On the complexity of the k-chain subgraph cover problem

Chang-Wu Yu[a], Gen-Huey Chen[b,*], Tze-Heng Ma[c]

[a] Department of Information Management, Ming Hsin Institute of Technology and Commerce,
Hsin-Chu, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, National Taiwan University,
Taipei, Taiwan, ROC
[c] Institute of Information Science, Academia Sinica, Nankang, Taipei, Taiwan, ROC

Abstract

The k-chain subgraph cover problem asks if the edge set of a given bipartite graph G is the union of the edge sets of k chain graphs, where each chain graph is a subgraph of G. Although the k-chain subgraph cover problem is known to be NP-complete for the class of bipartite graphs, it is still unknown whether this problem is NP-complete or polynomial-time solvable for subclasses of bipartite graphs. In this paper, we answer this question partially by showing that this problem for an important subclass of bipartite graphs, termed convex bipartite graphs, belongs to not only the class P, but also the class NC. More specifically, we show that the k-chain subgraph cover problem on the convex bipartite graph can be solved in $O(m^2)$ time sequentially or $O(\log^2 n)$ time in parallel using $O(m^3)$ processors on the CRCW PRAM, where n and m denote the number of vertices and edges, respectively. © 1998—Elsevier Science B.V. All rights reserved

1. Introduction

In this paper, we study a problem concerning bipartite graphs and chain graphs. A bipartite graph $G = (S, T, E)$ is a chain graph if each pair of edges either share an end vertex or are connected by an edge (see Fig. 1). A chain graph is also called a non-separable bipartite graph [10]. The *k-chain subgraph cover problem* [18] asks if the edge set of a given bipartite graph G is the union of the edge sets of k chain graphs, where each chain graph is a subgraph of G. This problem has a close relation to the

---

* Corresponding author. Fax: (886)-(2)-3628167; e-mail: ghchen@csie.ntu.edu.tw.
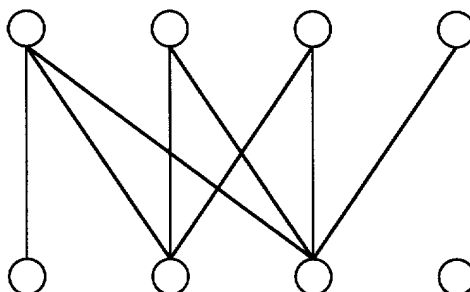
Fig. 1. A chain graph.

partial order dimension problem [18]. This problem is known to be NP-complete for $k \geqslant 3$ and polynomial-time solvable for $k = 2$ [18].

Recently, Ma and Spinrad [16] presented an $O(n^2)$ time algorithm for the 2-chain subgraph cover problem. Moreover, they showed that many problems can be reduced to the k-chain subgraph cover problem. Among those, there is a problem called the *biorder dimension problem* [7]. This problem arises from applications in social science, where multidimensional scaling is called for. There is a problem, named the *union biorder dimension k problem* [7], related to the biorder dimension problem. This problem asks whether a given bipartite partially ordered set can be covered by k biorders. It is exactly the k-chain subgraph cover problem interpreted in another context.

Although the k-chain subgraph cover problem which is defined on bipartite graphs is NP-complete, it is still unknown whether this problem is NP-complete or polynomial-time solvable for subclasses of bipartite graphs. In this paper, we answer this question partially by showing that this problem for an important subclass of bipartite graphs, termed convex bipartite graphs (explained below), belongs to not only the class P, but also the class NC (Nick's Class [11]). Previously, Cozzens and Leibowitz [7] have proposed a polynomial-time algorithm to solve the union biorder dimension k problem on the bipartite posets whose underlying graphs do not contain any induced $3k_2$ (three independent edges). This subclass of bipartite graphs and the convex bipartite graphs do not contain each other.

For a bipartite graph $G = (S, T, E)$, an ordering on S (or T) has the *adjacency property* if for each vertex $v \in T$ (or S), $N(v)$ contains consecutive vertices in the ordering. A bipartite graph $G = (S, T, E)$ is called a *convex bipartite graph* [15] if there is an ordering of S or T which has the adjacency property (see Fig. 2). In [13], Glover showed a practically important application of the convex bipartite graph in industry. In [15], Lipski and Preparata solved the maximum matching problem on the convex bipartite graph in $O(|S| + |T|)$ time. Dekel and Sahni [9] designed an efficient parallel algorithm to obtain maximum matchings in convex bipartite graphs. Their algorithms can be used to obtain parallel algorithms for several scheduling problems. Damaschke et al. [8] showed that many domination problems are polynomial-time solvable in convex bipartite graphs. Recently, Chen and Yesha [5] devised a parallel algorithm for
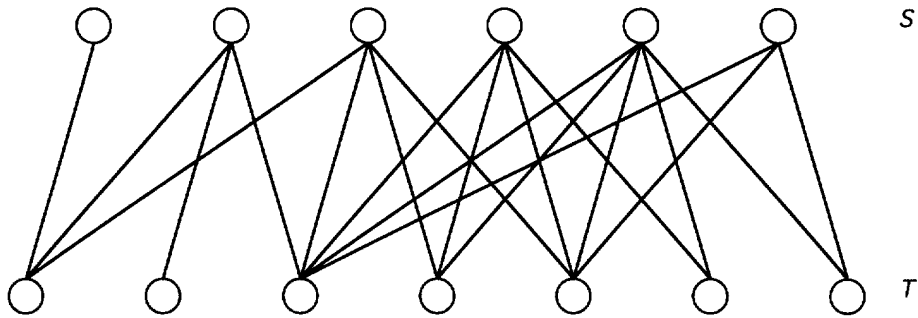
Fig. 2. A convex bipartite graph which S has the adjacency property.

recognizing the consecutive 1's property. This algorithm can be applied to recognize a convex bipartite graph in $O(\log^2 n)$ time using $O(n^3)$ processors.

In order to solve the k-chain subgraph cover problem on convex bipartite graphs, we show that it can be reduced to the coloring problem on comparability graphs, to be defined below. The reduction is detailed in Section 3. The resulting algorithm and complexity analysis are shown in Section 4. In the next section, we introduce some notations and definitions that are used throughout this paper.

2. Notations and definitions

For each vertex x in $G = (S, T, E)$, we define $N(x) = \{y \mid xy \in E\}$, which is called the *neighborhood* of x. Given a graph $G = (V, E)$, the *induced subgraph* of G by $V' \subset V$, which is denoted by $G_{V'}$, is the subgraph of G whose vertex set is V' and whose edge set contains those edges in E having both end vertices in V'. We say that two edges xy, wz of G are *independent* of each other if vertices x, y, w, z are distinct and $G_{\{x,y,w,z\}}$ contains only two edges xy and wz. A bipartite graph is a chain graph if it does not contain any pair of independent edges. Equivalently, $G = (S, T, E)$ is a chain graph if and only if $N(u) \subseteq N(v)$ or $N(v) \subseteq N(u)$ holds for any pair of vertices $u, v \in S$ (or T) [17].

A partially ordered set $(P, <)$ is a set of transitive and irreflexive binary relations on P. For a given $(P, <)$, we can construct a directed graph $G_{(P,<)}$ as follows: the vertex set of $G_{(P,<)}$ is P, and there exists an arc from a to b in $G_{(P,<)}$ if and only if $a < b$ holds. The *underlying graph* of a directed graph is an undirected graph having the same vertex set, but replacing each arc with an edge having the same end vertices. A graph G is called a *comparability graph* if there exists a partially ordered set $(P, <)$ such that G is the underlying graph of $G_{(P,<)}$ [12]. Moreover, the directed graph $G_{(P,<)}$ is called a *transitive orientation* of G. We define $\chi(G)$ as the minimum number of colors needed to color the vertices of G so that adjacent vertices are assigned different colors. An *independent set* in a graph G is a subset of vertices of G such that no two vertices in this subset are adjacent.

3. Reductions

For a bipartite graph G, we use $ch(G)$ to denote the smallest k so that G is k chain subgraphs coverable. Since $ch(G_1 \cup G_2 \cup \cdots \cup G_r) = ch(G_1) + ch(G_2) + \cdots + ch(G_r)$ if $G_1, G_2, \ldots, G_r$ are connected components of G, we assume the input graph G is connected.

Let $G = (S, T, E)$ be a convex bipartite graph such that $S = \{s_1, s_2, \ldots, s_k\}$ has the adjacency property. That is, the vertices adjacent to each t in T have consecutive indices in S. Let $\min(t)$ denote the minimum i such that s_i is adjacent to t. We sort T into $t_1, t_2, \ldots, t_l$ such that $i < j$ if $\min(t_i) < \min(t_j)$. Then, an adjacency matrix M of G is constructed as follows: $M[i, j] = 1$ if and only if t_i and s_j are adjacent. M is called a *canonical adjacency matrix* of G, which plays an important role in decomposing the edge set of G. An example is shown in Fig. 3. As usual, the element with the smallest row and column indices is located at the upper-left corner of a matrix. An *induced submatrix* in M is a matrix with elements $\{M[i, j] \mid i \in R, j \in C\}$, where R is a set of row indices and C is a set of column indices. From the definition, a bipartite graph G is a chain graph if and only if there are no induced submatrix

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

in the adjacency matrix of G. For a particular adjacency matrix, a pair of edges are called an α pair if they induce the former submatrix, and a β pair if they induce the latter submatrix.

As a consequence of the adjacency property of the convex bipartite graph, the following lemma results.
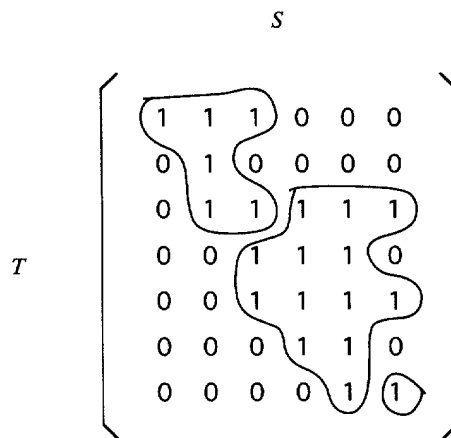

Fig. 3. The corresponding canonical adjacency matrix of Fig. 2 and the resulting partition generated by the greedy decomposition algorithm.

Lemma 1. *Let* G *be a convex bipartite graph. Then any canonical adjacency matrix does not contain any of the following induced submatrices*:

$$[\,1 \quad 0 \quad 1\,] \quad and \quad \begin{bmatrix} 0 & 1 \\ 1 & x \end{bmatrix},$$

where $x = 0$ *or* 1.

It follows that there is no β pair in M.

Then we construct $G^* = (V^*, E^*)$ from $G = (S, T, E)$ such that $V^* = E$ and two vertices in V^* are adjacent if and only if the two edges in G are independent. It is not difficult to see that the edge set of every chain subgraph of G induces an independent set of G^*, but the reverse is not always true. For example, let us consider the bipartite graph $G = (S, T, E)$ shown in Fig. 4, where $S = \{s_1, s_2, s_3, s_4\}$ and $T = \{t_1, t_2, t_3, t_4\}$. Although the edge subset $E' = \{(s_1, t_1), (s_2, t_2), (s_3, t_3), (s_4, t_4)\}$ induces an independent set of G^*, no subgraph of G whose edge set contains E' can be a chain graph. However, when G is restricted to a convex bipartite graph, we will show that there exists a special decomposition of the edge set of G (i.e. the vertex set of G^*) such that each edge subset of G can be extended to a chain subgraph of G.

Theorem 2. G^* *is a comparability graph if* G *is a convex bipartite graph.*

Proof. Let M be a canonical adjacency matrix of G. The 1's in M represent the vertices in G^*. Since each pair of independent edges in G induce the submatrix

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

in M by Lemma 1, we orient the edge from u to v as (u, v) if the entry representing u in M has smaller column and row indices than the entry representing v in M. Suppose there is a violation to the transitivity, i.e. after the orientation we have (u, v) and (v, w) but no (u, w). The submatrix of M induced by u, v, and w has the form

$$\begin{bmatrix} u & 0 & x \\ 0 & v & 0 \\ y & 0 & w \end{bmatrix}.$$

Since (u, w) does not exist, either $x = 1$ or $y = 1$ results in a forbidden submatrix of Lemma 1, which is a contradiction. $\square$

Since the edges of a chain subgraph of G induce an independent set in G^*, we have $\chi(G^*) \leqslant ch(G)$. By a greedy algorithm (shown below), we can decompose the vertices of G^* into k independent sets. Therefore, we derive a k-coloring of G^*. Since G^* is a comparability graph and the ordering (adopted by the greedy algorithm) of the vertices is actually a perfect ordering [3] on G^* (a topological sort of the transitively oriented graph), the greedy algorithm yields an optimal coloring [3, 14]. If each of
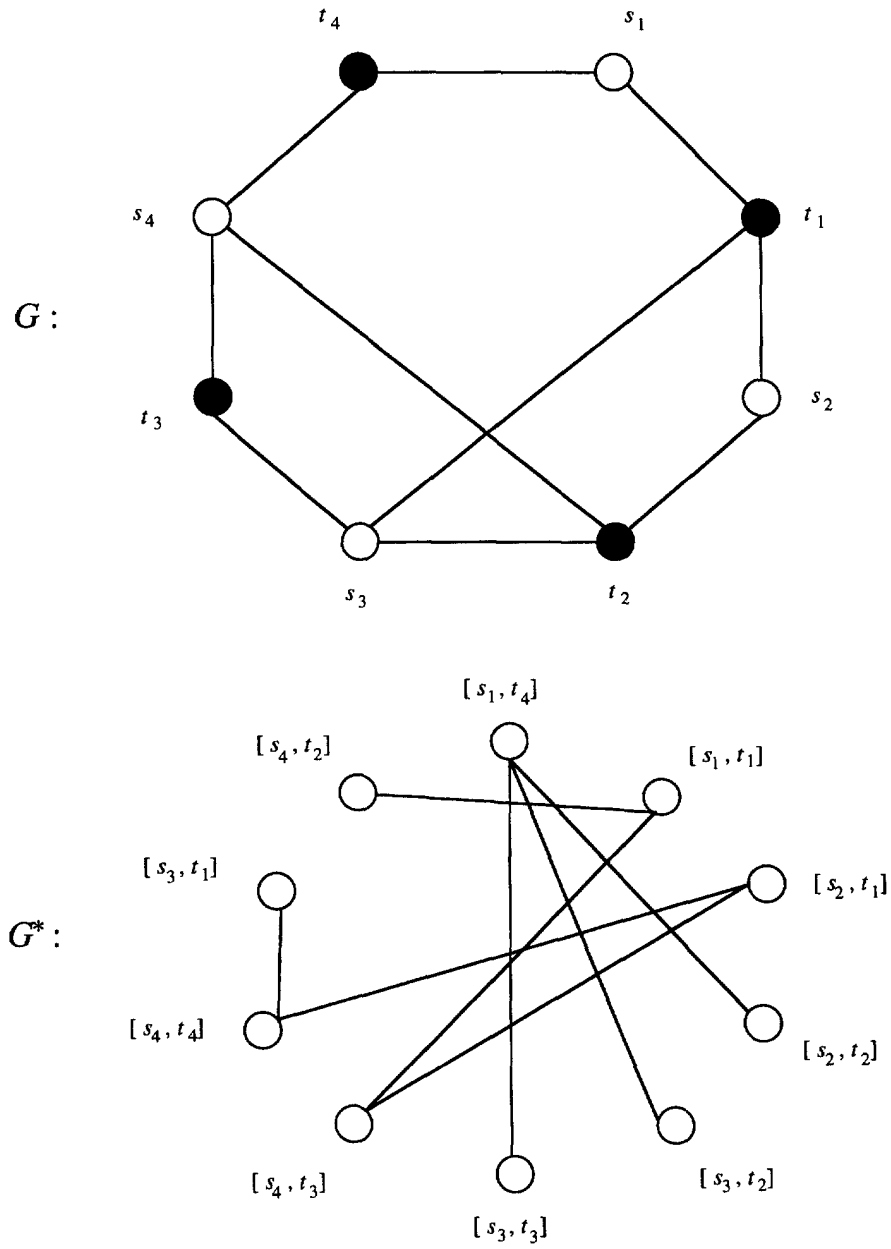
Fig. 4. A graph G and its G^*.

the k color classes can be extended into an independent set (in G^*) which forms a chain subgraph of G, then $\chi(G^*) \geqslant ch(G)$. Consequently, we have $\chi(G^*) = ch(G)$. The greedy decomposition algorithm is shown as follows.

Greedy Decomposition Algorithm.

Input: A canonical adjacency matrix M of a convex bipartite graph $G = (X, Y, E)$.

Output: A partition of E (the 1's in M) into $S_1, S_2, \ldots, S_k$ such that each subset does not contain an induced α pair.

begin

    scanned := 0;

    k := 0;

    while *scanned* < $|E|$ **do**

        begin

            $k := k + 1$

            $S_k := \emptyset$;

            for $i := 1$ **to** $|Y|$ **do**

            for $j := 1$ **to** $|X|$ **do**

                if $M[i, j] = 1$ and $x_j y_i$ combining with each edge in S_k does

                not induce an α pair

                then begin

                      add $x_j y_i$ to S_k;

                      $M[i, j] := 0$;

                      scanned := *scanned*+1

                end

        end

end.

We call each S_i $(1 \leqslant i \leqslant k)$ a *greedy independent set*. It is worthwhile to note that there is no α pair in each greedy independent set, but there might exist some β pairs, which prevent a greedy independent set in G^* from becoming a chain subgraph of G. For example, refer to Fig. 3 where a β pair can be found in the second greedy independent set S_2.

Our approach to overcoming these obstacles is that for each β pair $M[x, y], M[w, z]$ ($x < w$ is assumed) in S_i, add $M[x, z], M[x, z + 1], \ldots, M[x, y - 1]$ into S_i. We use S_i' to denote the set of these newly added elements, and we say that the β pair $M[x, y], M[w, z]$ introduce $M[x, z], M[x, z+1], \ldots, M[x, y-1]$ into S_i. Since $M[x, z] = 1$ (otherwise, $M[x, z], M[x, y]$, and $M[w, z]$ form a forbidden submatrix), we do destroy this β pair. We claim that no new α or β pair will be generated by this operation. The remaining work of this section is to prove that for $1 \leqslant i \leqslant k$ ($= \chi(G)$), the edge set $S_i \cup S_i'$ forms a chain subgraph.

In subsequent discussions we assume that G is a convex bipartite graph and $S_1, S_2, \ldots,$ S_k are the output of the greedy decomposition algorithm. Moreover, we use $M[S_i]$ to

denote the smallest induced submatrix of M that contains S_i. The submatrix $M[S_i]$ contains 1's in the entries specified by S_i, and 0's otherwise. For example, $M[S_2]$ with respect to Fig. 3 contains 13 1's. Note that the two entries at the left-up corner and the right-bottom corner of $M[S_2]$ are 0's, although they are 1's in M. We say that $M[i,j] \ll M[k,l]$ if $i \leqslant k$ and $j \leqslant l$ ($M[i,j]$ is scanned before $M[k,l]$ in the greedy decomposition algorithm). Next we show some properties of S_i and $S_i \cup S_i'$. In our subsequent proofs, a lot of discussions are made on the relative positions of elements of a canonical adjacency matrix M. We shall use lines, if necessary, to indicate elements with the same row or column indices. To avoid tediousness, some boundary and/or trivial cases are left to the readers.

Lemma 3. *There is no induced submatrix* [1 0 1] *contained in each* $M[S_i]$.

Proof. It is sufficient to prove that the greedy decomposition algorithm selects consecutive 1's for each row of M. Conversely, we suppose it is not true. Consider the [1 0 1] ($= [a\,b\,c]$) submatrix in some $M[S_r]$. Since [1 0 1] is forbidden in M by Lemma 1, we have $b = 1$ initially in M and $b \notin S_r$. Without loss of generality, we assume $b \in S_l$.

If $l > r$ (refer to Fig. 5(a)), there exists $d = 1 \in S_r$ such that b and d form an α pair. Then we have $o_1 = o_2 = 0$ in M. Since $d \in S_r$ and $c \in S_r$, d and c do not form an α pair, and we have $e = 1$ in M. Thus there is a forbidden submatrix $[d, o_2, e]$, which is a contradiction.

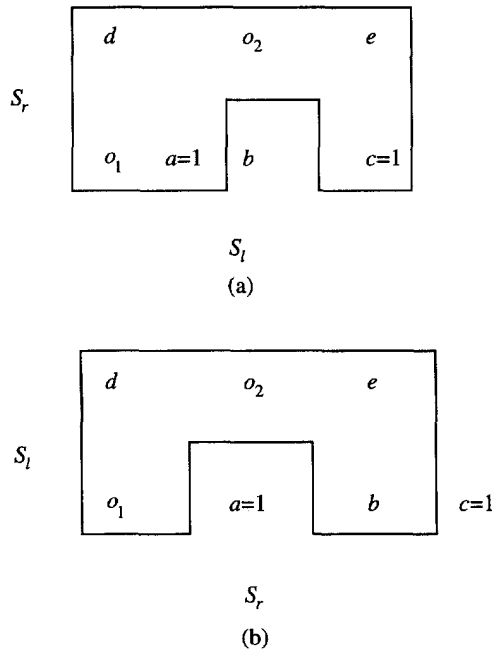Similarly there is a forbidden submatrix $[d, o_2, e]$ when $l < r$ (refer to Fig. 5(b)). □

Fig. 5. The proof of Lemma 3. (a) $l > r$. (b) $l < r$.

$$x \qquad o_2 \qquad a$$

$$o_1 \qquad y \qquad b$$

$$c \qquad d \qquad z$$

Fig. 6. The proof of Lemma 5.

Lemma 4. *There is no induced submatrix* $[1\ 0\ 1]$ *contained in each* $M[S_i \cup S_i']$.

Proof. Note that the set S_i' contains 1's distributed over one or more rows. These 1's are consecutive if they belong to the same row. Moreover, in each row there is a 1 (in S_i) next to the rightmost 1 (in S_i'). Hence $M[S_i \cup S_i']$ does not contain $[1\ 0\ 1]$ as an induced submatrix. $\square$

Lemma 5. *Suppose* $y \in S_i$ *and* $z \in S_j$. *If* $y \ll z$, *then* $i \leqslant j$.

Proof. Suppose $i > j$. There exists an $x \in S_j$ such that x and y form an α pair. Refer to Fig. 6, and we have $o_1 = o_2 = 0$. Since x and z are in S_j, either a or c is 1. Then either $\{x, o_2, a\}$ or $\{o_1, y, c\}$ forms a forbidden submatrix in M, which is a contradiction. $\square$

Lemma 6. *If* $x \ll y \ll z$ *and* $x, z \in S_i$, *then* $y \in S_i$.

Proof. Suppose $y \in S_j$. Since $x \ll y \ll z$, by Lemma 5 we have $i \leqslant j \leqslant i$. Hence, $j = i$. $\square$

To prove the edge set $S_i \cup S_i'$ forms a chain subgraph, it is then sufficient to prove that there are no new α pairs and β pairs in $S_i \cup S_i'$, which can be assured by subsequent lemmas.

Lemma 7. *There are no induced* α *pairs in* $S_i' - S_i$.

Proof. Suppose $p, q \in S_i' - S_i$, and they form an α pair in $S_i \cup S_i'$. Without loss of generality, we assume $p \ll q$. Moreover let the β pair x, y introduce p, and the β pair z, w introduce q (refer to Fig. 7), where $x, y, z, w \in S_i$. We note that according to the property of consecutive 1's, x cannot be on the right of o_2 ($=0$). Thus we have $x \ll q \ll z$. Since $x, z \in S_i$, by Lemma 6 we have $q \in S_i$, which is a contradiction. $\square$

Lemma 8. *There are no induced* α *pairs between* S_i *and* $S_i' - S_i$.

Proof. Suppose $z \in S_i' - S_i$, $w \in S_i$, and they form an α pair in $S_i \cup S_i'$. Moreover, let the β pair x, y introduce z. Refer to Fig. 8. We consider all possible positions of w,
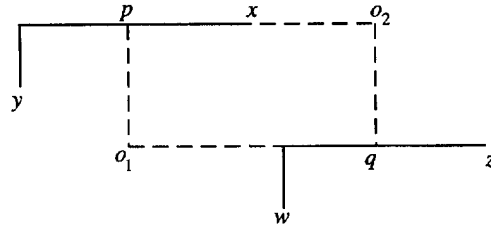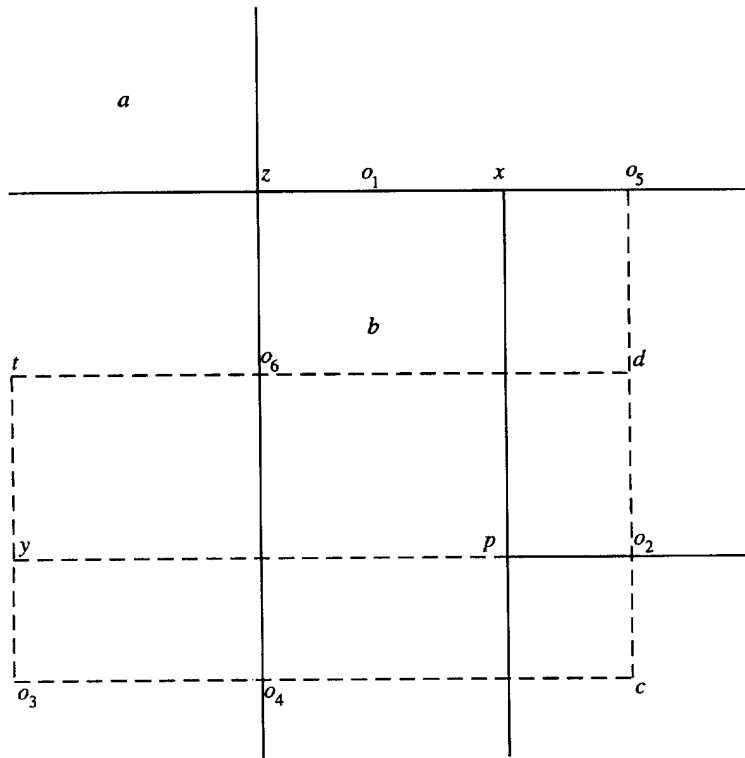
Fig. 7. The proof of Lemma 7.

Fig. 8. The proof of Lemma 8.

relative to z. Since z and w form an α pair, there are four possible positions of w, i.e. a, b, c, and d.

We claim $w \neq a$. Otherwise there is a contradiction because by Lemma 6 $a \ll z \ll x$ and $a, x \in S_i$ imply $z \in S_i$. Also we claim $w \neq b$. Otherwise $[z, o_1, x]$ $(=[1\ 0\ 1])$ forms a forbidden submatrix in $M[S_i \cup S_i']$.

Next, we claim $w \neq c$. The reason is as follows. If $w = c$, we have $o_4 = o_5 = 0$ in $M[S_i \cup S_i']$. We then consider the possible values of p in M. If $p = 1$, by Lemma 6 p belongs to S_i because $x \ll p \ll c$. Then x and y do not form a β pair in S_i, which
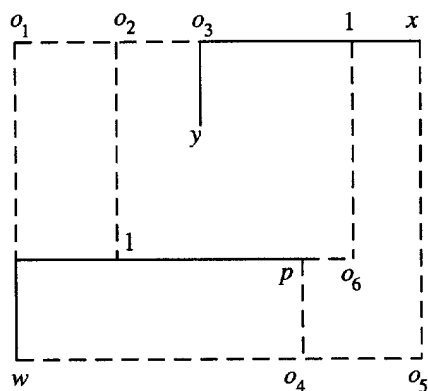
Fig. 9. The proof of Lemma 9.

is a contradiction. If $p = 0$, we have $o_2 = 0$ according to the property of consecutive 1's. Moreover, we have $o_3 = 0$ in $M[S_i \cup S_i']$, because otherwise $[o_3, o_4, c]$ ($= [1\ 0\ 1]$) forms a forbidden submatrix in $M[S_i \cup S_i']$. Since $o_3 = o_2 = 0$ in $M[S_i \cup S_i']$, y and c ($= w$) form an α pair in S_i, which is a contradiction.

Last, we claim $w \neq d$. Conversely, we suppose $w = d$. Since z and d form an α pair in $S_i \cup S_i'$, we have $o_6 \notin S_i \cup S_i'$ and by Lemma 6 $t \notin S_i$ because $t \ll o_6 \ll d$ and $d \in S_i$. Besides, since x and y form a β pair in S_i, we have $p \notin S_i$ and by Lemma 6 $o_2 \notin S_i$ because $y \ll p \ll o_2$. Hence, y and d form a β pair in S_i, and according to our process of eliminating β pairs, we have $o_6 \in S_i \cup S_i'$, which is a contradiction.

Since all four cases lead to contradictions, the lemma follows. □

**Lemma 9.** *There are no new β pairs generated in $S_i' - S_i$.*

Proof. Suppose there is a new β pair (the two 1's in Fig. 9) generated in $S_i' - S_i$. Then $o_2 = 0$ in $M[S_i \cup S_i']$. Let the two 1's be introduced by two β pairs x, y and p, w (refer to Fig. 9), respectively. Since x, y, p, and w are all in S_i, we have $o_3 = o_4 = 0$ in $M[S_i]$. Then we have $o_1 = o_5 = 0$ in $M[S_i]$. Otherwise either $[o_1, o_3, x]$ or $[w, o_4, o_5]$ forms the forbidden submatrix $[1\ 0\ 1]$. Note that o_4 is on the left of o_5 because if p has a greater column index than x, then $o_6 \in S_i'$ and the new β pair would not exist. It is implied (from $o_1 = o_5 = 0$) that x and w form a β pair in S_i, and they introduce $o_2 = 1$ into $S_i \cup S_i'$. This contradicts our assumption about o_2. □

Lemma 10. *There are no new β pairs generated between S_i and $S_i' - S_i$.*

Proof. Suppose there is a new β pair p, q, where $p \in S_i$ and $q \in S_i' - S_i$. Also q is assumed to be introduced by the β pair x, y. The subsequent discussion is made according to the relative positions of p and q.
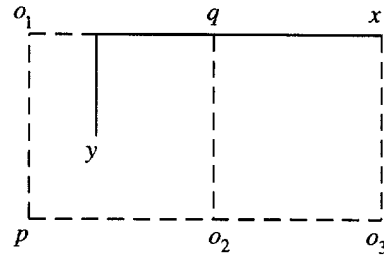
(a)
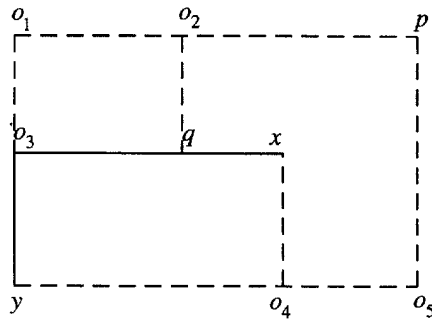
(b)

Fig. 10. The proof of Lemma 10. (a) q has a greater column index than p. (b) p has a greater column index than q.

Case 1: q has a greater column index than p (refer to Fig. 10(a)).

We have $o_1 = o_2 = 0$ in $M[S_i \cup S_i']$. If $o_3 \notin S_i$, then p and x form a β pair in S_i, and they introduce o_1 (=1) into $S_i \cup S_i'$, which is a contradiction. If $o_3 \in S_i$, then $[p, o_2, o_3] = [1\ 0\ 1]$ forms a forbidden submatrix.

Case 2: p has a greater column index than q (refer to Fig. 10(b)).

We have $o_2 = 0$ in $M[S_i \cup S_i']$. Since $o_3 = o_4 = 0$ in $M[S_i]$, we have $o_1 = o_5 = 0$ in $M[S_i]$. Otherwise, either $[o_1, o_2, p]$ or $[y, o_4, o_5]$ forms the forbidden submatrix $[1\ 0\ 1]$. Hence, y and p form a β pair in S_i, and consequently $o_2 = 1 \in S_i \cup S_i'$. This implies that p and q do not form a β pair in $S_i \cup S_i'$, which is a contradiction.   □

Thus far we have proved by Lemmas 7–10 that no new independent edge in G (an α pair or a β pair in M) would be generated by adding S_i', $1 \leqslant i \leqslant k$. Therefore, each edge set $S_i \cup S_i'$, $1 \leqslant i \leqslant \chi(G^*)$, forms a chain subgraph. Summarizing the discussion of this section, we obtain the main result of this paper as stated below.

Theorem 11. *If G is a convex bipartite graph, then $ch(G) = \chi(G^*)$.*

4. Algorithm and complexity analysis

According to Theorem 11, the k-chain subgraph cover problem on the convex bipartite graph can be solved by the following algorithm.

Input: A convex bipartite graph $G = (X, Y, E)$.
Output: Find minimum number of chain subgraphs of G that can cover G.
Step 1: Determine an ordering of X having the adjacency property.
Step 2: Generate the canonical adjacency matrix M of G by sorting Y.
Step 3: Construct G^* and its transitive orientation from M.
Step 4: Partition E into $S_1, S_2, \ldots, S_k$ by coloring vertices of G^*.
Step 5: Determine $S_1 \cup S_1', S_2 \cup S_2', \ldots, S_k \cup S_k'$, and generate k chain subgraphs from them.

Let n and m denote the numbers of vertices and edges, respectively, in G. The complexity of the algorithm is analyzed as follows. Step 1 can be completed in $O(m)$ sequential time by Booth and Lueker's work [1]. If Chen and Yesha's algorithm [5] is applied, Step 1 can be completed in $O(\log^2 n)$ parallel time using $O(n^3)$ processors on the CRCW PRAM.

Step 2 can be done by (1) computing all $\min(y_i)$'s, (2) sorting Y according to $\min(y_i)$'s, and (3) constructing M from sorted Y. Substep (1) takes $O(m)$ sequential time or $O(\log n)$ parallel time using $O(n^2/\log n)$ processors on the EREW PRAM [11]. Substep (2) takes $O(n \log n)$ sequential time or $O(\log n)$ parallel time using $O(n)$ processors on the EREW PRAM [6]. It is quite easy to complete substep (3) in $O(n^2)$ sequential time or $O(1)$ parallel time using $O(n^2)$ processors on the EREW PRAM. So totally Step 2 takes $O(n^2)$ sequential time or $O(\log n)$ parallel time using $O(n^2/\log n)$ processors on the EREW PRAM.

Step 3 can be done in $O(m^2)$ sequential time or $O(1)$ parallel time using $O(m^2)$ processors on the CREW PRAM. Step 4 can be completed in $O(m^2)$ sequential time [14] or $O(\log n)$ parallel time using $O(m^3)$ processors on the CREW PRAM [4]. With the aid of Lemma 4, Step 5 can be completed in $O(m^2)$ sequential time or $O(\log n)$ parallel time using $O(m^2)$ processors on the CREW PRAM.

To sum up, the execution of the algorithm needs $O(m^2)$ time sequentially or $O(\log^2 n)$ time in parallel using $O(m^3)$ processors on the CRCW PRAM.

Theorem 12. *The k-chain subgraph cover problem on the convex bipartite graph can be solved in $O(m^2)$ time sequentially or $O(\log^2 n)$ time in parallel using $O(m^3)$ processors on the CRCW PRAM, where n and m denote the numbers of vertices and edges, respectively. Besides, if the answer is "yes", the k-chain subgraphs can be constructed with the same time complexity and processor complexity.*

Corollary 13. *The k-chain subgraph cover problem on the convex bipartite graph belongs to not only the class P, but also the class NC.*

5. Conclusion and open problems

The containment relationships for some subclasses of bipartite graphs are known as follows [2]: bipartite permutation graphs $\subset$ doubly convex bipartite graphs $\subset$ convex bipartite graphs $\subset$ chordal bipartite graphs $\subset$ perfect elimination bipartite graphs $\subset$ bipartite graphs. The k-chain subgraph cover problem has been proved to be NP-complete for the bipartite graph. In this paper we further show that it belongs to the class P (also in the class NC) for the convex bipartite graph. The immediate open problem resulting from this paper is to decide whether the k-chain subgraph cover problem is solvable in polynomial time for chordal bipartite graphs.

References

[1] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (1976) 335–379.

[2] A. Brandstadt, J. Spinrad, L. Stewart, Bipartite permutation graphs are bipartite tolerance graphs, Congr. Numer. 58 (1987) 165–174.

[3] V. Chvatal, Perfectly ordered graphs, Ann. Discrete Math. 21 (1984) 63–65.

[4] L. Chen, Logarithmic time NC algorithms for comparability graphs and circle graphs, Lecture Notes in Computer Science: Advances in Computing and Information, vol. 497, 1991, pp. 383–394.

[5] L. Chen, Y. Yesha, Parallel recognition of the consecutive ones property with applications, J. Algorithms 12 (1991) 375–392.

[6] R. Cole, Parallel merge sort, SIAM J. Comput. 17 (1988) 770–785.

[7] M.B. Cozzens, R. Leibowitz, Multidimensional scaling and threshold graphs, J. Math. Psychol. 31 (1987) 179–191.

[8] P. Damaschke, H. Muller, D. Kratsch, Domination in convex and chordal bipartite graphs, Inform. Process. Lett. 36 (1990) 231–236.

[9] E. Dekel, S. Sahni, A parallel matching algorithm for convex bipartite graphs and application for scheduling, J. Parallel Distrb. Comput. 1 (1991) 185–205.

[10] G. Ding, Covering the edges with consecutive sets, J. Graph Theory 15 (5) (1991) 559–562.

[11] A. Gibbons, W. Rytter, Efficient Parallel Algorithms, Cambridge University Press, Cambridge, 1988.

[12] P.C. Gilmore, A.J. Hoffman, A characterization of comparability graphs and of interval graphs, Can. J. Math. 16 (1964) 539–548.

[13] F. Glover, Maximum matching in a convex bipartite graph, Naval Res. Logist. Quart. 14 (1967) 313–316.

[14] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1981.

[15] J.W. Lipski, F.P. Preparata, Efficient algorithms for finding maximum matching in convex bipartite graphs and related problems, Acta Inform. 15 (1981) 329–346.

[16] T. Ma, J. Spinrad, On the 2-chain subgraph cover and related problems, J. Algorithms 17 (1994) 251–268.

[17] M. Yannakakis, Node-deletion problems on bipartite graphs, SIAM J. Algebraic Discrete Methods 10 (2) (1981) 310–327.

[18] M. Yannakakis, The complexity of the partial order dimension problem, SIAM J. Algebraic Discrete Methods 3 (1982) 351–358.