

O conteúdo do presente relatório é de única responsabilidade do(s) autor(es).  
(The contents of this report are the sole responsibility of the author(s).)

**A Linear Time Algorithm for Binary  
Phylogeny using PQ-Trees**

*João Meidanis*      *Erasmio G. Munuera*

**Relatório Técnico DCC-95-21**

Dezembro de 1995

# A Linear Time Algorithm for Binary Phylogeny using PQ-Trees

João Meidanis

Erasmus G. Munuera

## Abstract

The Binary Phylogeny Problem is to reconstruct a tree reporting the evolutionary history of a group of taxa for which a binary matrix of characteristics is given. Gusfield [5] presented an  $O(mn)$ -time algorithm for this problem with  $n$  taxa and  $m$  characteristics. This bound is tight if the input is given as an  $n \times m$  matrix. In this paper we show that a linear time algorithm is possible provided the input is given as a list of the “1” positions in the matrix. The PQ-trees introduced by Booth and Leuker [1] are used here. More precisely, we show that a binary phylogeny exists if and only if the input admits a PQ-tree without Q nodes. This immediately gives a linear time algorithm for the problem.

*Keywords: combinatorial problems, phylogenetic trees, data structures, design of algorithms.*

## 1 Introduction

A *phylogenetic tree* gives an interpretation of the evolutionary history of a group of taxa. An important problem in science is to build such a tree from data relating the taxa. In one version of this problem, which we call the *Binary Phylogeny Problem*, the tree is constructed from a  $n \times m$  binary matrix where each row corresponds to a taxon and each column to a character, with “1” in position  $i, j$  if taxon  $i$  has character  $j$ . An  $O(nm^2)$  algorithm to solve this problem was proposed by Camin and Sokal [2]. Later, a simple property of the matrix allowed an  $O(n^2m)$  algorithm. Recently, Gusfield [5] presented an  $O(nm)$  algorithm.

The input to this problem can be represented in two ways: as a binary  $n \times m$  matrix or as a bipartite graph.

The matrix format needs  $\Omega(nm)$  space to store it, and in this case Gusfield’s upper bound is tight because an algorithm must look at all positions in the matrix before it can even decide whether there is a phylogeny or not.

The bipartite graph format is defined as follows. Let  $X$  be the set of lines of the matrix  $M$  and  $Y$  the set of columns. There is an edge between  $i$  and  $X$  and  $j \in Y$  if and only if  $M[i, j] = 1$ . This format requires only  $O(n + m + r)$  space, where  $r$  is the number of ones in  $M$ . If  $M$  is sparse this represents a big savings. Furthermore, for the bipartite graph format one can design a linear time algorithm, that is, one running in  $O(n + m + r)$  time. In this paper we show one such algorithm based on PQ-Trees.

PQ-trees were introduced by Booth and Leuker [1] to represent all permutations with certain properties and to test the consecutive ones property for a binary matrix. Later this structure was shown to have a large range of applications in combinatorial problems such as planarity testing [3], sequential arrangements of records [4], and VLSI design [6]. Here we use PQ-trees to solve the perfect phylogeny problem.

The rest of this paper is organized as follows. In Section 2 we present some definitions used later on and a brief explanation on PQ-trees. In Section 3 we present the main results. Concluding remarks appear in Section 4.

## 2 Definitions

A **binary matrix** is a matrix whose entries are either 0 or 1. In this paper, we will always regard a binary matrix  $M$  as a taxa versus characters matrix, with  $M[i, j] = 1$  if and only if taxon  $i$  has character  $j$ . We use the terms “row” and “taxon” interchangeably, as well as the terms “column” and “character”. We usually identify a column with the set of rows that have “1” in that column. A column  $A$  of  $M$  is **trivial** if either  $|A| = n$  or  $|A| \leq 1$ .

We say that a binary matrix  $M$  **has a phylogenetic tree** when there is a binary rooted tree  $T$  such that (1) the leaves of  $T$  are in 1–1 correspondence with the rows of  $M$ , and (2) for each column  $j$  of  $M$  the rows that have “1” in column  $j$  correspond exactly to the leaves in the subtree rooted at a certain node  $p(j)$  of  $T$ . We will use extensively the following well-known characterization of this property [5].

**Theorem 1** *A binary matrix  $M$  has a phylogenetic tree if and only if for each pair of columns  $A$  and  $B$  of  $M$  either  $A \subseteq B$  or  $B \subseteq A$  or else  $A \cap B = \emptyset$ .*

If  $p$  and  $q$  are tree nodes, the notation  $p \rightarrow q$  means  $p$  is the parent of  $q$ , and  $p \xrightarrow{*} q$  means  $p$  is an ancestor of  $q$  or  $p = q$ .

We say that a binary matrix  $M$  **has the consecutive ones property** if there is a permutation of the rows of  $M$  that leaves the “1”s consecutive in every column of  $M$ .

We end this section with a brief overview on PQ-trees.

Let  $U = \{a_1, a_2, \dots, a_m\}$  be a universal set. A **PQ-tree** over  $U$  will be a rooted tree where the leaves are elements  $a_i \in U$  and the internal (nonleaf) nodes are either P-nodes or Q-nodes. Internal nodes represent possible permutations of their children, as we will see shortly. A PQ-tree is **proper** if the following conditions hold:

1. Each  $a_i$  appears exactly once as a leaf.
2. Every P-node has at least two children.
3. Every Q-node has at least three children.

We will deal exclusively with proper PQ-trees.

The *frontier* of a PQ-tree  $T$  is a reading of the leaves of  $T$  from left to right. Notice that the order of children in a node is important.

Two PQ-trees are **equivalent** when one can be obtained from the other by applying transformations of the following kind a finite number of times:

1. permute arbitrarily the children of a P-node.
2. reverse the children of a Q-node.

We say that a permutation  $\alpha$  of the elements of  $U$  is **represented** in a PQ-tree  $T$  when there is a PQ-tree  $T'$  equivalent to  $T$  whose frontier is  $\alpha$ .

The following result from [1] relates PQ-trees to the consecutive ones property.

**Theorem 2** *A binary matrix  $M$  has the consecutive ones property if and only if there is a PQ-tree  $T$  over the set of rows of  $M$  such that the permutations that leave the ones consecutive in  $M$  are exactly the permutations represented in  $T$ .*

If  $M$  has the consecutive ones property, the tree  $T$  mentioned in the above result is called a **PQ-tree of  $M$** . The PQ-tree of  $M$  is unique up to tree equivalence [1].

### 3 Algorithm

In this section we will present a linear time algorithm for the phylogeny problem, based on PQ-trees. We begin with an important result.

**Theorem 3** *A binary matrix  $M$  has a phylogenetic tree if and only if  $M$  has the consecutive ones property and its PQ-tree has internal nodes of type P only.*

*Proof:* ( $\Rightarrow$ ) Suppose that  $M$  has a phylogenetic tree. We need to show that  $M$  has a PQ-tree with all internal nodes of type P.

Gusfield [5] shows that if  $M$  has a phylogenetic tree then  $M$  has the consecutive ones property for columns. It remains to prove that the PQ-tree of  $M$  has internal nodes of type P only. We do this by induction on the number of nontrivial columns of  $M$ .

**Base Case:**  $M$  has zero nontrivial columns. In this case, any permutation of rows will leave the ones consecutive in all columns. Hence, the PQ-tree of  $M$  will have a single internal node of type P with all leaves as children.

**General Case:**  $M$  has at least one nontrivial column. Let  $H$  be any nontrivial column of  $M$ . We build two matrices  $M_1$  and  $M_2$  from  $M$  in such a way that a tree of  $M$  can be obtained for composing the trees of  $M_1$  and  $M_2$ , as shown in Figure 1

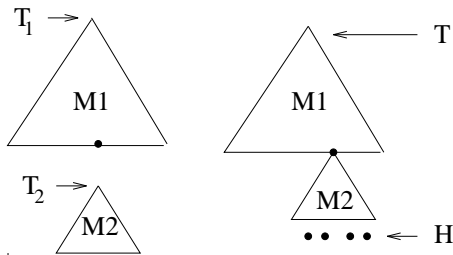


Figure 1: Composition of PQ-tree

The matrix  $M_1$  will be a matrix of taxa versus characters as well. The taxa of  $M_1$  will be all taxa of  $M$  that do not have the character  $H$  plus an extra taxon  $h$  representing all taxa of  $M$  that do have  $H$ . The characters of  $M_1$  will be all columns of  $M$  that contain  $H$  or are disjoint from  $H$ . The matrix  $M_1$  has at least one less trivial column than  $M$  does and hence we can apply the induction hypothesis to it.

The matrix  $M_2$  have as taxa the taxa in  $M$  that have  $H$ . The characters of  $M_2$  will be those contained in  $H$ . Note that  $H$  is trivial in  $M_2$ , therefore, we can apply the induction hypothesis to  $M_2$  as well.

By induction,  $M_1$  and  $M_2$  have PQ-trees where the internal nodes are all of type P. Let  $T_1$  and  $T_2$  be PQ-trees of  $M_1$  and  $M_2$ , respectively. We claim that the tree  $T$  obtained replacing the leaf  $h$  in  $T_1$  by  $T_2$  is a PQ-tree of  $M$ . To prove this claim we need to show that (1) all permutations represented in  $T$  leave the ones consecutive in  $M$ 's columns, and (2) all permutations that leave the ones of  $M$  consecutive in columns are represented in  $T$ .

Let us prove (1). Let  $\alpha$  be a permutation represented in  $T$ . We need to show that  $\alpha$  leaves the ones of each column in  $M$  consecutive. Note that the taxa of  $H$  appear consecutive in  $\alpha$ , because these are leaves of a subtree of  $T$ . Hence the ones of column  $H$  are consecutive.

The columns contained in  $H$  will be consecutive too, because the part of  $\alpha$  that has the taxa of  $H$  form an permutation represented by  $T_2$ , the PQ-tree of  $M_2$ . Thus, for all columns contained in  $H$  we have the desired result. On the other hand, the columns that contain  $H$  or are disjoint from  $H$  are contemplated in  $M_1$ . The permutation  $\alpha$ , if we collapse  $H$  into  $h$ , is represented in  $T_1$ , so all these columns will have their ones consecutive. We don't have other types of columns to consider because of Theorem 1.

Let us now switch to claim (2). Let  $\alpha$  be a permutation that leaves the ones consecutive. In particular, the taxa of  $H$  are consecutive in  $\alpha$ . Then we can write

$$\alpha = \alpha_1 \beta \alpha_2$$

where  $\beta$  is the part of  $\alpha$  involving taxa of  $H$ .

Note that  $\beta$  leaves the columns of  $M_2$  consecutive. These are exactly those columns of  $M$  contained in  $H$ . Hence,  $\beta$  is represented in  $T_2$ , because  $T_2$  is the PQ-tree of  $M_2$ .

Consider now  $\alpha' = \alpha_1 h \alpha_2$ . This is a permutation valid for  $M_1$ . Note that  $\alpha'$  leaves each column of  $M_1$  with the ones consecutive, because these columns are exactly the columns of  $M$  where the positions corresponding to  $H$  were condensed in a single element. Besides, this continuous portion contains only ones or only zeros and the element substituted for is a one or a zero, respectively.

We conclude that  $\alpha'$  is represented in  $T_1$ . Since  $\alpha$  is obtained from  $\alpha'$  by replacing of  $h$  by  $\beta$ , we conclude that  $\alpha$  is represented in  $T$ .

This shows that  $T$  is indeed a PQ-tree of  $M$ . Since by hypothesis  $T_1$  and  $T_2$  have internal nodes of type P only, the same occurs in  $T$ .

( $\Leftarrow$ ) We have a PQ-tree of  $M$  with all internal nodes of type P. We must show that  $M$  has a phylogenetic tree. In fact, the very PQ-tree is the phylogenetic tree sought, except for the fact that a PQ-tree is not necessarily binary. However, we can easily transform it into a binary tree replacing each node P by as many binary nodes needed to accommodate its children, as shown in Figure 2.

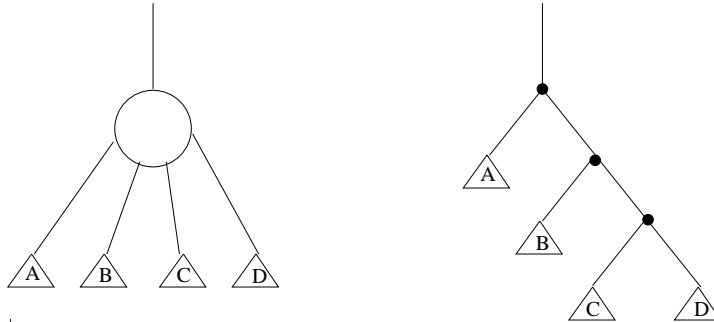


Figure 2: Example of binaryzation of P-node.

Note that under this transformation, for each subtree of the PQ-tree there is a subtree of the phylogenetic tree with the same leaves. This fact will be important in the sequel. To show that the transformed PQ-tree is our phylogeny tree, we prove that each character  $C$  of  $M$  corresponds to a P-node  $p$  in the PQ-tree such that  $C$  is exactly the set of leaves that descend from  $p$ .

Consider a nonempty character  $C$  and let  $T$  be a PQ-tree of  $M$ . Mark all leaves of  $T$  that have character  $C$ . Let  $p$  be the lowest common ancestor in  $T$  of all marked leaves.

Suppose that there exists an unmarked leaf  $a$  in the subtree rooted at  $p$ . Let  $q$  be the lowest node in  $T$  such that  $q \xrightarrow{*} a$  and that has at least one marked descendent. Since  $p \xrightarrow{*} a$  and  $p$  has descendent leaves marked, we have  $p \xrightarrow{*} q$ . We have two cases to consider:

**Case 1**  $p = q$

In this case,  $p$  has at least two descendent leaves marked otherwise it is not the lowest common ancestor of all marked leaves. Since by hypothesis  $T$  has nodes of type P only, we can permute the children of  $p$  in a way that puts two marked leaves around  $a$ . This will generate a permutation where  $C$  is not consecutive, violating the hypothesis that  $T$  is a PQ-tree of  $M$ .

**Case 2**  $p \neq q$

In this case it is possible to put  $a$  between two marked leaves permuting the children of  $p$  and  $q$ . Permute the children of  $q$  so that at least one of its marked leaves appears after  $a$  in the frontier. Then put the unique child of  $p$  which is an ancestor of  $q$  after all other children of  $p$ . This generates a permutation where  $C$  is not consecutive, violating the hypothesis.  $\square$

### 3.1 An algorithm to test phylogeny with PQ-trees

In this section we sketch an algorithm to test binary perfect phylogeny based on PQ-trees, commenting details of the implementation.

Given a binary matrix  $M$ , we attempt to build its PQ-tree as in [1]. Familiarity with this paper is necessary to understand what follows.

We need only a few of the templates used in procedure REDUCE described in [1], namely  $P_0$ ,  $P_1$ ,  $P_2$ ,  $P_3$ , and  $L_1$ . In addition, we need to alter one of them,  $P_3$ , and this

occurs because when it is applicable then the matrix  $M$  doesn't have a phylogeny. We need therefore to modify the replacement of template  $P_3$  to return a NULL TREE.

Templates not on the above list are not needed, because they work with Q-nodes, which are forbidden internal nodes in the binary phylogeny problem.

## 4 Conclusions

We introduced PQ-trees to solve binary phylogeny. With this viewpoint we are able to give a new upper bound for this problem: linear  $O(m + n + r)$  time was obtained.

The number of templates required for this test is very small, compared to the original implementation of PQ-trees. One template needed a simple alteration. We thus avoid the major headache in PQ-trees: the nodes of type Q. This fact simplifies dramatically the implementation.

## References

- [1] K. S. Booth and G. S. Leuker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Sys. Sci.*, 13(3):335–379, December 1976.
- [2] J. Camin and R. Sokal. A method for deducing branching sequences in phylogeny evolution. *Evolution*, 19:311–326, 1965.
- [3] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa. A linear algorithm for embedding planar graphs using PQ-trees. *J. Comput. Syst. Sci.*, 30(1):54–76, February 1985.
- [4] S. P. Ghosh. On the theory of consecutive storage of relevant records. *Journal Inform. Sci.*, 6:1–9, 1973.
- [5] D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.
- [6] H. Heeb and W. Fichtner. A module generator based on the PQ-tree algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(7):876–884, july 1992.

## Relatórios Técnicos – 1992

- 92-01 **Applications of Finite Automata Representing Large Vocabularies,** *C. L. Lucchesi, T. Kowaltowski*
- 92-02 **Point Set Pattern Matching in  $d$ -Dimensions,** *P. J. de Rezende, D. T. Lee*
- 92-03 **On the Irrelevance of Edge Orientations on the Acyclic Directed Two Disjoint Paths Problem,** *C. L. Lucchesi, M. C. M. T. Giglio*
- 92-04 **A Note on Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams,** *W. Jacometti*
- 92-05 **An  $(l, u)$ -Transversal Theorem for Bipartite Graphs,** *C. L. Lucchesi, D. H. Younger*
- 92-06 **Implementing Integrity Control in Active Databases,** *C. B. Medeiros, M. J. Andrade*
- 92-07 **New Experimental Results For Bipartite Matching,** *J. C. Setubal*
- 92-08 **Maintaining Integrity Constraints across Versions in a Database,** *C. B. Medeiros, G. Jomier, W. Cellary*
- 92-09 **On Clique-Complete Graphs,** *C. L. Lucchesi, C. P. Mello, J. L. Szwarcfiter*
- 92-10 **Examples of Informal but Rigorous Correctness Proofs for Tree Traversing Algorithms,** *T. Kowaltowski*
- 92-11 **Debugging Aids for Statechart-Based Systems,** *V. G. S. Elias, H. Liesenberg*
- 92-12 **Browsing and Querying in Object-Oriented Databases,** *J. L. de Oliveira, R. de O. Anido*



## Relatórios Técnicos – 1993

- 93-01 **Transforming Statecharts into Reactive Systems**, *Antonio G. Figueiredo Filho, Hans K. E. Liesenberg*
- 93-02 **The Hierarchical Ring Protocol: An Efficient Scheme for Reading Replicated Data**, *Nabor das C. Mendonça, Ricardo de O. Anido*
- 93-03 **Matching Algorithms for Bipartite Graphs**, *Herbert A. Baier Saip, Cláudio L. Lucchesi*
- 93-04 **A lexBFS Algorithm for Proper Interval Graph Recognition**, *Celina M. H. de Figueiredo, João Meidanis, Célia P. de Mello*
- 93-05 **Sistema Gerenciador de Processamento Cooperativo**, *Ivonne. M. Carrazana, Nelson. C. Machado, Célio. C. Guimarães*
- 93-06 **Implementação de um Banco de Dados Relacional Dotado de uma Interface Cooperativa**, *Nascif A. Abousalh Neto, Ariadne M. B. R. Carvalho*
- 93-07 **Estadogramas no Desenvolvimento de Interfaces**, *Fábio N. de Lucena, Hans K. E. Liesenberg*
- 93-08 **Introspection and Projection in Reasoning about Other Agents**, *Jacques Wainer*
- 93-09 **Codificação de Seqüências de Imagens com Quantização Vetorial**, *Carlos Antonio Reinaldo Costa, Paulo Lício de Geus*
- 93-10 **Minimização do Consumo de Energia em um Sistema para Aquisição de Dados Controlado por Microcomputador**, *Paulo Cesar Centoducatte, Nelson Castro Machado*
- 93-11 **An Implementation Structure for RM-OSI/ISO Transaction Processing Application Contexts**, *Flávio Moraes de Assis Silva, Edmundo Roberto Mauro Madeira*
- 93-12 **Boole's conditions of possible experience and reasoning under uncertainty**, *Pierre Hansen, Brigitte Jaumard, Marcus Poggi de Aragão*
- 93-13 **Modelling Geographic Information Systems using an Object Oriented Framework**, *Fatima Pires, Claudia Bauzer Medeiros, Ardemiris Barros Silva*
- 93-14 **Managing Time in Object-Oriented Databases**, *Lincoln M. Oliveira, Claudia Bauzer Medeiros*
- 93-15 **Using Extended Hierarchical Quorum Consensus to Control Replicated Data: from Traditional Voting to Logical Structures**, *Nabor das Chagas Mendonça, Ricardo de Oliveira Anido*

- 93-16  **$\mathcal{LL}$  – An Object Oriented Library Language Reference Manual**, *Tomasz Kowaltowski, Evandro Bacarin*
- 93-17 **Metodologias para Conversão de Esquemas em Sistemas de Bancos de Dados Heterogêneos**, *Ronaldo Lopes de Oliveira, Geovane Cayres Magalhães*
- 93-18 **Rule Application in GIS – a Case Study**, *Claudia Bauzer Medeiros, Geovane Cayres Magalhães*
- 93-19 **Modelamento, Simulação e Síntese com VHDL**, *Carlos Geraldo Krüger e Mário Lúcio Côrtes*
- 93-20 **Reflections on Using Statecharts to Capture Human-Computer Interface Behaviour**, *Fábio Nogueira de Lucena e Hans Liesenberg*
- 93-21 **Applications of Finite Automata in Debugging Natural Language Vocabularies**, *Tomasz Kowaltowski, Cláudio Leonardo Lucchesi e Jorge Stolfi*
- 93-22 **Minimization of Binary Automata**, *Tomasz Kowaltowski, Cláudio Leonardo Lucchesi e Jorge Stolfi*
- 93-23 **Rethinking the DNA Fragment Assembly Problem**, *João Meidanis*
- 93-24 **EGOLib — Uma Biblioteca Orientada a Objetos Gráficos**, *Eduardo Aguiar Patrocínio, Pedro Jussieu de Rezende*
- 93-25 **Compreensão de Algoritmos através de Ambientes Dedicados a Animação**, *Rackel Valadares Amorim, Pedro Jussieu de Rezende*
- 93-26 **GeoLab: An Environment for Development of Algorithms in Computational Geometry**, *Pedro Jussieu de Rezende, Welson R. Jacometti*
- 93-27 **A Unified Characterization of Chordal, Interval, Indifference and Other Classes of Graphs**, *João Meidanis*
- 93-28 **Programming Dialogue Control of User Interfaces Using Statecharts**, *Fábio Nogueira de Lucena e Hans Liesenberg*
- 93-29 **EGOLib – Manual de Referência**, *Eduardo Aguiar Patrocínio e Pedro Jussieu de Rezende*

## Relatórios Técnicos – 1994

- 94-01 **A Statechart Engine to Support Implementations of Complex Behaviour**, *Fábio Nogueira de Lucena, Hans K. E. Liesenberg*
- 94-02 **Incorporação do Tempo em um SGBD Orientado a Objetos**, *Ângelo Roncalli Alencar Brayner, Cláudia Bauzer Medeiros*
- 94-03 **O Algoritmo KMP através de Autômatos**, *Marcus Vinícius A. Andrade e Cláudio L. Lucchesi*
- 94-04 **On Edge-Colouring Indifference Graphs**, *Celina M. H. de Figueiredo, João Meidanis, Célia Picinin de Mello*
- 94-05 **Using Versions in GIS**, *Claudia Bauzer Medeiros and Geneviève Jomier*
- 94-06 **Times Assíncronos: Uma Nova Técnica para o Flow Shop Problem**, *Hélvio Pereira Peixoto e Pedro Sérgio de Souza*
- 94-07 **Interfaces Homem-Computador: Uma Primeira Introdução**, *Fábio Nogueira de Lucena e Hans K. E. Liesenberg*
- 94-08 **Reasoning about another agent through empathy**, *Jacques Wainer*
- 94-09 **A Prolog morphological analyser for Portuguese**, *Jacques Wainer, Alexandre Farcic*
- 94-10 **Introdução aos Estadogramas**, *Fábio N. de Lucena, Hans K. E. Liesenberg*
- 94-11 **Matching Covered Graphs and Subdivisions of  $K_4$  and  $\overline{C}_6$** , *Marcelo H. de Carvalho and Cláudio L. Lucchesi*
- 94-12 **Uma Metodologia de Especificação de Times Assíncronos**, *Hélvio Pereira Peixoto, Pedro Sérgio de Souza*

## Relatórios Técnicos – 1995

- 95-01 **Paradigmas de algoritmos na solução de problemas de busca multidimensional**, *Pedro J. de Rezende, Renato Fileto*
- 95-02 **Adaptive enumeration of implicit surfaces with affine arithmetic**, *Luiz Henrique de Figueiredo, Jorge Stolfi*
- 95-03 **W3 no Ensino de Graduação?**, *Hans Liesenberg*
- 95-04 **A greedy method for edge-colouring odd maximum degree doubly chordal graphs**, *Celina M. H. de Figueiredo, João Meidanis, Célia Picinin de Mello*
- 95-05 **Protocols for Maintaining Consistency of Replicated Data**, *Ricardo Anido, N. C. Mendonça*
- 95-06 **Guaranteeing Full Fault Coverage for UIO-Based Methods**, *Ricardo Anido and Ana Cavalli*
- 95-07 **Xchart-Based Complex Dialogue Development**, *Fábio Nogueira de Lucena, Hans K.E. Liesenberg*
- 95-08 **A Direct Manipulation User Interface for Querying Geographic Databases**, *Juliano Lopes de Oliveira, Claudia Bauzer Medeiros*
- 95-09 **Bases for the Matching Lattice of Matching Covered Graphs**, *Cláudio L. Lucchesi, Marcelo H. Carvalho*
- 95-10 **A Highly Reconfigurable Neighborhood Image Processor based on Functional Programming**, *Neucimar J. Leite, Marcelo A. de Barros*
- 95-11 **Processador de Vizinhança para Filtragem Morfológica**, *Ilka Marinho Barros, Roberto de Alencar Lotufo, Neucimar Jerônimo Leite*
- 95-12 **Modelos Computacionais para Processamento Digital de Imagens em Arquiteturas Paralelas**, *Neucimar Jerônimo Leite*
- 95-13 **Modelos de Computação Paralela e Projeto de Algoritmos**, *Ronaldo Parente de Menezes e João Carlos Setubal*
- 95-14 **Vertex Splitting and Tension-Free Layout**, *P. Eades, C. F. X. de Mendonça N.*
- 95-15 **NP-Hardness Results for Tension-Free Layout**, *C. F. X. de Mendonça N., P. Eades, C. L. Lucchesi, J. Meidanis*
- 95-16 **Agentes Replicantes e Algoritmos de Eco**, *Marcos J. C. Euzébio*
- 95-17 **Anais da II Oficina Nacional em Problemas Combinatórios: Teoria, Algoritmos e Aplicações**, *Editores: Marcus Vinicius S. Poggi de Aragão, Cid Carvalho de Souza*

- 95-18 **Asynchronous Teams: A Multi-Algorithm Approach for Solving Combinatorial Multiobjective Optimization Problems**, *Rosiane de Freitas Rodrigues, Pedro Sérgio de Souza*
- 95-19 **wxWindows: Uma Introdução**, *Carlos Neves Júnior, Tallys Hoover Yunes, Fábio Nogueira de Lucena, Hans Kurt E. Liesenberg*

*Departamento de Ciência da Computação — IMECC  
Caixa Postal 6065  
Universidade Estadual de Campinas  
13081-970 – Campinas – SP  
BRASIL  
reltec@dcc.unicamp.br*