교육학석사학위논문

A study on the clique partitioning problem in some weighted chordal graphs

(가중 현그래프의 클릭 분할 문제에 관한 연구)

2016년 2월

서울대학교 대학원

수학교육과

조 창 성

A study on the clique partitioning problem in some weighted chordal graphs

(가중 현그래프의 클릭 분할 문제에 관한 연구)

지도교수 김 서 령

이 논문을 교육학석사 학위논문으로 제출함

2015년 12월

서울대학교 대학원

수학교육과

조 창 성

조창성의 교육학석사 학위논문을 인준함

2015년 12월

위 원 장	정 상 권	(인)
부위원장	유 연 주	(인)
위 원	김 서 령	(인)

A study on the clique partitioning problem in some weighted chordal graphs

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Mathematics Education
to the faculty of the Graduate School of
Seoul National University

by

Chang Seong Jo

Dissertation Director : Professor Suh-Ryung Kim

Department of Mathematics Education
Seoul National University

February 2016

Abstract

In this thesis, we study the minimum clique partitioning problem with constrained bounds in weighted chordal graphs. Recently, Myung (2008) proposed an algorithm for the minimum clique partitioning problem with constrained bounds in weighted interval graphs. We extend the family of graphs to which Myung's algorithm is applicable to some chordal graph.

Key words: Interval graph; Chordal graph; Good Chordal Graph; Clique partitioning; Approximation algorithms
Student Number: 2014-20943

Contents

Chapter 1

Introduction

1.1 Minimum clique partitioning problem

A graph is called a *weighted graph* if a real value is assigned to each vertex. A subset of vertices of a graph is called a *clique* if any two vertices in the subset are adjacent. Given a weighted graph G and a positive real number B, the *minimum clique partitioning problem (MCPB for short)* in G is to find the minimum number of cliques such that the weight sum of vertices in the clique of G is no more than B. The *the bin-packing problem with conflicts* (BPPC) is one of the combinatorial optimization problems which have been studied a lot. It is defined as follows. Given a set of items $V = \{1, \ldots, n\}$ with size $s_1, \cdots, s_n \in \mathbb{R}^+$ and a conflict graph $G = (V, E)$ which has the edge set E in such way that $(i, j) \in E$ if and only if the items i and j cannot be packed into the same bin, the BPPC is to find a packing for the items into bins of size B (by definition, adjacent items must be assigned to different bins). The BPPC becomes the MCPB by taking complement $\bar{G}$ of the conflict graph G. In this context, MCPB has as many applications as BPPC does. One of real-world applications concerns scheduling problem such as

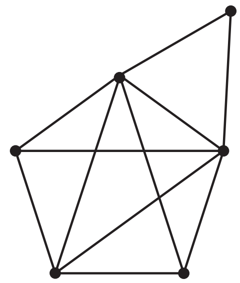

Figure 1.1: A chordal graph

job distribution where some jobs cannot be assigned to the same machine (see [5]). In this case, all jobs are represented by vertices and any two nonadjacent vertices are not allowed to be proposed on the same machine. Other applications arise in the problems including goods delivery (refer to [3]), seating guests on tables, storing digital musics on CDs (see [6]) and so on. For a graph G, taking the complement graph of G and assigning 0 to all vertices as weights lets us view the graph coloring problem as a special case of MCPB.

1.2 Interval graphs and chordal graphs

A graph G is a *chordal graph* if every cycle of length greater than or equal to 4 has a chord, that is, an edge connection two vertices that are not consecutive on the cycle. For example, the graph given in Figure 1.1 does not have any cycle of length greater than 3 without chord, so it is chordal. It is well-known that chordal graph has a “simplicial” vertex. A vertex v is called *simplicial* if its neighborhood $N(v) = \{w \in V(G) : (v, w) \in E(G)\}$ is clique, that is, every pair of neighbors of v is connected by an edge of the graph.

An intersection graph of intervals is called an *interval* graph. Gilmore and Hoffman [4] show that interval graphs are a special class of chordal graphs, that is, every

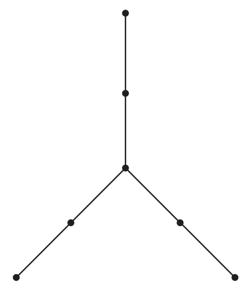

Figure 1.2: A chordal graph which is not an interval graph

interval graph is a chordal graph. However, the converse is not true. It is well known fact that there is a chordal graph which is not a interval graph as shown by the graph given in Figure 1.2.

1.3 The algorithm Relax

Myung [7] proposed an approximation algorithm called Relax with “factor” 2 for the MCPB in special case for weighted interval graphs. The Relax is described as Algorithm 1. We say that an approximation algorithm has *factor* k when the approximated solution by the algorithm is equal or less than k multiple of the optimal solution. Chen et al. [1] showed that for any $\epsilon > 0$, there is no approximation algorithm having a factor $3/2 - \epsilon$ for the MCPB with interval graphs unless $P = NP$. Based on their result, we may say that Myung's algorithm is near-optimal.

Algorithm 1: Relax

Input : a graph $G(V, E)$ with a weight function

Output: r cliques : $C_1, C_2, \cdots, C_r$

$j \leftarrow 1$
while $G \neq \emptyset$ **do**
    $i \leftarrow$ the vertex of G which has the smallest number
    for $k \in N[i]$ **do**
        if $|C_j \cup \{k\}| \leq B$ **then**
            $C_j \leftarrow C_j \cup \{k\}$ $G \leftarrow G \backslash \{k\}$
        else
            $w_k \leftarrow |C_j \cup \{k\}| - |C_j|$
            $j \leftarrow j + 1$
        end
    end
end

Myung [7] proved that an arbitrary optimal partition of the relaxed version of the MCPB in weighted interval graph can be modified to a partition generated by Relax without increasing the number of cliques. He assumed that the vertices are ordered in the nondecreasing order of right end of each interval. The following two properties of satisfied by intervals are used in the proof:

(1) If two intervals overlap with an interval with the right end point left to both of their right end points, they overlap;

(2) Suppose that, for an interval I overlapping with intervals J and K, the right end point of I is right to that of J and the right end point of J is right to that of K. Then if an interval whose right end point is left to the right end point I overlaps with J, then it also overlaps with K.

By the way, these are the only properties of intervals used in the proof. We can rewrite these properties in terms of the ordering $\lambda : V(G) \to \{1, \ldots, |V(G)|\}$ as follows:

(GC-I) For a vertex u_1 and vertices u_2, u_3 which are neighbors of u_1, if $\lambda(u_3) > \lambda(u_1)$ and $\lambda(u_2) > \lambda(u_1)$, then u_2 and u_3 are adjacent.

(GC-II) For a vertex v_1 and vertices v_2, v_3 which are neighbors of v_1, if $\lambda(v_1) < \lambda(v_2) < \lambda(v_3)$ and a vertex v_4 satisfying $\lambda(v_4) > \lambda(v_1)$ is adjacent to v_2, then v_4 is adjacent to v_3.

We say that the graph G is a *good chordal graph* if there is an ordering on $V(G)$ satisfying the properties (i) and (ii).

1.4 A preview of thesis

This thesis proves that chordal graphs without a particular configuration are good chordal graphs in Chapter 2. In Chapter 3, we design an algorithm which gives a vertex labeling satisfying the properties (i) and (ii) for some chordal graph by using the idea in the proof of Theorem 1. The algorithm consists of two parts.

1. Finding a labeling satisfying GC-I;

2. Exchanging the labels of two vertices which violate GC-II.

In Chapter 4, we summarize the thesis and then present an open problem.

Chapter 2

Good chordal graphs

Given a labeling λ on $V(G)$, suppose that λ does not satisfy the property (ii) in the definition a good chordal graph. Then there exist vertices v_1, v_2, v_3, v_4 such that v_2 and v_3 are neighbors of v_1 while v_4 is not, $\lambda(v_1) = 1 < \lambda(v_2) < \lambda(v_3)$, $1 < \lambda(v_4)$, and v_4 is adjacent to v_2 but not to v_3. See Figure 2.2 for an illustration. In such a cases, we say that λ *violates the property (ii) in the definition of a good chordal graph with the ordered quaternary* (v_1, v_2, v_3, v_4). We also denote by $\lambda_{v,w}$ the labeling obtained from λ by switch the labels for v and w, that is,

$$\lambda_{v,w}(u) = \begin{cases} \lambda(u) & \text{if } u \neq v, w; \\ \lambda(w) & \text{if } u = v; \\ \lambda(v) & \text{if } u = w. \end{cases} \tag{2.1}$$

Theorem 2.0.1. *If a chordal graph does not contain a configuration given in Figure 2.1, then it is a good chordal graph.*

Proof. By induction. A trivial graph vacuously satisfies the statement. Now suppose that the statement is true for the chordal graphs with $n - 1$ vertices that do

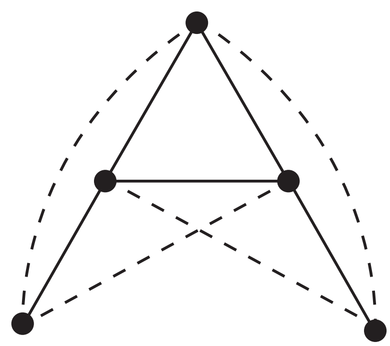

Figure 2.1: A forbidden configuration

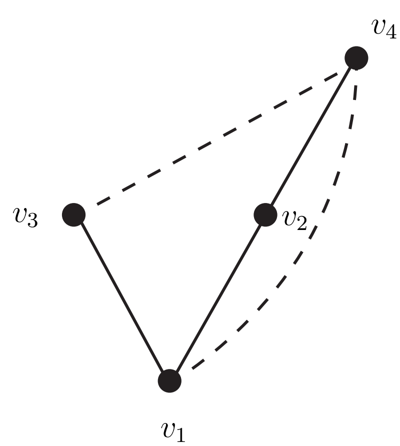

Figure 2.2: Four vertices violating the condition (ii).

not contain a configuration given in Figure 2.1. To reach a contradiction, suppose that there is a chordal graph G with n vertices that do not contain a configuration in Figure 2.1 but is not a good chordal graph. Since G is chordal, it contains a simplicial vertex, say w. Let G' be the graph obtained by deleting w from G. Then it is easy to check that G' is chordal and does not contain a configuration in Figure 2.1. Thus, by the induction hypothesis, G' is a good chordal graph, so there is a labeling $\ell' : V(G') \to [n-1]$ that satisfies the properties (i) and (ii) defining good chordal graphs. Now we define a labeling $\ell : V(G) \to [n]$ in the following way.

$$\ell(v) = \begin{cases} 1 & \text{if } v = w; \\ \ell'(v) + 1 & \text{otherwise.} \end{cases}$$

The condition (i) is immediately satisfied by ℓ since w is a simplicial vertex of G and $\ell(w) = 1$, which is the smallest label. Therefore ℓ violates (ii) as G is not a good chordal graph. Suppose that ℓ violates (ii) with the ordered quaternary (w, x, y, z). That is, x and y are neighbors of w while z is not, $\ell(w) = 1 < \ell(x) < \ell(y)$, $1 < \ell(z)$, and z is adjacent to x but not to y.

Since G does not contain a configuration in Figure 2.1, any neighbor of y that is not adjacent to w is adjacent to x. Since w is a simplicial vertex, any vertex adjacent to both w and y is also adjacent to x and vice versa. Thus

$$N_G(y) \subsetneq N_G(x). \tag{2.2}$$

Now we consider the labeling $\ell^* := \ell_{x,y}$ defined in (2.1). Then $\ell^*(y) < \ell^*(x)$. Furthermore we observe that

$\ell(x) < \ell(v)$ for any vertex v satisfying $\ell^*(y) < \ell^*(v)$, and $\ell(y) > \ell(v)$ for any vertex v satisfying $\ell^*(x) > \ell^*(v)$; (§)

for any vertex $v \neq x, y$, $\ell^*(x) < \ell^*(v) \Rightarrow \ell(x) < \ell(y) < \ell(v)$, and $\ell^*(v) < \ell^*(y) \Rightarrow \ell(v) < \ell(x) < \ell(y)$. ($\star$)

The observation (§) and ($\star$) still hold for ℓ' since ℓ' is the restriction of ℓ to $V(G) \setminus \{w\}$. Suppose to the contrary that ℓ^* does not satisfy (i). Then there exist three vertices u_1, u_2, u_3 of G such that u_2 and u_3 are neighbors of u_1,

$$\ell^*(u_1) < \ell^*(u_2) \quad \text{and} \quad \ell^*(u_1) < \ell^*(u_3), \tag{2.3}$$

but u_2 and u_3 are not adjacent in G. Since ℓ satisfies (i), $\ell(u_2) < \ell(u_1)$ or $\ell(u_3) < \ell(u_1)$. Without loss of generality, we may assume $\ell(u_2) < \ell(u_1)$. Then $u_2 = x$ or $u_1 = y$. Suppose that $u_2 = x$. Then $u_1 \neq y$ by (2.2). In addition, $\ell(y) > \ell(u_1)$ by the observation (§). Since u_2 and u_3 are not adjacent, $u_3 \neq y$. Therefore $\ell^*(u_1) < \ell^*(u_3)$ is equivalent to $\ell(u_1) < \ell(u_3)$ since $\{u_1, u_3\} \cap \{x, y\} = \emptyset$. Now u_1 and y are adjacent to x and $\ell(y) > \ell(x)$ and $\ell(u_1) > \ell(x)$. Then, since ℓ satisfies (i), u_1 and y are adjacent. Thus u_1 is adjacent to x and y, $\ell(u_1) < \ell(y)$, and $\ell(u_1) < \ell(u_3)$. Hence u_3 and y are adjacent, which contradicts (2.2). Now suppose $u_1 = y$. Then, by (2.2), $u_2 \neq x$ and $u_3 \neq x$. Then $\ell(x) < \ell(u_2)$ and $\ell(x) < \ell(u_3)$ by (2.3) and (§), which implies that ℓ violates (i). Hence ℓ^* satisfies (i).

Obviously, ℓ^* does not violate (ii) anymore with the quaternary (w, x, y, z). Suppose that ℓ^* violates (ii) with (v_1, v_2, v_3, v_4) where $v_1 \neq w$. Then $\ell^*(v_3) > \ell^*(v_2) > \ell^*(v_1)$, $\ell^*(v_1) < \ell^*(v_4)$, v_2 and v_3 are neighbors of v_1, v_4 is adjacent to v_2 but not to v_3. Since $\ell^*(w) = \ell(w) = 1$, w cannot be any of v_2, v_3, v_4. Moreover, since we have shown that ℓ^* satisfies (i), v_2 and v_3 are adjacent (see Figure 2.3).

If $\ell^*(v_4) > \ell^*(v_2)$, then v_4 and v_3 would be adjacent since $\ell^*(v_3) > \ell^*(v_2)$ and the condition (i) holds for ℓ^*. Therefore $\ell^*(v_4) < \ell^*(v_2)$. Thus

$$\ell^*(v_1) < \ell^*(v_4) < \ell^*(v_2) < \ell^*(v_3). \tag{2.4}$$

We will show that v_2 and v_3 are neighbors of w by contradiction. Suppose one of v_2, v_3 is not a neighbor of w. If none of v_1, v_2, v_3, v_4 equals x or y, then $\ell^*(v_i) = \ell'(v_i)$

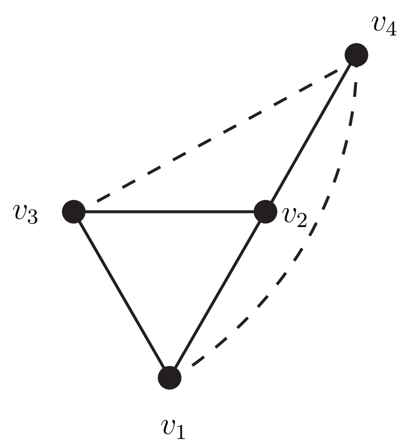

Figure 2.3: Four vertices violating the condition (ii).

for $i = 1, \ldots, 4$ and so ℓ' violates the property (ii) with (v_1, v_2, v_3, v_4), which is a contradiction. Thus $\{v_1, v_2, v_3, v_4\} \cap \{x, y\} \neq \emptyset$. Suppose that $\{v_1, v_2, v_3, v_4\} \cap \{x, y\} = \{x, y\}$. Since $\ell^*(y) < \ell^*(x)$, by (2.4),

$$(y, x) \in \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_4, v_2), (v_4, v_3), (v_2, v_3)\}.$$

Since v_3 and v_4 are not adjacent and v_1 and v_4 are not adjacent while x and y are adjacent, (y, x) cannot be (v_4, v_3) or (v_1, v_4). If $(y, x) = (v_1, v_2)$, then, $\ell(x) = \ell^*(y) < \ell^*(v_4) = \ell(v_4)$ by (2.4), and so v_1 and v_4 must adjacent by (i) since $\ell(y) > \ell(x)$, which is a contradiction. If $(y, x) = (v_4, v_2)$, then, $\ell(v_3) > \ell(x)$ by ($\star$) since $\ell^*(v_3) > \ell^*(v) = \ell^*(x)$ by (2.4), and so v_3 and v_4 must be adjacent by (i) since $\ell(v_4) = \ell(y) > \ell(x)$, which is a contradiction. If $(y, x) = (v_1, v_3)$, then $\ell'(x) < \ell'(v_4) < \ell'(v) < \ell'(y)$ by (2.4) and (§) and so ℓ' violates the property (ii) with (x, v_2, y, v_4), a contradiction. If $(y, x) = (v_2, v_3)$, then we reach a contradiction to our assumption that one of v_2, v_3 is not a neighbor of w. Therefore we conclude that only one of v_1, v_2, v_3 and v_4 is possible for x or y.

Suppose that $v_3 = y$. Then, by ($\star$) and (2.4),

$$\ell'(v_1) < \ell'(v_4) < \ell'(v_2) < \ell'(y) \tag{2.5}$$

and ℓ' violates the property (ii) with (v_1, v_2, y, v_4). Thus $v_3 \neq y$. Suppose that $v_3 = x$. Then $\ell'(v_1) < \ell'(v_4) < \ell'(v_2)$ by (2.4). Moreover, by the assumption, v_2 is not adjacent to w. Thus v_1 must be adjacent to w to avoid the configuration given in Figure 2.1. Since w is a simplicial vertex, v_1 is also adjacent to y. Moreover $\ell'(v_3) < \ell'(v_2)$, for otherwise, (2.5) holds, which implies that ℓ' violates the property (ii) with (v_1, v_2, v_3, v_4). Now $\ell'(v_3) = \ell'(x) < \ell'(y)$ and $\ell'(v_3) < \ell'(v_2)$, so y is adjacent to v_2 since ℓ' satisfies the condition (i). Since $\ell^*(v_2) < \ell^*(v_3) = \ell^*(x) = \ell'(y)$, $\ell'(v_2) < \ell'(y)$. Thus $\ell'(v_1) < \ell'(v_4) < \ell'(v_2) < \ell'(y)$. Since ℓ' satisfies the condition (ii), the vertex y should be adjacent to v_4. Since x is not adjacent to v_4, we reach a contradiction to (2.2). Thus we can conclude that v_3 is neither x nor y.

Next we show that v_2 is neither x nor y. Suppose that $v_2 = y$. If $\ell'(v_2) < \ell'(v_3)$, then (2.5) holds, which implies that ℓ' violates the property (ii) with (v_1, v_2, v_3, v_4). Thus $\ell'(v_3) < \ell'(v_2)$. Since $\ell^*(y) < \ell^*(v_3)$, $\ell'(x) < \ell'(v_3)$ by (§). Moreover, since the vertices v_4, v_3, and v_1 are adjacent to v_2, the vertices v_4, v_3, and v_1 are adjacent to x by (2.2). Then ℓ' violates (ii) with (v_1, x, v_3, v_4). Now we consider the case $v_2 = x$. If $\ell'(v_4) < \ell'(v_2)$, then (2.5) holds, which violates the property (ii) with (v_1, v_2, v_3, v_4). Thus $\ell(v_2) < \ell(v_4)$. Then, however, v_4 is not adjacent to v_3 even if $\ell'(v_2) < \ell'(v_3)$ and $\ell'(v_2) < \ell'(v_4)$, which contradicts the fact that ℓ' satisfies the condition (i).

Now we suppose that v_4 is x or y. We consider the case in which v_4 is y. Then $\ell^*(v_1) < \ell^*(y)$, so $\ell'(v_1) < \ell'(x)$ by ($\star$). In addition, $\ell^*(y) < \ell^*(v_2) < \ell^*(v_3)$, so $\ell'(x) < \ell'(v_2) < \ell'(v_3)$ by (§). We note that y is adjacent to x and v, $\ell^*(y) < \ell^*(x)$, and $\ell^*(y) < \ell^*(v_2)$. Since ℓ^* satisfies (i), x and v_2 are adjacent. Thus x and v_3

are adjacent, for otherwise, ℓ' violates (ii) with (v_1, v_2, v_3, x). Then x is adjacent to y and v_3, $\ell'(x) < \ell'(y)$ and $\ell'(x) < \ell'(v_3)$. However, v_3 and y are not adjacent and we reach a contradiction to the fact that ℓ' satisfies (i). Now suppose that v_4 equals x. Then $\ell^*(x) < \ell^*(v_2) < \ell^*(v_3)$, so $\ell'(x) < \ell'(v_2) < \ell'(v_3)$ by ($\star$). Thus $\ell'(x) < \ell'(v_2)$ and $\ell'(x) < \ell'(y)$. Since $y \neq v_2$ and (i) is satisfied by ℓ', y and v_2 are adjacent. Since $N_G(y) \subset N_G(x)$ by (2.2), v_3 and y are not joined. Since $\ell^*(v_1) < \ell^*(x)$, $\ell'(v_1) < \ell'(y)$ by (§). Then ℓ' violates (ii) with (v_1, v_2, v_3, y) and we reach a contradiction.

Suppose $v_1 = x$. Then $\ell'(v_3) > \ell'(v_2) > \ell'(x)$ and $\ell'(v_4) > \ell'(x)$ by ($\star$). Therefore ℓ' violates (ii) with (x, v_2, v_3, v_4) and we reach a contradiction. Suppose $v_1 = y$. Then v_2 and v_3 are adjacent to x by (2.2). Moreover, $\ell'(x) < \ell'(v_4) < \ell'(v_2) < \ell'(v_3)$ by (§). Note that $\ell(x) < \ell'(v_2)$, $\ell'(x) < \ell'(v_3)$, aw and v_4 are not adjacent while x and v_3 are adjacent. Since ℓ' satisfies (i), x and v_4 are not adjacent. Thus ℓ' violates (ii) with (x, v_2, v_3, v_4) and reach a contradiction.

We have reached in each of the cases to consider and so we can conclude that v_2 and v_3 are adjacent to w.

For each labeli λ on $V(G)$ that satisfies (i) but violates (ii), we denote by $NG(\lambda)$ the set of quaternary (u_1, u_2, u_3, u_4) with which λ violates (ii), that is,

$$NG(\lambda) = \{(u_1, u_2, u_3, u_4) : \{u_2,u_3\} \subset N(u_1), 1 < \lambda(u_2) < \lambda(u_3),$$
$$1 < \lambda(u_4), u_4u_2 \in E(G), u_4u_3 \notin E(G), u_4 \notin N[u_1].\}$$

By our assumption, the first component of each quaternary belonging to $NG(\ell)$ is w and $(w, x, y, z) \in NG(\ell)$. Furthermore, we have shown that if ℓ_{u_2,u_3} satisfies (i) but violates (ii), then, for (u_2, u_3) satisfying $(u_1, u_2, u_3, u_4) \in NG(\ell_{u_2,u_3})$ for some vertices u_1 and u_4 of G, $NG(\ell_{u_2,u_3}) \neq \emptyset$ and the first component of each quaternary belonging to $NG(\ell_{u_2,u_3})$ is w.

Let x_1 be the vertex with the smallest value of ℓ among the second components of pairs in $NG(\ell)$ and y_1 be a vertex with the largest $\ell(y) - \ell(x)$ among the the third components of quaternaries in $NG(\ell)$ with x in the second component. Let x_2 be the vertex with the smallest value of ℓ_{x_1,y_1} among the second components of quaternaries in $NG(\ell_{x_1,y_1})$ and y_2 be a vertex with the largest $\ell_{x_1,y_1}(v)-\ell_{x_1,y_1}(u)$ among the third components of quaternaries in $NG(\ell_{x_1,y_1})$ with u in the second component. Then $(x_2, y_2) \neq (x_1, y_1)$ since $\ell_{x_1,y_1}(x_1) > \ell_{x_1,y_1}(y_1)$ while $\ell_{x_1,y_1}(x_2) < \ell_{x_1,y_1}(y_2)$. Now we show that $\ell(x_1) < \ell(x_2)$. For notational convenience, we denote ℓ_{x_1,y_1} by ℓ_1. Assume $x_1 = x_2$. Then $y_1 \neq y_2$ and $\ell(y_1) > \ell(y_2)$ since $\ell(y_1)-\ell(x_1) \geq \ell(y_2)-\ell(x_2)$ by the choice of y_2. However, $\ell_1(x_2) = \ell_1(x_1) = \ell(y_1) > \ell(y_2) = \ell_1(y_2)$, which contradicts the fact that $(w, x_2, y_2, z) \in NG(\ell_1)$ for some $z \in V(G)$. Thus $x_1 \neq x_2$. Assume $y_1 = y_2$. Then $x_1 \neq x_2$ and $\ell(x_1) < \ell(x_2)$ by the choice of x_1 as desired. Therefore it remains to consider the case $x_1 \neq x_2$ and $y_1 \neq y_2$. For notational convenience, we denote by $NG'(\lambda)$ the set of second and third components of each quaternary belonging to $NG(\lambda)$, that is, $NG'(\lambda) = \{(u_2, u_3) : (u_1, u_2, u_3, u_4) \in NG(\lambda)$ for some $u_1, u_4\}$. We consider the two cases under this condition: $(x_2, y_2) \in NG'(\ell) \cap NG'(\ell_1)$; $(x_2, y_2) \notin NG'(\ell)$.

First suppose that $(x_2, y_2) \in NG'(\ell) \cap NG'(\ell_1)$. Since $(x_2, y_2) \in NG'(\ell)$, $\ell(x_1) \leq \ell(x_2)$. As we have shown that $x_1 \neq x_2$, $\ell(x_1) < \ell(x_2)$.

Now assume $(x_2, y_2) \notin NG'(\ell)$. Then $\ell(x_2) > \ell(y_2)$. If $x_2 \neq y_1$ and $y_2 \neq x_1$, then, since $x_1 \neq x_2$ and $y_1 \neq y_2$, $\ell(x_2) = \ell_1(x_2) < \ell_1(x_2) < \ell_1(y_2) < \ell(y_2)$ and we reach a contradiction. Thus $x_2 = y_1$ or $y_2 = x_1$. If $x_2 = y_1$, $\ell(x_1) < \ell(y_1) = \ell(x_2)$. If $y_2 = x_1$, $\ell(x_1) = \ell(y_2) < \ell(x_2)$. Thus we have shown that $\ell(x_1) < \ell(x_2)$.

Now for $\ell_2 := (\ell_1)_{x_2,y_2}$, if $NG'(\ell_2) \neq \emptyset$, then we may take a vertex x_3 with the smallest value of ℓ_2 among the second components of quaternaries in $NG'(\ell_2)$ and a vertex y_3 with the largest $\ell_2(v) - \ell_2(u)$ among the third components of quaternaries in $NG'(\ell_2)$ with u in the second component. Then, by the argument above, $\ell(x_2) < \ell(x_3)$. Since G is finite, this process stops to obtain a labeling $\ell_p := (\ell_{p-1})_{x_p,y_p}$ on $V(G)$ with $NG'(\ell_p) = \emptyset$, which implies that ℓ_p satisfies the properties (i) and (ii) defining good chordal graphs. Thus we have reached a contradiction to our assumption that G is not a good chordal graph. Hence G is a good chordal graph and this completes the proof. $\square$

Choi *et al.* [2] showed the following lemma.

Lemma 2.0.2. *Let G be a block graph that is not a disjoint union of complete graphs. Then G has a maximal clique which contains exactly one cut vertex.*

In a block graph that is not a disjoin union of complete graphs, we call a maximal clique which contains exactly one cut vertex a *leaf clique*.

Theorem 2.0.3. *A block graph is a good chordal graph.*

Proof. Let G be a block graph. If G is a block graph that is not a disjoint union of complete graphs, then G is obviously a good chordal graph. Thus we may assume that G is not a disjoint union of complete graphs. By Lemma 2.0.2, G has a leaf clique. We take a leaf clique of G and denote it by A_1. Now we delete the vertices of A_1 exept the cut vertex of G in A_1 to obtain a subgraph G_1 of G. Then G_1 is still a block graph. If G_1 is a disjoint union of complete graphs, we are done. Otherwise, we take a leaf clique A_2 in G_1 and delete the vetices of A_2 except the cut vertex of G_1 in A_1 (in fact, the cut vertex of G_1 is a cut vertex of G). We continue in this way either to conclude that G is a good chordal graph or a sequence of A_1, A_2, $\ldots$, A_m where A_i is a leaf clique of G_{i-1} ($G_0 = G$), and m is the edge clique cover number of G.

Now let $B_i = A_i - \bigcup_{l>i} A_l$. By the choice of A_i, $B_i \neq \emptyset$ for all i. If $i < j$, then $A_j \cap \left(A_i - \bigcup_{l>i} A_l\right) = \emptyset$. Therefore $B_i \cap B_j = \emptyset$ if $i \neq j$ since $A_k - \bigcup_{l>k} A_l \subset A_k$

for any $k = 1, \ldots, m$. Since $\bigcup_{l=1}^{m} B_l = \bigcup_{l=1}^{m} A_l = V(G)$, $\Omega := \{B_1, B_2, \cdots, B_m\}$ is a partition of $V(G)$. Let λ be a labeling on $V(G)$ such that $\lambda(v) < \lambda(w)$ if $v \in B_i$, $w \in B_j$, and $i < j$. Since Ω is a patition of $V(G)$, such a labeling exist.

We observe that

$$\text{if } i < j \text{ and } A_i \cap A_j \neq \emptyset\text{, then } \lambda(x) < \lambda(y) \text{ for any } x \in A_i - A_j \text{ and any } y \in A_j. \tag{$*$}$$

To see why, fix $x \in A_i - A_j$ and $y \in A_j$. By the choice of A_i, A_i has no common element with any of $A_{i+1}, \ldots, A_{j-1}, A_{j+1}, \ldots, A_m$. Therefore $B_i = A_i - \bigcup_{l>i} A_l = A_i - A_j$, and so $x \in B_i$. On the other hand, $y \in B_k$ for some $k \in \{1, \ldots, m\}$. Then $y \in A_k$ and $y \notin \bigcup_{l>k} A_l$. Since $y \in A_j$, $j \leq k$. Thus $y \in B_k$ for some $k \geq j$. Then $i < k$, so we conclude that $\lambda(x) < \lambda(y)$ by the definition of λ.

We will show that λ satisfies the property (i) and (ii). To reach a contradiction, suppose that λ does not satisfy the property (i) or (ii). First, suppose that λ does not satisfy the property (i). Then there exist three vertices u, v, w of G such that v and w are neighbors of u, $\lambda(u) < \lambda(v) < \lambda(w)$, but v and w are not adjacent in G. Since G is a block graph and v is not adjacent to w, the cliques $\{u, v\}$ and $\{u, w\}$ are contained in distinct cliques. We suppose $\{u, v\} \subset A_s$ and $\{u, w\} \subset A_t$ for some s and t. If $s < t$, then $\lambda(v) < \lambda(u)$ by ($*$) since $A_s \cap A_t \neq \emptyset$, $v \in A_s - A_t$, and $u \in A_t$. If $t < s$, then $\lambda(w) < \lambda(u)$ by ($*$) since $A_s \cap A_t \neq \emptyset$, $w \in A_t - A_s$, and $u \in A_s$. In both cases, we reach a contradiction. Thus λ satisfies the property (i). Now suppose that λ does not satisfy the property (ii). Then there exist four vertices a, b, c, d of G such that $\lambda(a) < \lambda(b) < \lambda(c) < \lambda(d)$, c and d are neighbors of a, and b is adjacent to c but not to d. Since λ satisfies the property (i), c and d are adjacent. Then $\{a, c, d\}$ is a clique and so $\{a, c, d\} \subset A_{i'}$ for some i' and $\{b, c\} \subset A_{j'}$ for some j' distinct from i'. If $i' < j'$, then $\lambda(d) < \lambda(c)$ by ($*$) since $A_{i'} \cap A_{j'} \neq \emptyset$, $d \in A_{i'} - A_{j'}$, and $c \in A_{j'}$. If $j' < i'$, then $\lambda(b) < \lambda(a)$ by ($*$) since $A_{i'} \cap A_{j'} \neq \emptyset$,

$b \in A_{j'} - A_{i'}$, and $a \in A_{i'}$. In both cases we reach a contradiction. Thus λ satisfies the property (ii). Hence G is a good chordal graph with the labeling λ. $\square$

Chapter 3

Algorithm

We develop an algorithm called *Ordering* that labels the vertices of a connected graph without the configuration 2.1 so that the ordering of the vertices given by the labeling in our algorithm plays the same role as the ordering of the vertices given by Myung [7] for an interval graph in Relax.

Algorithm 2: Ordering

Input : a graph G and a labeling f satisfying the property (i)

Output: a labeling f from $V(G)$ to $[n]$

```
for i = 4, 5, ..., n do
    while FindNG (f^{-1}(i)) ≠ ∅ do
        f ← FindNG (f^{-1}(i)) ∘ f
    end
end
```

The function FindNG used in Algorithm 2 is defined as follows.

Algorithm 3: FindNG

Input : a vertex $v \in V(G)$

Output: a transposition (st) or $\emptyset$

$V \leftarrow \{u : f(u) > f(v)$ and u is neighbor of $v\}$
$W \leftarrow \{f(u) : u \in V\}$
$g \leftarrow \emptyset$
if $|V| = 1$ **then**
    return g
else
    for $s \leftarrow$ *minimum* **to** *maximum in* W **do**
        for $t \leftarrow$ *maximum* **to** *minimum in* W **do**
            if $N(f^{-1}(s)) \cap V \subset N(f^{-1}(t)) \cap V$ **then**
                next
            else
                $g \leftarrow$ a transposition (st)
                return g
            end
        end
    end
end

In Ordering, it must start when i has value 4 since the property (ii) can be violated with 4 vertices. The function FindNG is a sub-algorithm to find no-good neighbors of an input vertex. Each iteration in Ordering does not repeat infinitely because the number of no-good neighbors decreases whenever FindNG find them, and so the algorithm is well-defined. The input labeling f used in Algorithm 2 is given by Algorithm 4. Algorithm 4 is a sub-algorithm to find a labeling f satisfying the property (i).

Algorithm 4:

Input : a graph G with n vertices and a labeling f

Output: a labeling f from $V(G)$ to $[n]$

```
i ← 1
j ← 1
while i < n do
    if f^{-1}(i) is a simplicial vertex in G then
        f ← (ij) ∘ f
        j ← j + 1
        G ← G \ f^{-1}(i)
        i ← i + 1
    else
        i ← i + 1
    end
end
return f
```

Chapter 4

Conclusion

In this thesis, we have studied some chordal graphs which can be used in the algorithm Relax. We proved that a chordal graph which does not contain the particular configuration given in Figure 2.1 is good chordal graph. Furthermore, we designed an algorithm Ordering for labeling on some chordal graphs to extend the family of graph to which the Relax is applicable. For instance, by using our algorithm, we can deal with the graph in Figure 1.2 which could not be applied by the Relax. In this thesis, we have presented a sufficient condition for a graph being a good chordal graph. It would be interesting to find a necessary and sufficient condition for a graph being a good chordal graph.

Bibliography

[1] M. Chen, J. Li, J. Li, W. Li, and L. Wang, Some approximation algorithms for the clique partition problem in weighted interval graphs, *Theoretical computer science*, **381** (2007) 124-133.

[2] J. Choi, S.R. Kim, J. Y. Lee, and Y. Sano, On the partial order competition dimensions of chordal graphs, *arXiv:1601.01188* .

[3] N. Christofides, A. Mingozzi, and P. Toth, The vehicle routing problem, *Combinatorial Optimization*, Whiley, Chichester, (1979) 315-338.

[4] P. Gilmore and J. Hoffman, A characterization of comparability graphs and of interval graphs, *Canad. J. Math*, **16** (1964) 539-548.

[5] K. Jansen and S. Oehring, An approximation scheme for bin packing with conflicts, *Information and Computation*, **132** (1997) 85-108.

[6] B. McCloskey and A. Shankar, Approaches to bin packing with clique-graph conflicts, *Computer Science Division,*University of California (2005).

[7] Y. S. Myung, On the clique partitioning problem in weighted interval graphs, *Theoretical Computer Science*, **396** (2008) 290-293.

국문초록

최근에 Myung(2008)은 제한된 가중 구간 그래프의 최소 클릭 분할 문제에 대한 알고리즘을 발표하였다. 이 논문에서는 제한된 가중 그래프의 최소 클릭 분할 문제를 연구하여 Myung의 알고리즘이 적용 가능한 그래프족의 범위를 특정 현그래프로 확장시켰다.

주요어휘: 현그래프, 좋은 현그래프, 클릭 분할, 근사 알고리즘
학번: 2014-20943