

2020 年度
修士論文

k -極大独立集合検証問題の
分散計算複雑性

Distributed Complexity of k -Maximal
Independent Set Verification

名古屋工業大学 大学院工学研究科 博士前期課程

情報工学専攻

片山・金研究室所属

2019 年度入学

学生番号：31414050 氏名：佐藤 僚祐

目次

第1章	はじめに	1
1.1	研究背景	1
1.2	本研究の成果	2
1.3	関連研究	2
1.4	論文の構成	3
第2章	諸定義	4
2.1	$CONGEST$ モデル	4
2.2	2 者間通信複雑性	4
2.3	k -極大独立集合	6
第3章	1-MIS 検証問題の $O(1)$ ラウンドアルゴリズム	7
第4章	k-MIS 検証問題の下界	8
4.1	2-MIS 検証問題の下界	8
4.2	3-MIS 検証問題の下界	11
4.3	k -MIS 検証問題の下界	14
第5章	まとめと今後の課題	18
5.1	まとめ	18
5.2	今後の課題	18

第1章

はじめに

1.1 研究背景

ネットワーク上の最大独立集合を発見する最大独立集合問題に対する数多くの分散グラフアルゴリズムが研究されている。大きなサイズの独立集合は経済学, 計算生物学, 符号理論, 実験計画法など様々な分野への応用に用いられる [1] が, そもそも最大独立集合問題は NP 完全であり, 頂点数 n に対して n の多項式時間で解くことは, その近似を含めて絶望的であるとされている。内部計算に指数時間かかることを許した *CONGEST* モデルの下でラウンド数を n の多項式でおさえるような最大独立集合問題の複雑性の結果はいくつか知られている [1], [2], [3] が, 内部計算に指数時間かかることを許容して分散グラフアルゴリズムの複雑性の議論を行うことの妥当性にはやや疑問が残る。そこで今回, 我々は最大独立集合の局所最適解である k -極大独立集合 (k -Maximal Independent Set, k -MIS) について考える。ネットワーク上の k -MIS を発見する k -MIS 問題は集中型アルゴリズムによって $O(n^{k+2})$ 時間で解くことができ, $k = O(1)$ のとき n の多項式時間になるため, 上記の疑問点を解決することができる。

CONGEST モデルにおいて, ある 1 つのノードにグラフ全体のトポロジの情報を集め, そのノード上で逐次アルゴリズムを実行するという素朴なアプローチから自明に $O(n^2)$ ラウンドの上界を得ることができる。 *CONGEST* モデルにおける下界の証明では, 下界をこの n^2 にどれだけ近づけることができるかに興味もたれている。

今回は k -MIS の検証問題 (verification) に着目し, その複雑性について議論を行う。 k -MIS 検証問題とは入力としてネットワークと独立集合が与えられたとき, それが k -MIS かどうかを判定する問題である。 k -MIS 問題に対する素朴な局所探索アルゴリズムは, ある 1 つの独立集合からスタートし,

(I) 現在の状態が k -MIS であるか判定。

(II) yes であればそれを出力, no であれば解を更新

というフェーズを繰り返す. k -MIS 検証問題は上記の (I) に対応しており, k -MIS 問題と関連付いた問題ととらえることができる.

1.2 本研究の成果

今回,我々は k -MIS 検証問題に対するいくつかの複雑性を示した. その結果を表 1.1 にまとめた. 最初に,1-MIS 検証問題が $O(1)$ ラウンドで解けることを証明した. 次に,2-MIS 検証問題に対する $\tilde{\Omega}(\sqrt{n})$ ラウンドの下界と 3-MIS 検証問題に対する $\tilde{\Omega}(n)$ ラウンドの下界を証明した. 最後に,任意の $l \geq 1$ に対して $k = 4l + 5$ としたときに k -MIS 検証問題に対する $\Omega\left(n^{2-\frac{1}{l+1}}/l\right)$ ラウンドの下界を証明した. 特に,下界の証明のアイデアは 2 者間通信の枠組みにおける交叉判定問題からの帰着に基づいている.

表 1.1: 本研究が示した k -MIS 検証問題のラウンド複雑性

k	1	2	3	$4l + 5$
上界 or 下界	上界	下界	下界	下界
結果	$O(1)$	$\tilde{\Omega}(\sqrt{n})$	$\tilde{\Omega}(n)$	$\Omega\left(n^{2-\frac{1}{l+1}}/l\right)$

1.3 関連研究

CONGEST モデルにおける最大独立集合問題の通信複雑性としては,最大重み付き独立集合の $(1 + \epsilon) \cdot \Delta$ -近似を高確率で見つけるアルゴリズムに対する $\left(\frac{\text{poly}(\log \log n)}{\epsilon}\right)$ ラウンドの上界 (Δ は頂点の最大次数) [1] や,最大独立集合を見つけるアルゴリズムに対する $\Omega\left(\frac{n^2}{(\log n)^2}\right)$ ラウンドの下界 [2], 最大独立集合の $(\frac{1}{2} + \epsilon)$ -近似を見つけるアルゴリズムに対する $\Omega\left(\frac{n}{(\log n)^3}\right)$ ラウンドの下界, $(\frac{3}{4} + \epsilon)$ -近似を見つけるアルゴリズムに対する $\Omega\left(\frac{n^2}{(\log n)^3}\right)$ ラウンドの下界 [3] が知られている.

本研究の下界に関する結果は 2 者間通信の枠組みにおける交叉判定問題からの帰着に基づいているが,交叉判定問題からの帰着によって下界を示すという証明方法は多くの問題に対して用いられている. 一部の例として最小カット発見問題と最小全域木問題に対する $\Omega(D + \sqrt{n})$ の下界 (D はグラフの直径) [4] や部分グラフ H_k 検出問題に対する $\Omega\left(\frac{n^{2-1/k}}{bk}\right)$ の下界 [5], 近似最大クリーク K_l 検出問題に対する $\Omega\left(\frac{n}{(l+\sqrt{n})b}\right)$ の下界 [6] などが挙げられる.

1.4 論文の構成

本論文は全5章で構成される. 第2章ではグラフの構造と用語の定義をしている. 第3章では1-MIS 検証問題に対する $O(1)$ ラウンドアルゴリズムについて述べている. 第4章では k -MIS 検証問題 ($k = 2, 3, 4l + 5 (l \geq 1)$) に対する下界について述べている. 第5章ではまとめについて述べている.

第2章

諸定義

2.1 CONGEST モデル

本研究で考える *CONGEST* モデルは, 単純無向グラフ連結グラフ $G = (V, E)$ により表現される. ここで V はノードの集合で $|V| = n$ とし, E は通信リンクの集合である. *CONGEST* モデルでは計算機はラウンドに従って同期して動作するものとする. 1 ラウンド内で, 隣接頂点へのメッセージ送信, 隣接頂点からのメッセージ受信, 内部計算を行う. 各辺は単位ラウンドあたり $b = O(\log n)$ ビットを双方向に伝送可能であり, 各ノードは同一ラウンドに異なる接続辺に異なるメッセージを送信可能である. また, 各ノードには $O(\log n)$ ビットの自然数値による ID が付与されており, 自身の隣接ノードすべての ID を既知であるとする. 各ノードはグラフのトポロジに関する事前知識を持たないものとする.

2.2 2 者間通信複雑性

2 者間通信複雑性の枠組みでは, アリスとボブの二人のプレイヤーがそれぞれ k ビットの $\{0,1\}$ のデータ列で構成されるプライベートな入力 x および y を持っているとする. プレイヤーの目標は, 結合関数 $f(x, y)$ を計算することであり, 複雑性の尺度として $f(x, y)$ を計算するためにアリスとボブが通信によって交換する必要のあるビット数が用いられる.

この枠組みにおける重要な問題として, 交叉判定問題 (set-disjointness) がある. この問題では, アリスとボブはそれぞれ $x \in \{0,1\}^k$ と $y \in \{0,1\}^k$ を入力として持ち, 目的は $DISJ_k(x, y) := \bigvee_{i=1}^k x_i \wedge y_i$ を計算することである. k ビットの交叉判定問題を解くために, アリスとボブは通信によって $\Omega(k)$ ビット交換する必要があることが知られており [7], この事実を用いて最小カット発見問題や最小全域木問題 [4], 部分グラフ検出問題や [5], 近似最大クリーク検出問題 [6] といったさまざまな問題に対する下界の証明がされている.

CONGEST モデルにおいて, 入力グラフ上に特性 P があるかどうかの判定に対する下

限の証明を2者間交叉判定問題から帰着するアプローチは以下のとおりである。最初にアリスとボブは特殊なグラフ $G = (V, E)$ の構築と G を G_A と G_B に分割するカット辺 C の決定を行う。次に、アリスとボブは入力文字列に基づいてそれぞれ G_A に辺 E_A と頂点 V_A , G_B に辺 E_B と頂点 V_B を追加する。このとき、 $DISJ_k(x, y) = 1$ のときのみ、何らかの特性 P (例えば、 P : 「グラフに与えられた MIS が 2-MIS でない」) を持つように辺や頂点を追加する。また、カット辺 C は入力文字列に依存しないようにする。グラフ G に辺や頂点を追加したグラフを $G^{x,y} = (V', E')$ とすると $V' = V \cap (V_A \cup V_B)$, $E' = E \cap (E_A \cup E_B)$ 表すことができる。グラフ $G^{x,y}$ の構造の概要を図 2.1 に示す。

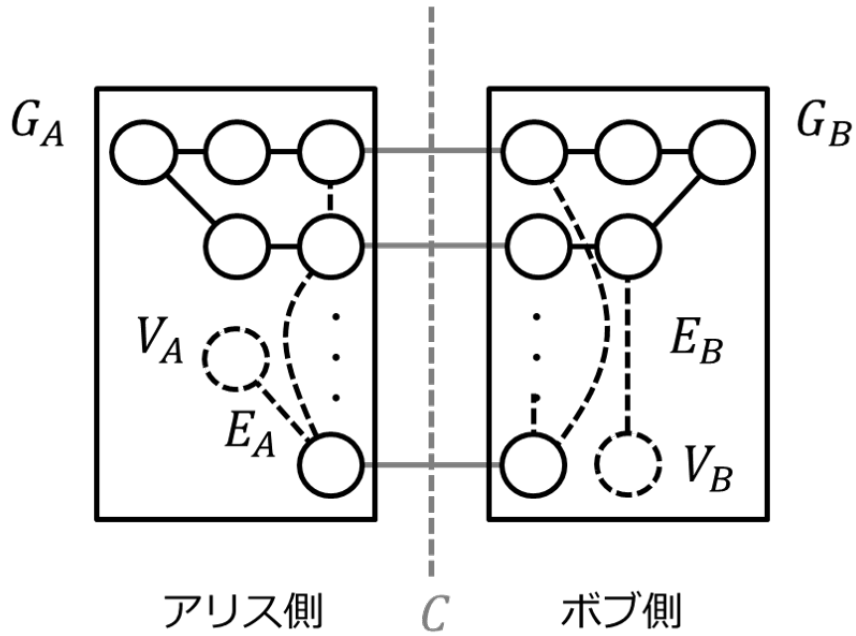


図 2.1: $G^{x,y} = (V', E')$

アリスとボブは、入力グラフ上に特性 P があるかを判定する分散アルゴリズムをシミュレートできる。アリスは G_A に含まれる頂点を、ボブは G_B に含まれる頂点をシミュレートする。2者間通信複雑性モデルでのシミュレートは、次のように実行される。 G_A 中の辺で送信されるメッセージ、あるいは G_B 中の辺で送信されるメッセージは、アリスとボブがそれぞれお互いと通信せずにシミュレートできる。カット辺 C を通じて送信されるメッセージに対しては、お互い情報を交換する必要がある。 $CONGEST$ モデルにおいてグラフ上に特性 P があるかどうかを r ラウンドで判定するアルゴリズム A が存在したとすると、アリスとボブは特性 P の判定のために $O(r \cdot |C| \cdot \log n)$ ビット通信したことになる。これは、各ラウンドで、アルゴリズムが各辺で $O(\log n)$ ビットの通信を行っているからである。このグ

ラフにおいてアルゴリズム A を実行すると同時に 2 者間交叉判定問題も解けていることになる. 例えばアルゴリズムを実行した結果, 入力グラフに特性 P があると判定されれば $DISJ_k(x, y) = 1$ であることがわかるからである. 交叉判定問題の通信複雑性よりアリスとボブは少なくとも $\Omega(k)$ ビットは通信しているはずである. したがって, $CONGEST$ モデルにおいて特性 P があるかどうかを判定する任意のアルゴリズムに対して $r = \Omega(k/|C| \cdot \log n)$ ラウンドの下界を得ることができる. カット辺の大きさが小さくなるほど下界が強くなる.

2.3 k -極大独立集合

定義 2.1. 頂点集合 I に対して, 以下を満たす頂点集合 $I' \subseteq I$ と $S \subset V \setminus I$ のペアが存在しないとき, I を k -極大独立集合と呼ぶ.

1. $|I'| = k$
2. $|S| \geq k + 1$
3. $(I \setminus I') \cup S$ は独立集合

つまり, ある独立集合 I に対して, サイズ k の I の部分集合 I' を取り除いてサイズ $k + 1$ 以上の V の部分集合 S を I に追加したものが新たな独立集合になり得ないとき, I を k -極大独立集合と定義する.

第3章

1-MIS 検証問題の $O(1)$ ラウンドアルゴリズム

この章では, 1-MIS 検証問題を $O(1)$ ラウンドで解くアルゴリズムについて述べる. *CONGEST* モデルにおいて, 入力としてグラフ G と独立集合 I が与えられたとき, 1-MIS 検証問題を解くために次のようなアルゴリズム A を実行する.

- 各頂点 $v \in I$ は, 自分の ID である $v.id$ を隣接頂点全員に送信する.
- 各頂点 $u \notin I$ のうち, 2 種類以上の ID をもらった頂点はアルゴリズムから離脱する.
- 離脱しなかった頂点 $u \notin I$ のうち 1 種類だけの ID ($v.id$ とする) を受信した頂点は離脱していない全隣接頂点へ $v.id$ を送信する.
- 各頂点 $v \in I$ は, 自身と同じ $v.id$ を返信してきた頂点の集合 ($v.X$ とする) のサイズを数え, そのサイズ $|v.X|$ を $v.X$ 中の頂点に送信する. 自身と違う $v.id$ を返信してきた頂点に対しては何もしない.
- メッセージを受け取った $v.X$ 中の頂点は, 送られたサイズ $|v.X| - 1$ と 3 番目のステップで $v.id$ を送信した頂点の個数を比較する. 全ての $v \in I$ に対する $v.X$ 中の全頂点でその値が等しければ与えられた独立集合 I は 1-MIS であり, どれか一つでも等しくなければ I は 1-MIS でないと判定する.

アルゴリズム A について, 以下の定理が成り立つ.

定理 3.1. *CONGEST* モデルにおいて, アルゴリズム A は 1-MIS 検証問題を $O(1)$ ラウンドで解く.

証明. ここの証明がなかなかうまく書けません... アルゴリズムが正しく 1-MIS がどうかを判定することと, 帯域幅の制限をクリアしていることを書けばよいのでしょーか...

第4章

k -MIS 検証問題の下界

この章では, k -MIS 検証問題の下界についての議論を行う. 4.1 節では,2-MIS 検証問題の下界についての定理とその証明を述べる. 4.2 節では,3-MIS 検証問題の下界についての定理とその証明を述べる. 4.3 節では, k -MIS 検証問題 ($k = 4l + 5, l \geq 1$) の下界についての定理とその証明を述べる.

4.1 2-MIS 検証問題の下界

この節では 2-MIS 検証問題の下界についての議論を行う. 具体的には, 次の定理を証明する.

定理 4.1. *CONGEST* モデルにおいて,2-MIS 検証問題を解く全てのアルゴリズムは $\tilde{\Omega}(\sqrt{n})$ の通信ラウンド数を必要とする.

証明. まず初めにアリスとボブが構築するグラフ $G = (V, E)$ を図 4.1 に示す.

図中の頂点のうち灰色のものは独立集合に含まれる頂点とする. 図中の辺を定式化すると以下ようになる.

- $\forall i((a_i^1, b_i^1), (a_i^2, b_i^2)) \in E$

このグラフが「 G 中に与えられている独立集合が, $DISJ_{N \times N}(x, y) = 1$ のときのみ 2-MIS でない」という特性 (*) を持つように, G_A に構造 H_A, G_B に構造 H_B を追加する. (なお, $DISJ_{N \times N}(x, y) := \bigvee_{i=1}^N \bigvee_{j=1}^N x_{i,j} \wedge y_{i,j}$ で定義される.) H_A と H_B の中身は以下のとおりである.

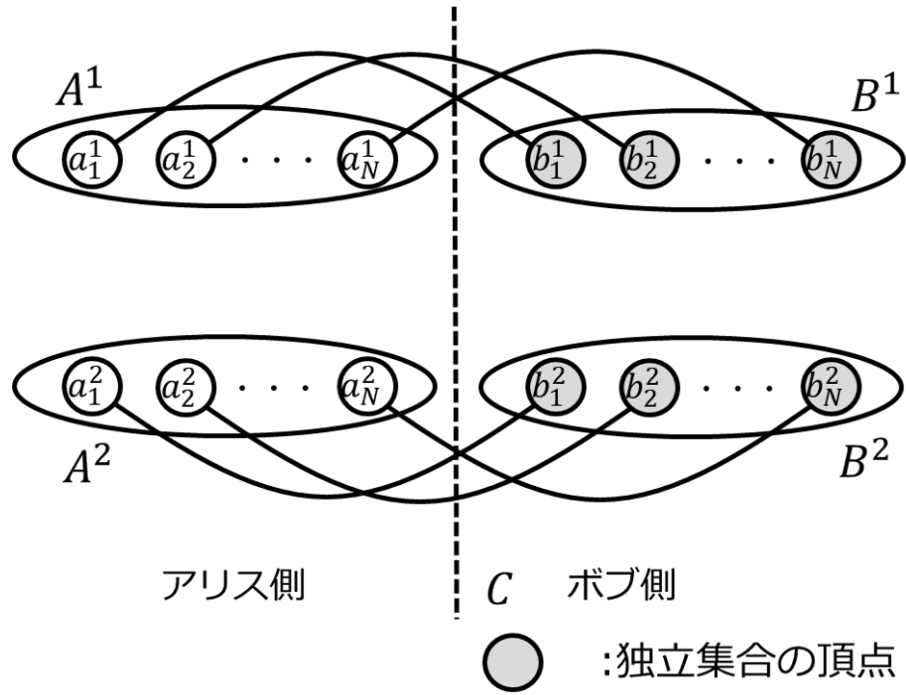


図 4.1: $G = (V, E)$

- $H_A: (a_i^1, a_j^2) \in E_A \Leftrightarrow x_{i,j} = 0$
- $H_B: W(y)$ 頂点のクリーク $K_{W(y)}(W(y))$ は 0/1 のデータ列 y 中に含まれる 1 の個数
を表す.) $K_{W(y)}$ 中の頂点 $c_{i,j} \in V_B$ は $y_{i,j} = 1$ であるような (i, j) でインデックスづ
けされるものとする.

このとき, $(c_{i,j}, b_i^1) \in E_B$ かつ $(c_{i,j}, b_j^2) \in E_B$

$G = (V, E)$ に辺と頂点を追加したグラフ $G^{x,y} = (V', E')$ を図 4.2 に示す.

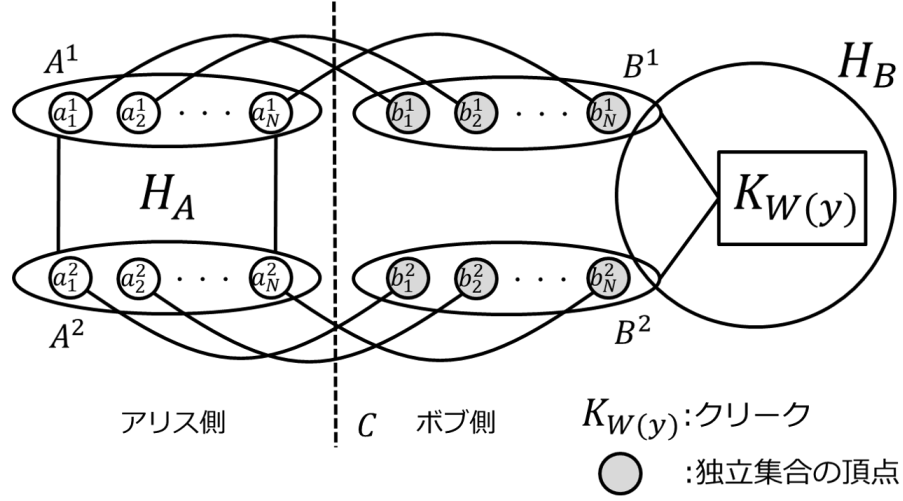
このグラフ $G^{x,y} = (V', E')$ が上記の特性 (*) を満たしていることを示すために, 次の 2
点を確認する.

(i) $DISJ_{N \times N}(x, y) = 1$ のとき, グラフに与えられている独立集合が 2-MIS でない:

$x_{i,j} = y_{i,j} = 1$ とすると, b_i^1 と b_j^2 の 2 点を取り除いて $a_i^1, a_j^2, c_{i,j}$ の 3 点を追加できることか
ら確認できる.

(ii) $DISJ_{N \times N}(x, y) = 0$ のとき, グラフに与えられている独立集合が 2-MIS である:

グラフに与えられている独立集合が 2-MIS でないと仮定する. このとき, ある 2 点を取り
除くことで独立集合に追加できる 3 点が存在する. 2 点の取り除き方は (1) b_i^1 と $b_j^1 (i \neq j)$,
(2) b_i^2 と $b_j^2 (i \neq j)$, (3) b_i^1 と b_j^2 が考えられる. (1) では a_i^1 と a_j^1 の 2 点しか追加できる可能性

図 4.2: $G^{x,y} = (V', E')$

がなく, (2) では a_i^2 と a_j^2 の 2 点しか追加できる可能性がない. (3) において, b_i^1 を取り除いて a_i^1 を追加し, b_j^2 を取り除いて a_j^2 を追加し, さらに $c_{i,j}$ を追加することを考える. a_i^1 と a_j^2 が両方とも追加できるのは $x_{i,j} = 1$ のときのみであり, $c_{i,j}$ が追加できる ($c_{i,j}$ が存在する) のは $y_{i,j} = 1$ のときのみであるが, これは $DISJ_{N \times N}(x, y) = 0$ に矛盾する. したがってグラフに与えられている独立集合から 2 点取り除いて 3 点追加することはできないため, この独立集合は 2-MIS である.

今回, $N \times N$ ビットの交叉判定インスタンスをグラフに埋め込んでおり, カット辺のサイズ $|C| = 2N$ であることがわかる. *CONGEST* モデルにおいてグラフ上に与えられた独立集合が 2-MIS であるかどうかを r ラウンドで判定するアルゴリズム \mathcal{A} が存在したとすると, アリスとボブは $O(r \cdot |C| \cdot \log n)$ ビット通信したことになる. このグラフにおいてアルゴリズム \mathcal{A} を実行すると同時に 2 者間交叉判定問題も解けていることになるので, 交叉判定問題の通信複雑性よりアリスとボブは少なくとも $\Omega(N \times N)$ ビットは通信しているはずである. よって, $r = \Omega(N/2 \log n) = \tilde{\Omega}(N)$ ラウンドの下界を得ることができる. 図 4.2 からわかる通り, A^1, A^2, B^1, B^2 はそれぞれ N 頂点で構成されており, $K_{W(y)}$ の頂点数は $O(N^2)$ であるため, グラフ全体の頂点数 n は $n = O(N + N^2)$ である. したがって $N = O(\sqrt{n})$ になるため, $\tilde{\Omega}(\sqrt{n})$ ラウンドの下界を得ることができる.

4.2 3-MIS 検証問題の下界

この節では 3-MIS 検証問題の下界についての議論を行う. 具体的には, 次の定理を証明する.

定理 4.2. *CONGEST* モデルにおいて, 3-MIS 検証問題を解く全てのアルゴリズムは $\tilde{\Omega}(n)$ の通信ラウンド数を必要とする.

証明. まず初めにアリスとボブが構築するグラフ $G = (V, E)$ を図 4.3 に示す.

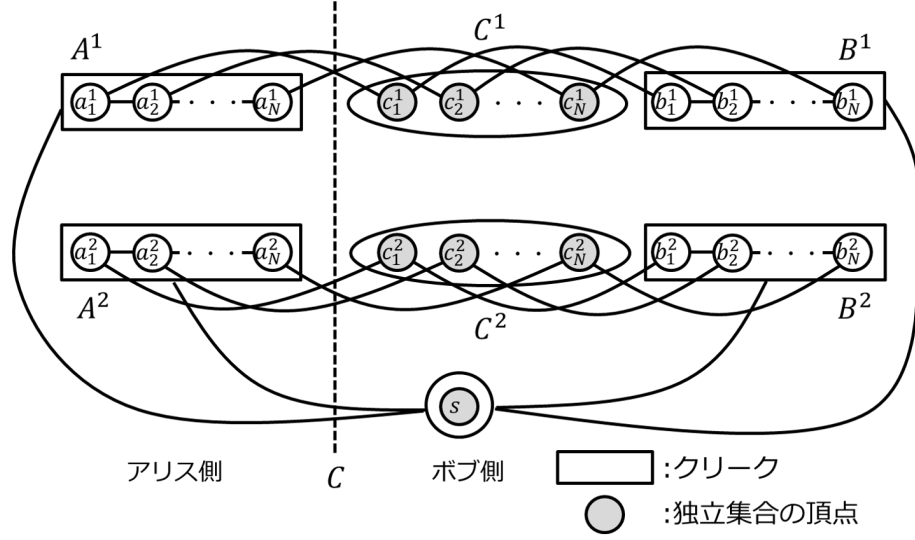


図 4.3: $G = (V, E)$

図中の頂点のうち灰色のものは独立集合に含まれる頂点とする. また, 四角で囲まれている部分はクリークを表す. 図中の辺を定式化すると以下ようになる.

- A^1, A^2, B^1, B^2 は N 頂点のクリーク K_N
- $\forall i((a_i^1, c_i^1), (a_i^2, c_i^2), (b_i^1, c_i^1), (b_i^2, c_i^2)) \in E$
- $\forall i((a_i^1, s), (a_i^2, s), (b_i^1, s), (b_i^2, s)) \in E$

このグラフが「 G 中に与えられている独立集合が, $DISJ_{N \times N}(x, y) = 1$ のときのみ 3-MIS でない」という特性(*)を持つように, G_A に構造 H_A , G_B に構造 H_B を追加する. H_A と H_B の中身は以下のとおりである.

- $H_A: (a_i^1, a_j^2) \in E_A \Leftrightarrow x_{i,j} = 0$
- $H_B: (b_i^1, b_j^2) \in E_B \Leftrightarrow y_{i,j} = 0$

$G = (V, E)$ に辺を追加したグラフ $G^{x,y} = (V, E')$ を図 4.4 に示す.

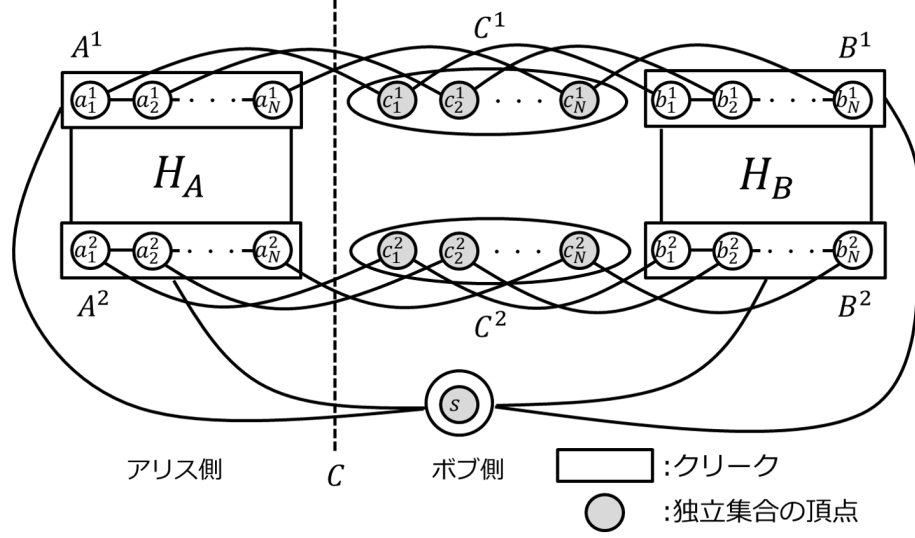


図 4.4: $G^{x,y} = (V, E')$

このグラフ $G^{x,y} = (V, E')$ が上記の特性 (*) を満たしていることを示すために, 次の 2 点を確認する.

(i) $DISJ_{N \times N}(x, y) = 1$ のとき, グラフに与えられている独立集合が 3-MIS でない:

$x_{i,j} = y_{i,j} = 1$ とすると, s と c_i^1 と c_j^2 の 3 点を取り除いて $a_i^1, b_i^1, a_j^2, c_j^2$ の 4 点を追加できることから確認できる.

(ii) $DISJ_{N \times N}(x, y) = 0$ のとき, グラフに与えられている独立集合が 3-MIS である:

グラフに与えられている独立集合が 3-MIS でないと仮定する. このとき, ある 3 点を取り除くことで独立集合に追加できる 4 点が存在するが, A^1, A^2, B^1, B^2 がそれぞれクリークであるため, 4 点を追加するためにはそれぞれから 1 点を選ぶ必要がある. c_i^1 を取り除いて a_i^1 と b_i^1 を追加し, c_j^2 を取り除いて a_j^2 と b_j^2 を独立集合に追加したとする. a_i^1 と a_j^2 が両方とも追加できるのは $x_{i,j} = 1$ のときのみであり, b_i^1 と b_j^2 が両方とも追加できるのは $y_{i,j} = 1$ のときのみであるが, これは $DISJ_{N \times N}(x, y) = 0$ に矛盾する. したがってグラフに与えられている独立集合から 3 点取り除いて 4 点追加することはできないため, この独立集合は 3-MIS である.

今回, $N \times N$ ビットの交叉判定インスタンスをグラフに埋め込んでおり, カット辺のサイズ $|C| = 4N$ であることがわかる. *CONGEST* モデルにおいてグラフ上に与えられた独立集合が 3-MIS であるかどうかを r ラウンドで判定するアルゴリズム \mathcal{A} が存在したとすると, アリスとボブは $O(r \cdot |C| \cdot \log n)$ ビット通信したことになる. このグラフにおいてアルゴリズム \mathcal{A} を実行すると同時に 2 者間交叉判定問題も解けていることになるので, 交叉判定問題の通信複雑性よりアリスとボブは少なくとも $\Omega(N \times N)$ ビットは通信しているはずである. よって, $r = \Omega(N/4 \log n) = \tilde{\Omega}(N)$ ラウンドの下界を得ることができる. 図 4.4 からわかる通り, $A^1, A^2, B^1, B^2, C^1, C^2$ はそれぞれ N 頂点で構成されているため, グラフ全体の頂点数 n は $n = O(N)$ である. したがって $N = O(n)$ になるため, $\tilde{\Omega}(n)$ ラウンドの下界を得ることができる.

4.3 k -MIS 検証問題の下界

このセクションでは k -MIS 検証問題の下界についての議論を行う. 具体的には, 次の定理を証明する.

定理 4.3. *CONGEST* モデルにおいて, 任意の $l \geq 1$ に対して $k = 4l + 5$ としたとき k -MIS 検証問題を解く全てのアルゴリズムは $\Omega\left(n^{2-\frac{1}{l+1}}\right)$ の通信ラウンド数を必要とする.

証明. 最初に $l = 1$ の場合を考える.

まず初めにアリスとボブが構築するグラフ $G = (V, E)$ を図 4.5 に示す.

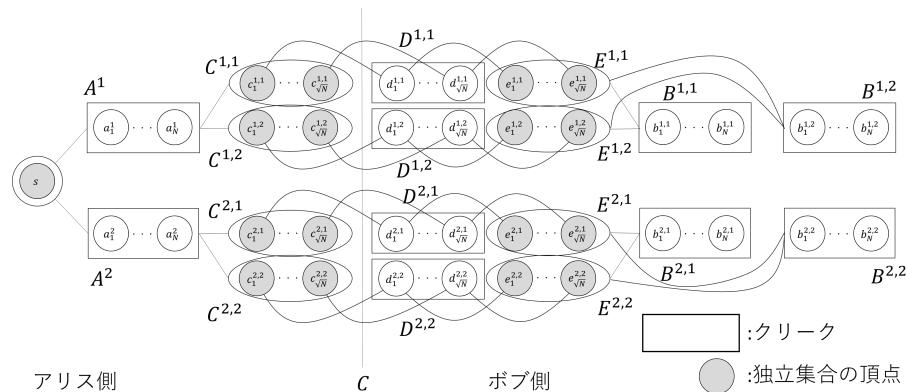


図 4.5: $G = (V, E)$

図中の頂点のうち灰色のものは独立集合に含まれる頂点とする. また, 四角で囲まれている部分はクリークを表す. 図中の辺を定式化すると以下ようになる.

- $A^1, A^2, B^{1,1}, B^{1,2}, B^{2,1}, B^{2,2}$ は N 頂点のクリーク K_N
- $D^{1,1}, D^{1,2}, D^{2,1}, D^{2,2}$ は \sqrt{N} 頂点のクリーク $K_{\sqrt{N}}$
- $\forall i((a_i^1, s), (a_i^2, s)) \in E$
- $\forall i, j((a_k^1, c_i^{1,1}), (a_k^1, c_j^{1,2}), (a_k^2, c_i^{2,1}), (a_k^2, c_j^{2,2}), \in E$ (ただし $k = (i - 1) \times \sqrt{N} + j$)
- $\forall i, j((b_k^{1,1}, e_i^{1,1}), (b_k^{1,1}, e_j^{1,2}), (b_k^{2,1}, e_i^{2,1}), (b_k^{2,1}, e_j^{2,2}), \in E$ (ただし $k = (i - 1) \times \sqrt{N} + j$)
- $\forall i, j((b_k^{1,2}, e_i^{1,1}), (b_k^{1,2}, e_j^{1,2}), (b_k^{2,2}, e_i^{2,1}), (b_k^{2,2}, e_j^{2,2}), \in E$ (ただし $k = (i - 1) \times \sqrt{N} + j$)
- $\forall i((c_i^{1,1}, d_i^{1,1}), (c_i^{1,2}, d_i^{1,2}), (e_i^{1,1}, d_i^{1,1}), (e_i^{1,2}, d_i^{1,2})) \in E$

- $\forall i((c_i^{2,1}, d_i^{2,1}), (c_i^{2,2}, d_i^{2,2}), (e_i^{2,1}, d_i^{2,1}), (e_i^{2,2}, d_i^{2,2})) \in E$

このグラフが「 G 中に与えられている独立集合が, $DISJ_{N \times N}(x, y) = 1$ のときのみ $k(= 4 \times 1 + 5 = 9)$ -MIS でない」という特性 (*) を持つように, G_A に構造 H_A, G_B に構造 H_B を追加する. H_A と H_B の中身は以下のとおりである.

- $H_A: (a_i^1, a_j^2) \in E_A \Leftrightarrow x_{i,j} = 0$
- $H_B: (b_i^{1,1}, b_j^{2,1}) \in E_B \Leftrightarrow y_{i,j} = 0, (b_i^{1,2}, b_j^{2,2}) \in E_B \Leftrightarrow y_{i,j} = 0$

$G = (V, E)$ に辺を追加したグラフ $G^{x,y} = (V, E')$ を図 4.6 に示す.

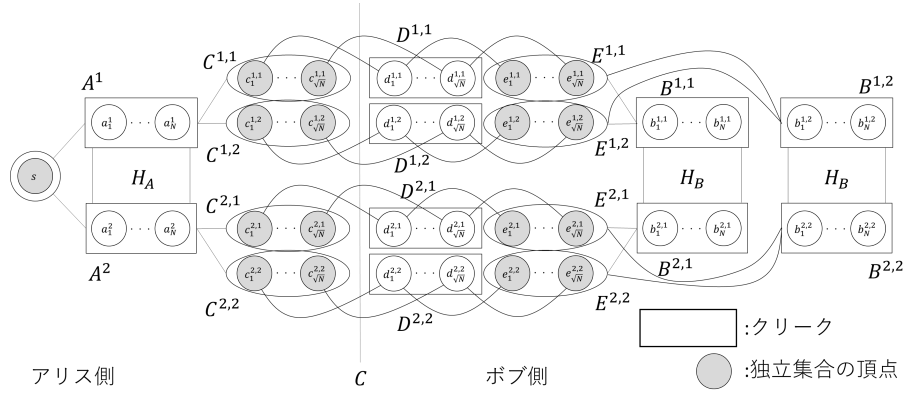


図 4.6: $G^{x,y} = (V, E')$

このグラフ $G^{x,y} = (V, E')$ が上記の特性 (*) を満たしていることを示すために, 次の 2 点を確認する.

(i) $DISJ_{N \times N}(x, y) = 1$ のとき, グラフに与えられている独立集合が $k(= 9)$ -MIS でない:
 $x_{i,j} = y_{i,j} = 1$ とすると, $s, c_k^{1,1}, c_l^{1,2} (i = (k-1) \times \sqrt{N} + l), c_p^{2,1}, c_q^{2,2} ((j = (p-1) \times \sqrt{N} + q), e_k^{1,1}, e_l^{1,2}, e_p^{2,1}, e_q^{2,2})$ の 9 点を取り除いて $a_i^1, a_j^2, b_i^{1,1}, b_j^{2,1}, b_i^{1,2}, b_j^{2,2}, d_k^{1,1}, d_l^{1,2}, d_p^{2,1}, d_q^{2,2}$ の 10 点を追加できることから確認できる.

(ii) $DISJ_{N \times N}(x, y) = 0$ のとき, グラフに与えられている独立集合が $k(= 9)$ -MIS である:
 グラフに与えられている独立集合が 9-MIS でないと仮定する. このとき, ある 9 点を取り除くことで独立集合に追加できる 10 点が存在するが,
 $A^1, A^2, B^{1,1}, B^{2,1}, B^{1,2}, B^{2,2}, D^{1,1}, D^{1,2}, D^{2,1}, D^{2,2}$ がそれぞれクリークであるため, 10 点を追加するためにはそれぞれから 1 点を選ぶ必要がある. $d_k^{1,1}$ を追加するために $c_k^{1,1}$ と $e_k^{1,1}$ を, $d_l^{1,2}$ を追加するために $c_l^{1,2}$ と $e_l^{1,2}$ を, $d_p^{2,1}$ を追加するために $c_p^{2,1}$ と $e_p^{2,1}$ を, $d_q^{2,2}$ を追加する

ために $c_q^{2,2}$ と $e_q^{2,2}$ を独立集合から取り除くとする. このとき, $i = (k-1) \times \sqrt{N} + l$ を満たす i に対して $b_i^{1,1}$ と $b_i^{1,2}$ を, $j = (p-1) \times \sqrt{N} + q$ を満たす j に対して $b_j^{2,1}$ と $b_j^{2,2}$ を追加できる可能性がある. A^1 と A^2 の頂点も独立集合に追加できるように s も取り除くと, 同じ i, j に対して a_i^1 と a_j^2 を追加できる可能性をもたせることができる. しかし, a_i^1 と a_j^2 が両方とも追加できるのは $x_{i,j} = 1$ のときのみであり, $b_i^{1,1}$ と $b_j^{1,2}$ (もしくは $b_i^{2,1}$ と $b_j^{2,2}$) が両方とも追加できるのは $y_{i,j} = 1$ のときのみであるが, これは $DISJ_{N \times N}(x, y) = 0$ に矛盾する. したがってグラフに与えられている独立集合から 9 点取り除いて 10 点追加することはできないため, この独立集合は 9-MIS である.

今回, $N \times N$ ビットの交叉判定インスタンスをグラフに埋め込んでおり, カット辺のサイズ $|C| = 4\sqrt{N}$ であることがわかる. *CONGEST* モデルにおいてグラフ上に与えられた独立集合が $k(=9)$ -MIS であるかどうかを r ラウンドで判定するアルゴリズム A が存在したとすると, アリスとボブは $O(r \cdot |C| \cdot \log n)$ ビット通信したことになる. このグラフにおいてアルゴリズム A を実行すると同時に 2 者間交叉判定問題も解けていることになるので, 交叉判定問題の通信複雑性よりアリスとボブは少なくとも $\Omega(N \times N)$ ビットは通信しているはずである. よって, $r = \Omega(N^{3/2}/4 \log n) = \tilde{\Omega}(N^{3/2})$ ラウンドの下界を得ることができる. 図 4.6 からわかる通り, $A^1, A^2, B^{1,1}, B^{2,1}, B^{1,2}, B^{2,2}$ はそれぞれ N 頂点で, C, D, E の頂点集合はそれぞれ \sqrt{N} 頂点で構成されているため, グラフ全体の頂点数 n は $n = O(N)$ である. したがって $N = O(n)$ になるため, $\tilde{\Omega}(n^{3/2})$ ラウンドの下界を得ることができる.

$l \geq 2$ については, 図 4.6 を図 4.7 のように拡張する.

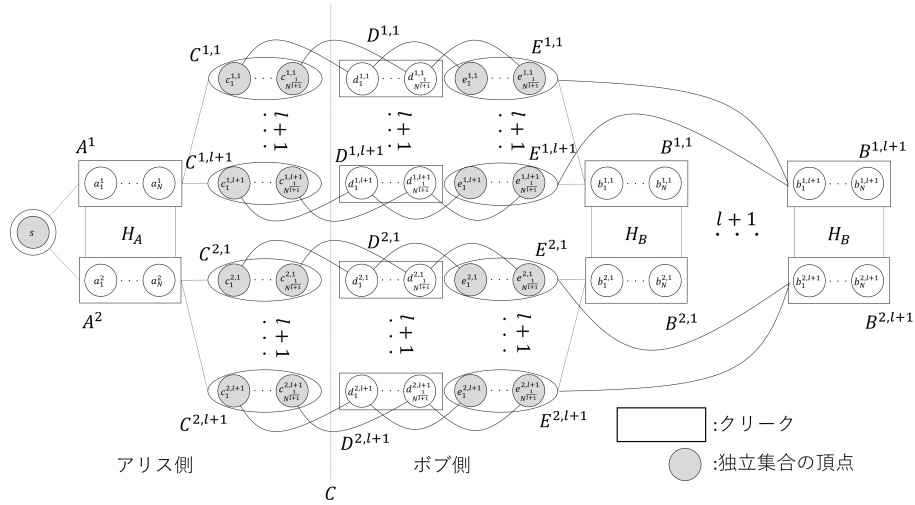


図 4.7: $l \geq 2$ における $G^{x,y} = (V, E')$

$4(l+1)+1 = 4l+5$ 頂点を取り除くことで $4(l+1)+2 = 4l+6$ 頂点を追加できるかどうか, つまりのグラフに与えられている独立集合が $k(= 4l+5)$ -MIS であるかどうかの議論を上記と同様に行うことができる. このときカット辺のサイズは $|C| = 2(l+1) \cdot N^{1/(l+1)}$ であることがわかるため, $r = \Omega\left(N^{2-\frac{1}{l+1}}/2(l+1)\log n\right) = \tilde{\Omega}\left(N^{2-\frac{1}{l+1}}/l\right)$ ラウンドの下界を得ることができる. A, B の頂点集合はそれぞれ N 頂点で, C, D, E の頂点集合はそれぞれ $N^{1/(l+1)}$ 頂点で構成されているため, グラフ全体の頂点数 n は $n = O(N)$ である. したがって $N = O(n)$ になるため, $\tilde{\Omega}\left(n^{2-\frac{1}{l+1}}/l\right)$ ラウンドの下界を得ることができる.

第5章

まとめと今後の課題

5.1 まとめ

本研究では極大独立集合検証問題に対するいくつかの複雑性を示した. 具体的には, 1-MIS 検証問題に対する $O(1)$ ラウンドの上界, 2-MIS 検証問題に対する $\tilde{\Omega}(\sqrt{n})$ ラウンドの下界, 3-MIS 検証問題に対する $\tilde{\Omega}(n)$ ラウンドの下界, k -MIS 検証問題 ($k = 4l + 5, l \geq 1$) に対する $\tilde{\Omega}\left(n^{2-\frac{1}{l+1}}/l\right)$ ラウンドの下界を証明した.

5.2 今後の課題

4.3 節で一般の k に対する k -MIS 検証問題の下界を証明したが, $k = 4, \dots, 8$ については現在 3-MIS 検証問題と同じ下界しか得られていない. この下界をよりタイトにできるかが今後の課題である.

謝辞

本研究の機会を与え、数々の御指導を賜りました泉泰介准教授に深く感謝致します。また、本研究を進めるにあたり多くの助言を頂き、様々な御協力を頂きました泉研究室の学生みなさんに深く感謝致します。

参考文献

- [1] Ken-ichi Kawarabayashi, Seri Khoury, Aaron Schild, and Gregory Schwartzman. Improved distributed approximation to maximum independent set. *arXiv preprint arXiv:1906.11524*, 2019.
- [2] Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and near-quadratic lower bounds for the congest model. *arXiv preprint arXiv:1705.05646*, 2017.
- [3] Yuval Efron, Ofer Grossman, and Seri Khoury. Beyond alice and bob: Improved inapproximability for maximum independent set in congest. *arXiv preprint arXiv:2003.07427*, 2020.
- [4] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM Journal on Computing*, 41(5):1235–1265, 2012.
- [5] Orr Fischer, Tzlil Gonen, Fabian Kuhn, and Rotem Oshman. Possibilities and impossibilities for distributed subgraph detection. In *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*, pages 153–162, 2018.
- [6] Artur Czumaj and Christian Konrad. Detecting cliques in congest networks. *Distributed Computing*, 33(6):533–543, 2020.
- [7] Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.