THE CLASSIFICATION OF COVERINGS OF PROCESSOR NETWORKS

H.L. Bodlaender

RUU-CS-85-11

March 1985

Rijksuniversiteit Utrecht

Vakgroep informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-531454
The Netherlands

THE CLASSIFICATION OF COVERINGS OF PROCESSOR NETWORKS

H.L. Bodlaender

Technical Report RUU-CS-85-11
March 1985

Department of Computer Science
University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht
the Netherlands

THE CLASSIFICATION OF COVERINGS OF PROCESSOR NETWORKS*

H.L. Bodlaender

Department of Computer Science, University of Utrecht
P.O.Box 80.012, 3508 TA Utrecht, the Netherlands

Abstract. Uniform emulations are known in the theory of parallel computing as a class of balanced, structure preserving simulations of large (processor-) networks on smaller (processor-) networks, possibly of the same type. The notion of a covering of one graph by another graph, derived from combinatorial topology, is strongly related to the notion of uniform emulation: every covering of a connected, undirected graph is a uniform emulation. A classification of the coverings of large networks on smaller networks of the same type is given for the following network types: the ring, the grid, the cube, the cube-connected cycles, the tree network and the complete network. A version of the notion of covering for directed graphs is introduced, and the directed coverings of the 4-pin shuffle and the shuffle-exchange networks are completely classified. The problem to decide whether there is a covering of a given graph $G=(V_G,E_G)$ on a given graph $H=(V_H,E_H)$ is shown to be at least as hard as GRAPH ISOMORPHISM, even if $|V_G|/|V_H|$ is fixed to a constant $c \in N^+$.

1. Introduction.

Parallel algorithms are normally designed for execution on a suitable network with N processors, with N depending on the size of the problem to be solved. In practice the size of the problem will be large and varying, whereas the size of the network

---

* This work was supported by the Foundation for Computer Science (SION) of the Netherlands Organisation for the Advancement of Pure Research (ZWO).

will be small and fixed. In [6] Fishburn and Finkel introduced the concept of uniform emulation. Uniform emulations are balanced, structure preserving sumulations of large (processor-) networks, on smaller (processor-) networks, possibly of the same type. Independently Berman [1] proposed a similar notion. An extensive analysis of the concept was made in [2, 3, 4, 5].

<u>Definition</u>. Let $G=(V_G,E_G)$ and $H=(V_H,E_H)$ be (undirected) graphs (processor networks). We say that $f\colon V_G \to V_H$ <u>emulates</u> G on H iff $(v,w)\in E_G$ implies that $f(v)=f(w)$ or $(f(v), f(w)) \in E_H$. f is <u>uniform</u> if there is a $c\in N^+$, such that for all $h\in V_H$ $|f^{-1}(h)|=c$. c is called the <u>computation factor</u> of f.

The concept of covering is a fundamental notion in combinatorial topology. Every covering is a mapping of a d-dimensional complex onto (another) d-dimensional complex. The notion of a one-dimensional complex corresponds with the notion of an undirected graph (with possibly parallel edges and selfloops). (Throughout this paper we will use the terminology of graphs.) In this paper we will study the coverings of graphs, for graphs representing the interconnection structure of processor networks. The following definition can be found in [8], and in many other standard books on combinatorial topology, often in generalized form.

<u>Definition</u>. Let $G=(V_G,E_G)$, $H=(V_H,E_H)$ be undirected graphs. We say that $f\colon V_G \to V_H$ covers G on H (or "G covers H") iff

1. f is surjective
2. $(v,w)\in E_G$ implies $((f(v), f(w)) \in E_H$
3. For all $v\in V_G$, let $w_1,\dots,w_{k_v}$ be the neighbours of v in G. Then $f(w_1),\dots,f(w_{k_v})$ are all different and each neighbour v' of f(v) is equal to a $f(w_i)$ $(1\le i\le k_v)$.

<u>Theorem 1.1</u>. [8] Every covering of a graph G on a connected graph H is uniform.

Corollary 1.2. Let $G=(V_G,E_G)$, $H=(V_H,E_H)$ be undirected graphs and let H be connected. Every covering of G on H is a uniform emulation of G on H.

In [8] the following method is given to describe all graphs that cover a given connected graph H with computation factor c:

- determine a spanning tree B of H,
- make c copies of B,
- for every edge in H that is not an edge in the tree B we have c copies of both its endpoints. These copies are connected on a one-to-one basis: every copy of first endpoint must be connected to a unique copy of the second endpoint. (This gives c! manners to make the connections.)

Up to isomorphism every graph G that covers H can be obtained in the described manner. For further details of the theory of coverings of graphs and more-dimensional complexes, see e.g. [8].

We introduce a directed version of the notion of covering:

Definition. Let $G=(V_G,E_G)$, $H=(V_H,E_H)$ be directed graphs. We say that $f: V_G \to V_H$ covers G on H (or "G covers H") iff

1. f is surjective
2. $(v,w) \in E_G$ implies $(f(v), f(w)) \in E_H$.
3. For all $v \in V_G$, let $w_1,\dots,w_{k_v}$ be the successors of v in G. Then $f(w_1),\dots,f(w_{k_v})$ are all different, and each successor v' of f(v) is equal to a $f(w_i)$ $(1 \le i \le k_v)$.
4. For all $v \in V_G$, let $w'_1,\dots,w'_{l_v}$ be the predecessors of v in G. Then $f(w'_1),\dots,f(w'_{l_v})$ are all different, and each predecessor v' of f(v) is equal to a $f(w'_i)$ $(1 \le i \le l_v)$.

Let $\overline{G}$ be the undirected graph obtained by ignoring the direction of the edges in the directed graph $G=(V,E)$, i.e. $\overline{G}=(V,\overline{E})$, with $\overline{E}=\{(v,w) \mid (v,w) \in E \text{ or } (w,v) \in E\}$.

Lemma 1.3. Let G,H be directed graphs. If G covers H, then $\overline{G}$ covers $\overline{H}$.

Proof. Clear from the definitions. □

Corollary 1.4. Every covering of a directed graph G on a directed graph H is a uniform emulation.

Similar to the description of the graphs that cover an undirected graph H with computation factor c, one can describe all graphs that cover a directed graph $H=(V_H,E_H)$ with computation factor c:

- let $B=(V_H,E_B)$ be a directed graph, such that:

  (i) $\overline{B}$ is a spanning tree of $\overline{H}$

  (ii) $|E_B|=|V_H|-1$, so for every pair of nodes $v_1,v_2 \in V_H$ at most one edge between these nodes (regardless to the direction) is an edge in E_B.
- make c copies of B
- for every edge $(v_1,v_2) \in E_H$ that is not an edge in E_B we have c copies of both its endpoints v_1,v_2. These copies are connected on a one-to-one basis: every copy of the first endpoint must be connected to a unique copy of the second endpoint: the direction of the edges is from a copy of v_1 to a copy of v_2.

Up to isomorphism every graph G that covers H can be obtained in the described manner. (The proof of this characterisation uses lemma 1.3. and the result for the undirected case, and is omitted.)

The notion of covering is much more restricted than the notion of uniform emulation: in many cases there are many more uniform emulations possible of some network G on some other network H then there are coverings possible, and in some cases there exist uniform emulations but no coverings of G on H. A (partial) classification of the uniform emulations of the shuffle exchange network, the 4-pin shuffle, the ring, the grid and the cube network on smaller networks of the same type has been carried out in [2,3]. The classification of coverings of networks on smaller networks of the same type is a special case of this problem. In this paper the classification of coverings is carried out for the following types of graphs; each type is used or proposed as the interconnection structure for current processor networks:

In section 6 the problem to decide whether a given connected graph $G=(V_G,E_G)$ covers a given connected graph $H=(V_H,E_H)$ is addressed and proven to be at least as hard as GRAPH ISOMORPHISM, even if the computation factor $c=|V_G|/|V_H|$ is fixed.

Let $b=b_1...b_n$ be a string with n bits $\in \{0,1\}$, and let $\alpha \in \{0,1\}$. We will use the following notation throughout:

$\frac{0}{1}$: a bit that can be 0 or 1
$\overline{\alpha}$: the complement of bit α. ($\overline{0}=1$, $\overline{1}=0$)
$\overline{b}$: the string one obtains by complementing every bit of b ($\overline{b_1...b_n} = \overline{b_1}...\overline{b_n}$)
$b|_i$: $b_1...b_i$ (truncation after the i^{th} bit)
$_i|b$: $b_i...b_n$ (truncation before the i^{th} bit)
$(b)_i$: b_i (the i^{th} bit)

For functions f defined on n-bit numbers b we use
$f_i(b)$: $(f(b))_i$ (projection on the i^{th} bit)

2. Coverings of ring, grid, tree, cube and complete networks.

2.1. Ring networks.

Definition. The ring with n nodes is the graph $R_n=(V_n,E_n)$ with $V_n=\{0,...,n-1\}$ and $E_n=\{(i, (i \pm 1) \bmod n) \mid i \in V_n\}$.

It is easy to classify the coverings of the ring networks. The following propositions are given without proof.

Proposition 2.1.1. Let G be a graph, consisting of connected components $G_1,\dots,G_j$ and let $k \in N^+$, $k \geq 3$. G covers R_k if and only if for all i, $1 \leq i \leq j$ G_i is graph isomorphic to a ring R_n with $k|n$.

Proposition 2.1.2. Let H be a graph, and $n \in N^+$. R_n covers H if and only if H is graph isomorphic to a ring R_k with $k|n$.

Proposition 2.1.3. The coverings of R_n on R_k with $k \geq 3$, $k|n$ are given by the following list:

(i) $f_j(m) = (m+j) \bmod k$ (for j, $0 \leq j < k$)

(ii) $\overline{f}_j(m) = (-m+j) \bmod k$ (for j, $0 \leq j < k$)

Proof. It is easy to verify that every f_j, $\overline{f}_j$ is a covering of R_n on R_k. Let f be a covering of R_n on R_k. Consider f(0) and f(1). If $f(1)=(f(0)+1) \bmod k$, then f is of type f_j; if $f(1)=(f(0)-1) \bmod k$, then f is of type $\overline{f}_j$. In both cases $j=f(0)$. With induction one can verify that f must be of the designated type. □

Corollary 2.1.4. For all k,n, $k|n$, $k \geq 3$ there are precisely 2k coverings of R_n on R_k.

2.2. Grid Networks.

Definition. The two-dimensional grid network with boundary connections is the graph $GR_n=(V_n,E_n)$ with $V_n=\{(i,j) \mid i,j \in N,\ 0 \leq i,j \leq n-1\}$ and $E_n = \{((i,j),(i',j')) \mid (i,j),(i',j') \in V_n$ and $(i=i' \wedge j = (j' \pm 1) \bmod n)$ or $(i=(i' \pm 1) \bmod n \wedge j=j')\}$. The two-dimensional grid network without boundary connections is the graph $GR_n'=(V_n,E_n')$ with $E_n' = \{((i,j),(i',j')) \mid (i,j),(i',j') \in V_n$ and $(i=i' \wedge j=j' \pm 1)$ or $(i=i' \pm 1 \wedge j=j')\}$.

Proposition 2.2.1. If $n \neq m$ then there exist no covering of GR_n' on GR_m'.

Proof. GR_n' and GR_m' both have exactly 4 nodes of degree 2. Because degree(v) = degree(f(v)) for all $v \in V_n$, and a covering of GR_n' on GR_m'

must be uniform, with a computation factor $\neq$ 1 because $n\neq m$, there exists no covering of GR_n' or GR_m'. □

One can view the two-dimensional grid network with boundary connections as the product of two ringnetworks: $GR_n \cong R_n x R_n$. Every covering of GR_n on GR_m ($m|n$, $m\geq5$) can also be written as the product of two coverings of R_n on R_m.

Theorem 2.2.2. Let $m|n$, $m\geq5$, and let f be a function $V_n \rightarrow V_m$. f is a covering of GR_n on GR_m, if and only if there are coverings f_1, f_2 of R_n on R_m, with

$$f((i,j)) = (f_1(i), f_2(j)) \text{ for all } (i,j) \in V_n$$

$$\text{or } f((i,j)) = (f_1(j), f_2(i)) \text{ for all } (i,j) \in V_n.$$

Proof. First note that every cycle with 4 nodes (a "square") in GR_n must be mapped on a cycle with 4 nodes (a "square") in GR_m. With induction one can prove that, after having fixed the image of one cycle with 4 nodes (square) in GR_n, the whole covering is fixed and necessarily of the described form. □

By the results of section 2.1. theorem 2.2.2. implies a complete characterization of the coverings of GR_n on GR_m, $m|n$, $m\geq5$.

2.3. Tree networks. The following theorem from combinatorial topology characterizes precisely which graphs cover a given tree network.

Theorem 2.3.1. [8] Let H be a tree, and let G cover H. Then G consists of a number (≥1) of connected components, each graph isomorphic to H. In particular, if G is connected then G and H are graph isomorphic.

2.4. Cube networks.

Definition. The cube network with 2^n nodes or n-cube is the graph $C_n=(V_n,E_n)$ with $V_n=\{(b_1..b_n)| \forall i,\ 1\leq i\leq n\ b_i=\frac{0}{1}\}$ and $E_n=\{(b,c)|\ b,c \in V_n$ and b,c differ in exactly one bitposition$\}$.

<u>Proposition</u> 2.4.1. Let C_n cover C_m. Then n=m.

<u>Proof</u>.

This follows because degree (f(v))=degree(v), for any covering f of C_n on C_m and $v \in C_n$. □

2.5. <u>Complete networks</u>. The same argument can be used for complete networks.

<u>Definition</u>. The complete network with n nodes is the graph $K_n=(V_n,E_n)$ with $V_n=\{0,..,n-1\}$ and $E_n=\{(i,j) \mid i,j \in V_n \wedge i\neq j\}$.

<u>Proposition</u> 2.5.1. Let K_n cover K_m. Then n=m.

3. <u>Coverings of the cube-connected cycles</u>. For denoting the 'cube-connected cycles' graph we use the notation introduced in [6]. Processors in the cube-connected cycles with $r.2^r$ nodes are addressed by r-bit strings with a "divide" in any one position between bits. The position to the left of the first bit is identified with the position to the right of the last bit, so there are r positions for the divide. A node $p_1...p_{i-1} | p_i...p_r$ is connected to the following 3 nodes:

$p_1...\overline{p_{i-1}} | p_i...p_r$

$p_1...p_{i-2} | p_{i-1}\ p_i...p_r$

and $p_1...p_{i-1}\ p_i | p_{i+1}...p_r$.

<u>Definition</u>. The cube-connected cycles graph with 2^r nodes is the (undirected) graph $CCC_r = (V_r,E_r)$, with $V_r = \{p_1...p_{i-1}|p_i...p_r \mid p_1...p_r \in (\frac{0}{1})^r,\ 1\leq i\leq r\}$ and $E_r = \{(p_1...p_{i-1}|p_i...p_r, p_1...\overline{p_{i-1}}|p_i...p_r) \mid p_1...p_{i-1}|p_i...p_r \in V_r\}$ $\{(p_1...p_{i-1}|p_i...p_r, p_1...p_i|p_{i+1}...p_r) \mid p_1...p_{i-1}|p_i...p_r,\ p_1...p_i|p_{i+1}...p_r \in V_r\}$. (Let - (+) be the subtraction (addition) modulo r.)

Edges of the type $(p_1...p_{i-1}|p_i...p_r,\ p_1...\overline{p_{i-1}}|p_i...p_r)$ are called exchange edges; edges of the type $(p_1...p_{i-1}|p_i...p_r, p_1...p_i|p_{i+1}...p_r)$ are called divide shift edges. Notice that every node is adjacent to 2 divide shift edges and one exchange edge. When a divide shift edge is passed in the direction from $p_1 \ldots p_{i-1} | p_i \ldots p_r$ to $p_1 \ldots p_i | p_{i+1} \ldots p_r$ we say it is passed in the positive direction, otherwise we say it is passed in the negative direction.

We will only consider coverings of CCC_r on CCC_s with $r \geq s \geq 9$.

<u>Lemma 3.1</u>. Every cycle in CCC_r with $r \geq 9$ has at least 8 nodes. For every cycle with exactly 8 nodes in CCC_r with $r \geq 9$ there are $p_1 \ldots p_r \in (\frac{0}{1})^r$, $i \in \{1,..,r\}$, such that the successive nodes visited by the cycle (assuming starting point and direction in which the cycle is traversed are well chosen) are:

$$\begin{array}{l}
p_1 \ldots p_i \,|\, p_{i+1}\; p_{i+2} \ldots p_r \\
p_1 \ldots p_i\; p_{i+1} \,|\, p_{i+2} \ldots p_r \\
p_1 \ldots p_i\; \overline{p_{i+1}} \,|\, p_{i+2} \ldots p_r \\
p_1 \ldots p_i \,|\, \overline{p_{i+1}}\; p_{i+2} \ldots p_r \\
p_1 \ldots \overline{p_i} \,|\, \overline{p_{i+1}}\; p_{i+2} \ldots p_r \\
p_1 \ldots \overline{p_i}\; \overline{p_{i+1}} \,|\, p_{i+2} \ldots p_r \\
p_1 \ldots \overline{p_i}\; p_{i+1} \,|\, p_{i+2} \ldots p_r \\
p_1 \ldots \overline{p_i} \,|\, p_{i+1}\; p_{i+2} \ldots p_r
\end{array}$$

Proof.

This is easy, but tedious to check. □

Now let f be a covering of CCC_r or CCC_s, with $r \geq s \geq 9$. Every cycle with 8 nodes of CCC_r must be mapped upon a cycle with 8 nodes of CCC_s. Notice that every edge that is adjacent to a cycle with 8 nodes but not part of this cycle, necessarily is a divide shift edge.

<u>Lemma 3.2</u>. Let f cover CCC_r on CCC_s, $r \geq s \geq 9$. Then f maps nodes connected by a divide shift edge on nodes connected by a divide shift edge and nodes connected by an exchange edge on nodes connected by an exchange edge.

Proof.

For every divide shift edge e there is a cycle of 8 nodes to which e is adjacent, but of which e is no part. Hence e must be mapped upon an edge, adjacent to a cycle of 8 nodes, that is: a divide shift edge. Because every node is adjacent to exactly 2 divide shift edges and one exchange edge, exchange edges must be mapped upon exchange edges. □

Now let $f(0|0..0)$ be given. There are two possible values for $f(00|0..0)$, the node reachable from $f(0|0..0)$ by a divide shift in positive direction, and the node reachable from $f(0|0..0)$ by a divide shift in negative direction. The choice of $f(0|0..0)$ and $f(00|0..0)$ fixes the covering f.

Lemma 3.3. Let b and c be nodes in CCC_r, that are connected by a divide shift edge. There is at most one covering f of CCC_r on CCC_s ($r \geq s \geq 9$) with $f(0|0..0) = b$ and $f(00|0..0) = c$.

Proof.

Suppose $f(0|0...0) = b = p_1...p_{i-1}|p_i...p_s$ and $f(00|0...0) = c = p_1...p_{i-1}p_i|p_{i+1}...p_s$. (If $b \to c$ is a divide shift in negative direction, then the proof is analogous.) Let f be a covering of CCC_r on CCC_s. If $f(q_1...q_{j-1}|q_j...q_r)$ reaches $f(q_1...q_j|q_{j+1}...q_r)$ by a divide shift in positive direction, then every $f(q_1...q_{j'-1}|q_{j'}...q_r)$ reaches $f(q_1...q_{j'}|q_{j'+1}...q_r)$ by a divide shift in positive direction ($1 \leq j' \leq r$). Similarly for negative directions.

Proposition 3.3.1. Let b',c' be nodes in CCC_r, and c' can be reached from b' by a divide shift in positive direction. Then f(c') can be reached from f(b') by a divide shift in positive direction.

Proof.

Suppose not. We now can divide $(\frac{0}{1})^r$ in two disjunctive sets A and B, with $A = \{q_1...q_r \mid f(q_1...q_{i-1}|q_i...q_r)$ reaches $f(q_1...q_i|q_{i+1}...q_r)$ by a divide shift in positive direction$\}$ and $B = \{q_1...q_r \mid f(q_1...q_{i-1}|q_i...q_r)$ reaches $f(q_1...q_i|q_{i+1}...q_r)$ by a divide shift in negative direction}. (Note that the choice of i is not important.) We know that A and B are not empty. For instance $0^n \in$ A.

There must be strings $a \in A$, $b \in B$, that differ in exactly one bitposition. So there are $p_1 \ldots p_r \in (\frac{0}{1})^r$, i, $1 \leq i \leq r$, $q_1 \ldots q_s \in (\frac{0}{1})^s$, $q_1' \ldots q_s' \in (\frac{0}{1})^s$, j, $1 \leq j \leq s$, j', $1 \leq j' \leq s$, with

- $f(p_1 \ldots p_{i-1} | \; p_i \; p_{i+1} \ldots p_r) = q_1 \ldots q_{j-1} | \; q_j \; q_{j+1} \ldots q_r$
- $f(p_1 \ldots p_{i-1} \; p_i | \; p_{i+1} \ldots p_r) = q_1 \ldots q_{j-1} \; q_j | \; q_{j+1} \ldots q_r$
- $f(p_1 \ldots p_{i-1} | \; \overline{p_i} \; p_{i+1} \ldots p_r) = q_1' \ldots q_{j'-1}' \; q_{j'}' | \; q_{j'+1}' \ldots q_r'$
- $f(p_1 \ldots p_{i-1} \; \overline{p_i} | \; p_{i+1} \ldots p_r) = q_1' \ldots q_{j'-1}' | \; q_{j'}' \; q_{j'+1}' \ldots q_r'$

From 3.2 we have

- $f(p_1 \ldots p_{i-1} \; \overline{p_i} | \; p_{i+1} \ldots p_r) = q_1 \ldots q_{j-1} \; \overline{q_j} | \; q_{j+1} \ldots q_r$, hence
- $f(p_1 \ldots p_{i-1} | \; \overline{p_i} \; p_{i+1} \ldots p_r) = q_1 \ldots q_{j-1} \; \overline{q_j} \; q_{j+1} | q_{j+2} \ldots q_r$.

Now notice that $p_1 \ldots p_{i-1} | p_i p_{i+1} \ldots p_r$; $p_1 \ldots p_{i-1} p_i | p_{i+1} \ldots p_r$; $p_1 \ldots p_{i-1} | \overline{p_i} p_{i+1} \ldots p_r$; $p_1 \ldots p_{i-1} \overline{p_i} | p_{i+1} \ldots p_r$ are 4 successive nodes in a cycle with 8 nodes in CCC_r. However, their images are not 4 successive nodes in a cycle with 8 nodes in CCC_s. So there is a cycle with 8 nodes that is not mapped upon a cycle with 8 nodes. Contradiction. □

This shows that f is completely determined by the choice of $f(0|0..0)$ and $f(00|0..0)$. Let d be a node of CCC_r. Look at the path from $0|0..0$ to d, and the f-images of the nodes on this path. If the path uses an exchange edge, a divide shift in positive direction or a divide shift in negative direction, respectively, then the path formed by the f-images must use the same type of edge. Hence f(d) is fully determined by $f(0|0..0)$ and $f(00|0..0)$. □

<u>Corollary 3.4</u>. Let $s|r$. There are at most $2s.2^s$ coverings of CCC_r on CCC_s.

Proof.

There are $s.2^s$ possible choices for $f(0|0..0)$. For $f(00|0..0)$ there are 2 choices left. □

Lemma 3.5. Let CCC_r cover CCC_s. Then $s|r$.

Proof.

The number of nodes of CCC_s must divide the number of nodes of CCC_r. I.e. $s.2^s|\ r.2^r$, so $s|r$. □

Consider the following graph isomorphisms of CCC_s.

1. Let $I \subseteq \{1,..,s\}$. The function X_I does not move the divide, but flips the bits with index in I:

   $X_I(p_1...p_j|p_{j+1}...p_s) = q_1...q_{j'}|q_{j'+1}...q_s \Leftrightarrow$
   $j=j'$; $q_i=p_i$ for all $i \notin I$; $q_i \neq p_i$ for all $i \in I$.

2. Let $t \in \{0,..,s-1\}$. The function S_t shifts the whole string, the divide incluis, t positions to the left:

   $S_t(p_1...p_i|p_{i+1}..p_s) = p_{t+1}..p_i|p_{i+1}..p_s\ p_1...p_t$
   (or, of course, $p_{t+1}...p_sp_1...p_i|p_{i+1}...p_t$).

3. The function R reverses the string. Notice that the divide is still placed <u>after</u> the same bit, not between the same bits, as originally:

   $R(p_1...p_i|p_{i+1}...p_s) = p_s...p_{i+1}p_i|p_{i-1}...p_1$.

It is easy to check that for every $I \subseteq \{1,..,s\}$, $t \in \{0,..,s-1\}$, X_I, S_t and R are graph isomorphisms (= coverings) of CCC_s onto itself.

Corollary 3.6. Every graph isomorphism of CCC_s is of one of the following forms:

$X_I \circ S_t$ $(I \subseteq \{1,..,s\},\ 0 \leq t \leq s-1)$

or $X_I \circ S_t \circ R$ $(I \subseteq \{1,..,s\},\ 0 \leq t \leq s-1)$.

Proof.

The described forms give $2s.2^s$ different graph isomorphisms of CCC_s. Corollary 3.4. shows there cannot be more (every isomorphism is a covering). □

Consider the following function $F: CCC_r \to CCC_s$:

$$F(p_1 \dots p_i | p_{i+1} \dots p_r) = q_1 \dots q_{i'} | q_{i'+1} \dots q_r$$
$$\Leftrightarrow q_j = \left(\sum_{k=0}^{r/s-1} b_{ks+j} \right) \bmod 2 \text{ (for all } j,\ 1 \leq j \leq s)$$

and $i' = i \bmod s$ (i.e. the divide is placed after the $i \bmod s^{th}$ bit).

Lemma 3.7. F is a covering of CCC_r on CCC_s.

Proof.

It is clear that F is surjective.

Consider a node $p_1 \dots p_i | p_{i+1} \dots p_r$ in CCC_r and its 3 neighbours. From the definition of F it is clear that each of these neighbours is mapped upon another neighbour of F(p). □

Theorem 3.8. Every covering f of CCC_r on CCC_s can be written as $f = J \circ F$, where F is given by the definition above, and J is a graph isomorphism of CCC_s onto itself.

Proof.

Every function of the form $J \circ F$ is a covering of CCC_r on CCC_s. There are $2s.2^s$ possible choices for J, so there are $2s.2^s$ possible functions of this form. It follows from corollary 3.4. that there cannot be more coverings of CCC_r on CCC_s. □

4. Coverings of the 4-pin shuffle. Stone [9] proposed a network, called the shuffle-exchange network, which has been successfully used as the interconnection network underlying a variety of parallel processing algorithms. There are two slightly different types of graphs, both realizing Stone's concept of a shuffle-exchange network. We use the terminology of [6] and call these graphs the shuffle-exchange

(graph) and the 4-pin shuffle.

Nodes in the 4-pin shufle are given n-bit addresses in the range $0..2^n-1$. A node $b_1..b_n$ has edges to the two nodes $b_2..b_n0$ and $b_2..b_n1$ (move the leading bit to tail position and possibly flip this bit).

The following definitions are taken from [2]:

Definition. The 4-pin shuffle is the graph $S_n=(V_n,E_n)$ with $V_n=\{b_1...b_n \mid \forall 1\leq i\leq n\ b_i=\frac{0}{1}\}$ and $E_n=\{(b,c) \mid b,\ c \in V_n \wedge \forall 2\leq i\leq n\ b_i=c_{i+1}\}$.

Definition. An emulation $f\colon S_n \to S_{n-k}$ is a step-simulation, iff $(b,c) \in E_n \Rightarrow (f(b),\ f(c)) \in E_k$

Lemma 4.1. Every covering of S_n on S_{n-k} is a uniform step-simulation.

Definition. Let $f\colon S_n \to S_{n-k}$ be a step-simulation. $\tilde{f}\colon S_{k+1} \to S_1$ is defined by $\tilde{f}(b_1...b_{k+1}) = f_1(b_1...b_{k+1}0..0)$.

Theorem 4.2. [2] The mapping π, defined by $\pi(f)=\tilde{f}$ from the step-simulations $S_n \to S_{n-k}$ to the step-simulations $S_{k+1} \to S_1$ is a bijection, with the following further properties:

1. If f is uniform, then $\pi(f)=\tilde{f}$ is uniform
2. $f(b_1...b_n)=\pi(f)(b_1...b_{k+1}).\pi(f)(b_2...b_{k+2})...\pi(f)(b_{n-k}...b_n)$.

Lemma 4.3. Let f cover S_n on S_{n-k}. Then:
(1) $\{\tilde{f}(0\ b_1..b_k),\ \tilde{f}(1\ b_1..b_k)\} = \{0,1\}$, for all $b_1..b_k \in (\frac{0}{1})^k$.
(2) $\{\tilde{f}(b_1..b_k\ 0),\ \tilde{f}(b_1..b_k\ 1)\} = \{0,1\}$, for all $b_1..b_k \in (\frac{0}{1})^k$.

Proof.

(1) $b_1..b_k0..0$ has predecessors $0b_1..b_k0..0$ and $1b_1..b_k0..0$. Hence $f(0b_1..b_k0..0)$ and $f(1b_1..b_k0..0)$ must be different: they can only be different in their first coordinate, because they are both predecessors of $f(b_1..b_k0..0)$, so $\tilde{f}(0b_1..b_k) \neq \tilde{f}(1b_1..b_k)$.

(2) is proved by similar argument. □

So, for coverings f of S_n on S_{n-k} we have that for all $x\in(\frac{0}{1})^{k-2}$ $\tilde{f}(0x0)$

$= \tilde{f}(1x1) \neq \tilde{f}(0x1) = \tilde{f}(1x0)$. However, this is also a sufficient condition for a step-simulation f to be a covering.

<u>Lemma 4.4</u>. Let f: $S_n \rightarrow S_{n-k}$ be step-simulating, and let for all $x \in (\frac{0}{1})^{k-2}$ $\tilde{f}(0x0) = \tilde{f}(1x1) \neq \tilde{f}(1x0) = \tilde{f}(1x0)$. Then f is a covering.

<u>Proof</u>.

Without much difficulty it can be checked that the conditions for the coverings of directed graphs are fulfilled. □

So we have a necessary and sufficient condition for a step-simulation $S_n \rightarrow S_{n-k}$ to be a covering. With the help of this condition we can completely classify the coverings of S_n on S_{n-k}.

<u>Theorem 4.5</u>.

a. If k=1 then the coverings of S_n on $S_{n-k} = S_{n-1}$ are the following:

$f(b_1..b_n) = c_1..c_{n-1}$ with $c_i = (b_i \equiv b_{i+1})$ $(1 \leq i \leq n-1)$

$\overline{f}(b_1..b_n) = \overline{c_1}..\overline{c_{n-1}}$ with $c_i = (b_i \equiv b_{i+1})$ $(1 \leq i \leq n-1)$

b. If $2 \leq k < n$, then every covering f of S_n on S_{n-k} can be found by choosing for each string $x \in (\frac{0}{1})^{k-1}$ whether

- $\tilde{f}(0x0) = \tilde{f}(1x1) = 0$ and $\tilde{f}(0x1) = \tilde{f}(1x0) = 1$

or - $\tilde{f}(0x0) = \tilde{f}(1x1) = 1$ and $\tilde{f}(0x1) = \tilde{f}(1x0) = 0$

c. For every $n > k \geq 1$ there are exactly $2^{2^{k-1}}$ coverings of S_n on S_{n-k}.

(In [2] it was proven that the functions f: $b \rightarrow b$ and f: $b \rightarrow \overline{b}$ are the only possible graph isomorphisms of S_n (= coverings of S_n on S_n).)

<u>5. Coverings of the shuffle-exchange network</u>. Nodes in the shuffle-exchange network are again given n-bit addresses in the range $0..2^{n-1}$. There is an edge from node b to node c if and only if b can be "shuffled" (move the leading bit to tail position) <u>or</u> "exchanged" (flip the tail bit) into c.

<u>Definition</u>. The shuffle-exchange network is the graph $SE_n = (V_n, \tilde{E}_n)$ with $V_n = \{(b_1,\ldots,b_n) \mid \forall 1 \leq i \leq n \; b_i = \frac{0}{1}\}$ and $\tilde{E}_n = \{(b,c) \mid b,c \in V_n \wedge$

$(\forall 2 \leq i \leq n\ b_i = c_{i-1} \wedge b_1 = c_n)$ or $(\forall 1 \leq i \leq n-1\ b_i = c_i \wedge b_n = \overline{c_n})\}$.

Theorem 5.1. There do not exists coverings of SE_n on SE_k with $k \nmid n$ and $k \geq 3$.

Proof.

Suppose $k \nmid n$, $k \geq 3$ and f is a covering of SE_n on SE_k. With $R^l(b)$ we denote the string $b_{l+1 (\mathrm{mod}\ n)} \dots b_n b_1 \dots b_{l (\mathrm{mod}\ n)}$, i.e. b rotated l positions to the left.

Let $b \in f^{-1}(0^{k-1}1)$.

Lemma 5.1.1. $f(R^l(b)) = R^l(f(b)) = R^l(0^{k-1}1)$, for all $l \geq 0$.

Proof.

With induction. For l=0 the lemma is trivially true. Suppose the lemma holds for certain l. Then it holds also for l+1:

$f(R^l(b)|_{n-1}\ \overline{(R^l(b))_n})$ must be mutually adjacent to $f(R^l(b))$, so must be connected with $f(R^l(b))$ via the exchange edge (we use that $f(R^l(b))$ cannot be of the form $(01)^{n/2}$ or $(10)^{n/2}$ (induction hypotheses)). This shows that $f(R^{l+1}(b))$ must be connected with $f(R^l(b))$ via a shuffle-edge, hence $f(R^{l+1}(b)) = R^1(f(R^l(b)) = R^{l+1}(f(b))$. □

In particular we now have: $0^{k-1}1 = f(b) = f(R^n(b)) = R^n(0^{k-1}1)$. So $k|n$. Contradiction. □

Theorem 5.2. Let $k \geq 3$, $k|n$. The coverings of SE_n on SE_k are given by the following list:

1. f', defined by $f'_i(b_1 \dots b_n) = (\sum_{j=0}^{n/k-1} b_{j.k+i}) \mathrm{mod} 2$ $(1 \leq i \leq k)$.

2. $\overline{f'}$, defined by $\overline{f'_i}(b_1 \dots b_n) = \overline{(\sum_{j=0}^{n/k-1} b_{j.k+i}) \mathrm{mod} 2}$ $(1 \leq i \leq k)$.

Proof.

Because 0^n in SE_n has a selfloop, and 0^k, 1^k are the only nodes in SE_k with a selfloop one has for every covering f of SE_n on SE_k: $f(0^n) = 0^k$ or $f(0^n) = 1^k$. We suppose we have a covering f of SE_n on SE_k with $f(0^n) = 0^k$, and will show that f is then necessarily of the form f'. If $f(0^n) = 1^k$ then f is of the form $\overline{f'}$.

Lemma 5.2.1. If $f(b) = f'(b)$ and $f(b) \notin \{(01)^{n/2}, (10)^{n/2}\}$,
then 1. $f(b_1 \ldots b_{n-1}\ \overline{b_n}) = f'(b_1 \ldots b_{n-1}\ \overline{b_n})$
2. $f(b_2 \ldots b_n\ b_1) = f'(b_2 \ldots b_n\ b_1)$
3. $f(b_n\ b_1 \ldots b_{n-1}) = f'(b_n\ b_1 \ldots b_{n-1})$

Proof.

First note $f(b_1 \ldots b_{n-1}\ \overline{b_n})$ must be mutually adjacent to f(b). Hence $f(b_1 \ldots b_{n-1}\ \overline{b_n}) = f'(b_1 \ldots b_{n-1}\ \overline{b_n})$. (Use the definition of f'.) Further use that $f(b_2 \ldots b_n\ b_1)$ must be a successor of f(b), unequal to $f(b_1 \ldots b_n\ \overline{b_n})$. This shows $f(b_2 \ldots b_n\ b_1) = f'(b_2 \ldots b_n b_1)$. Similarly $f(b_n\ b_1 \ldots b_{n-1}) = f'(b_n\ b_1 \ldots b_{n-1})$. □

Without proof we mention the following result:

Lemma 5.2.2. For every $b \in \binom{0}{1}^n$ there is a path from 0^n to b, such that for every node c on the path, except b: $f'(c) \notin \{(01)^{n/2}, (10)^{n/2}\}$.

(If n is odd, it will of course never be the case that $f'(c) \in \{(01)^{n/2}, (10)^{n/2}\}$.)

With induction to the length of the path, mentioned in 5.2.2. one now can prove that $f(b) = f'(b)$ for every $b \in \binom{0}{1}^n$. □

6. Complexity results. In this section we consider the following problem:

[GRAPH COVERING]

Instance: Connected, undirected graphs $G=(V_G,E_G)$, $H=(V_H,E_H)$

Question: Is there a covering of G on H?

The subproblem of GRAPH COVERING, in which the computation factor $c=|V_G|/|V_H|$ is fixed is called c-GRAPH COVERING. (The related problems UNIFORM EMULATION and c-UNIFORM EMULATION were addressed and proven NP-complete in [4,5].) The following lemma shows that 1-GRAPH COVERING and GRAPH ISOMORPHISM are virtually the same problem.

Lemma 6.1. [8] Let $G=(V_G,E_G)$ and $H=(V_H,E_H)$ be connected, undirected graphs and $|V_G|=|V_H|$. A function $f: V_G \to V_H$ covers G on H iff f is a graph isomorphism of G on H.

It is still an open problem whether GRAPH ISOMORPHISM is solvable in polynomial time or not, and whether it is NP-complete or not [7]. Theorem 6.2. shows that c-GRAPH COVERING is at least as hard as GRAPH ISOMORPHISM, indicating that it will be very hard, if not impossible to find a polynomial time algorithm for it: any polynomial time algorithm for c-GRAPH COVERING would enable us to solve GRAPH ISOMORPHISM in polynomial time.

Theorem 6.2. If there exists an algorithm that solves c-GRAPH COVERING in time $\leq f(|V_H|)$, then there exists an algorithm that solves GRAPH ISOMORPHISM in time $\leq f(|V_H|+4)+ O(\max\{|V_H|, |E_G|, |E_H|\})$, for every $c\geq 1$.

Proof.

For c=1 this follows from lemma 10.1. Let $c\geq 2$ be given, and let there exists an algorithm for c-GRAPH COVERING that uses time $\leq f(|V_H|)$, that is: it decides whether $G=(V_G,E_G)$ covers $H=(V_H,E_H)$ in time at most $f(|V_H|)$.

Now let an instance of GRAPH ISOMORPHISM be given, i.e. we have two connected graphs $G=(V_G,E_G)$ and $H=(V_H,E_H)$, with $|V_G|=|V_H|$ and $|E_G|=|E_H|$ and ask the question whether G and H are graph isomorphic.

We will now define graphs G', H' with $|V_{G'}|=|V_H|+4$, $|V_{G'}|=c.|V_{H'}|$ and G' covers H' if and only if G and H are graph isomorphic.

To H we add 4 extra nodes v_1', v_2', v_3', v_4', with extra edges between v_1' and every node in V_H, and (v_1', v_2'), (v_2', v_3'), (v_3', v_4'), and (v_4', v_2'). (See fig. 6.1.)

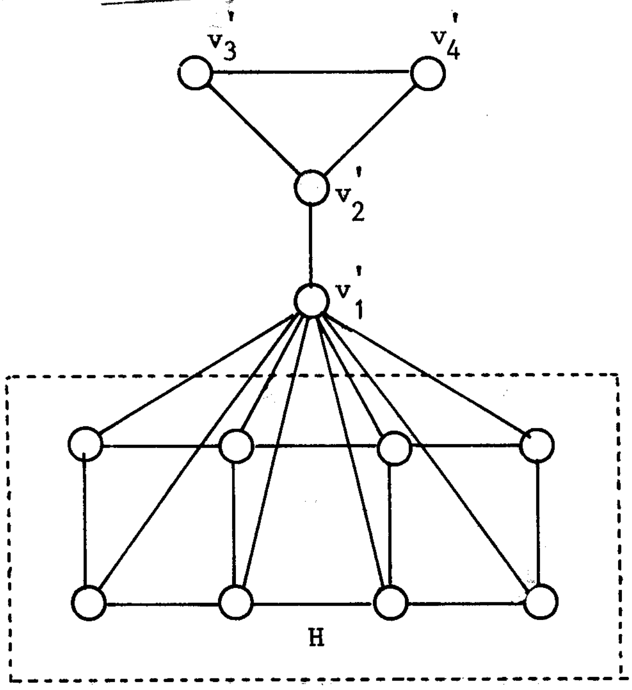

fig. 6.1.

G' is formed by taking c copies of G, adding 4c extra nodes v_j^i $(1 \leq i \leq c,\ 1 \leq j \leq 4)$, and adding the following edges:

- v_1^i is connected to every node in the i^{th} copy of G $(1 \leq i \leq c)$
- v_1^i is connected to v_2^i $(1 \leq i \leq c)$
- v_2^i is connected to v_3^i and to v_4^i $(1 \leq i \leq c)$
- v_3^i is connected to $v_4^{(i+1) \bmod c}$ $(1 \leq i \leq c)$

An example of the construction is shown in fig 6.2., with c=4.

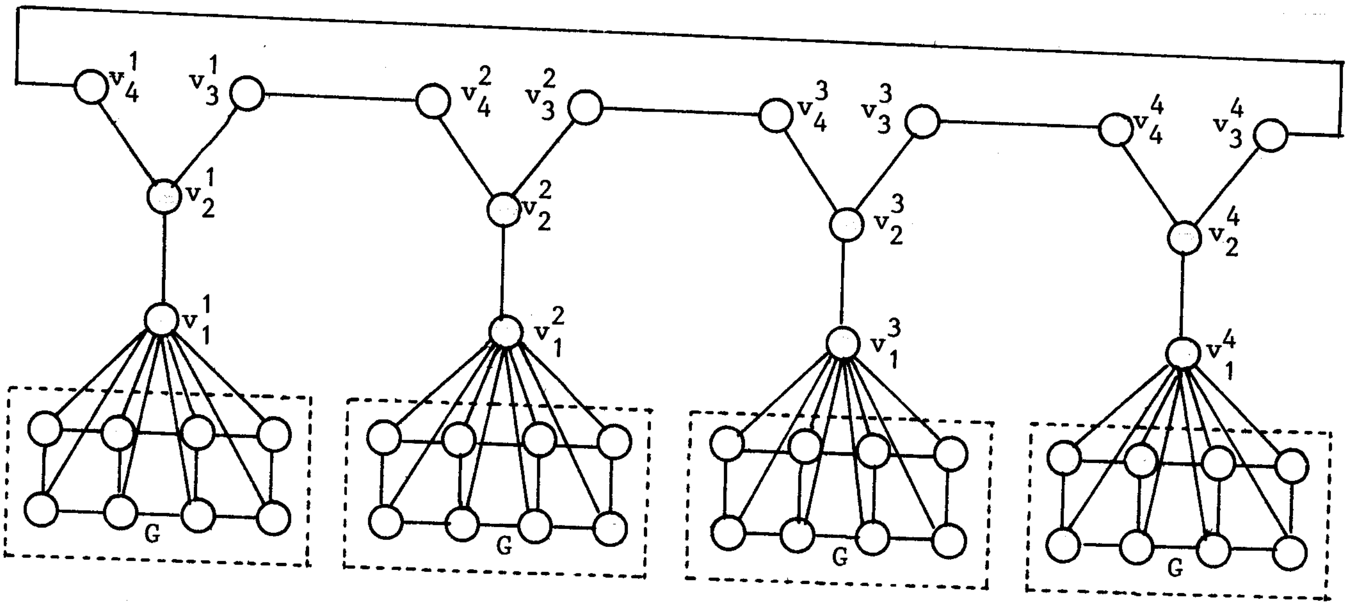

fig 6.2. G' with c=4

Claim 6.2.1. G' covers H' if and only if G and H are graph isomorphic.

Proof.

Let f cover G' on H'. Note that the only node with degree $|V_H|+1$ in H' is v_1'. Hence $f(v_1^i) = v_1'$ for every i, $1 \le i \le c$. This shows that the copy of G, connected to v_1^1, must be mapped upon the copy of H, connected to v_1' and f restricted to this copy of G gives a graph isomorphism of G on H.

It is easy to see that if G and H are graph isomorphic, then G' covers H'. □

G' and H' can be constructed in time $O(\max\{|V_H|, |E_G|, |E_H|\})$, hence there exists an algorithm that solves GRAPH ISOMORPHISM in time $\le f(|V_H|+4) + O(\max\{|V_H|, |E_G|, |E_H|\})$. □

It is an interesting open question open whether c-GRAPH COVERING is polynomially equivalent to GRAPH ISOMORPHISM for $c \ge 2$, and whether it is NP-complete.

References

[1] Berman, F., Parallel processing with limited resources, Proc. of the Conf. on Information Sciences and Systems, pp. 675-679, The Johns Hopkins University, 1983.

[2] Bodlaender, H.L. and J. van Leeuwen, Simulation of large networks on smaller networks, Tech. Rep. RUU-CS-84-4, Dept. of Computer Science, University of Utrecht, Utrecht, 1984. (Extended abstract in: K. Mehlhorn (ed.), Proc. of the 2nd Ann. Symp. on Theoretical Aspects of Computer Science (STACS 85), Lecture Notes in Computer Science, Vol. 182, pp. 47-58, Springer Verlag, Berlin 1985.)

[3] Bodlaender, H.L. Uniform emulations of two different types of shuffle exchange networks, Tech. Rep. RUU-CS-84-9, Dept. of Computer Science, University of Utrecht, 1984.

[4] Bodlaender, H.L. and J. van Leeuwen, The complexity of finding uniform emulations, Tech. Rep. RUU-CS-85-4, Dept. of Computer Science, University of Utrecht, Utrecht, 1985.

[5] Bodlaender, H.L., The complexity of finding uniform emulations on paths and ring networks, Tech. Rep. RUU-CS-85-5, Dept. of Computer Science, University of Utrecht, Utrecht, 1985.

[6] Fishburn, J.P. and R.A. Finkel, Quotient networks, IEEE Trans. on Comput. C-31 (1982) 288-295.

[7] Garey, M.R. and D.S. Johnson, Computers and Intractability, a guide to the theory of NP-completeness, W.H. Freeman, San Francisco, Calif., 1979.

[8] Reidemeister, K., Einführung in die Kombinatorische Topologie, Friedr. Vieweg & Sohn Akt. Ges., Braunschweig, 1932.

[9] Stone, H.S., Parallel processing with the perfect shuffle, IEEE Trans. on Comput. C-20 (1971) 153-161.