin{aligned} T(|V|) &= T(n) \\ &\leq \sum_{i=1}^{|L|} T(|SUBG_{p_i}|) + \sum_{j=1}^{|V-L|} T(|SUBG_{v_j}|) + Cn^2. \end{aligned}$$

式末尾の Cn^2 は, V 中の各節点を次数の昇順にソートし, 各 $SUBG_{p_i}$, 及び $SUBG_{v_j}$ を構成するために要する時間計算量の上界を表す.

上記式に対して, 先ず, L 中の各節点からの探索についての計算量上界を与える.

節点集合 L の定義から, $|SUBG_{p_i}| \leq 3.177d \lg n$ である. 従って, 適当な定数 E を用いて,

$$\begin{aligned} T(|SUBG_{p_i}|) &\leq E2^{0.3147|SUBG_{p_i}|} \\ &\leq E2^{0.3147 \cdot 3.177d \lg n} \\ &< En^d \end{aligned}$$

が成立する. よって,

$$\begin{aligned} \sum_{i=1}^{|L|} T(|SUBG_{p_i}|) \\ &< |L|En^d \\ &< En^{1+d}. \end{aligned}$$

次に, $V - L$ 中の各節点からの探索についての計算量上界を与える. いま G 中の節点のうち, 次数が $3.177d \lg n$ より大きい任意の節点を v とすると,

$$v \in V - L$$

である. また, $w \in \Gamma(v)$ に対して,

$$|\Gamma(w)| \leq 3.177d \lg n.$$

即ち,

$$w \in L$$

である. L 中の各節点からの探索終了後, V から L 中の各節点は全て削除される. 従って,

$1 \leq j \leq |V - L|$ に対して,

$$SUBG_{v_j} = \emptyset.$$

以上により,

$$T(|SUBG_{v_j}|) = T(|\emptyset|) = 0.$$

よって,

$$\begin{aligned} \sum_{j=1}^{|V-L|} T(|SUBG_{v_j}|) \\ = 0. \end{aligned}$$

以上から,

$$T(|V|)$$

$$\begin{aligned}
&= T(n) \\
&\leq \sum_{i=1}^{|L|} T(|SUBG_{p_i}|) + \sum_{j=1}^{|V-L|} T(|SUBG_{v_j}|) + Cn^2. \\
&< En^{1+d} + Cn^2 \\
&= O(n^{\max\{2,1+d\}}). \quad \square
\end{aligned}$$

この結果を基に、アルゴリズム MCP_{ex} に対して更に追加拡張を行うことにより、次の更なる多項式時間的可解性の拡張結果を得る。

8.4 拡張結果 (3)

アルゴリズム MCP_{ex} の計算量上界を用いて、最大クリーク抽出問題の多項式時間的可解性に関する拡張結果である、次の定理を与える。

定理 8.4.1 $|V| = n$ である一般グラフ $G = (V, E)$, 及びある定数 n_0 ($2 \leq n_0 \leq n/2$) に対して, $G(C)$ ($C \subseteq V$) をサイズ n_0 である任意の連結な誘導部分グラフ, C を,

$$C = \{c_1, c_2, \dots, c_{n_0}\}$$

とする。また,

$$\min d(C) = \min\{deg(c_1), deg(c_2), \dots, deg(c_{n_0})\}$$

とする。

これらの定義のもと, 以下が成立する。

$$\min d(C) \leq 3.177d \lg n \quad (d \geq 0 : \text{定数}).$$

であるならば, G に対する最大クリーク問題は,

$$O(n^{\max\{2,1+d\}}) \text{ なる多項式時間的で可解である.}$$

(証明) 前節, 定理 8.3.1 証明における定義と同様に,

$$L := \{v \in V \mid |F(v) \cap V| \leq 3.177d \lg n\},$$

$$= \{p_1, p_2, \dots, p_{|L|}\} \quad (\text{次数 昇順}),$$

$$V - L = \{v_1, v_2, \dots, v_{|V-L|}\},$$

$$SUBG_{p_i} := \Gamma(p_i) \cap (V - L[1, i - 1]), \quad (1 \leq i \leq |L|)$$

及び,

$$SUBG_{v_j} := \Gamma(v_j) \cap (V - L - ((V - L)[1, j - 1]))$$

とする. このとき,

$$\begin{aligned} T(|V|) &= T(n) \\ &\leq \sum_{i=1}^{|L|} T(|SUBG_{p_i}|) + \sum_{j=1}^{|V-L|} T(|SUBG_{v_j}|) + Cn^2. \end{aligned}$$

L 中の各節点からの探索については, 定理 8.3.1 証明と同様である.

$$\begin{aligned} &\sum_{i=1}^{|L|} T(|SUBG_{p_i}|) \\ &< |L|En^d \\ &< En^{1+d}. \end{aligned}$$

次に, $V - L$ 中の各節点からの探索についての計算量上界を与える.

前提から節点 $v_i \in V - L$ ($1 \leq i \leq |V - L|$) は, $V - L$ 中の節点のうち高々 $n_0 - 2$ 個と隣接する. これは, 以下による.

いま, 節点 v_i が $V - L$ 中の節点のうち $n_0 - 1$ 個以上と隣接するとして,

$$\Gamma(v_i) \cap (V - L) = \{w_1, w_2, \dots, w_m\} \quad (m \geq n_0 - 1)$$

とする. このとき $G(\{v_i\} \cup \Gamma(v_i))$ は, サイズ n_0 以上の連結な誘導部分グラフであり, かつ $\{v_i\} \cup \Gamma(v_i)$ 中の任意の節点の次数は $3.177d \lg n$ より大きい. これは前提に反する.

よって, 節点 L 中の各節点からの探索が終了後,

$$\max\{\deg(v_1), \deg(v_2), \dots, \deg(v_{|V-L|})\} \leq n_0 - 2.$$

ここで, $|SUBG| = S$ ($S \geq 1$: 定数) であるとき, MCP_{ex} において,

$$T(S) \leq C'2^{0.3144S}(S+1)^3 = O(1).$$

よって, 適切な定数 F を用いて,

$$\begin{aligned} &\sum_{j=1}^{|V-L|} T(|SUBG_{v_j}|) \\ &< |V - L|F \\ &< n \cdot F \\ &= Fn. \end{aligned}$$

以上から,

$$T(|V|)$$

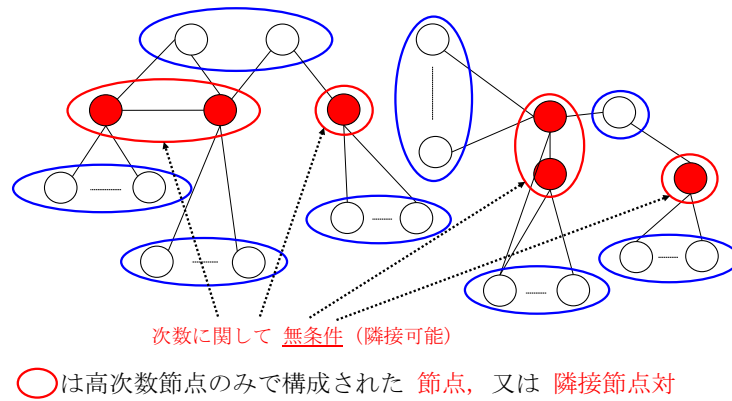


図 8.2 拡張結果 (3) ($n_0 = 3$ である場合)

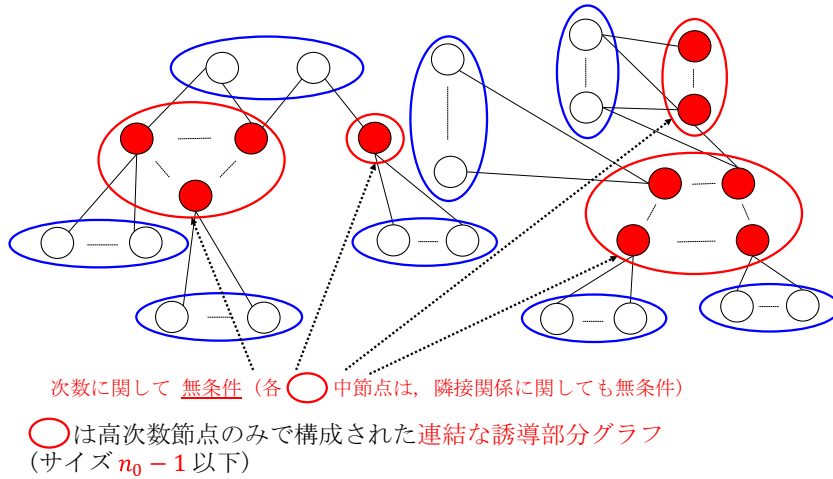


図 8.3 拡張結果 (3) ($n_0 \geq 2$ である場合)

$$\begin{aligned}
 &= T(n) \\
 &\leq \sum_{i=1}^{|L|} T(|SUBG_{p_i}|) + \sum_{j=1}^{|V-L|} T(|SUBG_{v_j}|) + Cn^2. \\
 &< En^{1+d} + Fn + Cn^2 \\
 &= O(n^{\max\{2, 1+d\}}). \quad \square
 \end{aligned}$$

この結果は, $n_0 = 2$ においては定理 8.3.1 と同一の結果である.

定理 8.3.1 の制限においては, 次数が $3.177d \lg n$ より大きい節点同士は必ず非隣接となる. これに対して, 定理 8.4.1 において, 例えば $n_0 = 3$ とする. このとき, サイズ 3 の連結な誘導部分グラフ中の節点のうち, 次数制限範囲内となるものはただ 1 個である. 従って

グラフ中には,

$$(v, w) \in E \text{ かつ } \min\{\deg(v), \deg w\} > 3.177d \lg n,$$

即ち次数の高い節点のみで構成された隣接節点对が存在し得る. (図 8. 2).

同様に考えれば, サイズ n_0 の連結な誘導部分グラフにおいて, 最小次数節点に制限を設けた場合, グラフ中にはサイズ $n_0 - 1$ 以下の高次数節点のみで構成された誘導部分グラフが存在し得る. 従って, n_0 の値が大きくなるにつれ, グラフの次数に関する条件は明確に緩和される (図 8. 3).

以上, 本章の結果により, 最大クリーク問題の多項式時間的可解性に関する新しい枠組を与えた.

第 9 章

まとめと今後の展望

本論文では、まず第 3 章において極大クリーク全列挙アルゴリズム CLIQUES を基礎として、非常に単純な最大クリーク抽出の基本アルゴリズム MCP_0 を提唱し、関連論文 (1) における次の結果を得た。

「節点数 n , 最大次数 $\Delta \geq 0$ なるグラフに対して, Δ が $\Delta \leq 2.493d \lg n$ ($d \geq 0$:定数) を満たすならば, 最大クリーク問題は $O(n^{1+d})$ -時間で可解である。」

このような制約条件を定量的に明確に示したのは、関連論文 (1) が初めてである。

第 4-7 章においては、アルゴリズム MCP_0 に独自手法による探索領域削減操作を順次追加、拡張した。これによりアルゴリズムの理論計算量上界を明確に改善し、第 7 章において次の結果を得た。

「節点数 n , 最大次数 $\Delta \geq 0$ なるグラフに対して, Δ が $\Delta \leq 3.177d \lg n$ ($d \geq 0$:定数) を満たすならば, 最大クリーク問題は $O(n^{1+d})$ -時間で可解である。」

更に、第 8 章においては第 7 章の結果を用いて、グラフの最大次数に対する制限を緩和した次の拡張結果を得た。

「サイズ $n_0 \geq 2$ なる任意の連結な誘導部分グラフ $G(C)$ ($C \subseteq V$) に対して, C 中の最小次数節点 v が,

$$\deg(v) \leq 3.177d \lg n \quad (d \geq 0:\text{定数})$$

を満たすとき, 最大クリーク問題は $O(n^{\max\{2, 1+d\}})$ の多項式時間で可解である。」

ここで、定数 $d \geq 0$ は、標準的には $d = 1$ を想定している。 $d = 1$ とした場合、この結果における多項式は $O(n^2)$ である。以下では、 $d = 1$ としてこの結果を扱う。

この制約条件のもとでは, G 中のサイズ $n_0 \geq 2$ なる任意の連結成分のうち, 最小次数節点を除く全ての節点の次数に関しては全く無条件でよい.

この結果に関連して, 本研究の最終段階において, 文献 [22] の結果の存在が知らされた.

同文献は, d -degenerate グラフなる概念を定義し, d -degenerate グラフにおける最大クリーク問題は, $O(nm + n2^{d/4})$ -時間で解決可能であることを示している. $O()$ 中の式における $2^{d/4}$ は, Robson の $O(2^{n/4})$ -時間アルゴリズム (文献 [94], Technical Report) をサブルーチンとして用いたことによる. この結果を利用すると, $d \leq 4\lg n$ である条件下において, $O(nm + n^2) = O(nm)$ なる多項式時間で解けることとなる.

本論文第8章において示した, 上記の制約条件下において, 与えられたグラフは $d_0 = 3.177\lg n$ に対して d_0 -degenerate グラフの範疇に入る. 従って本論文の上記結果は, 本論文中の d_0 -degenerate グラフが $O(n^2)$ -時間で解決可能となることを示したものである.

一般に, 連結なグラフにおいては,

$$n - 1 \leq m$$

が成立する. 枝数 m が大きくなる (グラフが密になる) に従い, nm は n^2 よりも大きくなる.

なお, このような拡張の出発点となる, 第8章, 8.2節の $n_0 = 2$ における定理 8.2.1 の結果は, 2011年に筆者らによって初めて示されている (文献 [79]).

以上, 本論文においては, 最大クリーク問題の多項式時間的可解性に関して, グラフ全体の最大次数に対する制約条件を緩和した新たな多項式時間的可解性の枠組を提唱した.

本論文の結果に対する今後の課題は以下の通りである.

本論文において示した最大クリーク問題の多項式時間的可解性は, 基本的にグラフの辺密度が低い場合を想定している. しかしながら, 本論文における最終的なアルゴリズム MCP_{ex} は, 次のような場合分けに基づく探索を行う.

- $k = |EXT_u|$ が十分大きい, あるいは小さい場合には, アルゴリズムは単純な操作のみを実行する.

- k がある定数範囲内である場合には, より強力な探索領域削減操作を実行する.

ここで、 k が定数範囲内である場合とは、入力 $SUBG$ 中の最大次数節点に隣接しない節点の個数が高々定数個となる場合、即ち u の次数が相対的に高い場合である。従って、 MCP_{ex} はグラフの辺密度が高い場合により有効であると考えられる。この点を考慮して、グラフの辺密度が高くなるような、グラフに対するより自然な多項式時間的可解性の制約条件を考えることも、今後の課題の一つである。

次に、定数値 c の値の改善に関して、本論文における成果をふまえた今後の課題は、以下の各点である。

本論文において、 MCP_0 に順次追加、拡張した探索領域削減操作は本論文独自で新規のものである。これらの各操作の基礎は全て補題 2.2.3, 補題 2.2.4, 及び 補題 2.2.5 となっている。従って本論文第 4-7 章におけるアルゴリズムの改良は、基本的に同じ操作の適用範囲を拡張したことにより達成されている。具体的には以下の通りである。

第 4 章から第 5 章においては、補題 2.2.4 に基づく探索領域削減手法に、補題 2.2.5 による拡張を行い、これにより計算量上界の改善を実現した。

第 5 章から第 6 章においては、先ずアルゴリズムにより分割された各部分問題のサイズが順次減少していくことを示した (第 6 章, 補題 6.2.1)。但し、補題 6.2.1 は条件 (即ち、 $G(EXT_u)$ がクリークである場合) 付きの結果である。この結果により、強力な探索領域削減操作の適用を、部分問題数が定数範囲内 (かつ、 $G(EXT_u)$ がクリーク) である場合のみに限定した。この限定の上で、第 5 章までには適用していなかった対象 (EXT_u 中の節点 v_i の、隣接節点集合) に対して新たに補題 2.2.4 を適用し、これにより計算量上界の改善を実現した。

第 6 章から第 7 章においては、第 6 章において示した部分問題のサイズ減少をより一般化した。これにより、一般に強力な探索領域削減操作の適用を、部分問題数が定数範囲内である場合のみに限定するという、アルゴリズムの基本的な構造を確立した。この限定の上で、補題 2.2.4 の適用対象を拡張 (即ち、 $G(EXT_u)$ が非クリークである場合にも適用) した。更に、補題 2.2.3, 及び、補題 2.2.4 を上記とは別の対象 (即ち、節点 v_i, v_{i+1} それぞれの隣接節点集合) に適用した。これらによって、計算量上界の改善を達成している。

このような適用範囲のより一層の拡張により、 MCP_{ex} の計算量上界を更に改善可能である展望を持っている。この方法により定数値 c を改善することは、今後の主要課題の 1

つである。

最大クリーク問題の理論計算量上界に関連して、文献 [37] において、Fomin らは Measure & Conquer 法という新たな解析手法を提唱した。同文献においては、Measure & Conquer 法以前の解析手法 (基本的に本論文において用いた解析手法と同様の手法。以下、従来手法と呼ぶ。) による計算量上界が $O^*(2^{0.406n})$ -時間である単純な最大独立節点集合抽出アルゴリズムに Measure & Conquer 法による解析を適用し、これにより全く同一のアルゴリズムに対して $O^*(2^{0.288n})$ -時間なる計算量上界を示した [37]。この結果は、ある解析手法によるアルゴリズムの計算量上界が、異なる解析手法により更に改善可能であることを示している。従って、本論文の定数値改善にあたっては、アルゴリズム自体の改善に加えて、異なる解析手法の導入による改善の可能性も検討する必要がある。

また、アルゴリズムの理論計算量評価においては、解析過程において計算機を利用する手法も採られてきた。これは、解析上生じる場合分けの数が有限個となることを示した上で、これら有限個の場合分け各々における計算量上界を計算機を用いて求めるものである [32]。本論文における各アルゴリズムの計算量評価過程においては、このような方法は採らなかった。しかしながら、前記の通り本論文の最終的な解析においては、部分問題数が定数範囲内である場合に計算量上界が最大となるという結果を得ている。従って、この場合において計算機を利用したより詳細な解析を行うことにより、本論文における計算量評価結果を改善可能であるかどうかとも検討が必要である。

本論文第3章における基本アルゴリズム MCP_0 は、文献 [98] において提唱された極大クリーク全列挙アルゴリズム CLIQUES を基にしている。

文献 [95] において、Shindo-Tomita はやはり文献 [98] のアルゴリズム CLIQUES を基に、単純な最大クリーク抽出アルゴリズム MAXCLIQUE を提唱し、その時間計算量上界が $O(2^{n/2.863}) = O(2^{0.349n})$ -時間であることを示した。このアルゴリズムは非常に単純な記述により表現可能であったが、理論的計算量上界に関しては Tarjan-Trojanowski の結果 $O(2^{n/3})$ -時間と比較してより小さいものではなかった。しかしながら、この論文中では Tarjan-Trojanowski のアルゴリズム、及び MAXCLIQUE のそれぞれを実働化した上、現実的に検証可能な範囲において計算機実験評価を行っている。実験の結果、MAXCLIQUE は計算機実験を行った範囲において、Tarjan-Trojanowski のアルゴリズムより大幅に高

速に動作することが示されている。

これらの結果に対して、同文献 [95] はこれまで国際的に幾つかの引用を受けている。このように、アルゴリズム評価においては、最大時間計算量評価の値が唯一絶対的な価値基準では必ずしもない。本論文のアルゴリズムは、MAXCLIQUE と同じく CLIQUES を基にして構成されている。更に MAXCLIQUE とは異なる新たな手法によるアルゴリズムの改善を行うことにより、第7章において MAXCLIQUE の理論計算量上界よりも良い結果を得た。一般にアルゴリズムの実働評価においては、辺密度の低いグラフは高速で解く事が可能であるが、この場合には MCP_0 のようなより単純なアルゴリズムが有効である。グラフの辺密度が高くなるにつれ、計算時間は大きく増加していくが、先述したように本論文第6章、7章における各アルゴリズムにおいて導入した手法は、辺密度が高いグラフに対して有効であることが期待できる。よって、本論文のアルゴリズムを実働化し、実行可能な範囲内においてその性能の評価を行うことも、今後の課題である。

謝辞

本研究を進めるにあたり、長年にわたり、多大なるご指導・ご鞭撻を賜りました、電気通信大学 富田悦次 名誉教授に心から感謝の意を表します。

ご指導，ご審査を頂きました電気通信大学大学院 情報理工学研究科 情報系 西野哲朗 教授，伊藤大雄 教授，高橋治久 教授，庄野逸 教授，垂井淳 準教授，並びに日頃よりご指導を頂いています，若月光夫 助教に深く感謝致します。

特に，伊藤教授には本研究の端緒となる貴重なコメントをいただき（当時，京都大学），垂井準教授にはご懇切な多くのご教示をいただき有難うございました。

また，日頃より多大なるご支援を頂いています，早稲田大学 教育学部数学科 守屋悦朗 教授に厚く御礼申し上げます。

参考文献

- [1] T. Akutsu, M. Hayashida, E. Tomita, J. Suzuki, and K. Horimoto, “Protein threading with profiles and constraints,” Proc. BIBI 2004, pp.537–544, 2004.
- [2] V.E. Alekseev, “Polynomial algorithm for finding the largest independent sets in graphs without forks,” Discrete Applied Mathematics, 135(1–3), pp.3–16, 2004.
- [3] N. Alon, R. Yuster, and U. Zwick, “Finding and counting given length cycles,” Algorithmica, 17(3), pp.209–223, 1977.
- [4] D. K. C. Bahadur, T. Akutsu, E. Tomita, T. Seki and A. Fujiyama, “Point matching under non-uniform distortions and protein side chain packing based on efficient maximum clique algorithms,” Genome Informatics, 13, pp.143–152, 2002.
- [5] D. K. C. Bahadur, E. Tomita, J. Suzuki, and T. Akutsu, “Protein side-chain packing problem: A maximum edge-weight clique algorithmic approach,” Journal of Bioinformatics and Computational Biology, 3, pp.103–126, 2005.
- [6] D. K. C. Bahadur, E. Tomita, J. Suzuki, K. Horimoto and T. Akutsu, “Protein threading with profiles and distance constraints using clique based algorithms,” Journal of Bioinformatics and Computational Biology, 4(1), pp.19–42, 2006.
- [7] B. Balasundaram, S. Butenko and I.V. Hicks, “Clique relaxations in social network analysis: The maximum k-plex problem,” Operations Research, 59(1), pp.133–142, 2011.
- [8] R. Beigel, “Finding maximum independent sets in sparse and general graphs,” Proc. SODA 1999, pp.856–857, 1999.
- [9] B. K. Bhattacharya, “An $O(m + n \log n)$ algorithm for the maximum-clique

- problem in circular-arc graphs,” *Journal of Algorithms*, 25(2), pp.336–358, 1997.
- [10] V. Boginski, S. Butenko and P.M. Pardalos, “Mining market data: a network approach,” *Computers & Operations Research*, 33(11), pp.3171–3184, 2006.
- [11] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, “The maximum clique problem,” *Handbook of Combinatorial Optimization*, Supplement vol.A, pp.1–74, 1999.
- [12] N. Bourgeois, B. Escoffer, V. T. Paschos, and J. M. M. van Rooij, “An $O^*(1.0977^n)$ exact algorithm for max independent set in sparse graphs,” *IWPEC 2008*, *Lecture Notes in Computer Science*, 5018, pp.55–65, 2008.
- [13] N. Bourgeois, B. Escoffer, V. T. Paschos, and J. M. M van Rooij, “Fast algorithms for max independent set in graphs of small average degree,” *arXiv:0901.1563 [cs.DM]*, 2009.
- [14] N. Bourgeois, B. Escoffer, V. T. Paschos, and J. M. M. van Rooij, “Fast algorithms for max independent set in graphs of small average degree,” *CoRR abs/0901.1563*, 2009.
- [15] N. Bourgeois, B. Escoffer, V. T. Paschos, and J. M. M. van Rooij, “Maximum independent set in graphs of average degree at most three in $O(1.08537^n)$,” *TAMC 2010*, *Lecture Notes in Computer Science*, 6108, pp.373–384, 2010.
- [16] N. Bourgeois, B. Escoffer, V. T. Paschos, and J. M. M. van Rooij, “A bottom-up method and fast algorithms for max independent set,” *SWAT 2010*, *Lecture Notes in Computer Science*, 6139, pp. 62–73, 2010.
- [17] N. Bourgeois, B. Escoffer, V. T. Paschos, and J. M. M. van Rooij, “Fast algorithms for MAX INDEPENDENT SET,” *Algorithmica*, 62(1), pp.382–415, 2012.
- [18] S. Bradde, A. Braunstein, H. Mahmoudi, F. Tria, M. Weigt and R. Zecchina, “Aligning graphs and finding substructures by a cavity approach,” *Europhysics Letters*, 89(3), 37009, 2010.
- [19] C. Bron and J. Kerbosch, “Algorithm 457, Finding all cliques of an undirected graph,” *Communications of the ACM*, 16, pp.575–577, 1973.
- [20] L. Brotcorne, G. Laporte and F. Semet, “Fast heuristic for large scale covering

- location problems,” *Computers & Operations Research*, 29(6), pp.651–665, 2002.
- [21] J. B. Brown, D. K. C. Bahadur, E. Tomita and T. Akutsu, “Multiple methods for protein side chain packing using maximum weight cliques,” *Genome Informatics*, 17(1), pp.3–12, 2006.
- [22] A. Buchanan, J. L. Walteros, S. Butenko, and P. M. Pardalos, “Solving maximum clique in sparse graphs: an $O(nm + n2^{d/4})$ algorithm for d-degenerate graphs,” *Optimization Letters*, 8(5), pp. 1611-1617, 2014.
- [23] S. Butenko and W. E. Wilhelm, “Clique-detection models in computational biochemistry and genomics - Invited Review - ,” *European Journal of Operational Research*, 173(1), pp.1–17, 2006.
- [24] M.W. Carter, G. Laporte and S.Y. Lee, “Examination timetabling: algorithmic strategies and applications,” *Journal of Operational Research Society*, 47(3), pp.373–383, 1996.
- [25] M.W. Carter and D.G. Johnson, “Extended clique initialisation in examination timetabling,” *Journal of Operational Research Society*, 52(5), pp.538–544, 2001.
- [26] J.Chen, I. A. Kanji and W. Jia, “Vertex cover: Further observation and further improvements,” *Journal of Algorithms*, 41(2), pp.280–301, 2001.
- [27] J.Chen, I. A. Kanji and G. Jia, “Labeled search trees and amortized analysis: improved upper bounds for np-hard problems,” *ISAAC 2003, Lecture Notes in Computer Science*, 2906, pp.148-157, 2003.
- [28] F. Chen, H. Zhai and Y. Fang, “Available bandwidth in multirate and multihop wireless ad hoc networks,” *IEEE Journal on Selected Areas in Communications*, 28(3), pp.299–307, 2010.
- [29] N. Chiba and T. Nishizeki, “Arboricity and subgraph listing algorithms,” *SIAM Journal of Computing*, 14(1), pp.210–223, 1985.
- [30] N. M. Dognin, R. Andonov and N. Yanev, “Maximum cliques in protein structure comparison,” *SEA 2010, Lecture Notes in Computer Science*, 6049, pp.106–117, 2010.
- [31] U. Dorndorf, F. Jaehn and E. Pesch, “Modelling robust flight gate scheduling as

- a clique partitioning problem,” *Transportation Science*, 42(3), pp.292–301, 2008.
- [32] D. Eppstein, “Quasiconvex analysis of multivariate recurrence equations for backtracking algorithms,” *ACM Transactions on Algorithms*, 2(4), pp.257–269, 2006.
- [33] D. Eppstein, M. Löffler and D. Strash, “Listing all maximal cliques in sparse graphs in near-optimal time,” *ISAAC 2010, Lecture Notes in Computer Science*, 6506, pp.403–414, 2010.
- [34] D. Eppstein and D. Strash, “Listing all maximal cliques in large sparse real-world graphs,” *SEA 2011, Lecture Notes in Computer Science*, 6630, pp.364–375, 2011.
- [35] D. Eppstein, M. Löffler and D. Strash, “Listing all maximal cliques in large sparse real-world graphs in near-optimal time,” *Journal of Experimental Algorithmics*, 18(3-1), 2013.
- [36] S. Even, A. Pnueli, and A. Lempel, “Permutation graphs and transitive graphs,” *Journal of the ACM*, 19(3), pp.400–410, 1972.
- [37] F. V. Fomin, F. Grandoni, and D. Kratsch, “Measure and conquer: A simple $O(2^{0.288n})$ independent set algorithm,” *Proc. SODA 2006*, pp.18–25, 2006.
- [38] F. V. Fomin, F. Grandoni, and D. Kratsch, “A measure & conquer approach for the analysis of exact algorithms,” *Journal of the ACM*, 56(5), pp.25:1–25:32, 2009.
- [39] F. V. Fomin and D. Kratsch, “Exact Exponential Algorithms,” Springer, 2010.
- [40] D. Fukagawa, T. Tamura, A. Takasu, E. Tomita, and T. Akutsu, “A clique-based method for the edit distance between unordered trees and its application to analysis of glycan structures,” *BMC bioinformatics*, 12(1), 1, 2011.
- [41] M. Fürer, “A faster algorithm for finding maximum independent sets in sparse graphs,” *LATIN 2006, Lecture Notes in Computer Science*, 3887, pp. 491–501, 2006.
- [42] M. R. Garey and D. S. Johnson, “Computers and intractability: A guide to the theory of NP-completeness,” W. H. Freeman and Company, New York, NY, 1979.
- [43] F. Gavril, “Algorithms for minimum coloring, maximum clique, minimum cover-

- ing by cliques, and maximum independent set of a chordal graph,” *SIAM Journal on Computing*, 1(2), pp.180–187, 1972.
- [44] F. Gavril, “Algorithms for a maximum clique and a maximum independent set of a circle graph,” *Networks*, 3(3), pp.261–273, 1973.
- [45] F. Gavril, “Algorithms on circular-arc graphs,” *Networks*, 4(4), pp.357–369, 1974.
- [46] F. Gavril, “Some NP-complete problems on graphs,” *Proc. CISS 1977*, pp.91–95, 1977.
- [47] K. Han, G. Cui and Y. Chen, “Identifying functional groups by finding cliques and near-cliques in protein interaction networks,” *Proc. FBIT 2007*, pp.159–164, 2007.
- [48] M. Hattori, Y. Okuno, S. Goto and M. Kanehisa, “Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways,” *Journal American Chemical Society*, 125(39), pp.11853–11865, 2003.
- [49] 堀田一宏, 富田悦次, 高橋治久, “最大クリーク抽出に基づく向きの変化に依存しない人物の顔検出,” *Transactions on IPSJ*, 44, SIG14(TOM9), pp.57–70, 2003.
- [50] D. Issac, R. Jaiswal, “An $O^*(1.0821^n)$ -time algorithm for computing maximum independent set in graphs with bounded degree 3,” *CoRR abs/1308.1351*, 2013.
- [51] 伊藤大雄, “クリーク列挙 - アルゴリズムと下限 - ,” *電子情報通信学会誌*, 95, pp.484–489 (小特集 「広がる列挙の技術 - 列挙による問題解決アプローチ - 」), 2012.
- [52] K. Iwaide and H. Nagamochi, “An improved algorithm for parameterized edge dominating set problem,” *WALCOM 2015, Lecture Notes in Computer Science*, 8973, pp. 234–245, 2015.
- [53] K. Iwaide and H. Nagamochi, “An improved algorithm for parameterized edge dominating set problem.” *Journal of Graph Algorithms and Applications*, 20(1), pp. 23-58, 2016.
- [54] T. Jian, “An $O(2^{0.304n})$ algorithm for solving maximum independent set prob-

- lem,” *IEEE Transactions on Computers*, 35(9), pp.847–851, 1986.
- [55] K. Jain, J. Padhye and V.N. Padmanabhan, L. Qiu, “Impact of interference on multihop wireless network performance,” *Wireless Networks*, 11(4), pp.471–487, 2005.
- [56] D. S. Johnson and M. A. Trick (Eds), “Cliques, Coloring, and Satisfiability,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26, 1996.
- [57] D. S. Johnson and M. Szegedy, “What are the least tractable instances of max independent set,” *Proc. SODA 1999*, pp.927–928, 1999.
- [58] H. C. Johnston, “Cliques of a graph—variations on the Bron-Kerbosch algorithm,” *International Journal of Computer Information Science*, 5(3), pp.209–238, 1976.
- [59] R. Karp, “Reducibility among combinatorial problems,” *Complexity of Computer Computations*, pp.85–103, 1972.
- [60] 木津隆史, 荒木俊郎, 柏原敏伸, “円筒上長方形交グラフの最大クリークを求めるアルゴリズム,” *信学論 (A)*, vol.J78-A, pp.1149–1156, 1995.
- [61] S. Kobayashi, T. Kondo, K. Okuda and E. Tomita, “Extracting globally structure free sequences by local structure freeness,” *Proc. DNA9*, p.206, 2003.
- [62] J. Kneis, A. Langer, P. Rossmanith, “A fine-grained analysis of a simple independent set algorithm,” *Proc. FSTTCS 2009, LIPIcs*, 4, pp. 287-298, 2009.
- [63] I. Koch, “Enumerating all connected maximal common subgraphs in two graphs,” *Theoretical Computer Science*, 250(1–2), pp.1–30, 2001.
- [64] E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, “Generating all maximal independent sets: NP-hardness and polynomial-time algorithms,” *SIAM Journal on Computing*, 9(3), pp.558–565, 1980.
- [65] G. Liu, L. Wong, H.-N. Chua, “Complex discovery from weighted PPI networks,” *Bioinformatics*, 25(15), pp.1891–1897, 2009.
- [66] X. Li, M. Wu, C. -K.Ng Kwoh and S.-K. Ng, “Computational approaches for detecting protein complexes from protein interaction networks: a survey,” *BMC Genomics*, 11(1), 1, 2010.

- [67] K. Makino and T. Uno, “New algorithms for enumerating all maximal cliques,” SWAT 2004, Lecture Notes in Computer Science, 3111, pp.260-272, 2004.
- [68] E. Maslov, M. Batsyn and P.M. Pardalos, “Speeding up branch and bound algorithms for solving the maximum clique problem,” Journal of Global Optimization, 59(1), pp.1–21, 2014.
- [69] T. Matsunaga, C. Yonemori, E. Tomita, M. Muramatsu, “Clique-based data mining for related genes in a biomedical database,” BMC Bioinformatics, 10(1), 205, 2009.
- [70] T. Matsunaga, S. Kuwata and M. Muramatsu, “Computational gene knockout reveals transdisease-transgene association structure,” Journal of Bioinformatics and Computational Biology, 8(05), pp.843–866, 2010.
- [71] 松野浩嗣, 田中都子, “定数次数のグラフの最大クリークを抽出するビット演算アルゴリズム,” 情報処理学会論文誌, 37, pp.1869–1872, 1996.
- [72] G. J. Minty, “On maximal independent sets of vertices in claw-free graphs,” Journal Combinatorial Theory, Series B, 28(3), pp.284–304, 1980.
- [73] S. Mohseni-Zadeh, A. Louis, P. Brézellec and J.-L. Risler, “PHYTOPROT: a database of clusters of plant proteins,” Nucleic Acids Research, 32(suppl 1), pp.D351–D353, 2004.
- [74] J. W. Moon and L. Moser, “On cliques in graphs,” Israel Journal of Mathematics, 3(1), pp.23–28, 1965.
- [75] T. Mori, T. Tamura, D. Fukagawa, A. Takasu, E. Tomita, and T. Akutsu, “A clique-based method using dynamic programming for computing edit distance between unordered trees,” Journal of Computational Biology, 19(10), pp. 1089–1104, 2012.
- [76] R. Mosca, “Polynomial algorithms for the maximum stable set problem on particular classes of P_5 -free graphs,” Information Processing Letters, 61(3), pp.137–143, 1997.
- [77] 中森眞理雄, 富田悦次, “数理モデルの応用技術,” 情報処理学会 50 年のあゆみ, 第 3 編, 情報処理学会, pp.227-229, 2010.

- [78] 中西裕陽, 富田悦次, “最大クリーク問題の多項式時間的可解性の一結果,” 信学論 (D), vol.J93-D, no.4, pp.417–425, 2010.
- [79] 中西裕陽, 富田悦次, “最大クリーク問題の多項式時間的可解性の拡張,” 信学技報, COMP2011-30, pp.15–22, 2011.
- [80] 中西裕陽, 富田悦次, “最大クリーク問題の多項式時間的可解性の改良結果,” 信学論 (D), vol.J94-D, no.5, pp.843–851, 2011.
- [81] 中西裕陽, 富田悦次, 若月光夫, 西野哲朗, “最大クリーク問題の多項式時間的可解性の更なる改良結果,” 信学論 (D), vol.J94-D, no.12, pp.2037–2046, 2011.
- [82] 中西裕陽, 富田悦次, 若月光夫, 西野哲朗, “最大クリーク問題の多項式時間的可解性の拡張,” 信学論 (D), vol.J95-D, no.9, pp.1716–1728, 2012.
- [83] 中西裕陽, 富田悦次, 若月光夫, 西野哲朗, “最大クリーク問題の多項式時間的可解性の拡張の改良,” 信学論 (D), vol.J97-D, no.6, pp. 1106–1121, 2014.
- [84] 名久井行秀, 西野哲朗, 富田悦次, 中村知倫, “節点重み最大クリーク抽出に基づく量子回路の深さ最小化,” 京都大学 数理解析研究所講究録, 1325, pp. 45–50, 2003.
- [85] H. Ogawa, “Labeled point pattern matching by Delaunay triangulation and maximal cliques,” *Pattern Recognition*, 19(1), pp. 35–40, 1986.
- [86] Y. Okubo, M. Haraguchi and E. Tomita, “Structural change pattern mining based on constrained maximal k-plex search,” *DS 2012, Lecture Notes in Artificial Intelligence*, 7569, pp.284-298, 2012.
- [87] Y. Okubo, M. Haraguchi and E. Tomita, “Relational change pattern mining based on modularity difference,” *MIWAI 2013, Lecture Notes in Artificial Intelligence*, 8271, pp.187–198, 2013.
- [88] Y. Okubo, M. Haraguchi and E. Tomita, “Enumerating maximal isolated cliques based on vertex-dependent connection lower bound,” *MLDM 2016, Lecture Notes in Computer Science*, 9729, pp. 569-583, 2016.
- [89] P. M. Pardalos and J. Xue, “The maximum clique problem,” *Journal of Global Optimization*, 4(3), pp.301–328, 1994.
- [90] J. Pattillo, N. Youssef and S. Butenko, “Clique relaxation models in social network analysis,” *Handbook of Optimization in Complex Networks: Theory and*

- Applications, Springer Optimization and Its Applications, 58, pp.143–162, 2012.
- [91] M.G. Ravetti and P. Moscato, “Identification of a 5-protein biomarker molecular signature for predicting alzheimer’s disease,” Plos One 3(9):e3111, 2008.
- [92] I. Razgon, “Faster computation of maximum independent set and parameterized vertex cover for graphs with maximum degree 3,” Journal of Discrete Algorithms, 7(2), pp.191–212, 2009.
- [93] J. M. Robson, “Algorithms for maximum independent sets,” Journal of Algorithms, 7(3), pp.425–440, 1986.
- [94] J. M. Robson, “Finding a maximum independent set in time $O(2^{n/4})$,” Technical Report 1251-01, Université Bordeaux I, LaBRI, 2001.
- [95] M. Shindo and E. Tomita, “A simple algorithm for finding a maximum clique and its worst-case time complexity,” Systems and Computers in Japan, 21(3), Wiley, pp.1–13, 1990.
(新道美喜男, 富田悦次, “最大クリークを抽出する単純なアルゴリズムとその最大時間計算量,” 信学論 (D), vol.J71–D–I, pp.472–481, 1988. の英訳版)
- [96] D. M. Strickland, E. Barnes, J. S. Sokol, “Optimal protein structure alignment using maximum cliques,” Operations Research, 53(3), pp.389–402, 2005.
- [97] R. E. Tarjan and A. E. Trojanowski, “Finding a maximum independent set,” SIAM Journal on Computing, 6(3), pp.537–546, 1977.
- [98] E. Tomita, A. Tanaka and H. Takahashi, “The worst-case time complexity for finding all the cliques,” Technical Report of the University of Electro-Communications, UEC-TR-C5(2), 1988. Available from: http://ir.lib.uec.ac.jp/infolib/user_contents/9090000351/9090000351.pdf
- [99] E. Tomita, A. Tanaka, H. Takahashi, “An optimal algorithm for finding all the cliques,” Technical Report of IPSJ, 1989-AL-12, pp. 91-98, 1989.
- [100] E. Tomita and T. Seki, “An efficient branch-and-bound algorithm for finding a maximum clique,” DMTCS 2003, Lecture Notes in Computer Science, 2731, pp.278–289, 2003.
- [101] E. Tomita, A. Tanaka and H. Takahashi, “The worst-case time complexity for

- generating all maximal cliques,” COCOON 2004, Lecture Notes in Computer Science, 3106, pp.161–170, 2004.
- [102] E. Tomita, A. Tanaka, and H. Takahashi, “The worst-case time complexity for generating all maximal cliques and computational experiments,” Theoretical Computer Science, 363(1) (Special Issue on COCOON 2004), pp.28–42, 2006.
- [103] E. Tomita, T. Kameda, “An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments, ” Journal of Global Optimization, Vol. 37, pp.95–111, 2007.
- [104] E. Tomita, T. Akutsu and T. Matsunaga, “Efficient algorithms for finding maximum and maximal cliques: Effective tools for bioinformatics, “ in “Biomedical Engineering, Trends in Electronics, Communications and Software,” A. N. Laskovski (Ed.), ISBN: 978-953-307-475-7, InTech, pp.625–640, 2011. Available from: <http://www.intechopen.com/articles/show/title/efficient-algorithms-for-finding-maximum-and-maximal-cliques-effective-tools-for-bioinformatics>
- [105] E. Tomita, Y. Sutani, T. Higashi, M. Wakatsuki, “A simple and faster branch-and-bound algorithm for finding a maximum clique with computational experiments,” IEICE Transactions on Information and Systems, vol. E96-D, pp.1286–1298, 2013.
- [106] E. Tomita, “Clique Enumeration,” in Ming-Yang Kao (Ed.), “Encyclopedia of Algorithms, 2nd Edition,” pp. 313–317, 2016.
- [107] E. Tomita, K. Yoshida, T. Hatta, A. Nagao, H. Ito and M. Wakatsuki, “A much faster branch-and-bound algorithm for finding a maximum clique,” FAW 2016, Lecture Notes in Computer Science, 9711, pp.215–226, 2016.
- [108] T. Uno, “An efficient algorithm for solving pseudo clique enumeration problem,” Algorithmica, 56(1), pp.3–16, 2010.
- [109] 宇野毅明, “高速クリーク・密部分グラフマイニングアルゴリズム,” 人工知能学会誌, 27, pp.246-251 (特集:「離散構造処理系 - 知能情報処理を支えるアルゴリズムの技法」), 2012.

- [110] O. Weide, D. Ryan and M. Ehrgott, “An iterative approach to robust and integrated aircraft routing and crew scheduling,” *Computers & Operations Research*, 37(5), pp.833–844, 2010.
- [111] G.J. Woeginger, “Exact algorithms for NP-hard problems: Survey,” *Combinatorial Optimization-Eureka! You shrink!*, Lecture Notes in Computer Science, 2570, pp.185–207, 2003.
- [112] Q. Wu, and J. K. Hao, “A review on algorithms for maximum clique problems,” *European Journal of Operational Research*, 242(3), pp. 693-709, 2015.
- [113] M. Xiao, “A note on vertex cover in graphs with maximum degree 3,” *Proc. COCOON 2010*, pp.150–159, 2010.
- [114] M. Xiao, “A simple and fast algorithm for maximum independent set in 3-degree graphs,” *WALCOM 2010*, Lecture Notes in Computer Science, 5942, pp.281–292, 2010.
- [115] M. Xiao and H. Nagamochi, “Confining sets and avoiding bottleneck cases: A simple maximum independent set algorithm in degree-3 graphs,” *Theoretical Computer Science*, 469, pp.92–104, 2013.
- [116] M. Xiao and H. Nagamochi, “Exact algorithms for maximum independent set,” *ISAAC 2013*, Lecture Notes in Computer Science, 8283, pp.328–338, 2013.
- [117] M. Xiao and H. Nagamochi, “Exact algorithms for maximum independent set,” *arXiv:1312.6260v1 [cs.DS]*, Dec. 2013.
- [118] 米森力, 松永務, 関根純, 富田悦次, “クリークを用いた企業間関係の構造分析,” *DBSJ Journal*, 7(4), pp.55–60, 2009.
- [119] H. Zhai, M. Haraguchi, Y. Okubo and E. Tomita, “Enumerating maximal clique Sets with pseudo-clique constraint,” *DS 2015*, Lecture Notes in Computer Science, 9356, pp. 324-339, 2015.
- [120] H. Zhai, M. Haraguchi, Y. Okubo, E. Tomita, “A fast and complete algorithm for enumerating pseudo-cliques in large graphs ,” *International Journal of Data Science and Analytics*, 2(3-4), pp.145–158, 2016.
- [121] B. Zhang, B.-H. Park, T. Karpinets and N. F. Samatova, “From pull-down

- data to protein interaction networks and complexes with biological relevance,”
Bioinformatics, 24(7), pp.979–986, 2008
- [122] D. Zuckerman, “Linear degree extractors and the inapproximability of max
clique and chromatic number,” Proc. STOC 2006, pp. 681-690, 2006.

関連論文の印刷公表方法と時期

(1) 中西裕陽, 富田悦次

題目「最大クリーク問題の多項式時間的可解性の一結果」

信学論 (D), Vol.J93-D, No.4, pp.417-425 (2010. 4).

(第3章の内容)

(2) 中西裕陽, 富田悦次

題目「最大クリーク問題の多項式時間的可解性の改良結果」

信学論 (D), Vol.J94-D, No.5, pp.843-851 (2011. 5).

(第4章の内容)

(3) 中西裕陽, 富田悦次, 若月光夫, 西野哲朗

題目「最大クリーク問題の多項式時間的可解性の更なる改良結果」

信学論 (D), Vol.J94-D, No.12, pp.2037-2046 (2011. 12).

(第5章の内容)

(4) 中西裕陽, 富田悦次, 若月光夫, 西野哲朗

題目「最大クリーク問題の多項式時間的可解性の拡張」

信学論 (D), Vol.J95-D, No.9, pp.1716-1728 (2012. 9).

(第6章の内容)

(5) 中西裕陽, 富田悦次, 若月光夫, 西野哲朗

題目「最大クリーク問題の多項式時間的可解性の拡張の改良」

信学論 (D), Vol.J97-D, No.6, pp.1106-1121 (2014. 6).

(第7章の内容)

その他の参考文献

- (1) 中西裕陽, 富田悦次

題目「最大クリーク問題の多項式時間的可解性 - 基本的結果」

FIT2008 講演論文集 第1分冊, RA-001, pp.1-4 (2008. 9).

(査読あり)

- (2) Etsuji Tomita, Hiroaki Nakanishi

題目「Polynomial-time solvability of the maximum clique problem」

Proc. the 3rd European Computing Conference (ECC 2009),

pp.203-208 (2009. 6).

(査読あり)

- (3) 中西裕陽, 富田悦次

題目「最大クリーク問題の多項式時間的可解性の改良結果」

信学技報, COMP2010-43, pp.29-36 (2010. 12).

- (4) 中西裕陽, 富田悦次

題目「最大クリーク問題の多項式時間的可解性の更なる改良結果」

信学技報, COMP2011-6, pp.41-48 (2011. 4).

- (5) 中西裕陽, 富田悦次, 若月光夫, 西野哲朗

題目「最大クリーク問題の多項式時間的可解性の拡張」

信学技報, COMP2011-30, pp.15-22 (2011. 10).

- (6) 中西裕陽, 富田悦次, 若月光夫, 西野哲朗

題目「最大クリーク問題の多項式時間的可解性の拡張の改良」

信学技報, COMP2012-28, pp.17-24 (2012. 9).

(7) 中西裕陽, 富田悦次, 若月光夫, 西野哲朗

題目「最大クリーク問題の多項式時間的可解性の拡張の更なる改良」

信学技報, COMP2014, pp.85-92 (2014. 6).

著者略歴

- 2004年3月 早稲田大学教育学部 理学科数学専修 卒業
- 2004年4月 電気通信大学大学院 電気通信学研究科 情報通信工学専攻 入学
- 2008年3月 電気通信大学大学院 電気通信学研究科 情報通信工学専攻 修了
- 2008年5月より
- 2016年11月現在 電気通信大学 先進アルゴリズム研究ステーション 協力研究員
- 2013年4月より
- 2015年3月 早稲田大学 教育学部 数学科 助手