Parallel and Distributed Computing and Systems
November 3-6, 1999 in Cambridge Massachusetts, USA

Efficiently Checking the Equivalence of Multistage Interconnection Networks

Tiziana Calamoneri* Annalisa Massini
Dip. di Scienze dell'Informazione, Università di Roma "La Sapienza"
via Salaria 113 - 00198 Rome, Italy.
{calamo,massini}@dsi.uniroma1.it

Abstract

In this paper we propose a new characterization for the topological equivalence of multistage interconnection networks (MINs). Then, we apply it both to $\log N$ stage MINs and to $2\log N - 1$ stage MINs. In the first case, we provide an algorithm for checking the equivalence with the Reverse Baseline running in $O(N \log N)$ time. In the second case, we give an algorithm running in $O(N \log N)$ time testing the equivalence of two MINs obtained as concatenation of two $\log N$ stage MINs, both equivalent to the Reverse Baseline. This result definitely closes the equivalence problem between these $2\log N - 1$ stage MINs and substantially improves the time complexity of the previously known algorithms.

Keywords: multistage interconnection networks, topological equivalence, layered cross product.

1 Introduction

In the past two decades the binary relation of topological equivalence between two different MINs have been investigated since the understanding of this relation make it possible to apply the routing scheme of a MIN to another MIN and to develop more general routing algorithms useful for all the MINs in the same equivalence class. Wu and Feng [5] presented the first formal definition of topological equivalence and proved that the six $\log N$ stage MINs mentioned above are topologically equivalent, but they showed the isomorphism between two networks and did not give any characterization of the equivalence class which these MINs belong to. Another approach was considered by Agrawal [8], but unfortunately it is correct only for $\log N$ stage MINs of small dimension. A revised version was proposed by Bermond, Fourneau and Jean-Marie [9]; they gave more general properties to check the topological equivalence. Hu, Shen and Yang [10] presented a simplified checking equivalence algorithm based on a marking scheme of nodes whose time complexity is $O(N \log N)$, that is optimal.

A lot of efforts have been expended in studying topological equivalence of $2\log N - 1$ stage MINs. Wu and Feng [11] extended their equivalence properties for $\log N$ stage MINs to prove that two-passes of a reverse baseline network has the same routing capability and is equivalent to the Beneš network. Lee [12] proved that an Omega network concatenated with its reverse is equivalent to the Beneš network. Yeh and Feng [13] proposed a coding scheme to check whether a given $2\log N - 1$ stage MIN is topologically equivalent to the Beneš network, but this scheme leaves many cases uncovered. Feng et al. [14] studied 36 common topologies obtained as concatenation of two $\log N$ stage MINs and classified them into two topologically equivalent classes. Hu, Shen and Yang [10] studied whether a $2\log N - 1$ stage MIN is the concatenation of two $\log N$ stage MINs and gave an algorithm to determine whether two given $2\log N - 1$ stage MINs are topologically equivalent in $O(N^4 \log N)$ time.

In this work a new characterization of MINs is proposed. This is a very general characterization, and achieves surprising results when applied to $\log N$ and $2\log N - 1$ stage MINs. Namely, the characterization for $\log N$ stage MINs makes it possible to design an optimal (i.e. running in $O(N \log N)$ time) algorithm to determine if a given MIN is topologically equivalent to the Reverse Baseline. About $2\log N - 1$ stage MINs, first we prove that all MINs obtained as concatenation of a $\log N$ stage MIN equivalent to the Reverse Baseline and of its reverse are topologically equivalent. Then, an optimal algorithm (i.e. running in $O(N \log N)$ time) testing the equivalence of two MINs obtained as the concatenation of two $\log N$ stage MINs, both equivalent to the Reverse Baseline, is provided. This result definitely closes the equivalence problem between these $2\log N - 1$ stage MINs and substantially improves the time complexity of the previously known algorithms.

2 Preliminary Definitions

In this section we give some basic definitions and preliminary results, useful for the comprehension of the last part of this paper.

Definition 2.1 *An* l-layered graph, $G = (V_1, V_2, \ldots, V_l, E)$

*Supported by the Italian Research Council (CNR)

consists of l layers of nodes; V_i is the (non-empty) set of nodes in layer i, where $1 \leq i \leq l$; E is a set of edges: every edge connects nodes of two adjacent layers.

Observe that a rooted tree T (or simply tree, when non confusion arises) of height h is a particular case of h-layered graph, where layer i is defined either as the set of nodes having distance $i-1$ from the root or as the set of nodes having distance $h-i$ from the root. From now on, we shall call a complete binary tree T by means of Δ or ∇ according to whether the first or the second way of defining layers is chosen (the notation comes from the usual graphic representation of such trees – see Fig. 1.a and 1.b).

The $N \times N$ *multistage interconnection network (N-MIN)* is another particular l-layered graph, where $|V_i| = \frac{N}{2}$ and any node in V_i, representing a switching element of size 2×2, is adjacent to exactly two nodes in V_{i-1} and $V_{i+1}, 2 \leq i \leq l-1$, and the edge set is partitioned into $l-1$ subsets $E_1, E_2, \ldots, E_{l-1}$ such that E_i contains N edges, that is all edges between V_i and V_{i+1}. Each node of the first stage is also connected to a pair of *inputs* and each node of the last stage is also connected to a pair of *outputs*. Then an N-MIN contains N inputs and N outputs.

Definition 2.2 *Any two l stage N-MINs G' and G'' are* topologically equivalent *(or simply* equivalent*) if and only if there is an isomorphic mapping ψ such that if $v \in V_i(G')$ then $\psi(v) \in V_i(G'')$, and if $\langle u, v\rangle \in E_i(G')$ then $\langle \psi(u), \psi(v)\rangle \in E_i(G''), i = 1 \ldots l$.*

Definition 2.3 *[15] Let $G' = (V_1', V_2', \ldots, V_l', E')$ and $G'' = (V_1'', V_2'', \ldots, V_l'', E'')$ be two l-layered graphs. Their* Layered Cross Product *(LCP for short) , $G' \otimes G''$ is an l-layered graph $G = (V_1, V_2, \ldots, V_l, E)$ where V_i is the cartesian product of V_i' and V_i'', $1 \leq i \leq l$, and an edge $\langle (u', u''), (v', v'')\rangle$ belongs to E if and only if $\langle u', v'\rangle \in E'$ and $\langle u'', v''\rangle \in E''$. G' and G'' are called the* first *and* second factor *of G, respectively.*

We will call *decomposition in factors* the inverse operation of the LCP.

The LCP is commutative, therefore from now on, when we speak about isomorphism between the first and the second factors of two N-MINs $G' = G_1' \otimes G_2'$ and $G'' = G_1'' \otimes G_2''$ we mean that G_1' and G_2' are equivalent either to G_1'' and G_2'', or to G_2'' and G_1'', respectively.

Definition 2.4 *A N-MIN has the* **Banyan property** *if and only if for any input node and any output node there exists a unique path connecting them.*

Lemma 1 *[15] The LCP operation yields a Banyan graph if and only if each of its factors is Banyan.*

Lemma 2 *Given two l-layered graphs, having c_1 and c_2 connected components, respectively, then their LCP has $c_1 \cdot c_2$ connected components.*

Proof It immediately follows from the definition of LCP. **Q.E.D.**

Lemma 3 *[15] The LCP of a Δ and a ∇, both of them with $\frac{N}{2}$ leaves, outputs an N input Butterfly network.*

In the rest of the paper we will point out our attention on N-MINs that are decomposable as LCP of two graphs, therefore from now on when we speak about LCP we implicitly assume that the result is a N-MIN. Observe that the inputs and outputs of N-MINs are not involved in the LCP, but it is not restrictive to add them at the end of the computation of the LCP.

3 A New Characterization for the Equivalence of MINs Based on LCP

In this section we propose a new general characterization of MINs' topological equivalence, that is based on the following theorem.

Theorem 4 *Let G' and G'' be two l stage N-MIN, and let G' decomposable as $G_1' \otimes G_2'$. Then G'' is topologically equivalent to G' if and only if G'' can be decomposed as $G_1' \otimes G_2'$.*

Proof If G'' can be decomposed as $G_1' \otimes G_2'$, then trivially G' and G'' are equivalent.

If G' is equivalent to G'', then there exists the isomorphic mapping ψ between G' and G''. Let us construct a decomposition for G'' and let us prove that the factors are equivalent to G_1' and G_2'.

Node decomposition: consider $v' \in V_i(G'), 1 \leq i \leq l$, in view of the hypothesis $v' = (v_1', v_2')$; add vertices v_1'' to G_1'' and v_2'' to G_2'', if they do not exist, yet, and decompose $\psi(v')$ as (v_1'', v_2'').

Edge decomposition: if there exists an edge $\langle u', v'\rangle \in E(G')$ it is obtained as product of edges $\langle u_1', v_1'\rangle$ and $\langle u_2', v_2'\rangle$ belonging to G_1' and G_2', respectively. In view of the decomposition of nodes of G'', $\psi(u') = (u_1'', u_2'')$ and $\psi(v') = (v_1'', v_2'')$. Add the edge $\langle u_1'', v_1''\rangle$ in G_1'' and the edge $\langle u_2'', v_2''\rangle$ in G_2'', respectively, so that edge $\langle \psi(u'), \psi(v')\rangle$ is decomposed as LCP of $\langle u_1'', v_1''\rangle$ and $\langle u_2'', v_2''\rangle$. By construction, G_1'' is the same as G_1' and G_2'' is the same as G_2'. **Q.E.D.**

Corollary 5 *Given two N-MINs $G' = G_1' \otimes G_2'$ and $G'' = G_1'' \otimes G_2''$, they are topologically equivalent if and only if their factors are topologically equivalent.*

This characterization may be very helpful since the factors of a N-MIN G are simpler graphs than G itself, and therefore to check the equivalence between factors may be much easier than checking the equivalence between N-MINs.

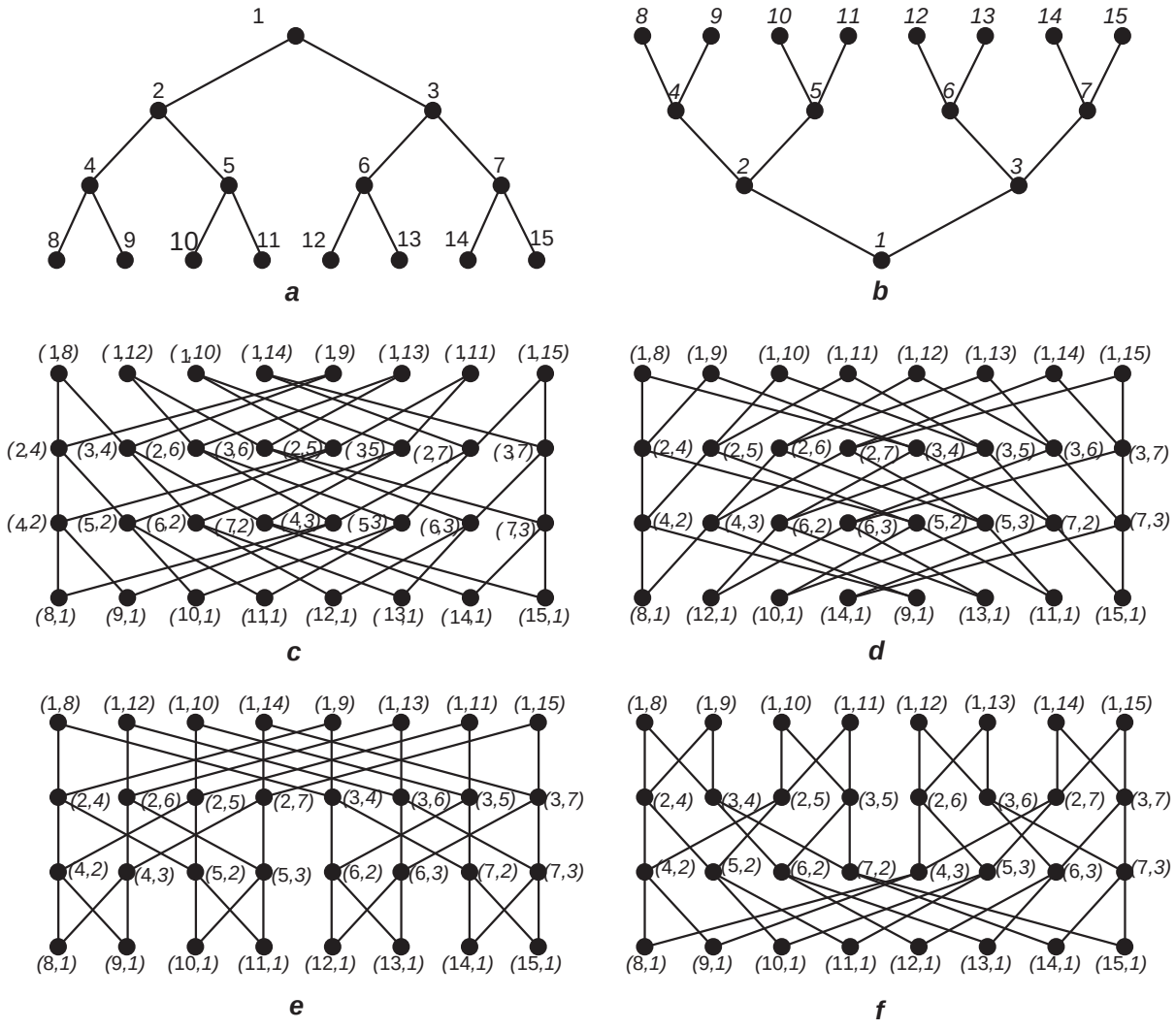

Figure 1: a. a Δ; b. a ∇; c. an Omega network; d. a Flip network; e. a Butterfly network; f. a Reverse Baseline Network. All networks from c. to f. are the LCP of Δ and ∇, as highlighted by the labels, and are all topologically equivalent.

4 On the Equivalence of $\log N$ Stage N-MINs

In this section we deal with $\log N$ stage N-MINs, therefore –where no confusion arises– when we speak about N-MINs we mean $\log N$ stage N-MINs. For this class of networks we specify a particular decomposition that will be on the basis of the characterization of the topological equivalence and of the design of the algorithm to check it.

Bermond, Fourneau and Jean-Marie [9] gave an interesting characterization of N-MINs topologically equivalent to the Reverse Baseline network. It is based on the Banyan property and on the $P(*,*)$ property, that we briefly remind here.

Property P(i,j). A N-MIN has property $P(i,j)$ for $1 \leq i \leq j \leq \log N$ if the subgraph $G_{i,j}$ induced by the nodes of the stage from i to j has exactly $2^{\log N-1-j+i}$ connected components.

Property P(*,*). A N-MIN has property $P(*,*)$ if and only if it satisfies $P(i,j)$ for every ordered pair i,j such that $1 \leq i \leq j \leq \log N$.

Theorem 6 *[9] All the N-MINs satisfying the Banyan Property and $P(*,*)$ are topologically equivalent to the Reverse Baseline.*

Basing on this theorem and on properties of the LCP, we can state the following new characterization for all N-MINs that are topologically equivalent to Reverse Baseline network.

Theorem 7 *A N-MIN G satisfies the Banyan Property and $P(*,*)$ if and only if it can be decomposed as $\Delta \otimes \nabla$.*

Proof If G can be decomposed as $\Delta \otimes \nabla$ then G satisfies the Banyan property, in view of Lemma 1. G satisfies also $P(*,*)$ indeed for any i,j such that $1 \leq i \leq j \leq \log N$, the subgraph of Δ induced by the nodes of the stages from i to j has exactly 2^{i-1} connected components and the subgraph of ∇ induced by the nodes of the stages from i to j has exactly $2^{\log N-j}$ connected components. It follows from Lemma 2 that the subgraph of G induced by the nodes of the stages from i to j has exactly $2^{\log N-j+i-1}$ connected components, i.e. G satisfies $P(i,j)$ for any i,j and therefore G satisfies $P(*,*)$.

If G satisfies the Banyan and $P(*,*)$ properties, then – in view of Theorem 6 – G is topologically equivalent to the Reverse Baseline. It is well known that the Reverse Baseline is topologically equivalent to the Butterfly network [6], which can be decomposed as LCP of $\Delta \otimes \nabla$, in view of Lemma 3. Our characterization (Corollary 5) ensures that also the Reverse Baseline can be decomposed as LCP of $\Delta \otimes \nabla$ and, consequently G is the LCP of the same factors. **Q.E.D.**

As a consequence of this theorem and of the known equivalence of Butterfly, Omega, Flip, Reverse Baseline, etc. we deduce that these networks can all be decomposed as $\Delta \otimes \nabla$, as shown in Fig. 1. Furthermore, since the LCP is commutative and the reverse of a Δ is a ∇ (and vice versa), it follows that each N-MIN decomposable as $\Delta \otimes \nabla$ is equivalent to its reverse.

Checking the topological equivalence of a N-MIN to the Reverse Baseline using the characterization of Bermond, Fourneau and Jean-Marie requires $O(N^2 \log N)$ time, since the Banyan and $P(*,*)$ properties can be checked in $O(N^2 \log N)$ and $O(N \log^2 N)$ time, respectively. This time complexity has been improved by Hu, Shen and Yang [10], who presented a simplified checking equivalence algorithm based on a marking scheme of nodes requiring $O(N \log N)$ time, that is optimal, since it is the same order of the number of nodes in the N-MIN.

We will show that also the time complexity of checking the property of Theorem 7 is $O(N \log N)$. Although this result does not improves the previously known one, we detail it since it is preliminary to the algorithm described in the next section, where $2 \log N - 1$ stage N-MINs are considered.

Theorem 8 *The topological equivalence between a given a N-MIN G and the Reverse Baseline network is checkable in $O(N \log N)$ time.*

Proof We prove the claim in a constructive way, by giving an algorithm trying to decompose G as $\Delta \otimes \nabla$; if the algorithm is successful, then G is topologically equivalent to the Reverse Baseline network.

The algorithm assigns a label to nodes of G; the label of node v is a pair (v', v''), where v' and v'' represent the factors of v according to its decomposition. If the algorithm succeeds in completing the labeling, then it provides the decomposition as $\Delta \otimes \nabla$; if, during the execution, two different labels are assigned to the same node, then the algorithm returns and we deduce that the MIN cannot be decomposed as $\Delta \otimes \nabla$, and therefore it is not topologically equivalent to the Reverse Baseline.

Let $d_1, d_2, \ldots, d_{N-1}$ and $n_1, n_2, \ldots, n_{N-1}$ be the names of the nodes in Δ and ∇, respectively. As usual, let nodes d_{2i} (n_{2i}) and d_{2i+1} (n_{2i+1}) be the children of d_i (n_i).

The algorithm consists of two steps: it starts assigning the first elements of the labels, corresponding to labels of factor Δ. In the second step, it assigns the second elements of the labels corresponding to labels of factor ∇.

Assignment of the first element of the labels: consider any node v at the first level of G and assign to it d_1, corresponding to the root of Δ. Recursively consider an already labeled node d_i and assign labels d_{2i} and d_{2i+1} to its adjacent nodes at next level, until the last level of the N-MIN is reached. At the end of this procedure we have assigned the first elements of the labels of $N-1$ nodes, inducing a complete binary tree and, in particular, all nodes at the last level have been labeled. Now, the algorithm starts from the last level to label all other unlabeled nodes. Sequentially, each

node v at the last level of the N-MIN, labeled d_i, is considered and the same label $d_{\lfloor i/2 \rfloor}$ is assigned to its adjacent nodes at the previous level. Go on if these nodes are either not labeled or already labeled with $d_{\lfloor d/2 \rfloor}$, return if at least one of them is labeled differently by $d_{\lfloor i/2 \rfloor}$. Recursively repeat this procedure for each level until all nodes of the first level are labeled (or the algorithm returns).

Assignment of the second element of the labels: if all the first elements of the labels have been assigned, the algorithm begins to assign the second elements, corresponding to labels of factor ∇. Proceeding exactly as in the first step, but reversing the order of levels; namely, it starts from the last level, goes up to the first one and down again.

Now we prove the correctness of the algorithm. The part assigning the labels to a subgraph of G representing a tree consists in considering the LCP between Δ and a simple path P from a leaf to the root in ∇. We have freedom of choice in view of the symmetry of the tree structure. Then, we need to start from nodes of G corresponding to the leaves of Δ in order to consider all the other paths in ∇ and to respect the connections of such paths with P.

It is easy to see that the algorithm processes each node a constant number of times, then the time complexity is linear in the number of nodes of G, i.e. $O(N \log N)$. **Q.E.D.**

5 On the Equivalence of $2 \log N - 1$ Stage Networks

A $2 \log N - 1$ stage N-MIN is classically obtained as concatenation of two N-MINs with $\log N$ stages, by merging the last stage of the first one with the first stage of the second one. It is typical to concatenate all the combinations of pairs of networks among Butterfly, Omega, Flip, Baseline, etc. and their reverses to obtain a new N-MIN. In all this section we call N-MIN^2 a network G with $2 \log N - 1$ stages obtained concatenating two N-MINs with $\log N$ stages each, both of them equivalent to the Reverse Baseline.

In view of the considerations done in the previous section, both the first and the second networks can be decomposed as LCP of $\Delta \otimes \nabla$. As a consequence, we obtain that the factors of G are the concatenation of a Δ and a ∇ and of a ∇ and a Δ, respectively. It is obvious how to merge the last stage of a ∇ with the first stage of a Δ, but there are many ways of merging the last stage of a Δ and the first stage of a ∇. In fact, different ways of connecting the leaves of the two trees lead to non equivalent structures, as highlighted in Figs. 2 and 3.

Theorem 9 *Given two $\log N$ stage N-MINs G_1 and G_2, both topologically equivalent to the Reverse Baseline, the two N-MIN^{2}s obtained as concatenation of G_1 and G_1^{-1}, and of G_2 and G_2^{-1}, repsectively, are topologically equivalent.*

Proof The statement follows from the decomposition of both G_1 and G_2 in $\Delta \otimes \nabla$ and from the fact that the concatenation of a network and its reverse is symmetric with respect to the line passing through the conjunction level. **Q.E.D.**

In the following, we will indicate by $\begin{smallmatrix}\nabla\\ \Delta\end{smallmatrix}$ ($\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$) the concatenation of a ∇ and a Δ (a Δ and a ∇).

Theorem 10 *Given an N-MIN^2 G, $O(N \log N)$ time is sufficient to decompose it into a $\begin{smallmatrix}\nabla\\ \Delta\end{smallmatrix}$ and a $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$, where the way of concatenating the two binary trees of $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$ is completely specified.*

Proof The algorithm is divided into three different phases, whose the first one decompose the first $\log N$ stages of the N-MIN^2 and the sets of edges in between, the second one decomposes the last $\log N - 1$ stages and the sets of edges in between, the third one decompose the set of edges between stages $\log N$ and $\log N + 1$.

The first phase consists of the algorithm used to decompose a $\log N$ stage N-MIN as $\Delta \otimes \nabla$, and outputs a label for each node v of the first $\log N$ stages.

Second phase also utilizes the same procedure, but it handles only the last $\log N$-1 stages of G and of its factors. Then, the subgraph ∇' of $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$ is a $\frac{N}{4}$ leaf complete binary tree since the leaves of the complete ∇ are not considered. Differently, the subgraph of $\begin{smallmatrix}\nabla\\ \Delta\end{smallmatrix}$ is a pair, Δ'_1 and Δ'_2, of $\frac{N}{4}$ leaf complete binary trees, since the root of Δ is not considered. Therefore, this phase decomposes the last $\log N - 1$ stages as LCP of Δ'_1 times ∇' and Δ'_2 times ∇'.

At the end of these two phases all nodes of G have been labeled, and all nodes and all edges, but the edges between stages $\log N$ and $\log N + 1$, of the N-MIN^2 are decomposed as LCP.

Third phase uses the N-MIN^2 to construct the edges between stages $\log N$ and $\log N + 1$ of $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$. Namely, an edge in $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$ is added between nodes u and v if an edge in the N-MIN exists between nodes having as second element of their label u and v, respectively. It is always possible to perform this third phase, in view of the definition of the input N-MIN^2, and therefore it leads to a labeling of ∇ nodes.

From Theorem 8 and from the definition of G the correctness of the algorithm follows.

For what concerns the time complexity, the first two phases run in $O(N \log N)$ time; the third phase considers all edges between stages $\log N$ and $\log N + 1$ of G once, and therefore runs in $O(N)$ time. **Q.E.D.** Let us study the $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$ factor given in output by the algorithm in the proof of Theorem 10. It is possible to label each node of $\begin{smallmatrix}\Delta\\ \nabla\end{smallmatrix}$ in the following way:

- the root of the Δ has null label;

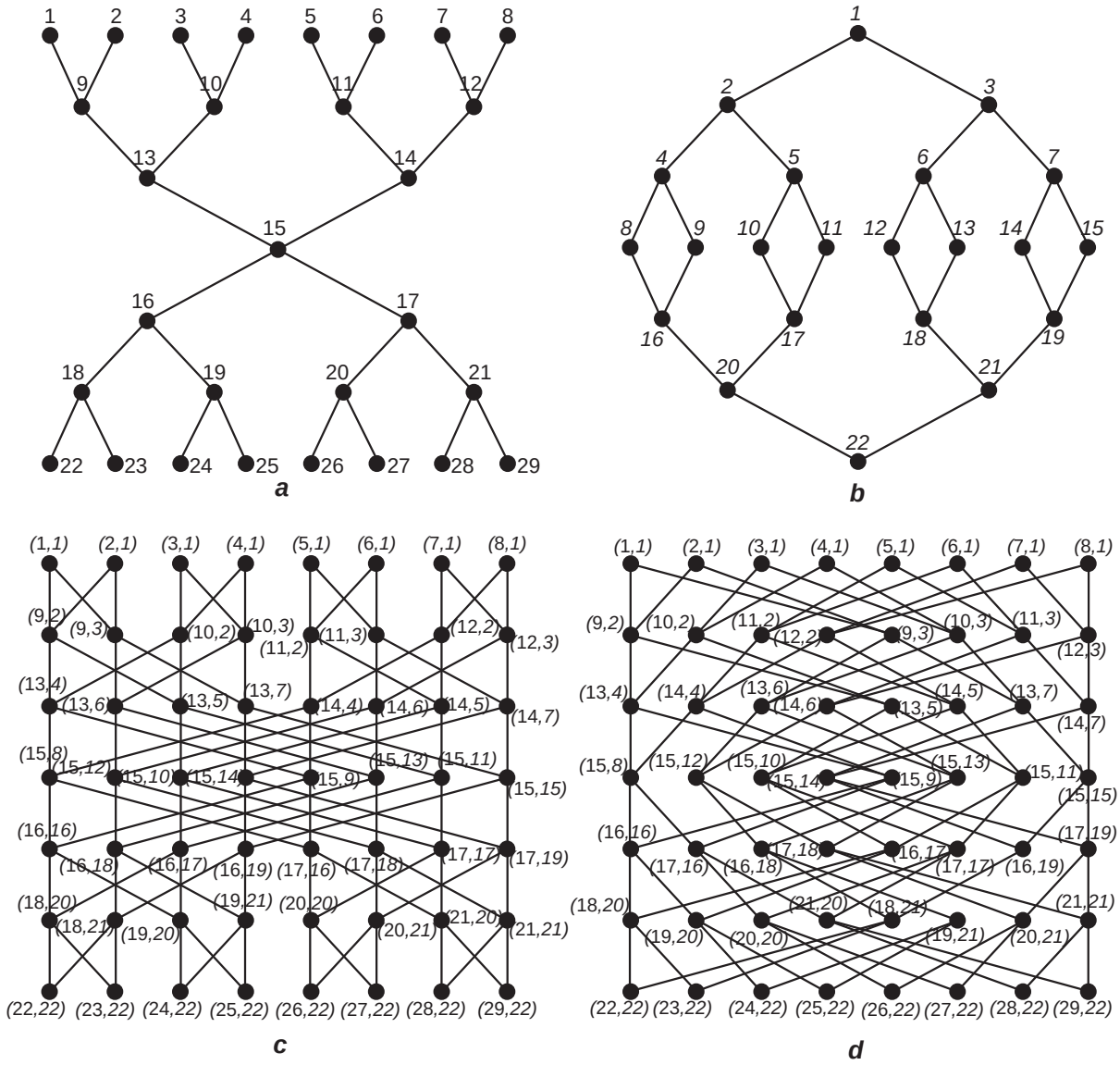

Figure 2: a. a $\overset{\nabla}{\Delta}$; b. a $\overset{\Delta}{\nabla}$; c. the concatenation of a reverse Butterfly and a Butterfly; d. the concatenation of a Flip and an Omega. The networks in c. and d. are equivalent.

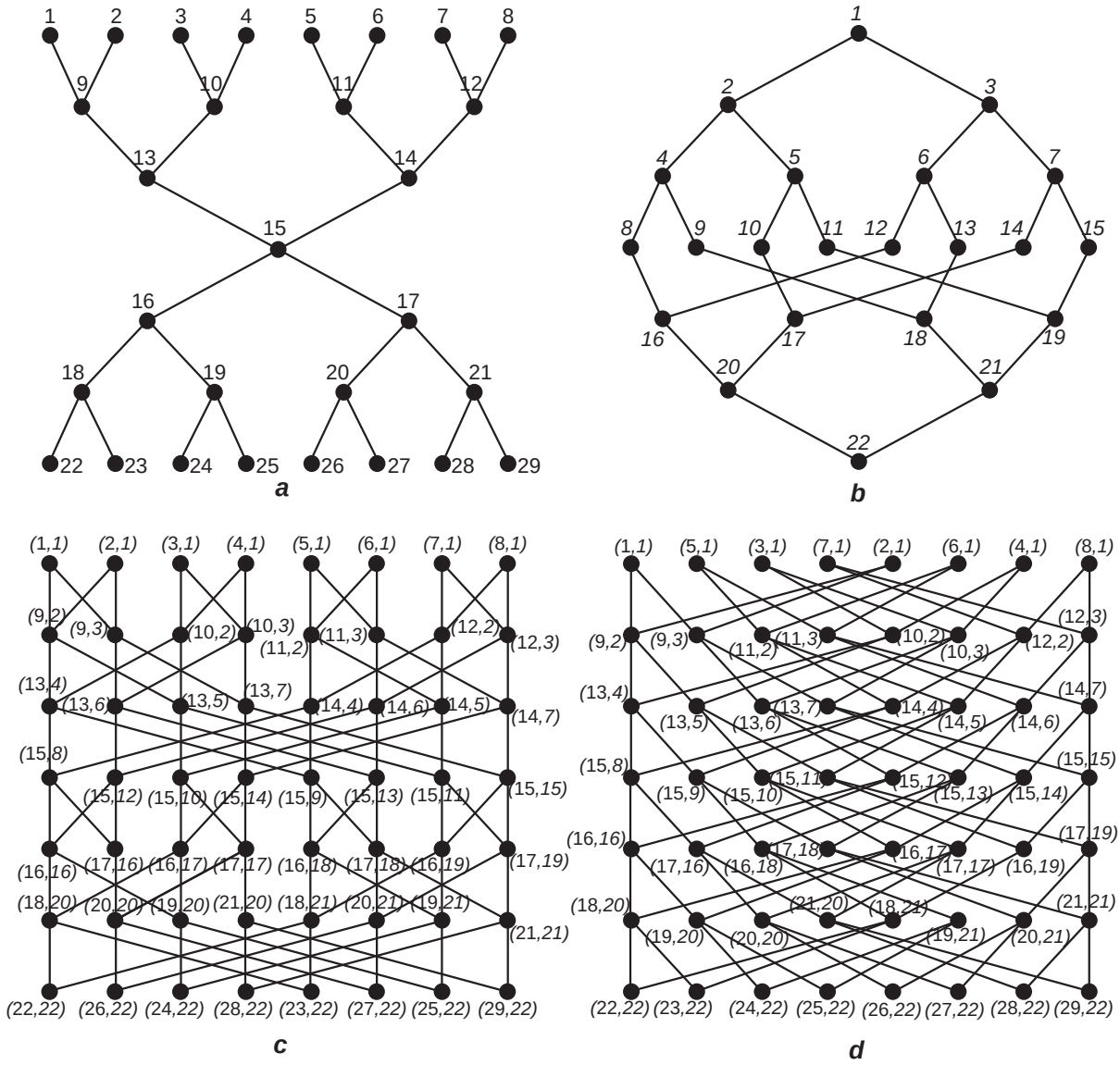

Figure 3: a. a $\bigtriangledown$; b. a $\bigtriangleup$; c. the concatenation of two reverse Butterflies; d. the concatenation of two Omega networks. The networks in c. and d. are equivalent.

- from each node v labeled with $l(v)$, recursively, label v' left child with $l(v)0$ and v' right child with $l(v)1$; go on until all nodes of Δ are labeled;

- for each couple of nodes in ∇, u and w, labeled $l(u)$ and $l(w)$ that are children of the same unlabeled node v, label v with a label obtained by eliminating from $l(u)$ the only bit where it is different from $l(w)$; go on until the root of ∇ is reached (and labeled with a null label).

Observe that the labeling of $\overset{\Delta}{\nabla}$ is such that each Δ node at level i has a label $i-1$ bit long, each ∇ node at level i has a label $2\log N - i - 1$ bit long. The first phase of the labeling scheme leads to a labeling of Δ nodes on each level corresponding to their sorted binary numbering from left side to right side.

Furthermore, the definition of the N-MIN^2 guarantees also that all couples of siblings at the same levels have their label differing in the same bit. For this reason, differently from Δ nodes, the set of labels on a level of ∇ does not correspond anymore to a sorted binary numbering but to a permutation.

From these observations, we can deduce: i) a fact giving an upper bound on the number of equivalence classes of N-MIN^{2}s, for a fixed N, and ii) an optimal algorithm to check the equivalence of two $\overset{\Delta}{\nabla}$. In particular, this last result leads to an efficient algorithm to check the topological equivalence of two N-MIN^{2}s.

Fact 11 *Given a $N = 2^n$, the number of non equivalent $\overset{\Delta}{\nabla}$ that are factor of an N-MIN is $(\log N - 1)!$.*

Proof The labels of the leaves of any $\overset{\Delta}{\nabla}$ are $\log N - 1$ bit long. In order to build the labels of the $(\log N + 1)$-th level of $\overset{\Delta}{\nabla}$ we eliminate a bit, among $\log N - 1$, in the labels of the leaves. We can generalize this reasoning for each level of ∇. The thesis follows observing that the elimination of different bits leads to non equivalent graphs. **Q.E.D.**

It is straightforward to notice that, for each fixed N, the number of equivalence classes is different. This implies that any two networks belonging to different classes for a certain N can fall in the same class for a smaller N.

Lemma 12 *Given two $\overset{\Delta}{\nabla}$graphs, both factors of two different N-MIN^{2}s, $O(N)$ time is sufficient to check their topological equivalence.*

Proof Let us label the two $\overset{\Delta}{\nabla}$ by means of the labeling scheme described in the proof of Theorem 10. During the third step of the labeling scheme, associate to each level of ∇ an integer corresponding to the index of the eliminated bit. This procedure allows one to associate to each $\overset{\Delta}{\nabla}$ a sequence $(\log N - 1)$ long. It is easy to see that the two $\overset{\Delta}{\nabla}$ are equivalent if and only if their sequences are the same.

The time complexity is $O(N)$ since each $\overset{\Delta}{\nabla}$ has $\frac{3}{2}N - 2$ nodes and each of them is processed a constant number of times by the labeling scheme. Comparing the two sequences takes $O(\log N) = o(N)$. **Q.E.D.**

Theorem 13 *Given two N-MIN^{2}s, $O(N \log N)$ time is sufficient to check their topological equivalence.*

6 Conclusions and Open Problems

In this paper we have proposed a new general characterization of topological equivalence of N-MINs based on the LCP. Then, we have applied this characterization to $\log N$ and $2\log N - 1$ stage N-MINs.

Observe that the results for $\log N$ and $2\log N - 1$ stage networks are different, since the first one checks the belonging to the equivalence class whose the Reverse Baseline is a representative, while the second one checks the equivalence of any two N-MIN^{2}s. Nevertheless, it does not seem easy to cover this gap using the LCP, since it is not clear how to specify the properties of the complement of the equivalence class which the Reverse Baseline belongs to.

Finally, the characterization of the equivalence classes of $2\log N - 1$ stage N-MINs allows one to do some considerations about the rearrangeability of this set of N-MINs; this is the subject of a further paper [16].

References

[1] D. H. Lawrie: Access and Alignment of Data in an Array Processor, *IEEE Trans. Comput.*, C24, 1975, 1145-1155.

[2] K.E. Batcher: The Flip Network in STARAN,*Proc. Internat. Conf. on Parallel Processing*, 1981, 65-71.

[3] M. C. PEASE: The Indirect Binary Cube Microprocessor Array, *IEEE Trans. Comput.*, C26, 1977, 458-473.

[4] T. Feng: Data Manipulating Functions in Parallel Processors and Their Implementations, *IEEE Trans. Comput.*, C23, 1974, 309-318.

[5] C. WU, T. FENG: On a Class of Multistage Interconnection Networks, *IEEE Trans. Comput.*, C29, 1980, 694-702.

[6] F.T. Leighton: *Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes*. Morgan Kaufmann Publ. San Mateo, CA, 1992.

[7] Beneš, V. E.: On Rearrangeable Three-Stage Connecting Networks, *Bell Syst. Tech. J.*, XLI, 1962,1481-1492.

[8] D.P. Agrawal: Graph Theoretical Analysis and Design of Multistage Interconnection Networks, *IEEE Trans. Comput.*, C32, 1983, 637-648.

[9] J.C. Bermond,J.M. Fourneau, A. Jean-Marie: Equivalence of Multistage Interconnection Networks. *Inform. Proc. Letters* 26, 1987, 45-50.

[10] Q. Hu, X. Shen, J. Yang: Topologies of Combined $(2\log N - 1)$-Stage Interconnection Networks, *IEEE Trans. Comput.*, 46, 1997, 118-124.

[11] C. Wu, T. Feng: The Reverse-Exchange Interconnection Networks, *IEEE Trans. Comput.*, C29, 1980, 801-811.

[12] K. Y. Lee: On the Rearrangeability of $2\log N - 1$-Stage Permutation Networks, *IEEE Trans. Comput.*, C34, 1985, 412-425.

[13] Y. Yeh, T. Feng: On a Class of Rearrangeable Networks, *IEEE Trans. Comput.*, C41, 1992, 1361-1379.

[14] T. Feng,Y. Kim, S. SEO: Interstage Correlation in a Class of Multistage Interconnection Networks, *Proc. 1994 Int. Computer Symp*, 1994.

[15] S. Even, A. Litman: Layered Cross Product - A technique to construct interconnection networks. 4^{th} *ACM SPAA*, 1992, 60-69.

[16] T. Calamoneri, A. Massini: On the Rearrangeability of Multistage Interconnection Networks. *submitted*, 1999.