Vol.2022-AL-187 No.3
2022/3/14

クリーク幅の構成表現に対するグラフ分解手法の適用の検討

鈴木　絢香[1,a)]

概要：本稿では，クリーク幅の近似値を求める問題に対して，グラフの分解手法を適用した場合の知見を与える．ここでグラフの分解手法の適用とは，クリーク幅を求めたいグラフを小さなグラフに分解して，そのグラフのクリーク幅を求め，小さなグラフから元のグラフのクリーク幅を求める方法である．具体的には，グラフの分解手法としてスプリット分解に着目し，次を明らかにした．(1) スプリット分解を用いて小さく分解したグラフを再び構成する際に，構成表現数が増えないような分解順序の性質を明らかにした．さらにそのような分解順序を導出するアルゴリズムを提案した．(2) スプリット分解を拡張した概念として擬スプリット分解を定義し，スプリット分解で分解できないグラフに対する分解手法を与えた．またモジュール連結度という概念を定義し，グラフを再構成するために追加で必要なラベル数に関して，そのラベル数が少ない分解手法から多い分解手法への順序づけを可能とした．これらの定義を用いてクリーク幅が定まるグラフにまで分解できると仮定した場合の構成表現数を求めた．また，擬スプリット分解とランク幅との関係を考察した．

キーワード：クリーク幅，ランク幅，スプリット分解

Graph decompositions for approximating clique-widths

Abstract: This work studies decomposition methods of graphs for approximating clique-widths. In these methods, clique-width expressions are obtained by decomposing a given graph, obtaining clique-width expressions of the decomposed graphs, and reconstructing the given graph. We focus on split decompositions and show two main results. (1) We prove properties of the decomposition orders that do not increase the number of labels of clique-width expressions and propose an algorithm to compute such a decomposition order. (2) We introduce pseudo-split decompositions by extending split decompositions. They are able to decompose graphs with no split. Then, we define the module connectivity, which allows to select fewer labels that are used to reconstruct a given graph. By using these concepts, we show an upper bound of the number of labels of clique-width expressions under the hypothesis that clique-width expressions of the decomposed graphs can be obtained. We also consider the relation between pseudo-split decompositions and rank-widths.

Keywords: clique-width, rank-width, split decomposition

1. はじめに

クリーク幅は，グラフの複雑さを表すパラメータである [3]．入力のグラフのクリーク幅が定数以下である場合に，NP-困難な複数の問題を線形時間で解くことができることがわかっている [4]．似たパラメータに木幅があるが，クリーク幅はより密なグラフに対しても定数の値をとることができる [10]．しかし，グラフ G と正の整数 k が与えられたとき，G のクリーク幅が k 以下であるか否かの判定問題は NP-完全である [6]．$k = 3$ の場合，k 以下かを判定してその構成表現を出力する多項式時間アルゴリズムが提案されている [2]．$k \geq 4$ の場合，ランク幅という別のパラメータを用いてクリーク幅の近似値を出力する FPT アルゴリズムがいくつか提案されている [7], [11], [12]．しかし，計算時間が早く，かつタイトな近似値を得ることは未解決な問題である．

本稿では，クリーク幅の近似値を求める問題に対して，グラフの分解手法を適用した場合の知見を与える．ここでグラフの分解手法の適用とは，クリーク幅を求めたいグラフを小さなグラフに分解して，そのグラフのクリーク幅を

[1] 電気通信大学
University of Electro-Communications, Japan
[a)] s2031087@edu.cc.uec.ac.jp

求め，小さなグラフから元のグラフのクリーク幅を求める方法である．なお，本稿で挙げる定理の証明については，[15] を参照されたい．

定義 1 (k-ラベル付きグラフ)．k を正の整数とする．k-ラベル付きグラフ $((V,E),l)$ とは，グラフ (V,E) と写像 $l : V \to \{1,2,\ldots,k\}$ との組である．

定義 2 (クリーク幅)．グラフ G が与えられたとき，次の 4 つの操作によって G のラベル付きグラフを構成することを考える．

- 頂点の導入：ラベル i ($i \in \{1,\ldots,k\}$) が割り当てられた頂点 v を導入する操作．この操作を $(v)_i$ と表す．
- ラベルの更新： ラベル i をラベル j へ更新する操作．この操作を $\rho_{i\to j}$ と表す．
- 辺の挿入：$i \neq j$ で，ラベル i のすべての頂点とラベル j のすべての頂点を辺で結ぶ操作．この操作を $\eta_{i,j}$ と表す．
- 非交和：ラベル付きグラフ同士の非交和を生成する操作．この操作を $\oplus$ と表す．

G を構成するための操作手順を表したものを構成表現と呼ぶ．G を構成するために頂点の導入としてラベル数 k を用いた場合，その構成表現を k-表現と呼ぶ．また，この k の値を構成表現数と呼ぶ．G のクリーク幅 $\mathrm{cwd}(G)$ とは，G のラベル付きグラフを構成するのに必要な最小のラベルの数である．

2. スプリット分解の適用の検討

グラフの分解手法としてまず，スプリット分解 [5] に着目した．スプリット分解は，クリーク幅との関連がいくつか明らかになっている分解手法である．例えば，グラスクラスの一つである距離遺伝的グラフ [1] に含まれるグラフのクリーク幅は高々 3 で [8]，このグラフクラスのグラフはスプリット分解を行うと，すべての素グラフの頂点数が 3 以下である [9]．

スプリット分解を定義するために，まずスプリットを定義する．スプリットに単分解を再帰的に適用したグラフがスプリット分解である．図 1 はその例である．

定義 3 (連結グラフのスプリット)．連結グラフ $G=(V,E)$ のスプリットとは V の分割 $\{V_1,V_2\}$ で，次を満たすもののことである．

- $|V_1| \geq 2, |V_2| \geq 2$.
- $W_1 \subseteq V_1$ と $W_2 \subseteq V_2$ が存在して，任意の $v \in W_1$ に対して $N_G(v) \cap V_2 = W_2$ であり，かつ，任意の $v \in V_1 \backslash W_1$ に対して $N_G(v) \cap V_2 = \emptyset$.

定義 4 (単分解)．$G=(V,E)$ のスプリット $\{V_1,V_2\}$ に関する単分解とは，グラフ $G'=(V',E')$ である．

$$V' = V \cup \{v_1, v_2\},$$
$$E' = E \cup \{v_1, v_2\} \cup \{\{u, v_1\} \cup \{v_2, v\} \mid u \in W_1, v \in W_2\}.$$

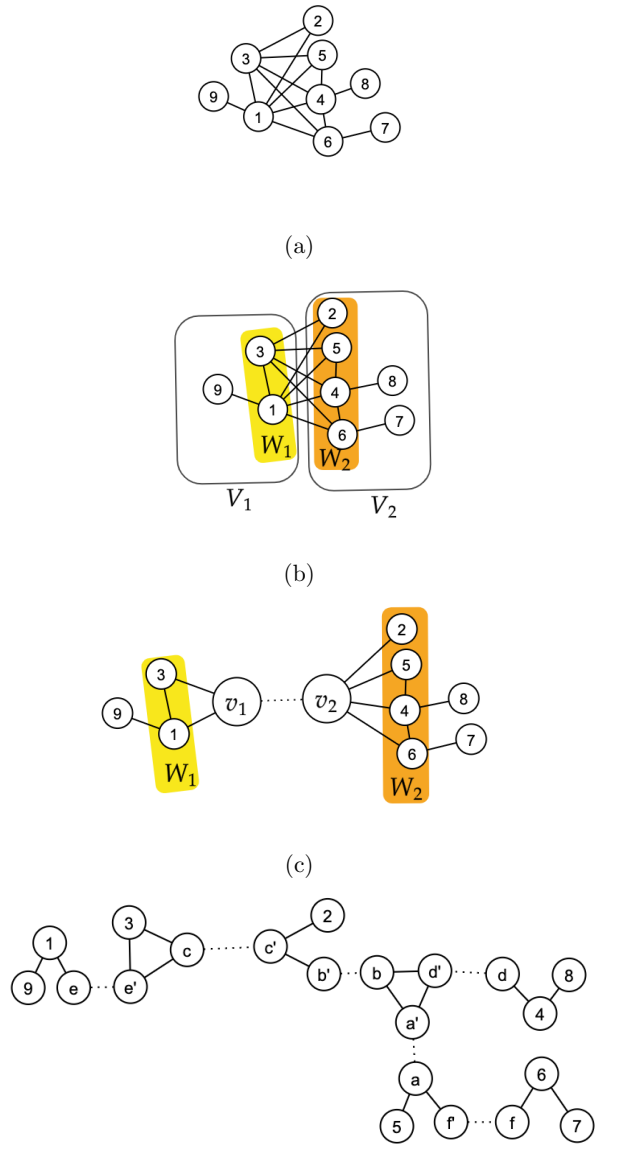

図 1: スプリットと単分解，スプリット分解の例．グラフ G を図 1a としたとき，$V(G)$ の分割として $\{V_1,V_2\} = \{\{1,3,9\},\{2,4,5,6,7,8\}\}$ を考えると，これはスプリットである．スプリット $\{V_1,V_2\}$ に関する単分解は図 1c のグラフである．G のスプリット分解は図 1d である．

上記のような v_i をマーカー要素，2 頂点の組 $\{v_1,v_2\}$ をマーカー要素の組，辺 $\{v_1,v_2\}$ をマーカー辺と呼ぶ．

定義 5 (スプリット分解)．連結グラフ G のスプリット分解 $D(G)$ とは，スプリットが存在しなくなるまで G に単分解を再帰的に適用してできた連結グラフである．また分解の要素集合 $\mathcal{D}$ とは，$D(G)$ からすべてのマーカー辺を取り除いてできた連結グラフの集合である．

定義 6 (スプリット分解木)．グラフ G のスプリット分解木 $T(G)$ とは，次で定義する木である．$D(G)$ の分解の要素集合 $\mathcal{D}$ に含まれる一つのグラフに対して一つのノード h

を対応させることを考える．ノード h に対応させた $\mathcal{D}$ のグラフを G_h^* と表す．

$$V(T(G)) = \{h \mid G_h^* \in \mathcal{D}\},$$
$$E(T(G)) = \{\{h,h'\} \mid \exists\{m,m'\} \in (D(G) \text{のマーカー要素の組}),$$
$$m \in V(G_h^*) \wedge m' \in V(G_{h'}^*)\}.$$

2.1 構成表現数が増えないための分解順序の性質

グラフ G が与えられたとき，G のクリーク幅を求めるために，スプリット分解を用いることを考える．このとき問題となるのは，分解してから元のグラフを再構成する際に，追加でラベルを用いる必要が生じて，実際の値よりも，G の構成表現数が大きくなる可能性があることである．事実，分解の要素集合に含まれるグラフの最大クリーク幅が k であるとき，G のクリーク幅は $2k+1$ 以下である [14]．

そこで本研究では，スプリット分解を用いて分解したグラフを再構成する際に，構成表現数が増えないような分解順序の性質を明らかにした．なお，分解順序の逆をたどったものが，構成順序である．

定義 7 (分解順序)．G をグラフ，$D(G)$ を G のスプリット分解とする．$D(G)$ のマーカー要素の組の集合を $U = \{\{u_1,u_1'\},\ldots,\{u_n,u_n'\}\} = \{\tilde{u}_1\ldots,\tilde{u}_n\}$ とする．ここで $\tilde{u}_i = \{u_i,u_i'\}$ を表す．このとき分解順序 $(v_1,v_2,\ldots,v_n)$ とは，デカルト積 $U\times\cdots\times U$ の部分集合の要素で，$i\neq j$ である任意の i と j に対して $v_i \neq v_j$ であるものと定義する．

定理 1. G を一つ以上スプリットが存在するグラフ，$k\geq 2$ とする．分解の要素集合に含まれるグラフにすべて $k+1$ 表現が存在して，条件を満たす分解順序が存在するならば，$\mathrm{cwd}(G)\leq k+1$ である．(条件の詳細は [15] を参照されたい)

2.2 分解順序を導出するアルゴリズムの提案

定理 1 を満たすための十分条件として，スプリット分解木の各ノードに関する 2 つの条件を明らかにした．この十分条件をもとに定理 1 を満たす分解順序を出力するアルゴリズムを与え，その正当性を証明した．具体的には，入力として，グラフ G のスプリット分解木 T を与えたとき，出力として次で定義するトポロジカル擬順序が得られる場合，その順序が定理 1 の性質を満たす分解順序であることを示した．

アルゴリズムのアイデアは次の通りである．まず，有向グラフを構成する．頂点集合として，マーカー要素すべてからなる集合，弧集合として，二つのマーカー要素間で十分条件を満たすならばそのときに限り弧を与えてその集合に含める．その有向グラフから定義できるトポロジカル擬順序が存在するならば，十分条件を満たしながら，もとの与えられたグラフを分解の要素集合まで分解することができる．

定義 8. $\overrightarrow{G} = (\overrightarrow{V},\overrightarrow{E})$ は有向グラフで，$\overrightarrow{V} = \{v_1,\ldots,v_n\}$ とする．$i\neq j$ を満たす任意の $i,j\in\{1,\ldots,n\}$ に対して，次が成り立つとき $v_i \precsim v_j$ であると定義する．

$$\text{弧 } (v_i,v_j)\in\overrightarrow{E}\text{が存在して,} (v_j,v_i)\notin\overrightarrow{E} \Rightarrow i<j.$$

さらに，$i\neq j$ を満たす $i,j\in\{1,\ldots,n\}$ に対して，次が成り立つとき $v_i \prec v_j$ であると定義する．v_i から v_j への道 $v_i = v_{i_0},v_{i_1},\ldots,v_{i_l} = v_j$ が存在して，任意の $k\in\{0,\ldots,l-1\}$ に対して，$v_{i_k}\precsim v_{i_{k+1}}$ が成り立つ．一方 $v_i\prec v_j$ が成り立たないとき，$v_i\not\prec v_j$ と表す．また，v_i と v_j が比較可能であるとは $(v_i\prec v_j\wedge v_i\not\succ v_j)\vee(v_i\not\prec v_j\wedge v_i\succ v_j)\vee(v_i\prec v_j\wedge v_i\succ v_j)$ を表す．

定義 9 (トポロジカル擬順序)．$\overrightarrow{G} = (\overrightarrow{V},\overrightarrow{E})$ は有向グラフで，$\overrightarrow{V} = \{v_1,\ldots,v_n\}$ とする．次を満たすとき，$(v_1,v_2,\ldots,v_n)\in\overrightarrow{V}\times\cdots\times\overrightarrow{V}$ は $\overrightarrow{G}$ に関するトポロジカル擬順序であるいう．

- $i\neq j$ を満たす任意の i と j に対して，$v_i\neq v_j$ である．
- v_i と v_j がともに葉であるならば，v_i は v_j 以外のすべての頂点と比較可能である．ここで $v\in\overrightarrow{V}$ が葉であるとは，v が出次数 0 の頂点であることとする．
- 任意の $1\leq i<j\leq n$ に対して，v_i と v_j がともに葉ではないならば，$v_i\prec v_j$ が成り立つ．

3. 擬スプリット分解の適用の検討

3.1 スプリット分解を拡張した概念の提案

分解手法としてスプリット分解を用いると，スプリット分解では分解できないグラフの箇所が存在して，その箇所の頂点数が大きい場合がある．その場合，クリーク幅の近似値の決定に，スプリット分解を用いることが難しい．

そこで本研究では，さらに分解する操作の提案として，スプリット分解を拡張した概念である擬スプリット分解を定義した．そのためにまず，スプリットを拡張して擬スプリットを定義した．擬スプリットを定義するために，k-モジュールの概念を用いた．

定義 10 (k-モジュール [13])．G をグラフ，$M\subseteq V(G)$ とする．次を満たすとき M は G の k-モジュールであるという．M の分割 $\{M^1,\ldots,M^k\}$ が存在して，すべての $i\in\{1,\ldots,k\}$ において，任意の $x,y\in M^i$ と任意の $z\in V(G)-M$ に対して，

$$\{x,z\}\in E(G)\Leftrightarrow\{y,z\}\in E(G)$$

が成り立つ．G に関する $M\subseteq V(G)$ の擬モジュールとは，M の部分集合 M' で，任意の $z\in V(G)-M$ に対して，z は M' のすべての頂点と隣接しているか，M' のいかなる頂点とも隣接していないもののうち極大な集合であることである．また，$M\subseteq V(G)$ について $\mathrm{bm}_G(M)$ とは，G に関する M の擬モジュールの数を表す．

擬スプリットは，頂点の分割を考えて，分割うち，片方

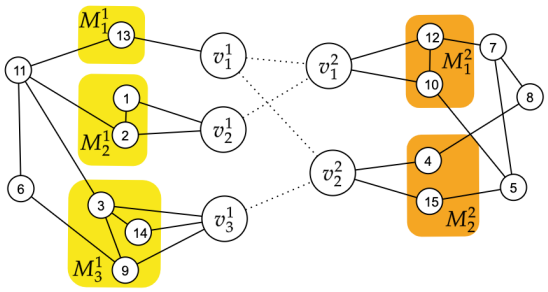

図 2: 3-擬スプリット $\{\{1,2,3,6,9,11,13,14\}, \{4,5,7,8,10,12,15\}\}$ に関する単分解の例.

の頂点集合がもう片方の頂点集合に対して同じ隣接頂点を持つ頂点を一つの頂点集合とみなしている概念である．加えて，冗長な分解が生じないよう条件をつけている．

定義 11 (k-擬スプリット)**.** 連結グラフ $G=(V,E)$ の k-擬スプリットとは，V の 2 分割 $\{V_1,V_2\}$ で次の 3 点を満たすものである．

- $k = \max\{\mathrm{bm}_G(V_1)-\chi_1, \mathrm{bm}_G(V_2)-\chi_2\}$．ここで，$V_1 \setminus N_G(V_2) \neq \emptyset$ のとき $\chi_1=1$，$V_1 \setminus N_G(V_2)=\emptyset$ のとき $\chi_1=0$ として，また，$V_2 \setminus N_G(V_1) \neq \emptyset$ のとき $\chi_2=1$，$V_2 \setminus N_G(V_1)=\emptyset$ のとき $\chi_2=0$ とする．
- $|V_1| \geq 2, |V_2| \geq 2$.
- $V_1 \cap N_G(V_2)$ と $V_2 \cap N_G(V_1)$ のどちらも自明なモジュールではない．ここで集合 $V_i \cap N_G(V_j)$ が自明なモジュールであるとは，V_i の擬モジュールのうち，$V_i \cap N_G(V_j)$ に含まれるすべての擬モジュールの頂点数が 1 であることを表す．

次に，擬スプリットに対しても単分解を与え，それを再帰的に適用してできた連結グラフを擬スプリット分解と定義した．なお，1-擬スプリット分解はスプリット分解である．

さらに，k-擬スプリットの k を選ぶ指標としてモジュール連結度を定義する．

定義 12 (モジュール連結度)**.** 連結グラフ G のモジュール連結度 $\mathrm{con}(G)$ とは，k-擬スプリットが存在するような最小の k であるとする．なお，いかなる $k \in \{1,2,\ldots,|V(G)|\}$ の値でも k-擬スプリットが存在しない場合 $\mathrm{con}(G)=0$ と定義する．

3.2 擬スプリット分解を適用した場合の構成表現数

グラフ G が与えられたとき，G のクリーク幅の近似値を求めるために，k-擬スプリット分解を用いることを考える．$k=1$ の場合，スプリット分解における分解の要素集合に含まれるグラフの最大クリーク幅が k であるとき，G のクリーク幅は $2k+1$ 以下である [14] という事実を用いて近似値を求めることができる．しかし $k \geq 2$ の場合，クリーク幅の近似値は明らかになっていない．

Algorithm 1 連続的な分解操作

Require: グラフ G
Ensure: 構成表現数が何らかの値であると明らかなグラフの集合 S

```
S ← ∅
con(G)-擬スプリット分解を行い分解の要素集合 D_G を得て，D ← D_G
while D ≠ ∅ do
    for all H ∈ D do
        if con(H) = 0 または H の構成表現数がある値に定まる then
            S ← S ∪ {H}
            D ← D \ {H}
        else
            con(H)-擬スプリット分解を行い分解の要素集合 D_H を得て，D ← D ∪ D_H
        end if
    end for
end while
```

そこで本研究では，k-擬スプリット分解を用いた場合のクリーク幅の近似値を求め，定理 2 の値になることを明らかにした．

定理 2. 連結グラフ G が与えられたとき，$\mathrm{con}(G)=k$ とする．このとき k-擬スプリット分解を行い，その分解の要素集合 $\mathcal{D}$ を得たとする．$m=\max\{\mathrm{cwd}(H) \mid H \in \mathcal{D}\}$ ならば，

$$m \leq \mathrm{cwd}(G) \leq (k+1)m+k+1.$$

グラフ G が与えられたとき，擬スプリット分解を繰り返し適用し続けることで，クリーク幅が明らかなグラフか，または擬スプリットが存在ないグラフを得ることができると仮定する．なおここでは，擬スプリットが存在ないグラフのクリーク幅は求まると仮定する．この操作をアルゴリズム 1 で与える．このとき，G の構成表現数の上界は定理 3 の値となる．

定理 3. グラフ G に対して，アルゴリズム 1 を実行し，出力 $S=\{G_1^0, G_2^0, \ldots, G_m^0\}$ を得たとする．G_i^0 はあるグラフ G_i^1 に対して擬スプリット分解をして得られた分解の要素の一つである．もし $G_i^1 \neq G$ ならば，G_i^1 はあるグラフ G_i^2 に対して擬スプリット分解をして得られた分解の要素の一つである．同様の議論を G になるまで考える．その過程に現れるグラフを $G_i^0, \ldots, G_i^{N-1}, G_i^N = G$ とする．また，グラフ $G^0 \in S$ は S のうち，$\prod_{j\in\{1,\ldots,N\}} \mathrm{con}(G^j)$ が最大になるグラフであるとする．このとき，次の式で定まる f_N は G の構成表現数の上界である．

$$f_i = \begin{cases} (\mathrm{con}(G^i)+1)f_{i-1} + \mathrm{con}(G^i) + 1 & \text{if } i \geq 1, \\ \mathrm{cwd}(G^0) & \text{if } i = 0. \end{cases}$$

観察 1 から，分解の各段階で得られるグラフに関して，分解していくごとのモジュール連結の値の定まり方に単調性があるわけではない．その例を図 3 に挙げる．

観察 1. 連結グラフ G のモジュール連結度 $\mathrm{con}(G)=k$ で

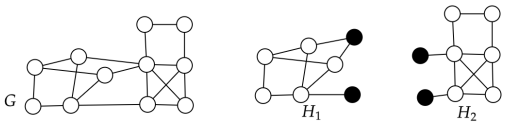

図 3: 観察 1 の例．G は $\mathrm{con}(G) = 2$ を満たす．2-スプリット分解を行いその分解の要素集合 $\mathcal{D} = \{H_1, H_2\}$ を得る．これらのグラフは $\mathrm{con}(H_1) = 1$，$\mathrm{con}(H_2) = 1$ であるので，$\mathrm{con}(H_1) < \mathrm{con}(G)$，$\mathrm{con}(H_2) < \mathrm{con}(G)$．

あるとき，k-スプリット分解を行い分解の要素集合 $\mathcal{D}$ を得る．このとき $H \in \mathcal{D}$ は，$\mathrm{con}(G) \leq \mathrm{con}(H)$ を満たすとは限らない．

また，クリーク幅が非有界なグラフクラスに含まれるグラフの場合でも，モジュール連結度が 0 とは限らない．例えば，$n \times n$ 正方グリッド G_n $(n \in \mathbb{N})$ のクリーク幅は，$\mathrm{cwd}(G_2) = 2, \mathrm{cwd}(G_n) = n + 1$ $(n \geq 3)$ である [8]．一方，対角線上に存在する頂点とそれ以外の頂点との分割を考えると，それらは n-擬スプリット であるので，$1 \leq \mathrm{con}(G_n) \leq n$ である．

3.3 擬スプリットとランク幅およびクリーク幅との関係

擬スプリット分解を適用した場合の構成表現数がクリーク幅の実際の値よりもどれほど大きいものなのかを比較することは，グラフの分解手法では難しい．そこで，擬スプリットとランク幅との関係，および擬スプリットとクリーク幅との関係を考察した．なお，ランク幅とクリーク幅との関係として $\mathrm{rwd}(G) \leq \mathrm{cwd}(G) \leq 2^{\mathrm{rwd}(G)+1} - 1$ という事実 [11] がある．

定理 4. 任意のグラフ G で，

$$\mathrm{rwd}(G) \leq \max_{\{V_1,V_2\} \in \{G \text{ が取りうる分割 }\}} \{\min\{\mathrm{bm}_G(V_1) - \chi_1, \mathrm{bm}_G(V_2) - \chi_2\}\}.$$

定理 5. 任意のグラフ G で，$\mathrm{con}(G) \leq \mathrm{cwd}(G)$．

また，後述の定理 6 からわかるように，擬スプリットの単分解を行う前のグラフと行った後のグラフでは，その分割に関して，カットランク関数が同じ値である．なお，カットランク関数はランク幅を定義するために用いている概念である [11]．

定義 13 (カットランク関数 [11])**.** G をグラフとする．$X \subseteq V(G)$ に対する G のカットランク関数 cutrk_G を次のように定義する．

$$\mathrm{cutrk}_G(X) = \mathrm{rk}(A(G)[X, V(G) \setminus X]).$$

ここで $A(G)[X, V(G) \setminus X]$ とは $|X|$ 行 $|V(G) \setminus X|$ 列の GF(2) 上の隣接行列で，$A(G)[X, V(G) \setminus X] = (m_{ij})$ とする．なお，$i \in X$ の頂点と $j \in V(G) \setminus X$ の頂点が隣接しているとき，$m_{ij} = 1$，$i \in X$ の頂点と $j \in V(G) \setminus X$ の頂点が隣接していないとき，$m_{ij} = 0$ であるとする．また rk は行列の階数を返す関数であるとする．

定理 6. グラフ G の頂点分割 $\{V_1, V_2\}$ は k-擬スプリットであるとし，$\{V_1, V_2\}$ に対する単分解を行い得たグラフを G' とする．G' の頂点分割 $\{V_1', V_2'\}$ を，$V_1' = V_1 \cup \{v_1^1, \ldots, v_{k^1}^1\}, V_2' = V_2 \cup \{v_1^2, \ldots, v_{k^2}^2\}$ とする．ここで，$\{v_1^1, \ldots, v_{k^1}^1\}$ と $\{v_1^2, \ldots, v_{k^2}^2\}$ はそれぞれ単分解で新たに導入した頂点のうち，V_1 に隣接する頂点と V_2 に隣接する頂点である．このとき，

$$\mathrm{rk}(A(G)[V_1, V_2]) = \mathrm{rk}(A(G')[V_1', V_2']).$$

4. まとめ

スプリット分解を用いる手法が，構成表現数の上界を抑えることに有効な場合があることがわかった．また，スプリット分解を拡張した分解手法を提案した．今後の課題として，以下を挙げる．

- 擬スプリット分解を繰り返し適用し続けることで得られるグラフ，または $\mathrm{con}(G) = 0$ であるグラフ G は，どのようなグラフか？ ある特定のグラフクラスに含まれるか？ 含まれるとしたらどのようなグラフクラスか？そのクリーク幅を求めることは可能か？
- グラフ G が与えられたとき，G のモジュール連結度 $\mathrm{con}(G)$ とその $\mathrm{con}(G)$-スプリット分解を効率的に求めることは可能か？

参考文献

[1] Hans Jürgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, Vol. 41, No. 2, pp. 182–208, 1986.

[2] Derek G. Corneil, Michel Habib, Jean Marc Lanlignel, Bruce Reed, and Udi Rotics. Polynomial-time recognition of clique-width 3 graphs. *Discrete Applied Mathematics*, Vol. 160, No. 6, pp. 834–865, 2012.

[3] Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *Journal of Computer and System Sciences*, Vol. 46, No. 2, pp. 218–270, 1993.

[4] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, Vol. 33, No. 2, pp. 125–150, 2000.

[5] William H. Cunningham. Decomposition of directed graphs. *SIAM Journal on Algebraic and Discrete Methods*, Vol. 3, pp. 214–228, 1982.

[6] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width minimization is NP-hard. *Proceedings of the Annual ACM Symposium on Theory of Computing*, Vol. 2006, pp. 354–362, 2006.

[7] Fedor V. Fomin and Tuukka Korhonen. Fast fpt-approximation of branchwidth, 2021.

[8] Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, Vol. 11, No. 3, pp. 423–443, 2000.

[9] Peter L. Hammer and Frédéric Maffray. *Discret. Appl. Math.*, Vol. 27, No. 1-2, pp. 85–99, 1990.
[10] Petr Hliněný, Sang il Oum, Detlef Seese, and Georg Gottlob. Width parameters beyond tree-width and their applications. *Computer Journal*, Vol. 51, No. 3, pp. 326–362, 2008.
[11] Sang il Oum and Paul Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory. Series B*, Vol. 96, No. 4, pp. 514–528, 2006.
[12] Sang il Oum. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, Vol. 5, No. 1, dec 2008.
[13] Michaël Rao. *Décompositions de graphes et algorithmes efficaces*. Theses, Université Paul Verlaine - Metz, June 2006.
[14] Michaël Rao. Solving some NP-complete problems using split decomposition. *Discrete Applied Mathematics*, Vol. 156, No. 14, pp. 2768–2780, 2008.
[15] 鈴木絢香. クリーク幅の構成表現に対するグラフ分解手法の適用の検討. 電気通信大学大学院情報理工学研究科修士論文, 2022.