A Fully Dynamic Recognition Algorithm for Weak-bisplit graphs.

J.M. Vanherpe, S. ZITI
Université d'Orléans, LIFO

Rapport N^o **RR 2005-03**

Abstract

We provide efficient fully dynamic recognition algorithms involving vertex or edge insertions or deletions for the class of Weak-bisplit graphs, a class of bipartite graphs closely related to cographs and split graphs.

1 Introduction

Dynamic graph algorithms are algorithms that maintain properties of a graph while the graph is changing. These algorithms are potentially useful in a number of application areas, including communication networks, VLSI design, distributed computing, and graphics. The problem of dynamically recognizing a graph Property Π is to determine if a given graph satisfies the Property under repeated vertex or edge operations and to maintain a characteristic representation of the graph as long as the Property Π holds.

More precisely, when a graph $G = (V, E)$ satisfying Property Π and a representation of G are given the concerned vertex or edge operations are described below.

Vertex Insertion Given a new vertex $x \notin V$ and a set E_x of edges between x and vertices of G, determine whether the graph $G' = (V \cup \{x\}, E \cup E_x)$ satisfies Property Π. If yes give a representation of the graph G' or stop otherwise.

Vertex Deletion Given a vertex $x \in V$ and the set E_x of edges of G incident to x, determine whether the graph $G' = (V\backslash\{x\}, E\backslash E_x)$ satisfies Property Π. If yes give a representation of the graph G' or stop otherwise

Edge Insertion Given a nonedge $xy \notin E$ determine whether the graph $G' = (V, E \cup \{xy\})$ satisfies Property Π. If yes give a representation of the graph G' or stop otherwise.

Edge Deletion Given a edge $xy \in E$ determine whether the graph $G' = (V, E\backslash\{xy\})$ satisfies Property Π. If yes give a representation of the graph G' or stop otherwise.

Many researchers have investigated this problem, for example Demetrescu and Italiano present in [5], the first fully dynamic algorithm for maintaining all pairs shortest paths in directed graphs with real-valued edge weights and King and Sager dynamically maintain the transitive closure of a directed graph (see [12]).

In a large scale of applications Property Π is to belong to some graph family. Hence in [11] Ibarra gives a dynamic algorithm for the recognition of chordal graphs which performs only edge operations within $O(n)$ time per operation where n denotes the number of vertices of the graph. For proper interval graphs, in [10], Hell, Shamir and Sharon handle each update in $O(d + logn)$ where d is the number of edges involved in a edge operation and the degree of the vertex involved in a vertex operation.

Complement reducible graphs (*Cographs* for short) were designed in [2] as the graphs that can be decomposed into single vertices by recursively considering the connected components of the graph or those of the complementary graph. In this paper it is shown that a cograph admits a unique tree representation, called the *cotree*, which is the framework for algorithmic solutions of many optimization problems known to be NP-hard for general graphs. Using the cotree as a data structure for representing a cograph, Corneil, Perl and Stewart provide in [3] a linear time recognition algorithm for cographs using only vertex insertions. This algorithm was extended by Shamir and Sharan in [14] to a fully dynamic recognition algorithm, each operation being done within $O(d)$ time complexity where d denotes the number of edges involved in the operation. More recently, Crespelle and

PAUL proposed a fully dynamic recognition algorithm for *directed* cographs within the same time complexity per operation (see [4]).

Bipartite graphs form a important model for a large number of applications and adapting general graph results to the bipartite case has motivated many researchers. For example MCKEE in [13] presents a bipartite version of various graph concepts, BRANDSTÄDT in [1] is interested to relations between chordal, strongly chordal, split graphs and bipartite graphs, while FROST ET AL. in [7] examine bipartite analogs of split graphs.

In this mind GIAKOUMAKIS and VANHERPE introduced in [8] the *Bi-cographs* graphs as the bipartite analogous of cographs and FOUQUET ET AL. proposed to extend the results of [8] and [7] in studying a wider class of graphs, namely the *Weak-bisplit graphs*. In [9] a linear time recognition algorithm for weak-bisplit graphs is given, this algorithm follows the ideas developed in [3] for Cographs recognition and uses an incremental approach since the vertices are inserted one by one in the graph. But, in order to handle each vertex insertion in $O(d)$ time, where d is the degree of the inserted vertex, a preprocessing is done on the vertices, thus in the form proposed in [9] this algorithm is not fully dynamic.

In this paper we propose a fully dynamic recognition algorithm for weak-bisplit graphs. In this algorithm each vertex insertion is done in $O(n)$ time, here n denotes the number of vertices of the graph, each vertex deletion takes a constant time while an edge operation takes $O(d(x) + d(y))$ where $d(x)$ and $d(y)$ denote the degree of the endpoints of the edge involved in the operation.

The paper is organized as follows. Section 2 below gives the required definitions, notations and background information. Sections 3 and 4 respectively deal with vertex and edge operation.

2 Basics, notations and background information

In the following we only deal with simple undirected loopless graphs .

A bipartite graph $G = (V, E)$ is a graph whose chromatic number is at most 2, G is then also denoted by $G = (B, W, E)$ where B stands for the set of *black* vertices while the vertices of W are said to be *white*; when B nor W is the empty set G is said to be be-chromatic (*mono-chromatic* otherwise). G is said to be *complete bipartite* when E contains all possible edges connecting a vertex of B to a vertex of W. A *biclique* is a set of vertices inducing a complete bipartite graph. Given a bipartite graph $G = (B \cup W, E)$, the bi-complemented graph of G is defined by $\overline{G}^{bip} = (B, W, B \times W \backslash E)$.

If x and y are two vertices of G such that $xy \in E$, x and y are said *adjacent* or neighbors, *independent* or non-neighbors otherwise. For a vertex x, the *degree* of x, denoted $d(x)$ is the number of its neighbors. A vertex of G having no neighbor is said to be *isolated* and a vertex which is isolated in $\overline{G}^{bip}$ is said to be *universal* in G.

A graph having only isolated vertices is said to be *edgeless* while the graph without any vertex is the *empty graph*. A set of vertices inducing an edgeless graph is a *stable set*.

A chord-less path on k vertices is called a P_k and is denoted $x_1x_2 \ldots x_k$, the vertices x_1 and x_k are the *endpoints* of the P_k.

2.1 $K + S$ graphs

In [6], FOUQUET, GIAKOUMAKIS and VANHERPE have defined the $K + S$ graphs as the bipartite graphs having either an isolated vertex or allowing a decomposition of their vertex set into a biclique and a stable set. Let $G = (B, W, E)$ be a $K + S$

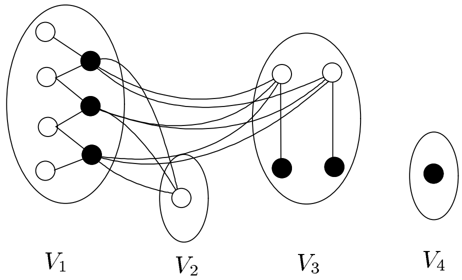

Figure 1: A $K+S$ graph and its $K+S$-decomposition (V_1, V_2, V_2, V_4).

graph, an *associated partition* of G, namely (V_1, V_2), is an ordered partition of the vertex set of G into two non empty classes V_1 and V_2 such that the black vertices of V_1 are connected to the white vertices of V_2 while the white vertices of V_1 are independent of the black vertices of V_2.

Proposition 1 [6] Let $G = (V, E)$ be a bipartite graph that is not a $K+S$ graph. Then any vertex of G belongs to a $2K_2$ of G.

The following decomposition theorem is proved in [6].

Theorem 2 Let G be a bipartite graph of order at least 2. G is a $K+S$ graph if and only if G admits a unique (up to isomorphism) partition into non empty sets of its vertex set, say $(V_1, V_2, \ldots, V_k)$ $(k > 1)$, satisfying :
1) $\forall i = 1, \ldots k-1$, $(V_1 \cup \ldots V_i, V_{i+1} \cup \ldots V_k)$ is an associated partition to the graph G.
2) $\forall i = 1, \ldots k$, V_i the subgraph induced by V_i is not a $K+S$ graph.

The partition $(V_1, V_2, \ldots, V_k)$ $(k > 1)$ is said to be the $K+S$-decomposition of G while the sets $V_1, \ldots, V_k$ are the *$K+S$-components* and can be found in $O(|V(G)|)$ time once the degree sequences of the vertices of G are given (see an example in Figure 1).

2.2 Canonical decomposition of bipartite graphs

The $K+S$-decomposition together with the decomposition of a bipartite graph G following its connected components (parallel decomposition) or those of $\overline{G}^{bip}$ (series decomposition in the bipartite sense) led in [6] to a decomposition scheme for G, called *canonical decomposition*.

In [6], it is shown that a unique set of indecomposable (or *prime*) graphs with respect to canonical decomposition is obtained independently of the order in which $K+S$, parallel and series decomposition are recursively applied to G.

Obviously, a tree $T(G)$ is associated to this decomposition. This tree contains $O(|V(G)|)$ nodes which correspond to subgraphs of G. The internal nodes are labelled by the type of decomposition applied, while its leaves correspond to indecomposable graphs.

Thus, there is 3 different types of internal nodes : Parallel (labelled 0), Series (labelled 1) or $K+S$ (labelled 2). An internal node having label i, $i \in \{0, 1, 2\}$ will be called a i-node.

In order to obtain a unique (up to isomorphism) decomposition tree, the recursive process for canonical decomposition of a bipartite graph G described in Algorithm Canonical Decomposition will be used.

Algorithm Canonical Decomposition
Input: A bipartite graph G
Output: The canonical decomposition tree $T(G)$.

if G is a $K+S$ graph **then** $\{K+S$-decomposition$\}$
 decompose G into its ordered $K+S$ components
else if G is not connected **then** {Parallel decomposition}
 decompose G into its connected components
else if $\overline{G}^{bip}$ is not connected **then** {Series decomposition}
 decompose G into the connected components of $\overline{G}^{bip}$
else
 G is indecomposable with respect to canonical decomposition.

This decomposition process implies some properties of the decomposition tree that we list now. In the following the meanings of usual terms like leaf, internal node, child,father, descendant, ancestor are as expected.

Observation 3 [6] Properties of the decomposition tree.

1. Two internal nodes having the same label cannot be consecutive.
2. The leaves are the vertices of the graph while internal nodes correspond to decomposition operations.
3. Internal nodes labelled 0 or 1 cannot have a trivial leaf. *Such a node would have either a universal or an isolated vertex and thus, would induce a* $K+S$ *graph.*
4. The father of a trivial leaf is labelled 2. *(consequence of Property 3.)*
5. If G is bi-chromatic then for any 2-node, say α, G_α is bi-chromatic too. *In fact the father of a* 2*-node corresponding to a mono-chromatic graph would have universal or isolated vertices and would induce a* $K+S$ *graph. A contradiction with property* 1.
6. The children of a 2-node are ordered following the order given by the $K+S$-decomposition (see Observation 2).
7. Let δ be a 2-node, If its father, say δ' is labelled 0 then
 a) The first $K+S$-component of δ cannot be a trivial white leaf.
 b) The last $K+S$-component of δ cannot be a trivial black leaf.
 (otherwise G_δ *would have isolated vertices and would induce a* $K+S$ *graph.*
8. Let δ be a 2-node, if its father, say δ', is labelled 1 then
 a) The first $K+S$-component of δ cannot be a trivial black leaf.
 b) The last $K+S$-component of δ cannot be a trivial white leaf.

Notations Let α be an internal node of $T(G)$.

Let $E=\{l_1,\dots l_k\}$ be a set of children of α in $T(G)$ and let G_i, $(i=1,\dots k)$ be the subgraph of G induced by the vertices of G having l_i as ancestor in $T(G)$. Then the graph induced by $V(G_1)\cup\ldots\cup V(G_k)$ will be denoted G_E, when all the children of α belong to E, G_E is simply denoted G_α while V_α denotes its vertex set and $T(G_\alpha)$ its canonical decomposition tree.

Given a vertex x of G_α the unique child of α in $T(G_\alpha)$ which is an ancestor of x is denoted $direct-descendant(\alpha,x)$.

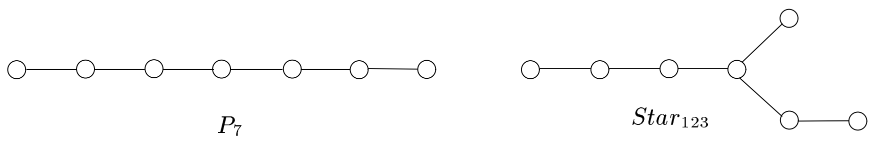

Figure 2: The forbidden configuration for weak-bisplit graphs.

2.3 Weak-bisplit graphs

The weak-bisplit graphs are defined in [6] to be the bipartite graphs that can be reduced to single vertices using canonical decomposition (i.e. the indecomposable graphs in a weak-bisplit graph are only trivial ones). Those graphs are characterized with two forbidden configurations (see Theorem 4 and figure 2).

Theorem 4 [6] Let G be a bipartite graph, G is a weak-bisplit graph if and only if G is $(P_7, Star_{123})$-free.

It is easy to see that the canonical decomposition tree of a weak-bisplit graph G is a characteristic representation of the graph. Moreover two vertices $b \in B$ and $w \in W$ of G are adjacent if an only if their least common ancestor in $T(G)$, say α is either a 1-node or a 2-node such that $direct - descendant(\alpha, w)$ succeeds $direct - descendant(\alpha, b)$.

3 Inserting or removing a vertex

In this section we consider the vertex insertion and vertex deletion operations. Let $G = (B \cup W, E)$ be a bi-chromatic weak-bisplit graph and $T(G)$ its canonical decomposition tree.

3.1 Vertex insertion

For this problem we consider a vertex x outside of G and a set of edges, say E_x, connecting x to some vertices of G, the vertex insertion operation is to determine whether the graph $G' = (B \cup W \cup \{x\}, E \cup E_x)$ remains to be a weak-bisplit graph and if yes to give the canonical decomposition tree of G'.

This is precisely the work done in the main step of the recognition algorithm for weak-bisplit graphs given in [9]. Unfortunately, for complexity reasons, this algorithm assumes that the vertices are inserted following the increasing order of their degree, thus this algorithm needs global information. However, without this assumption the main step of the algorithm given in [9] solves the vertex insertion problem within $O(|B \cup W|)$ time complexity.

3.2 Vertex deletion

When x is a vertex of G the deletion of x is to compute the subgraph of G induced by $B \cup W - \{x\}$, say G'. This will not create any forbidden configuration in G', in other words in any case G' remains to be a weak-bisplit graph. We give below the construction when G' remains to be bi-chromatic.

Let α be the father of x in $T(G)$, $T(G')$ is obtained by just removing x as a child of α. It may happen that G_α becomes mono-chromatic after this operation. Then we use the so-called *node-exchange* that is to exchange in $T(G)$ the node α together with its children with the father of α. It may also happen that this operation leaves α with a unique internal child, say β, in this case we apply the following *tree contraction*. The node α is deleted while β is set a new child of the father of α (if exists). When β and its new father have the same label, those two nodes are collapsed together.

Obviously, vertex deletion can be done in constant time.

4 Inserting or removing an Edge

Since removing an edge in a bi-chromatic weak-bisplit graph is equivalent to insert this edge in the bi-complemented graph we only consider here the edge insertion problem.

Let $G = (B \cup W, E)$ a bipartite bi-chromatic weak-bisplit graph and $T(G)$ its canonical decomposition tree, let $x \in B$ and $y \in W$ be two non adjacent vertices of G. Let's denote $G' = (B \cup W, E \cup \{xy\})$ the graph obtained by inserting the edge xy into G.

In this section we give an algorithm which determines whenever the graph G' remains to be a weak-bisplit graph, in this case our algorithm provides the canonical decomposition tree of G' as a by-product. Our algorithm works within $O(d(x) + d(y))$ time complexity.

We denote by α the least common ancestor to x and y in $T(G)$. All additional work for solving the edge deletion problem will be done in the subgraph G_α whose decomposition tree is the subtree of $T(G)$ rooted on α. For convenience let's set $\alpha_x = direct - descendant(\alpha, x)$ and $\alpha_y = direct - descendant(\alpha, y)$. Since x and y are not adjacent the internal node α cannot be a 1-node.

We discuss in the following on the type of the node α which can be either a 0-node or a 2-node.

4.1 The node α denotes a parallel decomposition

In this case G_{α_x} and G_{α_y} are non-trivial connected graphs (see Observation 3, Property 3).

Proposition 5 If G' is a weak-bisplit graph then α_x nor α_y can be a 1-node.

Proof Assume w.l.o.g that α_x is a 1-node. Let b in G_{α_y} such that yb is an edge.

Claim 1 The node α_x has exactly two children.

Proof Let α_1 and α_2 be two children of α_x which are distinct from $direct - descendant(\alpha_1, x)$. Let b_1, w_1 (resp. b_2, w_2) be independent vertices of G_{α_1} (resp. G_{α_2}) and let xw be a non-edge of $direct - descendant(\alpha_x, x)$. But now the set $\{x, w_2, b, y, w_1, b_2, w\}$ would induce a $Star_{123}$ in G', a contradiction. ∎

Let α_1 and α_2 be the two children of α_x. Assume w.l.o.g that x is a member of V_{α_1}, recall that $\overline{G_{\alpha_1}}^{bip}$ and $\overline{G_{\alpha_2}}^{bip}$ are connected. Let xw be a non edge of G_{α_1}.

Claim 2 The graph G_{α_2} has no edge.

Proof Assume not. Let b_2w_2 be an edge of G_{α_2}. Since $\overline{G_{\alpha_2}}^{bip}$ is connected there is at least 4 vertices on a chordless chain connecting b_2 to w_2 in $\overline{G_{\alpha_2}}^{bip}$, thus b_2 is an endpoint of a $\overline{P_4}^{bip}$ in this graph, namely $b_2w'_2b'_2w''_2$. The vertices x, w'_2, y, b, w''_2, w, b_2 induce a $Star_{123}$ in G_{α_x}, a contradiction. ∎

Let b_2w_2 be a non edge of G_{α_2}.

Claim 3 The graph G_{α_1} has no edge.

Proof Let's first state the following fact.

Fact 1 The vertex x has no neighbor in G_{α_1}.

Proof Let xw be an edge in G_{α_1}. Since $\overline{G_{\alpha_1}}^{bip}$ is connected x is an endpoint of some $\overline{P_4}^{bip}$ in this graph, namely xw_1b_1w'. Then the set $\{x, w_2, y, b, w', b_2, w_1\}$ where b_2 and w_2 are non adjacent vertices of G_{α_2} induces a $Star_{123}$, a contradiction. $\diamond$

Suppose now that b_1w_1 is an edge in G_{α_1}. The vertices b, y, x, b_1, w_1 together with two independent vertices of G_{α_2} would induce a P_7, a contradiction that proves the Claim. ■

It follows from Claim 2 and Claim 3 that G_{α_x} has two connected components, a contradiction with the fact that G_{α_x} is a connected graph. □

Proposition 6 If G' is a weak-bisplit graph then the node α_x is the father of x and the node α_y is the father of y.

Proof By Observation 3 (Property 1) and Proposition 5, α_x denotes a $K + S$ decomposition. Suppose that α_x is not the father of x, we state $\beta_x = direct-descendant(\alpha_x, x)$.

We know by Proposition 1 that x belongs to a $2K_2$ of G_{β_x}, namely xw_1, b_2w_2.

When β_x is not the last child of α_x, there must exist a white vertex, say w in this last child (Observation 3 Property 7). But now, the vertices of the $2K_2$ xw_1, b_2w_2 together with w, y and b would induce a $Star_{123}$ in G_α, a contradiction.

When β_x is the last child of α_x, there must exist a black vertex, say b', in the first child of α_x (Observation 3 Property 7). In this conditions $byxw_1b'w_2b_2$ would be a P_7 in G_α, a contradiction.

We can prove in a similar way that α_y is the father of y. □

When G' is a weak-bisplit graph the vertices x and y are respectively children of α_x and α_y. In this situation we denote by G_1 (resp. G_2) the graph induced by the vertices of G_{α_x} having a child of α_x which precedes x (resp. y) as ancestor. We denote by G'_1 (resp. G'_2) the graph induced by the vertices of G_{α_x} having a child of α_x which succeeds x (resp. y) as ancestor (see figure 3).

Since G_{α_x} and G_{α_y} are both connected by Observation 3, Properties 7 and 8 we know that all white (resp. black) vertices of G_1 (resp. G'_2) have a neighbor in G_1 (resp. G'_2). Moreover G'_1 contains at least one white vertex and G_2 contains at least one black vertex. Let w'_1 be a white vertex of G'_1 and b_2 be a black vertex of G_2, we have that b_2y is an edge of G_{α_y} and xw'_1 is an edge of G_{α_x}.

Proposition 7 If G' is a weak-bisplit graph then the following hold :

- G_1 has no white vertex or G'_2 is empty.
- G'_2 has no black vertex or G_1 is empty.

Proof Let w_1 be a white vertex of G_1 and b_1 be a neighbor of w_1 in G_1.

Claim G'_2 has no white vertex.

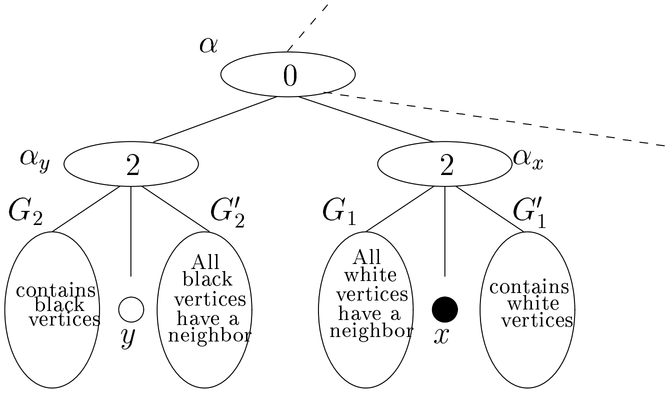

Figure 3: A description of $T(G)$ when α is a 0-node.

Proof A white vertex of G_2' together with b_2, y and the following vertices of G_{α_x} x, w_1', b_1, w_1 would induce a P_7 in G', a contradiction. ■

It follows from this Claim that G_2' cannot have a black vertex since a such vertex would have a neighbor in G_2'. Thus G_2' is the empty graph as claimed.

On the same manner we can prove that G_2' has no black vertex or G_1 is empty.
□

We assume henceforth that the necessary conditions of Propositions 5, 6 and 7 are verified and we will now consider several cases according to the fact that G_1 has a white vertex or not and to the fact that G_2' has a black vertex or not.

4.1.1 Case : G_1 contains a white vertex

We suppose here that G_1 contains a white vertex.

Theorem 8 G' is a weak-bisplit graph if and only if G_1 is bipartite complete, G_1' has no black vertex, G_2 has no white vertex and G_2' is the empty graph.

Proof Assume first that G' is a weak-bisplit graph. We know by Proposition 7 that G_2' is the empty graph.

Claim 1 G_1 is a complete bipartite graph.

Proof Let b_1w_1 be a non edge in G_1, let b_1' be a neighbor of w_1 in G_1. But now the vertex set $\{w_1', x, b_1', w_1, b_1, y, b_2\}$ induces a $Star_{123}$ in G', a contradiction ■

Let w_1b_1 be an edge in G_1, a such edge must exist since G_1 contains a white vertex and all the white vertices of G_1 have a neighbor in G_1.

Claim 2 G_1' has no black vertex

Proof Let b be a black vertex of G_1', G_{α_x} being connected, b must have a neighbor, say w in G_1'. But the vertices w, b, b_1, w_1, x, y, b_2 would induce a $Star_{123}$ in G', a contradiction. ■

Claim 3 G_2 has no white vertex

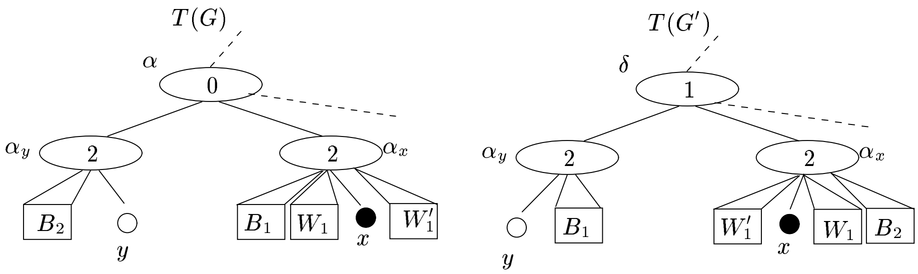

Figure 4: A construction of $T(G')$ when G_1 has white vertices.

Proof Let w be a white vertex of G_2, G_{α_y} being connected, w must have a neighbor, say b in G_2. But the vertices w, b, y, x, w_1', b_1, w_1 would induce a P_7 in G', a contradiction. ■

For the converse part of the Theorem we describe the canonical decomposition tree $T(G')$ of G'. This tree is obtained from $T(G)$ by inserting or deleting few internal nodes of $T(G_\alpha)$.

By assumption G_1 is a complete bipartite graph, let's denote B_1 its set of black vertices while W_1 denotes its set of white vertices, observe that B_1 and W_1 may be empty sets. Moreover, G_1' has only white vertices, let's denote W_1' its vertex set. The graph G_2 has only black vertices, let B_2 be its vertex set, recall that G_2' is the empty graph.

The decomposition tree $T(G')$ is obtained from $T(G)$ as follows :

Observe that B_1 is not an empty set, in fact G_2 is supposed to have white vertices and we know that every white vertex in G_2 has a neighbor in G_2.

Let δ be a new internal 1-node that replaces α, the children of α_y are y and the members of B_1 in this order while the children of α_x are set to the vertices of W_1, the vertex x, the vertices of W_1' and the vertices of B_2 in this order (see Figure 4). □

4.1.2 Case : G_2' has a black vertex

This situation is very similar to the case studied in section 4.1.1, thus we only give the recognition Theorem without its proof, the construction of $T(G')$ is depicted in Figure 5.

Theorem 9 G' is a weak-bisplit graph if and only if G_2' is bipartite complete, G_1' has no black vertex, G_2 has no white vertex and G_1 is the empty graph.

4.1.3 G_1 has no white vertex and G_2' has no black vertex

In this case we denote by B_1 the vertex set of G_1 and by W_2' the vertex set of G_2'.

Theorem 10 G' is a weak-bisplit graph if and only if one of the following conditions holds.

1. The graph G_1' is complete bipartite or G_2 has no white vertex and G_2' is the empty graph.

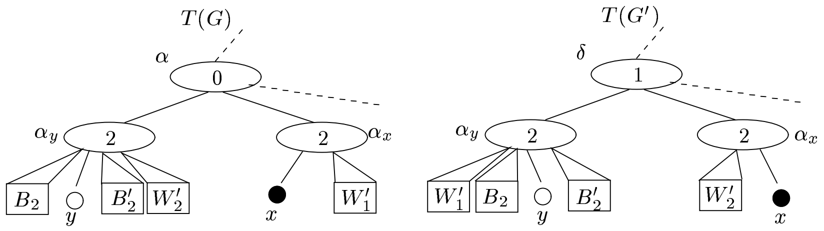

Figure 5: A construction of $T(G')$ when G'_2 has black vertices.

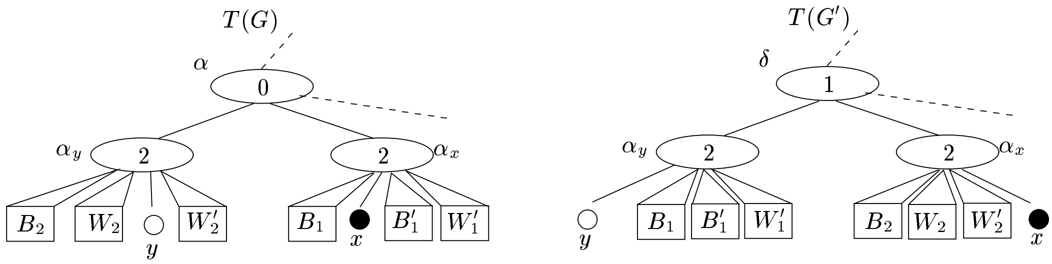

Figure 6: A construction of $T(G')$ when G'_1 and G_2 are complete bipartite.

2. The graph G_2 is complete bipartite or G'_1 has no white vertex and G_1 is the empty graph.

Proof When both G'_1 and G_2 are complete bipartite graphs the construction of $T(G')$ is described in Figure 6.

Without loss of generality we suppose now that G'_1 is not a complete bipartite graph. A very similar proof holds when G_2 is not complete bipartite. Let b'_1 and w'_1 be two non adjacent vertices in G'_1, G_{α_x} being connected, b_1 must have a neighbor, say w''_1 in G'_1. If G_2 has a white vertex or if G'_2 is not empty there must be in G_{α_y} a white vertex w distinct from y. But now a black member of G_2 together with y and w and the vertices x, w'_1, b'_1 and w''_1 would induce a $Star_{123}$ in G', a contradiction.

Conversely, let's describe the construction of $T(G')$ in Figure 7 where G'_1 is not complete bipartite, in this case B_2 denotes the vertex set of G_2. The construction when G_2 is not complete bipartite holds similarly. □

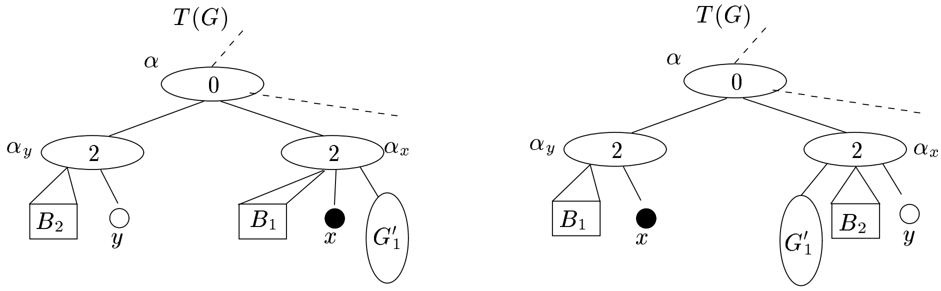

Figure 7: A construction of $T(G')$ when G'_1 is not complete bipartite.

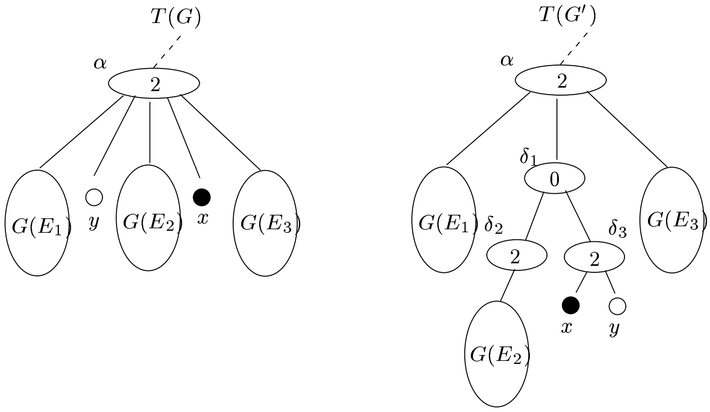

Figure 8: A construction of $T(G')$ when α is the father of x and y.

4.2 The node α denotes a $K + S$ decomposition

The vertices x and y being independent in G, it follows that $direct-descendant(\alpha, y)$ precedes $direct - descendant(\alpha, x)$.

Proposition 11 If G' is a weak-bisplit graph then x or y is a child of α.

Proof Assume on the contrary that G_{α_x} nor G_{α_y} are trivial graphs. V_{α_x} and V_{α_y} being non-trivial $K + S$ components of G_α, by Proposition 1, we have on one hand that the vertex x belongs to a $2K_2$ of G_{α_x}, namely xw_x, $b_xw'_x$; on the other hand the vertex y belongs to a $2K_2$ in G_{α_y}, say yb_y, $w_yb'_y$. But now the set $\{b_y, w_y, w'_x, b_x, w_x, x, y\}$ induces a $Star_{123}$ in G', a contradiction. □

Let's give the construction of $T(G')$ when x and y are both children of α. Let E_1 be the set of children of α that precede y, E_2 the set of children of α that succeed y and precede x and let E_3 be the set of children of α that succeed x. Figure 8 describes the construction of $T(G')$ in this conditions : we create one new internal 0-node, namely δ_1 and two new internal 2-nodes say δ_2 and δ_3. The node δ_1 is set as a child of α in replacement of $E_2 \cup \{y, x\}$ while the children of δ_2 are the members of E_2 and the children of δ_3 are set to x and y in this order. When E_2 is a singleton we apply a tree contraction as described in Section 3.2.

We suppose in the following that x and y have not the same father and without loss of generality we assume that α is the father of y and we denote by β the father of x.

Definition 12 Let γ be an ancestor of β in $T(G_\alpha)$, the node γ is said to be *1-ancestor-less* when no ancestor of γ in $T(G_\alpha)$ denotes a series decomposition (i.e. is labelled 1).

Observe that any ancestor of a *1-ancestor-less* node in $T(G_\alpha)$ is itself *1-ancestor-less.* In particular α is *1-ancestor-less.*

Notation 13 Let γ be a *1-ancestor-less* 2-node of $T(G_\alpha)$.

When $\gamma = \alpha$ we denote by C_γ the set of children of γ that precede $direct - descendant(\gamma, x)$ and that succeed y.

When $\gamma \neq \alpha$ we denote by C_γ the set of children of γ that precede $direct - descendant(\gamma, x)$.

Let's denote G_{C_γ} the subgraph of G_γ induced by the vertices which have an ancestor in C_γ.

Observation 14 When γ is a *1-ancestor-less* 2-node distinct from α, its father in $T(G_\alpha)$ denotes a parallel decomposition, consequently a non empty G_{C_γ} must contain black vertices, otherwise G_γ would not be connected. If on the contrary G_{C_γ} is the empty graph, $direct - descendant(\gamma, x)$ is the first child of γ.

Proposition 15 If G' is a weak-bisplit graph then there is at most one 2-node γ which is *1-ancestor-less* and such that G_{C_γ} has black vertices, in addition G_{C_γ} is an edgeless graph.

Proof Let γ and γ' be distinct 2-nodes, both of them being *1-ancestor-less*. Suppose that γ' is an ancestor of γ, let b' be a black vertex in $G_{C_{\gamma'}}$ and b be a black vertex in G_{C_γ}. Let δ be the father of γ, we consider b_1 and w_1 two adjacent vertices of G_δ that do not belong to G_γ. We also consider w, a neighbor of x in G_γ (Recall that G_γ is connected), by construction of G_{C_γ}, the vertex b is adjacent to w. Thus the set $\{w, b, x, y, b', w_1, b_1\}$ induces a $Star_{123}$ in G', a contradiction.

When γ is a *1-ancestor-less* 2-node such that G_{C_γ} contains an edge, say bw, by Proposition 1, x belongs to a $2K_2$ in the subgraph rooted at direct-descendant(γ, x). The vertices of this $2K_2$ together with b, w and y would induce a $Star_{123}$ in G', a contradiction. □

Proposition 16 Let γ be a *1-ancestor-less* 2-node distinct from α such that $C_\gamma \neq \emptyset$. Then C_γ has only black vertices and $direct - descendant(\gamma, x)$ is the last child of γ.

Proof The node γ being *1-ancestor-less* G_γ is connected, thus a white vertex of C_γ must have a neighbor in C_γ, a contradiction since by Proposition 15 C_γ is an edgeless graph.

If $direct - descendant(\gamma, x)$ is not the last child of γ, we denote by w a white vertex in the last child of γ (see Observation 3, Property 7). By Proposition 1, x belongs to a $2K_2$ in the subgraph rooted at direct-descendant(γ, x). The vertices of this $2K_2$ together with a black member of G_{C_γ}, w and y would induce a $Star_{123}$ in G', a contradiction. □

We suppose henceforth that the necessary conditions of Propositions 15 and 16 are verified and we discuss on the existence of a 1-node among the ancestors of x in $T(G_\alpha)$.

4.2.1 Case : No ancestor of x in $T(G_\alpha)$ denote a series composition

Recall that β denotes the father of x. By Observation 3, Property 4, β is a 2-node, consequently any ancestor of β in $T(G_\alpha)$ is a *1-ancestor-less* 2-node.

Theorem 17 The graph G' is a weak-bisplit graph if and only if one of the following holds.

1. For any *1-ancestor-less* 2-node γ, G_{C_γ} has no black vertex.
2. The vertex x is universal in G_β.

Proof Let γ be a *1-ancestor-less* 2-node and let b be a black vertex in G_{C_γ}. Suppose in addition that x has a non-neighbor in G_β, say w. G_β being connected there must exist in this graph a neighbor of x say w'. Let δ be the father of β in

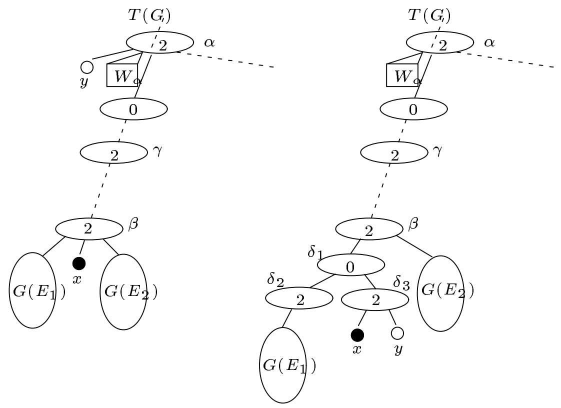

Figure 9: A construction of $T(G')$ when x is not a universal vertex of G_β.

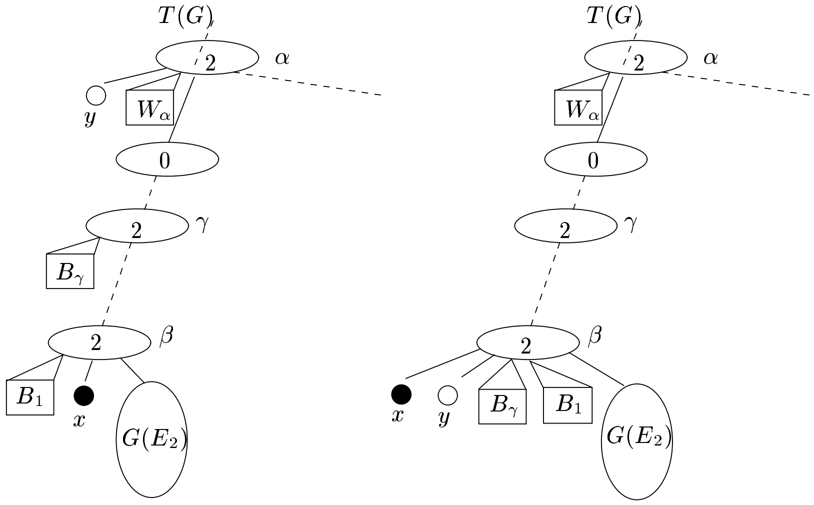

Figure 10: A construction of $T(G')$ when x is a universal vertex of G_β.

$T(G_\alpha)$, we consider b_1 and w_1 two adjacent vertices of G_δ that do not belong to G_β. The set $\{b, w, w_1, b_1, w', x, y\}$ induces a $Star_{123}$ in G', a contradiction.

Conversely let E_1 be the set of children of β that precede x and E_2 be its set of children that succeed x.

If the subgraph $G(E_1)$ contains a white vertex, x is not universal in G_β and then for any 2-node γ ancestor of β in $T(G_\alpha)$ we have $C_\gamma = \emptyset$. For the construction of $T(G')$ in this case we create three new internal nodes namely δ_1 which is labelled 0, δ_2 and δ_3 which are labelled 2 (see Figure 9).

If on the other hand the vertex set of $G(E_1)$ is reduced to a set of black vertices, namely B_1, we possibly have one 2-node ancestor of β, say γ such that C_γ is not empty. We know by Proposition 15 that G_{C_γ} is an edgeless graph, we denote by B_γ its set of black vertices while W_γ denotes its set of white vertices; observe that W_γ can be not empty only when $\gamma = \alpha$. The construction of $T(G')$ where y, the members of B_γ and those of B_1 become children of β is depicted in Figure 10. $\square$

4.2.2 Case : some ancestor of x in $T(G_\alpha)$ denotes a series composition

Let θ be a such node.

Theorem 18 The graph G' is a weak-bisplit graph if and only if the following hold.

1. θ is the father of β.
2. The graph G_β is an edgeless graph.
3. The vertex x is the only black vertex in G_β.
4. If γ is a *1-ancestor-less* 2-node of $T(G_\alpha)$ such that C_γ has black vertices then γ is the father of θ in $T(G_\alpha)$.

Proof For convenience we denote here $direct-descendant(\theta, x)$ by θ_x. Let b and w be two non adjacent vertices in G_θ that do not belong to G_{θ_x}.

Claim 1 The vertex x has no neighbor in G_{θ_x}.

Proof If x has a neighbor w' in G_{θ_x}, this graph being co-connected there must be a $\overline{P_4}^{bip}$, say $xw_1b_1w_2$ on a chord-less path connecting x to w' in $\overline{G_{\theta_x}}^{bip}$. Now we have a $Star_{123}$ induced in $T(G')$ by the vertices of $\{x, y, w, b_1, w_2, b, w_1\}$, a contradiction. ■

The vertex x being isolated in G_{θ_x}, θ_x must be the father of x and thus θ is the father of β. It follows that θ is the unique 1-node among the ancestors of β in $T(G_\alpha)$.

Claim 2 G_β is an edgeless graph.

Proof Let $b'w'$ be an edge in G_β, we know by Claim 1 that b' is distinct from x (recall that $\beta = \theta_x$). Since G_β is co-connected, b' must be an endpoint of some $\overline{P_4}^{bip}$, namely $b'w_1b_2w_2$ in G_β. The path $yxwb'w_2bw_1$ is a P_7 of G', a contradiction. ■

Claim 3 The vertex x is the only black vertex in G_β.

Proof Assume not, let b be a black member of of G_β distinct from x. By Property 5 in Observation 3 there must be a white vertex say w in G_β.

Fact θ has exactly two children in $T(G_\alpha)$.

Proof Let θ_1 and θ_2 be two children of θ distinct from β. Let b_1 and w_1 be two independent vertices of G_{θ_1} while b_2 and w_2 are independent vertices of G_{θ_2}. We have a $Star_{123}$ induced in $T(G')$ by the vertices w_2, b, x, y, b_1, w, b_2, a contradiction. ◇

Let θ' be the second child of θ. Since G_β is an edgeless graph there must be an edge in $G_{\theta'}$ otherwise G_θ would not be connected, thus this graph contains an induced $\overline{P_4}^{bip}$. The vertices of this $\overline{P_4}^{bip}$ together with b, x and y would induce a $Star_{123}$, a contradiction. ■

Recall that θ and all of its ancestors in $T(G_\alpha)$ are *1-ancestor-less*, let γ be a 2-node among the ancestors of θ in $T(G_\alpha)$ such that C_γ has black vertices, let b be a such vertex in G_{C_γ}.

Claim 4 γ is the father of θ.

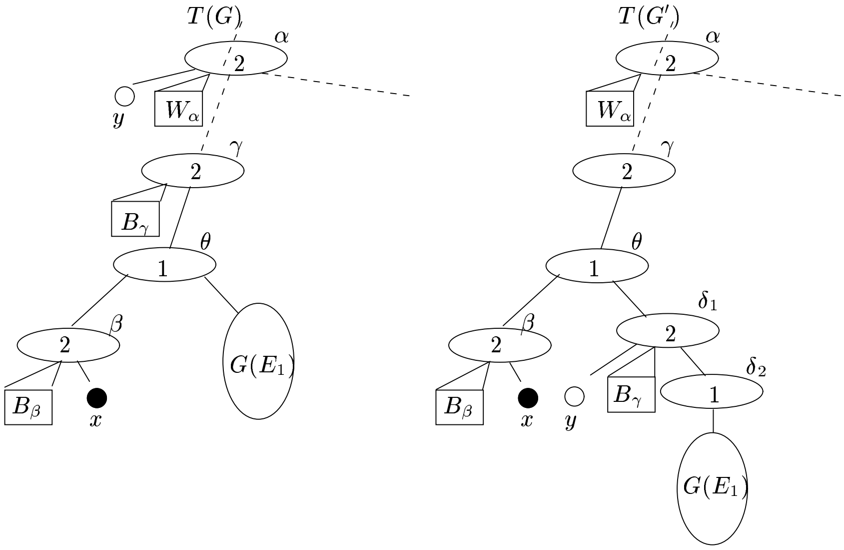

Figure 11: A construction of $T(G')$ when the father of G_β is a 1-node.

Proof Assume not. We denote γ_x the node $direct - descendant(\gamma, x)$, let b_1 and w_1 be two adjacent vertices of G_{γ_x} which do not belong to the subtree of $T(G_{\gamma_x})$ rooted on $direct - descendant(\gamma_x, x)$. Moreover, x belongs to a $2K_2$ of G_δ, namely xw, $w'b'$. We have a $Star_{123}$ induced in $T(G')$ by the vertices b, w', w_1, b_1, w, x, y, a contradiction. ∎

Conversely, we consider the set of black vertices of G_{C_γ}, namely B_γ, the set W_α of white vertices of G_{C_α} while W_β denotes the whites vertices of G_β and E_1 is for the set of children of θ distinct from β. Recall that the sets W_α or B_γ may be empty (see Figure 11).

The construction of $T(G')$ follows those different steps. We create a new internal 2-node δ_1 and another internal 1-node δ_2, E_1 becomes the set of children of δ_2 while the children of δ_1 are set to y, the black vertices of B_γ and δ_2 in this order, finally δ_1 becomes a child of θ in replacement of the members of E_1. □

4.3 The algorithm

We are now in a position to give our algorithm for the edge insertion problem. This algorithm takes as input the weak-bisplit graph G, its canonical decomposition tree $T(G)$ and the two independent vertices x and y of G.

The first step of the algorithm is to determine the least common ancestor to x and y in $T(G)$, say α. Then following the case distinction given in sections 4.1 and 4.2 the algorithm checks whether the necessary conditions of Propositions 5, 6, 7, Theorems 8, 9, 10, Propositions 11, 16, 15 and Theorems 17, 18 are verified. In case of success the algorithm ends with the construction of $T(G')$ as described in Theorems 8, 9, 10, 17 or 18.

For an algorithmic complexity point of view the most costly work is to find the node α in $T(G)$, all additional stuff being done on the path from x (or y) to α.

A simple analysis of the structure of $T(G)$ shows that α is not so far from x nor y. More precisely, in section 4.1 the nodes α must be the grand-father of x and y in $T(G)$ and thus can be found in constant time. In section 4.2 the distance from α to x linearly depends of the number of *1-ancestor-less* 2-nodes in the path from x to α, those nodes being connected they must have a white vertex into their last child, by construction this white vertex is a neighbor of x. Consequently the distance from x to α linearly depends of the degree of x.

Finally, when considering the symmetric case the whole algorithm runs within $O(d(x) + d(y))$ time complexity.

References

[1] A. Brandstädt. Classes of bipartite graphes related to chordal graphs. *Disc. Appl. Math.*, 32:51–60, 1991.

[2] D.G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981.

[3] D.G. Corneil, Y. Perl, and L.K Stewart. A linear recognition algorithm for cographs. *SIAM Journal of computing*, 14:926–934, 1985.

[4] C. Crespelle and C. Paul. Fully dynamic recognition algorithm and certicate for directed cographs. In Lecture Notes in Computer Science, editor, *30th International WorkShop on Graph Theoretical Concepts in Computer Science (WG'04)*, 2004.

[5] C. Demetrescu and G.F. Italiano. Fully dynamic all pairs shortest paths with real edge weights. In *42st Annual IEEE Symposium on Foundations of Computer Science (FOCS'01)*, October 14-17 2001.

[6] J.L. Fouquet, V. Giakoumakis, and J.M. Vanherpe. Bipartite graphs totally decomposable by canonical decomposition. *International Journal of Fundations of Computer Science*, 10(4):513–534, 1999.

[7] H. Frost, M. Jacobson, J. Kabell, and F.R. McMorris. Bipartite analogues of split graphs and related topics. *ARS Combinatorica*, 29:283–288, 1990.

[8] V. Giakoumakis and J.M. Vanherpe. Bi-complement reducible graphs. *Advances in Applied Mathematics*, 18:389–402, 1997.

[9] V. Giakoumakis and J.M. Vanherpe. Linear time recognition and optimization for weak-bisplit graphs, bi-cographs and bipartite P_6-free graphs. *International Journal of Fundations of Computer Science*, 14(1):107–136, 2003.

[10] P. Hell, R. Shamir, and R. Sharan. A fully dynamic algorithm algorithm for recognizing and representing proper interval graphs. *SIAM Journal on Computong*, 31(1):289–305, 2002.

[11] L. Ibarra. Fully dynamic algorithms for chordal graphs. In *10th ACM-SIAM Annual Symposium on Discrete Algorithms (SODA '03)*, pages 923–924.

[12] V. King and G. Sager. A fully dynamic algorithm for maintaining the transitive closure. *Journal of Computer and System Sciences archive*, 65(1):150 – 167, 2002.

[13] T.A. McKee. Bipartite analogs of graph theory. *Congressus Numerantium*, 60:261–268, 1987.

[14] R. Shamir and R. Sharan. A fully dynamic algorithm for modular decomposition and representation of cographs. *Discrete Appied Mathematics*, 136(2-3):329–340, 2004.