Title	多項式時間グラフ再構築問題に関する研究
Author(s)	菅原，祐介
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/4346
Rights	
Description	Supervisor:上原　隆平，情報科学研究科，修士

修 士 論 文

多項式時間グラフ再構築問題に関する研究

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

菅原 祐介

2008 年 3 月

修 士 論 文

多項式時間グラフ再構築問題に関する研究

指導教官　上原隆平 准教授

審査委員主査　上原隆平 准教授
審査委員　浅野哲夫 教授
審査委員　金子峰雄 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

0610047 菅原 祐介

提出年月: 2008 年 2 月

概 要

「グラフの再構築予想」とは，n 頂点のグラフ G^* から 1 頂点ずつ取り除かれた n 個のグラフ（$Deck$) が与えられたときに，グラフ G^* が一意に決定されるという予想である．グラフの再構築予想は，数十年解かれていないよく知られた未解決問題である．グラフの再構築予想と与えられた 2 つのグラフが同型かどうか判定するグラフの同型性判定問題の複雑さについては，深い関係があることが知られている．本研究ではまず，グラフ再構築問題を定義し，C を遺伝的でかつ同型性判定問題が多項式時間で解くことができるグラフのクラスとするとき，C に属する任意の非連結グラフ G^* のグラフ再構築問題は，C に属する連結なグラフの再構築問題に多項式時間で帰着できることを示した．また，proper interval graph の $O(n^3m)$ 時間再構築アルゴリズムと quasi threshold graph の $O(nm)$ 時間再構築アルゴリズムを提案した．ここで n は頂点数，m は枝数である．

目 次

第1章　序章

1.1　背景と目的

計算機で扱う多くの問題は，グラフ構造でモデル化することができる．こうした問題を効率よく解くには，グラフ理論とアルゴリズム理論がともに重要な役割を果たす．

与えられた2つのグラフが同型であるかどうかを判定する「グラフ同型性判定問題」は，こうしたアルゴリズム理論とグラフ理論に深く関係する問題の中でも，もっとも基本的なものの1つである．しかし，この問題を効率よく解くアルゴリズムが存在するかどうかは，数十年間解かれていない．

グラフの同型性判定問題と同様，数十年解かれていない未解決問題に，**「グラフの再構築予想」** がある．まず，グラフの再構築予想に必要な言葉の定義をする．

定義 1.1 n 頂点 $v_1, v_2, ..., v_n$ からなるグラフ G^* を考える．ここで $i = 1, 2, ..., n$ に対して，G^* から v_i と v_i に隣接する辺を取り除いて得られるグラフを G_i とする．これら n 個の G_i から構成される集合を G^* の $Deck$ と呼ぶ．

グラフ再構築予想とは以下のような予想である．
[**グラフの再構築予想**]
頂点にラベルがついていない $(n-1)$ 頂点からなるグラフが n 個与えられた時，これらを $Deck$ として持つような n 頂点のグラフは高々1つである．

この再構築予想は，$n > 2$ のときに例外なく成立すると予想されているが，ごく簡単な場合にしかその正しさは証明されていない [1].

一方で，グラフの再構築予想と，グラフの同型性判定問題の複雑さについては，深い関係があることが知られている [2]．そこで，グラフの再構築を実際に行うアルゴリズムと，その計算量や正当性を研究することは，グラフの再構築予想が成立するグラフクラスの拡張や，グラフの同型性判定問題の複雑さの解析にもつながる研究であると考えられる．

本論文で扱うグラフ再構築問題を以下のように定義する．

定義 1.2 グラフ再構築問題
入力：頂点数 $(n-1)$ の n 個のグラフ　　$G_1, G_2, ..., G_n$
出力：$\{G_1, G_2, ..., G_n\}$ を $Deck$ とする頂点数 n のグラフ G^*.
　　　ただし，$\{G_1, G_2, ..., G_n\}$ を $Deck$ として持つグラフが存在しないときは No を返す.

本論文では，与えられた n 個のグラフを $Deck$ として持つグラフ G^* を再構築するアルゴリズムを提案する．より正確には，本論文の再構築アルゴリズムでは，与えられた $Deck$ に対応する G^* が存在しなければそれを判定し，存在すれば G^* を 1 つ出力する．出力される G^* の一意性は本研究では保障されないことに注意する．

与えられた $Deck$ に対応する G^* を見つけるアルゴリズムは次のように構成することができる．

- $Deck$ の中の G_1 を適当に選ぶ
- 頂点 v と G_1 の $n-1$ 個の頂点の結び方 2^{n-1} 通りのそれぞれについて対応する G^+ を作る
  - G^+ の $Deck$ が入力の $Deck$ と一致するかどうかを同型性を判定するアルゴリズムを繰り返し用いてチェックする．
- $Deck$ に合致する G^+ があれば G^+ を G^* として出力し，合致するものがなければ No を出力する．

しかし，上記のアルゴリズムは同型性判定を入力サイズに対して指数関数回行う必要がある．本研究では，グラフクラスを限定し，その上で与えられた $Deck$ からもとのグラフを再構築する多項式時間アルゴリズムを研究する．

あるグラフクラスに対して，多項式時間で $Deck$ からグラフ G^* を構築するアルゴリズムが存在するのであれば，そのグラフクラスに対する再構築予想の証明も相対的に簡単であることが予想される．また，こうしたアルゴリズムの解析を通じて，再構築予想に対する知見を得られることが期待できる．これまではグラフの再構築予想は，成立するかどうか，という静的でかつおおまかな指標しか与えることができなかったが，多項式時間計算可能性という新しい指標を与えることで，再構築予想の困難さの階層構造を提案することができる．

本研究では，遺伝的でかつ属する 2 つのグラフの同型性判定がそれらのサイズの多項式時間でできるグラフクラスを対象とする．ただし，グラフクラス C が遺伝的であるとは，C に属する 2 頂点以上からなる任意のグラフの $Deck$ 中のグラフはすべて C に属することと定義する．遺伝的で同型性判定問題が多項式時間で判定できるグラフクラスとして代表的なものに interval graph[4] や cograph[7]，distance-hereditary graph[8] があげられる．

本研究では，まず，C を遺伝的でかつ同型性判定問題が多項式時間で解くことができるグラフのクラスとするとき，C に属する任意の非連結グラフのグラフ再構築問題は，C に属する連結なグラフの再構築問題に多項式時間で帰着できることを示した．

本研究では，具体的なグラフクラスとして **proper interval graph** と **quasi threshold graph** を取り上げる．proper interval graph は **interval graph** のサブクラスであり quasi threshold graph は **cograph** のサブクラスであり，cograph は **distance-hereditary**

graph のサブクラスである．また，本論文で取り上げたこれらのグラフクラスは遺伝的である．これらのグラフクラスに対して proper interval graph の $O(n^3m)$ 時間再構築アルゴリズムと quasi threshold graph の $O(nm)$ 時間再構築アルゴリズムを示した．

1.2　本論文の構成

本論文では，2 章をグラフの基本的な用語の定義，3 章を非連結グラフの再構築問題，4 章を **proper interval graph** の再構築アルゴリズム，5 章を **quasi threshold graph** の再構築アルゴリズムとし，6 章をまとめとする．

第2章　準備

2.1　基本用語

グラフに関する基本的な用語を説明する．

- グラフ
  グラフ $G = (V, E)$ とは頂点の有限集合 V と V の 2 元部分集合である辺の集合 E から成る．グラフ $G = (V, E)$ が与えられたとき，V の部分集合 U と辺集合 $E' = \{\{u, v\} \in E \mid u \in U$ かつ $v \in U\}$ から構成されるグラフ $G[U] = (U, E')$ を U による G の**誘導部分グラフ**という．

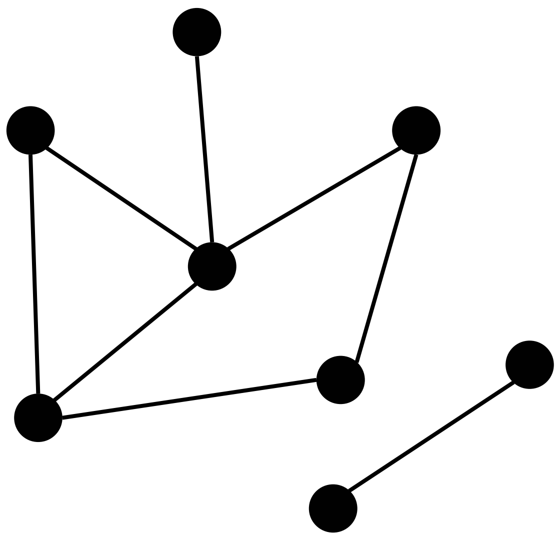

図 2.1: グラフの例

- 隣接点と頂点の次数と完全グラフ
  グラフ $G = (V, E)$ に対して，ある辺 $e = \{x, y\} \in E$ が存在するとき，G の 2 つの頂点 x, y は**隣接する**という．このとき辺 e は頂点 x 及び頂点 y に**接続している**という．$G = (V, E)$ における頂点 v の隣接点集合は $N_G(v) = \{u \in V \mid \exists v \in V$ ただし $\{u, v\} \in E\}$ と表す．頂点 v の**次数** $\deg_G(v)$ とは，頂点 v に接続している辺の数である．また，グラフ G のすべての頂点が互いに隣接しているとき，G は**完全グラフである**という．

- 孤立点と頂点の追加と挿入と削除
  グラフ $G = (V, E)$ において，V 中のどの頂点にも接続していない次数 0 の頂点を**孤立点**という．本論文において，グラフ G に孤立点を加えることを**頂点の追加**という．また，G に頂点を追加し，追加した頂点から G の頂点への辺をいくつか張ることを**頂点の挿入**という．さらに，頂点とその頂点に接続しているすべての辺を削除することを**頂点の削除**という．

- パスと 2 頂点間の距離
  グラフ $G = (V, E)$ において，$i = 1, 2, ..., l$ に対して $(v_{i-1}, v_i) \in E$ であるとき，頂点の列 $[v_0, v_1, ..., v_l]$ は v_0 から v_l への長さ l の**パス**であるという．2 頂点 u, v に対して，u, v 間の**距離**は u から v への最短のパスの長さであり，$d(u, v)$ と表記される．

- 連結グラフと連結成分
  無向グラフ $G = (V, E)$ に対して，V の任意の 2 つの頂点を結ぶパスが存在するとき，グラフ G は**連結である**という．また，グラフ G の極大な連結部分グラフを G の**連結成分**と呼ぶ．図 2.2 は連結成分の例を示す．

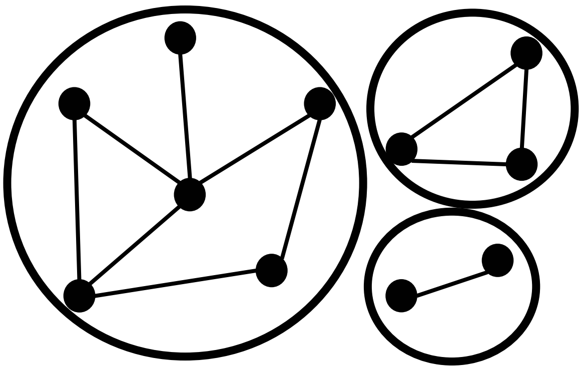

図 2.2: 連結成分

- グラフの同型性
  $G=(V,E)$ と $G'=(V',E')$ を 2 つのグラフとする．その頂点集合の間の一対一対応 $\varphi : V \to V'$ が存在し，すべての $x,y \in V$ に対して $\{x,y\} \in E \Leftrightarrow \{\varphi(x),\varphi(y)\} \in E'$ を満たすとき，G と G' は**同型である**といい，$G \cong G'$ と表記する．また，この写像 φ を G，G' の**同型写像**と呼ぶ．図 2.3 において，(a) と (b) は同型なグラフである．しかし，(a)

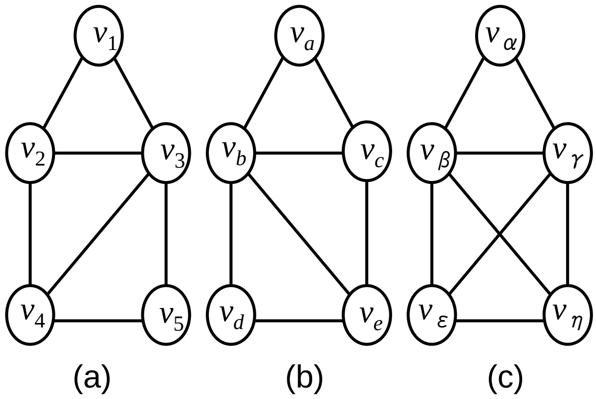

図 2.3: 同型なグラフと同型でないグラフの例

- クリーク
  グラフ $G=(V,E)$ において，頂点集合 $W \subset V$ に属する任意の 2 つの頂点が隣接するとき，W は**クリーク**であるという．また，G のどのクリークも W を真に含まないとき，W を**極大クリーク**と呼ぶ．

- **interval graph** と区間表現
  グラフ $G=(V,E)$ の**区間表現**とは，数直線上の区間の集合 I であり，以下の条件を満たすものである．

  - V の各頂点は I のある区間と対応する
  - G において頂点が隣接するための必要十分条件は，対応する区間同士が重なりを持つことである．

  区間表現を持つグラフを **interval graph** という．また，**interval graph** のクラスは，**chordal graph** のクラスに真に含まれる [9]．図 2.4 において，(a) は **interval graph** を表し，(b) は (a) の **interval graph** の区間表現を表している．

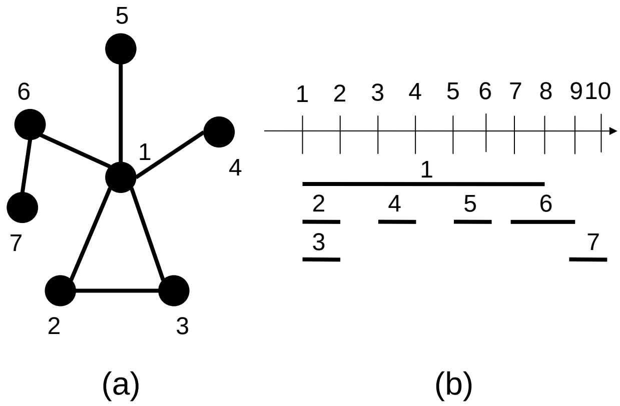

図 2.4: interval graph と対応する区間表現

- **distance-hereditary graph**
  グラフ $G = (V, E)$ が与えられ，U を V の部分集合とする．G の誘導連結部分グラフ $G[U]$ における任意の 2 頂点間の距離 $d(u, v)$ が，G における $d(u, v)$ と同じとき，$G[U]$ は **isometric** であるという．任意の誘導部分グラフが isometric なグラフを **distance-hereditary graph** という．

2.2 proper interval graph

proper interval graph は以下の定義で与えられる．

定義 2.1 **interval graph** $G = (V, E)$ が次の性質を満たすとき，G は **proper interval graph** であるという．

G の適当な区間表現 $\{[l_1, r_1], [l_2, r_2], ..., [l_n, r_n]\}$ に対して，$l_{s(1)} \leq l_{s(2)} \leq ... \leq l_{s(n)}$,
$r_{s(1)} \leq r_{s(2)} \leq ... \leq r_{s(n)}$
を満たす $(1, 2, ..., n)$ の置換 s が存在する．(図 2.5 参照)

また，このような区間表現を G の proper **な区間表現**と呼ぶ．

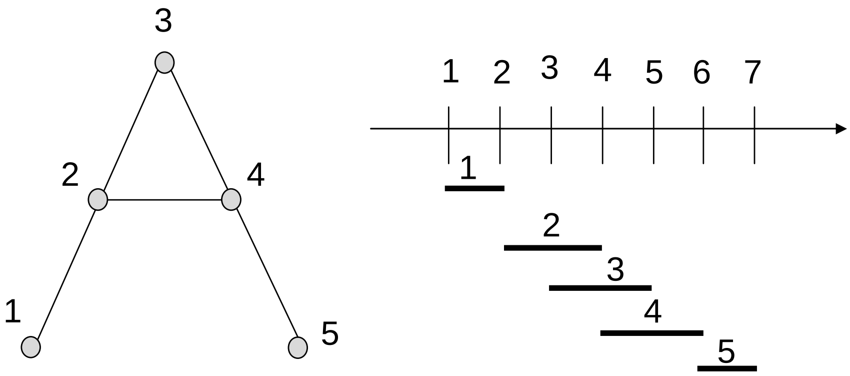

図 2.5: proper interval graph と対応する区間表現

また，本論文では，proper interval graph に関して，頂点の挿入，区間の挿入について以下の定義を与える．

定義 2.2 *proper interval graph* $G = (V, E)$ と G の *proper* な区間表現 $\{I_i\}_{i=i,\dots,|V|}$ が与えられ区間 I_v を $\{I_i\}$ に加えた区間の集合がある *proper interval graph* G‘ の *proper* な区間表現になっているとき，区間 I_v を $\{I_i\}$ に加えることを**区間** I_v **の** G **への挿入**という．このとき，得られた *proper interval graph* G' は I_v に対応する頂点 v を G に**挿入**して得られたグラフという．

また，**proper interval graph** に関して，以下の定理が成立する．

定理 2.3 proper interval graph から任意の 1 頂点を削除して得られるグラフは **proper interval graph** である．

証明 区間表現上で考える．ある区間はある頂点と対応しているので，明らかに，**proper interval graph** は 1 頂点取り除いても **proper interval graph** の性質を失わない．よって，**proper interval graph** から 1 頂点削除したグラフは **proper interval graph** である．

（証明終）

2.3 quasi threshold graph

quasi threshold graph は以下の定義で与えられる．

定義 2.4 **quasi threshold graph** は以下のように再帰的に定義される．

1. K_1 は **quasi threshold graph** である．（操作 1）
2. **quasi threshold graph** G に G のすべての頂点と隣接する新しい頂点を 1 つ挿入して得られるグラフは **quasi threshold graph** である．（操作 2）
3. 2 つのグラフ $G = (V, E)$，$G = (V', E')$ が **quasi threshold graph** であるとき，グラフ $(V \cup V', E \cup E')$ は **quasi threshold graph** である．（操作 3）

quasi threshold graph については，以下の性質が知られている．

定理 2.5 **quasi threshold graph** については，以下の包含関係が成立する *[10]*.
quasi threshold graphs $\subseteq$ *interval graphs*
quasi threshold graphs $\subseteq$ *distance-hereditary graphs*

quasi threshold graph の例を図 2.6 に示す．

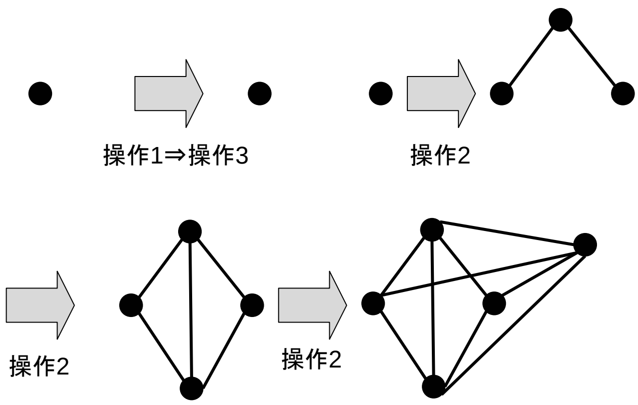

図 2.6: quasi threshold graph の例

2.4　グラフの同型性判定問題

グラフの同型性判定問題は以下のように定義される．

定義 2.6 グラフの同型性判定問題
入力：グラフ G_1 とグラフ G_2
質問：グラフ G_1 と G_2 は同型か？

一般のグラフ同型性判定問題は入力サイズの多項式時間で計算が可能なのか NP 完全なのかはわかっていない．入力のグラフに制限を加えた場合，同型性判定問題が多項式時間で解けるグラフクラスが存在する [3]．本論文では，同型性判定問題が入力サイズの多項式時間で解けるグラフクラスを対象としている．

第3章　非連結グラフの再構築問題

　非連結グラフの再構築問題を以下のように定義する．

定義 3.1 非連結グラフの再構築問題
入力：頂点数 $(n-1)$ の n 個のグラフ　$G_1, G_2, ..., G_n$
出力：$\{G_1, G_2, ..., G_n\}$ を $Deck$ とする頂点数 n の非連結グラフ G^*.
　　ただし，$\{G_1, ..., G_n\}$ を $Deck$ として持つような非連結グラフが存在しない
　　ときは No を返す.

本章では入力がある非連結グラフ G^* の $Deck$ であると仮定した場合の再構築アルゴリズムを示す.
　まず，G^* が頂点数 1 個の連結成分を含む非連結グラフである場合，以下の補題が成立する.

補題 3.2 G^* が頂点数 1 の連結成分を含む非連結グラフであるならば，$Deck$ 内に頂点数 1 の連結成分を含む G_i が少なくとも $n-1$ 個存在する.

証明　G^* の頂点数 1 の連結成分数を $k(1 \leq k \leq n)$ 個とするとき，以下の場合分けをすることができる.

- G^* に頂点数 2 の連結成分がない場合
- G^* に頂点数 2 の連結成分がある場合

G^* に頂点数 2 の連結成分がない場合，$Deck$ は孤立点が $k-1$ かつ孤立点以外の頂点数が $n-k$ で構成されている G_i が k 個と，孤立点が k かつ孤立点以外の頂点数が $n-k-1$ で構成されている G_i が $n-k$ 個で，構成されている．G^* に頂点数 2 の連結成分がある場合，s を頂点数 2 の連結成分の全体の頂点数とする．$Deck$ は孤立点 $k-1$ 個かつ孤立点以外の頂点数が $n-k$ 個で構成されている G_i が k 個と，孤立点が $k+1$ 個かつ孤立点以外の頂点数が $n-k-1$ 個で構成されている G_i が $s-1$ 個と孤立点が k 個かつ孤立点以外の頂点数が $n-k-s$ で構成されている G_i が $n-k-s$ 個で，構成されている．よって，頂点数 1 の連結成分を含む G_i が少なくとも $n-1$ 個存在する．したがって，G^* が頂点数 1 の連結成分を含む非連結グラフであるならば，$Deck$ 内の頂点数 1 の連結成分を含む G_i が少なくとも $n-1$ 個存在する.

（証明終）

G^* が頂点数 1 個の連結成分を含む非連結グラフである場合，補題 3.1 より，$Deck$ 内の頂点数 1 個の連結成分の個数が最小であるグラフに孤立した頂点を追加すれば，G^* を構築することができる．

以上の議論より以下の定理が成り立つ．

定理 3.3 G^* が頂点数 1 個の連結成分を含む非連結グラフである場合，G^* を再構築することができる．

したがって，これ以降は G^* の各連結成分は少なくとも 2 個以上の頂点を含むと仮定する．

グラフ G^* が頂点数 2 個以上の連結成分のみからなる非連結グラフである場合，以下のことが言える。

定理 3.4 グラフ G^* が頂点数 2 個以上の連結成分のみからなる非連結グラフであることの必要十分条件は入力 G_i の各々のグラフがすべて非連結グラフであることである．

証明 (必要性) グラフ G^* が頂点数 2 個以上の連結成分のみからなる非連結グラフならば，入力 G_i の各々のグラフがすべて非連結グラフであるということは，明らかである．
(十分性) G^* が連結の場合を考える．このとき，G^* は連結な全域木 T を持つ．G^* は全域木 T の葉になる頂点を取り除いても連結である．よって G^* が連結ならば $Deck$ の中に連結グラフが存在する．したがって，入力 G_i の各々のグラフがすべて非連結グラフならば，グラフ G^* は非連結グラフである．

（証明終）

次に 2 つの記号を定義する．

定義 3.5 非連結グラフ G において頂点数最大の連結成分の集合を $M(G)$ とし，$M(G)$ の各連結成分の頂点数を $P(G)$ とする．

すると，以下の捕題が成り立つ．

補題 3.6 G^* の連結成分の頂点数がすべて同数であるための必要十分条件は，すべての i に対して G_i は $M(G_i)$ と頂点数が $(P(G_i) - 1)$ 個の連結成分で構成されていることである．

証明 (必要性) G^* の連結成分の頂点数がすべて同数であるならば，$Deck$ の任意の i に対して G_i は $M(G_i)$ と頂点数が $(P(G_i) - 1)$ 個の連結成分で構成されるのは明らかである．
(十分性) G^* の連結成分の頂点数が少なくとも 2 種類あったと仮定する．そのとき，G^* の $Deck$ を構築すると，連結成分の頂点数の差の最大が 2 以上になる G_i が存在する．したがって，任意の i に対して G_i は $M(G_i)$ と頂点数が $(P(G_i) - 1)$ 個の連結成分で構成されているならば，G^* の連結成分の頂点数がすべて同数である．

（証明終）

G^* が遺伝的でかつ多項式時間で同型性判定問題を解くことのできるグラフクラスに属する場合の再構築問題を定義する.

定義 3.7 C を遺伝的でかつ同型性判定問題が多項式時間で解くことができるグラフのクラスとする．このとき，C に属する任意の非連結グラフの再構築問題を以下のように定義する.

C に属する任意の非連結グラフの再構築問題
入力：頂点数 $(n-1)$ の n 個の C に属する任意のグラフ　　$G_1, G_2, ..., G_n$
出力：$\{G_1, G_2, ..., G_n\}$ を $Deck$ とする頂点数 n の C に属する任意の非連結グラフ G^*.
　　ただし，$\{G_1, ..., G_n\}$ を $Deck$ として持つような C に属する任意の非連結グラフが存在しないときは No を返す.

G^* が遺伝的でかつ多項式時間で同型性判定問題を解くことのできるグラフクラスに属する場合，以下の定理が成り立つ.

定理 3.8 C を頂点の削除に関して遺伝的でかつ同型性判定問題が多項式時間で解くことができるグラフのクラスとする．この時，C に属する任意の非連結グラフ G^* のグラフ再構築問題は，C に属する連結なグラフの再構築問題に多項式時間で帰着できる.

証明　C に属する任意の非連結グラフ G^* のグラフ再構築問題は，以下のように場合分けをすることができる.

1. 非連結グラフ G^* の各連結成分の頂点数がすべて同数の場合

2. その他の場合

１．まず，グラフ G^* の連結成分の頂点数がすべて同数の場合を述べる．G^* の各連結成分を $G^1, G^2, G^3, ..., G^k$ とし，それぞれの頂点数を d とする.
まず，G^* の各連結成分が同型でない場合を考える．G^* の連結成分の頂点数がすべて同数であるので，以下の式が言える.

$$\max_i P(G_i) = P(G^*)$$

各 $M(G_i)$ 同士が同型となる G_i を対象とし，各 $M(G_i)$ が同型となる G_i から $M(G_i)$ を除く．この操作を繰り返し行うと，G_i の連結成分が 1 つになる.
次に，G^* が同型な連結成分を含む場合を考える．同型なグラフが k 個存在する場合を考える．この場合，各 $M(G_i)$ が同型である G_i が kd 個存在する．G_i の中の $M(G_i)$ が同型である G_i を G'_i とする．G'_i の中の G_i が同型であるグラフから 1 つずつ選ぶ．G'_i の中の G_i が同型であるグラフから 1 つずつ選んだグラフを G''_i とする．G''_i は d 個存在する．よって，$M(G''_i)$ が同型となる G''_i を対象とし，$M(G''_i)$ が同型となる G''_i から $M(G''_i)$ を除く．この操作を繰り返し行うと，G_i の連結成分が 1 つになる.
したがって，非連結グラフ G^* の各連結成分の頂点数がすべて同数の場合は非連結グラフの再構築問題が多項式時間で連結グラフの再構築問題に帰着可能である.

2．次に，グラフ G^* の連結成分の頂点数が同数でない場合について述べる．$\max_i P(G_i)$ は $M(G^*)$ の各連結成分の頂点数であるので以下の式が成り立つ．

$$\max_i P(G_i) = P(G^*)$$

$P(G_i) = P(G^*)$ となる i のうち，$M(G_i)$ の要素数が最大となる i を k とする．G^* は $M(G_k)$ に含まれる各連結成分を連結成分として含む．そこで，以下の操作を考える．

- $M(G_i)$ が $M(G_k)$ と同型になる G_i を探す．
- 探して得られた G_i から $M(G_i)$ を除く．

以上の操作を $n - |M(G_i)|$ 個のグラフについて行うと，最終的に G_i の連結成分が 1 つになるか，または G_i のすべての連結成分の頂点数が等しくなる．

以上の議論より，C を頂点の削除に関して遺伝的でかつ同型性判定問題が多項式時間で解くことができるグラフのクラスとするとき，C に属する任意の非連結グラフ G^* のグラフ再構築問題は，C に属する連結なグラフの再構築問題に $Deck$ のサイズの多項式時間で帰着できる．

（証明終）

第4章　proper interval graphの再構築アルゴリズム

本章では，G^* が連結な proper interval graph である場合について多項式時間再構築アルゴリズムを提案する．

4.1　問題の定義

連結な proper interval graph の再構築問題を以下のように定義する．

定義 4.1 連結な *proper interval graph* の再構築問題
入力：頂点数 $(n-1)$ の n 個の *proper interval graph*　$G_1, G_2, ..., G_n$
出力：$\{G_1, G_2, ..., G_n\}$ を $Deck$ とする頂点数 n の連結な *proper interval graph* G^*．
ただし，$\{G_1, ..., G_n\}$ を $Deck$ として持つような連結な *proper interval graph* が存在しないときは No を返す．

4.2　再構築アルゴリズム

連結な proper interval graph の場合，頂点の挿入に関して以下の定理が成り立つ．

定理 4.2 連結な *proper interval graph* において，*proper interval graph* の性質を満たすように挿入することのできる頂点の挿入の仕方は $O(n^2)$ 通り存在する．

証明　*proper interval graph* の区間表現中の区間を，$I_1, I_2, ..., I_{n-1}$ とし，$i \in 1, 2, ..., (n-1)$ に対して，区間の左端点を $L(i)$, 右端点を $R(i)$ とする．また，区間表現は数直線上 $[1, 2, ..., 2(n-1)]$ の各整数上に $I_1, I_2, ..., I_{n-1}$ の左端点・右端点が存在し，$L(i) \neq L(j)$，$L(i) \neq R(j)$ とする．このとき，挿入する区間を I_x とし，I_x の左端点を $L(x)$，右端点を $R(x)$ とする．挿入する区間の左端点 $L(x)$ を数直線上 0.5〜$2(n-1)+\frac{1}{2}$ の 1 刻みにスイープさせる．次に，下の式を満たす区間 I_k，I_h の左端点をチェックする．

$$\max_k L(I_k) < L(I_x) \qquad \min_h L(I_h) > L(I_x)$$

このとき，*proper interval graph* の関係から挿入する区間の右端点 R_x は，$R(I_k)+0.5$〜$R(I_h)-0.5$ の 1 刻みに挿入することが可能である．左端点の選び方は，$2n$ 通りであり，それぞれの左端点に対応する右端点の選び方は $O(n)$ 通りである．

よって，*proper interval graph* の頂点の挿入は $O(n^2)$ 通り存在する．

（証明終）

頂点の挿入に関して，ここでは線形個数しかないようにみえるが，実際には挿入可能な場所が $\Theta(n^2)$ 箇所あるような proper interval graph が存在する．$L(i)$ と $L(i+1)$ の間に区間 I_x の L_x を挿入するとき，次の場合を考える．$L(i)$ と $L(i+1)$ の間に $\frac{n}{2}-1$ の $R(j)$ が挟まっていて，かつ $R(i)$ と $R(i+1)$ の間に $\frac{n}{2}-1$ の $L(j)$ が挟まっている．この場合を考えると $L(x)$ の挿入の仕方は $\frac{n}{2}-1$ 通り，$R(x)$ の挿入の仕方は約 $\frac{n}{2}-1$ 通り存在する（図 4.1 参照）．よって，proper interval graph の頂点の挿入の仕方は，$\Theta(n^2)$ 通りより，少なくすることができない．

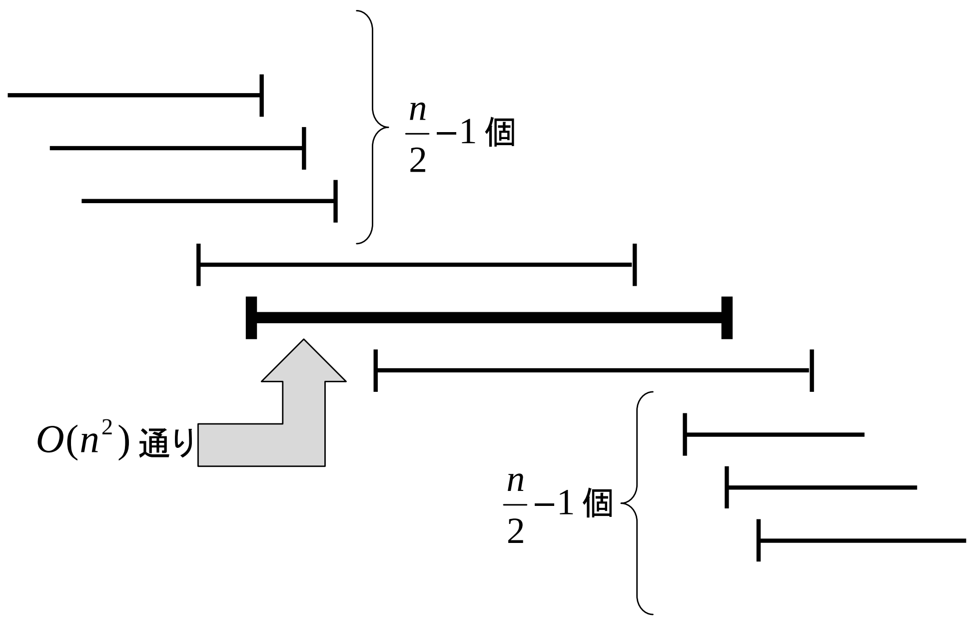

図 4.1: 区間の挿入

G^* が存在すると仮定する．定理 2.3 より区間を挿入して得られる区間表現に対応するグラフは proper interval graph の性質を失わない．つまり定理 4.2 では proper interval graph の可能な頂点の挿入をすべて網羅している．よって，G^* が存在すると仮定すると，定理 4.2 の頂点の挿入で構築されるグラフの中に G^* が存在することがわかる．よって，入力のグラフ $G_1, G_2, ..., G_n$ と頂点を挿入して得られたグラフの *Deck* がすべて同型なら，頂点を挿入して得られたグラフが G^* となり構築することができる．アルゴリズムの詳細は擬似コードとして **Algorithm 1** に示す．

入力されたグラフを区間表現にする線形時間アルゴリズムは存在し [5]，また，入力された区間表現からグラフ表現を線形時間で構成するアルゴリズムは自明である．**Algorithm 1** では，頂点の挿入は $O(n^2)$ 通りである．一般に同型性判定にかかる実行時間は入力サイズの線形時間で計算することができる [4]．しかし，**Algorithm 1** では $Deck$ の各々のグラフに対して同型性判定を行っているため入力サイズの多項式時間で計算される．以上の議論より，以下の定理が成り立つ．

定理 4.3 *proper interval graph* には入力サイズの多項式時間で実行可能な再構築アルゴリズムが存在する．

4.3 高速アルゴリズム

本節では，**Algorithm 1** より高速なアルゴリズム **Algorithm 2** を提案する．定理 2.3 より，区間を挿入して得られる区間表現に対応するグラフは proper interval graph の性質を失わない．つまり，定理 4.2 は，proper interval graph の可能な頂点の挿入をすべて網羅している．よって，G^* が存在すると仮定すると，定理 4.2 の頂点の挿入によって構築されるグラフの中に G^* が存在することがわかる．ここで，頂点の挿入によって構築されたグラフを G'_1 とする．**Algorithm 2** は入力の $Deck$ 及び G'_1 の $Deck$ を 1 つのグラフとみなし，同型性判定をする．このことにより，**Algorithm 1** より高速なアルゴリズム **Algorithm 2** を実現することができる．よって入力の $Deck$ と G'_1 が同型であれば，G'_1 を G^* とし，出力する．したがって G^* が存在すれば，G^* を構築することができる．

次に計算時間について考察する．まず，頂点の挿入について考える．定理 4.2 より 1 頂点を挿入する選び方は $O(n^2)$ 通りある．次に $Deck$ の構築について述べる．$Deck$ を構築する計算時間は $O(nm + n^2)$ 時間である．ここで，このアルゴリズムは連結グラフを対象としているので $n = O(m)$ である．したがって，$Deck$ の構築にかかる実行時間は $O(nm)$ 時間である．

最後に，このアルゴリズムの同型性判定について考える．2 つの interval graph $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ を入力とした場合，interval graph の同型性判定にかかる実行時間は $O((|V_1|+|V_2|)+(|E_1|+|E_2|))$ 時間である [4]．proper interval graph は interval graph のサブクラスである．よって，interval graph の同型性判定アルゴリズムは，proper interval graph の同型性判定アルゴリズムに用いることができる．$Deck$ の頂点数は高々n^2 個であり，枝数は高々nm 本である．よって，$Deck$ 同士の同型性判定にかかる実行時間は $O(n^2+nm)$ 時間である．ここで，D_1 を G'_1 の $Deck$ とすると，今 $n = O(m)$ であるので入力の $Deck$ と D_1 との同型性判定にかかる実行時間は $O(nm)$ 時間である．

定理 4.4 *proper interval graph* には，$O(n^3m)$ 時間の再構築アルゴリズムが存在する．

Algorithm 1

Input: 頂点数 $(n-1)$ の n 個の proper interval graph $G_1, G_2, ..., G_n$
Output: 頂点数 n の proper interval graph G^*．または，No
$G_1 :=$ 連結なグラフ $\in Deck$
$I(G_1) := G_1$ の区間表現の 1 つ
begin
  for $L(x) = 0.5$ **to** $\frac{4n+1}{2}$ **do**
    for $R(x) = L(x)$ **to** $\frac{4n+1}{2}$ **do**
      $I(G_1)$ に区間 $[L(x), R(x)]$ を挿入
      $G_1' := I(G_1)$ に区間 $[L(x), R(x)]$ を挿入して得られる区間表現に対応するグラフ
      if G_1' が proper interval graph でない **then**
        continue
      end
      $\{F_1, F_2, ..., F_n\} \leftarrow\ False$
      for each $g \in V(G_1')$ **do**
        G_1' から g を削除
        $G_g := G_1'$ から g を取り除いたグラフ
        if $(F_l = False) \wedge (G_g$ と G_l が同型 $)$ **then** $F_l \leftarrow True$
        else goto next
        end
      end
      $G^* \leftarrow G_1'$;   **return** G^*
      next
    end
  end
  return No
end.

Algorithm 2

Input: 頂点数 $(n-1)$ の n 個の proper interval graph $G_1, G_2, ..., G_n$
Output: 頂点数 n の proper interval graph G^*．または，No
$G_1 :=$ 連結なグラフ $\in Deck$
$D :=$ 入力の $Deck$
$I(G_1) := G_1$ の区間表現の 1 つ
begin
  for $L(x) = 0.5$ **to** $\frac{4n+1}{2}$ **do**
    for $R(x) = L(x)$ **to** $\frac{4n+1}{2}$ **do**
      $I(G_1)$ に区間 $[L(x), R(x)]$ を挿入
      $G_1' := I(G_1)$ に区間 $[L(x), R(x)]$ を挿入して得られる区間表現に対応するグラフ
      if G_1' が proper interval graph でない **then**
        continue
      end
      $D_1 := G_1'$ の $Deck$
      $D_1 := G_1'$ の $Deck$
      D と D_1 に対して同型性判定
      if D と D_1 が同型である **then**
        $G^* \leftarrow G_1'$; **return** G^*
      end
    end
  end
  return No
end.

第5章　quasi threshold graphの再構築アルゴリズム

本章では，G^* が連結な **quasi threshold graph** である場合についての多項式時間再構築アルゴリズムを提案する．

5.1　問題の定義

連結な **quasi threshold graph** の再構築問題を以下のように定義する．

定義 5.1 連結な *quasi threshold graph* の再構築問題
入力：頂点数 $(n-1)$ の n 個の *quasi threshold graph*　$G_1, G_2, ..., G_n$
出力：$\{G_1, G_2, ..., G_n\}$ を $Deck$ とする頂点数 n の連結な *quasi threshold graph* G^*.
　　ただし，$\{G_1, ..., G_n\}$ を $Deck$ として持つような連結な *quasi threshold graph* が存在しないときは No を返す.

5.2　再構築アルゴリズム

まず，グラフ G が連結な **quasi threshold graph** の場合，以下の定理が成り立つ．

定理 5.2 グラフ G が連結な **quasi threshold graph** ならば，G は少なくとも 1 つの次数 $n-1$ の頂点を持つ.

証明　G が少なくとも 1 つの次数 $n-1$ の頂点を持つならば，必ず定義 2.4 の操作 2 で終了し構築されているのは自明である．定義 2.4 の操作 3 で終了すると仮定する．定義 2.4 の操作 3 後に構築されるグラフは非連結グラフであり，G^* が連結であるという仮定と矛盾する．よって G^* が連結な **quasi threshold graph** ならば，G^* は少なくとも 1 回の定義 2.4 の操作 2 で終了し構築されたグラフである．したがって，グラフ G が連結な **quasi threshold graph** ならば，少なくとも 1 つの次数 $n-1$ の頂点を持つ.

（証明終）

G^* が存在すると仮定する．G^* が 1 つの次数 $n-1$ の頂点を持つ場合を考える．最後の操作で挿入された頂点を v_1 とする．v_1 の次数は $n-1$ である．G^* から $Deck$ を構築すると，v_1 以外の頂点を削除した場合は次数 $n-2$ の頂点が 1 つ存在するのは明らかである．また，v_1 を削除した場合，次数 $n-2$ の頂点の数は 0 である．次に G^* が 2 つの次数 $n-1$ の頂点を持つ場合を考える．最後の 2 回の操作 2 で挿入された頂点を v_1, v_2 とする．v_1 と v_2 の次数は双方とも $n-1$ である．G^* から $Deck$ を構築すると，v_1 と v_2 以外の頂点を削除した場合は次数 $n-2$ の頂点が 2 つ存在するのは明らかである．また，v_1 または v_2 を削除したグラフの場合，次数 $n-2$ の頂点の個数は 1 個である．

次に G^* が k 個の次数 $n-1$ の頂点を持つ場合を考える．$v_1 \sim v_k$ の次数は各々$n-1$ である．G^* から $Deck$ を構築すると，$v_1 \sim v_k$ 以外の頂点を削除した場合は次数 $n-2$ の頂点が k 個存在するのは明らかである．また，$v_1 \sim v_k$ のどれか 1 つの頂点を削除した場合，次数 $n-2$ の頂点の数は $k-1$ 個である．つまり，次数 $n-2$ の頂点の個数が $k-1$ 個のグラフは，最後の操作 3 または操作 1 のあとの操作 2 で挿入した頂点を取り除いたグラフである．したがって，入力 $Deck$ の中の次数 $n-2$ の頂点の個数は k か $k-1$ のいずれかである．よって，$\min\{$ 次数 $n-2$ の頂点の個数 $\} = k-1$ であるグラフに定義 2.4 の操作 2 を行えば，G^* を構築することが可能である．

$Deck$ の各グラフの次数 $n-2$ の頂点の個数を数えるのにかかる実行時間は $O(n^2)$ 時間であり，頂点を追加し辺を挿入する操作にかかる実行時間は $O(n)$ 時間である．そして，同型性判定にかかる実行時間は quasi threshold graphs は interval graphs のサブクラスであるので 4.2 節より，$O(nm)$ 時間である．

よって以上の議論より，以下の定理が成り立つ．

定理 5.3 quasi threshold graph には，$O(nm)$ 時間の再構築アルゴリズムが存在する．

詳細は擬似コードとして **Algorithm 3** に示す．

Algorithm 3

Input: 頂点数 $(n-1)$ の n 個の quasi threshold graph $G_1, G_2, ..., G_n$
Output: 頂点数 n の quasi threshold graph G^*．または，No

begin
 $J_i := Deck$ の各グラフ G_i の次数 $(n-2)$ の頂点の個数
 $G_1 := \min_{1 \leq i \leq n} J_i$ であるグラフ $\in Deck$
 G_1 に頂点 u を追加
 $G' := G_1$ に頂点 u を追加して得られるグラフ
 $G' = (V', E')$
 for $\forall v \in V'$ **then**
  $E' \leftarrow E' \bigcup \{v, g\}$
 end
 $D' := G'$ の $Deck$
 D と D' に対して同型性判定
 if D と D' が同型である **then**
  $G^* \leftarrow G'$;　**return** G^*
 else
  return No
 end
end

第6章　まとめ

本研究では，連結な *proper interval graph* と *quasi threshold graph* の多項式時間再構築アルゴリズムを提案した．C を頂点の削除に関して遺伝的でかつ同型性判定問題が多項式時間で解くことができるグラフのクラスとするとき，C に属する任意の非連結グラフ G^* のグラフ再構築問題は，C に属する連結なグラフの再構築問題に多項式時間で帰着できることも示した．

今後の課題として，同型性判定が多項式時間で可能である *interval graph* の多項式時間再構築アルゴリズムの開発と *distance-hereditary graph* の多項式時間再構築アルゴリズムの開発があげられる．

謝辞

本研究を行うにあたり，日頃より懇切丁寧な御指導を賜りました上原隆平准教授に，心から感謝いたします．浅野哲夫教授，元木光雄助教，清見礼助教にはゼミなどにおいて適切な御教示を頂き，厚く御礼申し上げます．また，情報基礎学講座の皆様には公私にわたりお世話になりました．この場を借りて感謝します．

参考文献

[1] W. T. Tutte, Graph Theory As I Have Known It, Oxford, 1998.

[2] E. Hemaspaandra and L. A. Hemaspaandra and S. P. Radziszowski and R. Tripathi, Complexity results in graph reconstruction, Discrete Applied Mathematics, 155(2), p.103-118, 2007.

[3] R. Uehara, S. Toda and T. Nagoya, Graph Isomorphism Completeness for Chordal Bipartite Graphs and Strongly Chordal Graphs, Discrete Applied Mathematics, 145(3), pp.479-482, 2005.

[4] G. S. Lueker and K. S. Booth, A Linear Time Algorithm for Deciding Interval Graph Isomorphism, Journal of ACM, 26(2), pp.183-195, 1979.

[5] B. S. Panda and Sajal K. Das, A linear time recongnition algorithm for proper interval graphs, Information processing Letters, 87, pp.153-161, 2003.

[6] R. Diestel, Graph Theory, Springer-Verlag, 1997.

[7] D. G. Corneil, Y. Perl and L. K. Stewart, A linear recognition algorithm for cographs, SIAM Journal on Computing, 14, pp.926-934, 1985.

[8] S. Nakano, R. Uehara and T. Uno, A New Approach to Graph Recognition and Applications to Distance Hereditary Graphs, 4th Annual Conference on Theory and Applications of Models of Computation (TAMC 07), Lecture Notes in Computer Science, Vol.4484, pp.115-127, 2007/5.

[9] M. C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, 1980.

[10] J. H. Yan, J. J. Chen, G. J. Chang, Quasi-threshold graphs, Discrete Appl. Math, 69, No.3, pp.247-255, 1996.