THE COMPLEXITY OF FINDING UNIFORM EMULATIONS ON PATHS AND RING NETWORKS

Hans L. Bodlaender

RUU-CS-85-5
April 1988

Rijksuniversiteit Utrecht

Vakgroep Informatica

Budapestlaan 6 3584 CD Utrecht
Corr. adres: Postbus 80.012 3508 TA Utrecht
Telefoon 030-53 1454
The Netherlands

THE COMPLEXITY OF FINDING UNIFORM EMULATIONS ON PATHS AND RING NETWORKS*

Hans L. Bodlaender
Department of Computer Science, University of Utrecht
P.O.Box 80.089, 3508 TB Utrecht, the Netherlands

Abstract

Uniform emulations are a method to obtain efficient, balanced simulations of large processor networks (guests) on smaller processor networks (hosts). Each uniform emulation has associated with it a constant c, called the computation factor, which is the size of the guest network divided by the size of the host network. In this paper we prove that the problem to decide whether a given connected graph (guest network) G can be uniformly emulated on a path or a ring network is NP-complete, even if G is required to be a binary tree. If the computation factor c is fixed, then these problems become solvable in polynomial time for arbitrary connected graphs G. However, if we keep the computation factor fixed, and at least 4 or 2 for the path- or ring-version, respectively, but allow disconnected graphs G, then the problem again is NP-complete. Similar results are shown for directed graphs.

1 Introduction.

Often the need arises to simulate one processor network on another processor network, for instance when a parallel algorithm is designed for execution on a certain network, but must be implemented on another network. In this paper we study a notion of simulation, termed emulation, proposed by Fishburn and Finkel [4], which is suitable to obtain efficient simulations of large networks on smaller networks. Independently, Berman [1] proposed a similar notion.

Definition.
Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be networks of processors (graphs). We say that G can be emulated on H if there exists a function $f : V_G \rightarrow V_H$ such that for every edge $(g, g') \in E_G : f(g) = f(g')$ or $(f(g), f(g')) \in E_H$. The function f is called an emulation function or, in short, an emulation of G on H.

Let f be an emulation of G on H. Any processor $h \in V_H$ must actively emulate the processors $\in f^{-1}(h)$ in G. When $g \in f^{-1}(h)$ communicates information to a neighbouring processor g', then h must communicate the corresponding information "internally", when

*This research was supported by the Foundation for Computer Science (SION) of the Netherlands Organization for the Scientific Research (NWO).

it emulates g' itself, or to a neighbouring processor $h' = f(g')$ in H otherwise. If all processors act synchronously in G, then the emulation will be slowed by a factor proportional to $\max|f^{-1}(h)|$.

Definition.
Let G, H and f be as above. The emulation f is said to be (computationally) uniform if for all $h, h' \in V_H : |f^{-1}(h)| = |f^{-1}(h')|$.

Every uniform emulation f has associated with it a fixed constant c, called: the computation factor, such that for all $h \in V_H$ $|f^{-1}(h)| = c$. It means that every processor of H emulates the same number of processors of G.

Graphs representing the interconnection pattern of a processor network will be connected (in the undirected case) or strongly connected (in the directed case). Therefore we will be mainly interested in uniform emulations of (strongly) connected graphs on (strongly) connected graphs.

In [3] variants of the following problem were considered:

> [UNIFORM EMULATION]
> Instance: Connected graphs $G = (V_G, E_G), H = (V_H, E_H)$.
> Question: Is there a uniform emulation of G on H?

This problem is NP-complete, even if the problem is restricted to instances with H a hypercube, a grid or a shuffle-exchange network, and the computation factor is fixed, or with G and H of bounded degree and the computation factor is fixed. Also, UNIFORM EMULATION is NP-complete for fixed host graphs H, for every connected graph H, that is not a complete graph [3]. Note that the related question whether G can be emulated on H (without the requirement of uniformity) is trivially solvable, as every constant function is an emulation.

In this paper we consider UNIFORM EMULATION for the case that H is a path or a ring (cycle). We will call these problems UNIFORM EMULATION ON A PATH and UNIFORM EMULATION ON A RING, respectively.

The UNIFORM EMULATION ON A PATH-problem is strongly related with the problem to determine whether a graph G has a bandwidth of at most K, for some $K \in N^+$. To be precise, the BANDWIDTH problem is the following:

> [BANDWIDTH]
> Instance: A graph $G = (V_G, E_G)$ and an integer $K \leq |V_G|$
> Question: Is there a bijection $f : V_G \to \{0, \ldots, |V_G| - 1\}$ such that for all $(v, w) \in E_G$ $|f(v) - f(w)| \leq K$, in other words: does there exist an arrangement of the nodes of G on a straight line, such that adjacent nodes in G have a distance of at most K on the line.

Leung, Vornberger and Witthoff [8] introduced a variant of BANDWIDTH, called CYCLIC BANDWIDTH. In this problem one is asked to arrange the nodes of G on a cycle (ring), such that adjacent nodes in G have a distance of at most K on the cycle. The CYCLIC BANDWIDTH problem is strongly related with the UNIFORM EMULATION ON A RING-problem.

This paper is organized as follows. In section 2 some preliminary definitions and results concerning the relation between (CYCLIC) BANDWIDTH and UNIFORM EMULATION ON A PATH (RING) are given. In section 3 we prove that UNIFORM EMULATION ON A PATH and ON A RING are NP-complete, even if G is required to be a binary tree. In section 4 similar results are derived for directed variants of the problems. In section 5 we give polynomial time algorithms for the UNIFORM EMULATION ON A PATH (RING) problems for the case that the computation factor is fixed. In section 6 we show that the problem becomes again NP-complete, if we keep the requirement that the computation factor is fixed, but drop the requirement that G is connected, for each computation factor $c \geq 4$ (UNIFORM EMULATION ON A PATH) or each computation factor $c \geq 2$ (UNIFORM EMULATION ON A RING).

2 Preliminaries.

We first give a number of basic definitions. The notion of cyclic bandwidth was introduced in [8].

Definition.
The path with n nodes is the (undirected) graph $P_n = (V_n, E_n^P)$ with $V_n = \{0, \ldots, n-1\}$ and $E_n^P = \{(b,c) | b, c \in V_n \text{ and } |b-c| = 1\}$. The ring (or cycle) with n nodes is the (undirected) graph $R_n = (V_n, E_n^R)$ with $E_n^R = \{(b,c)|b,c \in V_n \text{ and } |b-c| = 1 \text{ or } |b-c| = n-1\}$. The directed path with n nodes is the (directed) graph $\vec{P_n} = (V_n, \vec{E_n^P})$ with $\vec{E_n^P} = \{(b,c)|b,c \in V_n \text{ and } c = b+1\}$. The directed ring (or directed cycle) with n nodes is the (directed) graph $\vec{R_n} = (V_n, \vec{E_n^R})$ with $\vec{E_n^R} = \{(b,c)|b,c \in V_n \text{ and } c = (b+1) \bmod n\}$.

With $d_G(b,c)$ for $b, c \in V_n$ we denote the distance of b to c in the graph G, i.e., the length of the shortest (directed) path from b to c.

Definition.
Let $G = (V, E)$ be an undirected graph. A linear ordering of V is a bijection $f : V \to V_n$. f is said to have bandwidth K if $K = \max\{|f(u) - f(v)| | (u,v) \in E\} = \max\{d_{P_n}(f(u), f(v)) | (u,v) \in E\}$, i.e., the maximum distance in P_n between the image of adjacent nodes in G is K. The <u>bandwidth</u> of G is the minimum bandwidth over all linear orderings of V, i.e., Bandwidth $(G) = \min\{\text{bandwidth}(f) | f \text{ is a linear ordering of } V\}$.

For cyclic bandwidth, the nodes are mapped upon a ring, instead of on a path.

Definition.
Let $G = (V, E)$ be an undirected graph. A linear ordering f is said to have cyclic bandwidth K if $K = \max\{d_{R_n}(f(u), f(v)) | (u,v) \in E\}$. We write: Cyclic Bandwidth $(G) = \min\{\text{Cyclic Bandwidth } (f) | f \text{ is a linear ordering of } V\}$.

For directed (cyclic) bandwidth, if $(u, v) \in E$ then we want to have a path of length at most K from $f(u)$ to $f(v)$ in $\vec{P_n}$ ($\vec{R_n}$).

Definition.
Let $G = (V, E)$ be a directed graph. A linear ordering f is said to have directed bandwidth K, if for all $(u, v) \in E$ $f(v) > f(u)$, and $K = \max\{d_{\vec{P_n}}(f(u), f(v)) | (u, v) \in E\}$. f is said to have directed cyclic bandwidth K if $K = \max\{d_{\vec{R_n}}(f(u), f(v)) | (u, v) \in E\}$. If there exists a linear ordering with directed bandwidth K and no linear ordering with a smaller directed bandwidth then we write Dir Bandwidth $(G) = K$. Similarly we write Dir Cyclic Bandwidth$(G) = K$ if K is the least possible cyclic bandwidth for all linear orderings of V.

Note that there exists a K with Dir Bandwidth$(G) = K$, if and only if G is cycle free. The following lemma's relate the (cyclic) bandwidth of a graph and its capacity to be uniformly emulated on a path or a ring. Note that the computation factor of each of uniform emulations mentioned in the lemma's equals K.

Lemma 2.1
Let $G = (V, E)$ be an undirected graph and $K |\ |V|$. Let $n = |V|/K$. Then:

Bandwidth$(G) \leq K$
$\Rightarrow G$ can be uniformly emulated on P_n
$\Rightarrow$ Bandwidth$(G) \leq 2K - 1$.

Proof.

(i) Let f be a linear ordering of G with bandwidth $\leq K, K |\ |V|$ and $n = |V|/K$. Then $g : V \rightarrow V_n$, defined by $g(v) = \lfloor f(v)/K \rfloor$ is a uniform emulation of G on P_n : if $(u, v) \in E$ then $|f(u) - f(v)| \leq K$, hence $|g(u) - g(v)| \leq 1$, and $g(u)$ and $g(v)$ are adjacent in P_n.

(ii) Let g be a uniform emulation of G on P_n. For every $i \in V_N$ we can number the nodes in $g^{-1}(i)$ from 0 to $K - 1$. Let $nb(v)$ be the number given to v, i.e. $\forall v, w \in V v \neq w \wedge g(v) = g(w) \Rightarrow nb(v) \neq nb(w)$ and $\forall v \in V nb(v) \in \{0, \ldots, K - 1\}$. Now $f : V \rightarrow V_{|V|}$, defined by $f(v) = K.g(v) + nb(v)$ is a linear ordering with bandwidth at most $2K - 1$.

□

Lemma 2.2
Let $G = (V, E)$ be an undirected graph and $K |\ |V|$. Let $n = |V|/K$. Then

Cyclic Bandwidth $(G) \leq K$
$\Rightarrow G$ can be uniformly emulated on R_n
$\Rightarrow$ Cyclic Bandwidth $(G) \leq 2K - 1$.

Proof.
Similar to 2.1. □

Lemma 2.3
Let $G = (V, E)$ be a directed, acyclic graph. Let $K \mid |V|$ and $n = |V|/K$. Then

Dir Bandwidth $(G) \leq K$
$\Rightarrow G$ can be uniformly emulated on $\vec{P}_n$
$\Rightarrow$ Dir Bandwidth $(G) \leq 2K - 1$.

Proof.

(i) Similar to 2.1.

(ii) Let g be a uniform emulation of G on $\vec{P}_n$. For every $i \in V_n$ we can number the nodes in $g^{-1}(i)$ from 0 to $K-1$, such that if $(u,v) \in E, u,v \in g^{-1}(i)$, then $nb(u) < nb(v)$. Now $f : V \to V_{|V|}$, defined by $f(v) = K.g(v) + nb(v)$ is a linear ordering with directed bandwidth at most $2K - 1$.

□

Lemma 2.4
Let $G = (V, E)$ be a directed graph, let $K \mid |V|$, and $n = |V|/K$. Then

(i) Dir Cyclic Bandwidth $(G) \leq K$
$\Rightarrow G$ can be uniformly emulated on R_n and G does not contain a cycle with length less than n.

(ii) G can be uniformly emulated on R_n and G does not contain a cycle with length $\leq K$
$\Rightarrow$ Dir Cyclic Bandwidth $(G) \leq 2K - 1$.

Proof.

(i) Let f be a linear ordering of V with directed cyclic bandwidth at most K. A uniform emulation of G on $\vec{R}_n$ can be constructed similar to 2.1. Suppose G contains a cycle with length less than n, with nodes $v_0, \ldots, v_{l-1} (l < n)$. Then $\sum_{i=0}^{l-1} d_{\vec{R}_{|V|}}(f(v_i), f(v_{(i+1) \bmod l})) = n$, so there exists a $i, 0 \leq i \leq l-1$ such that $d_{\vec{R}_{|V|}}(f(v_i), f(v_{(i+1) \bmod l})) \geq |V|/l > |V|/n = K$: the directed cyclic bandwidth of f is greater than K. Contradiction.

(ii) Let g be a uniform emulation of G on $\vec{R}_n$, and suppose G does not contain cycles with length K or less. Then for every $i, 0 \leq i \leq n-1$, the subgraph of G, induced by node-set $g^{-1}(i)$ is cycle free. So there exists a bijection $nb_i : g^{-1}(i) \to \{0, \ldots, K-1\}$, such that $u, v \in g^{-1}(i) \wedge (u,v) \in E \Rightarrow nb_i(u) < nb_i(v)$. Now again $f : V \to V_{|V|}$, defined by $f(v) = Kg(v) + nb_{g(v)}(v)$ is a linear ordering with bandwidth $\leq 2K - 1$.

□

3 NP-completeness for finding emulations of trees.

The problem to determine whether a given connected graph $G = (V, E)$ has bandwidth $\leq K$ is NP-complete, even if G is required to be a binary tree [5]. On the other hand, when the bandwidth factor K is fixed, then there exists an algorithm that is polynomial in $|V|$, (but exponential in K), to determine whether a given graph G has bandwidth $\leq K$ [9]. In contrast, CYCLIC BANDWIDTH is NP-complete even for fixed $K \geq 2$, if we allow the graphs to be not connected. If G is required to be connected then CYCLIC BANDWIDTH can be solved in polynomial time for any fixed K [8]. It is not difficult to prove that the CYCLIC BANDWIDTH problem for connected graphs, but with variable K, is NP-complete, by transformation from BANDWIDTH.

The resemblance of these results to the results for UNIFORM EMULATION ON A PATH and UNIFORM EMULATION ON A RING, respectively, is not very surprising, given lemma's 2.1 and 2.2., but the details are tedious.

Theorem 3.1
The following problem is NP-complete:

> [UNIFORM EMULATION ON A PATH for binary trees]
> Instance: An undirected, connected graph G, that is a binary tree, and a positive integer $n \in \mathbb{N}^+$.
> Question: Is there a uniform emulation of G on P_n?

Proof.
Clearly the problem is in NP. To prove NP-hardness we transform the BANDWIDTH problem for binary trees to this problem. This version of BANDWIDTH is NP-complete [5].

Let $G = (V_G, E_G)$ be a binary tree; let $0 \leq K \leq |V_G|$, and $n = |V_G|$.

We will construct an undirected graph $G_1 = (V_1, E_1)$, that is again a binary tree, and a positive integer $m \in \mathbb{N}^+$, such that G_1 can be uniformly emulated on P_m, if and only if the bandwidth of G is $\leq K$.

Let $k \in \mathbb{N}^+$ be the smallest integer, such that $k+1$ is a power of 2, and $2^k \geq 42kK$. Let $c = 2^{k+1}/(k+1)$. Note that k is polynomially bounded in $\log n$, and c is polynomially bounded in n. c will be the compuation factor of a possible emulation of G_1 on P_m. Let $m = 4(k+1) + 6k(n-1) + \frac{3}{4}(k+1)n$.

We now describe the construction of G_1, together with a sketch of the main ingredients of the proof. The complete proof is given afterwards.

(a) Take two perfect binary trees of depth k (i.e. with 2^k leaves and $2^k - 1$ internal nodes each), and connect the roots (see figure 3.1.). We call this structure a "treeblock". G_1 contains two such treeblocks. We will show that these treeblocks must be mapped to the "ends" of P_m.

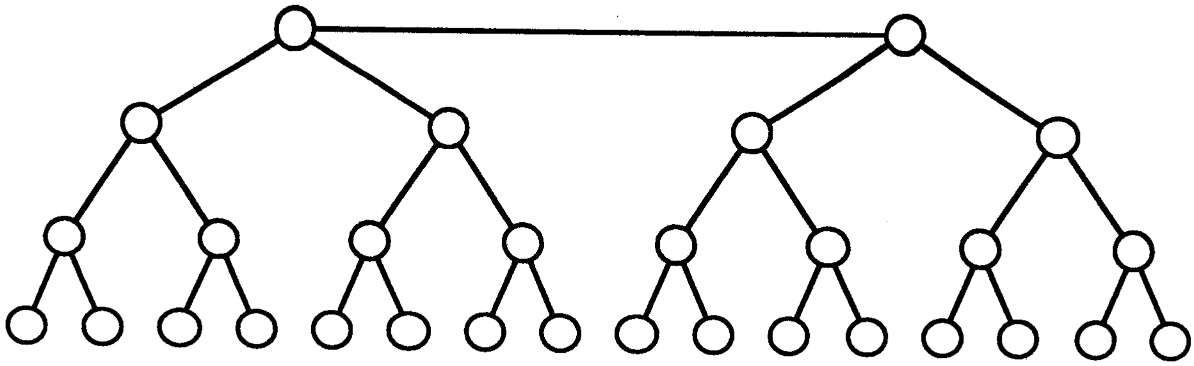

Figure 3.1

(b) Take a path with length $6k(n-1)+\frac{3}{4}(k+1)n$ between a leaf of one of the treeblocks and a leaf of the other treeblock. It will follow that the *i'th* node on this path must be mapped to node $2(k+1)+i-1$.

(c) Take a perfect binary tree of depth k, and remove $(k+4)k$ leaves. A copy of this tree is connected to the *j'th* node of the path of (b), for every $j \in \{\frac{3}{4}(k+1)i + 6k(i-1)+\alpha | 1 \le i \le n-1, \alpha \in \{k, k+1, 3k, 3k+1, 5k, 5k+1\}\}$. In this way each "interval" $3(k+1)i+6k(i-1)+1 \cdots 3(k+1)i+6k(i-1)+6k$ (for each $i, 1 \le i \le n-1$), has "many" nodes that must be mapped upon it from these trees, whereas no or only few nodes of these trees can be mapped to nodes in intervals $3(k+1)i+6ki+1 \cdots 3(k+1)i+6ki+3(k+1)$. In this way we have n "slots" of $\frac{3}{4}(k+1)$ consequtive nodes each, and $n-1$ "mountains" of $6k$ nodes each.

(d) Each vertex $x \in V_G$ is represented by a $(k-1)$-deep perfect binary tree T_x. It will follow that each of these trees must be mapped to a unique "slot".

(e) For each edge $(x,y) \in E_G$, connect a leaf of T_x and a leaf of T_y with a path of length $\frac{3}{4}(k+1)(K+1)+6kK$. This makes that T_x and T_y must be mapped "at most K slots away".

(f) Connect an arbitrary leaf of an arbitrary T_x to the first node of the path of (b), with a path of length $\frac{3}{4}(k+1)n+6k(n-1)$. This path serves to make G_1 connected.

(g) Add to the last node of the path of (b) a path with just enough nodes to total the total number of nodes in the constructed graph to cm.

Let G_1 be the graph, obtained in this manner. G_1 is a tree, and without difficulty, G_1 can be chosen such that no node has degree > 3.

Claim 3.1.1
There is a uniform emulation of G_1 on P_m, if and only if bandwidth $(G) \le K$.

Proof.
We will say that a node p in P_m has α *free places left*, if we have designated $c-\alpha$ nodes of G' so far to be mapped on p.

First let a linear ordering $f : V_G \to V_n$ with bandwidth $\leq K$ be given. We will show that there is a uniform emulation g of G_1 on P_m.

a) The treeblocks are mapped to the first $2k+2$ and the last $2k+2$ nodes of P_m. We will only give the mapping of the first treeblock to $\{0, 1, \ldots, 2k+1\}$, the other case is similar.

One of the two subtrees of the treeblock is conected to the path of (b), the other is not. We map c nodes of the latter subtree to 0, and $\frac{1}{2}c$ leaves of this tree to each of the nodes $1, 2, \ldots, k-1$. In this way we have mapped every leaf of this tree. We let nodes with similar ancestors be mapped as much as possible on the same node. Then we can map the parents of the nodes, that are mapped to i, to $i+1$, for each $i \in \{0, \ldots, k-1\}$. The rest of this tree we map to k. When we would map the other half of the tree in the same manner to $\{k+1, \ldots, 2k+1\}$, then every node in $\{0, \ldots, 2k+1\}$ would have mapped c nodes upon it, except k and $k+1$, which would have mapped $c-1$ nodes upon it. We can now alter this mapping slightly (we move some nodes one place to the left), such that every node in $\{0, \ldots, 2k\}$ has c nodes mapped upon it, and $2k+1$ has $c-2$ nodes mapped upon it. One has to take care that the leaf, connected to the path of (b) is mapped to $2k+1$.

b) We now can map the nodes of the path (b) to the nodes $2k+2, \ldots, m-2k-3$ (i.e. the $j'th$ node of the path is mapped to $2k+1+j$).

c) We can map the trees of (c) in a manner, similar to the mapping of the trees in (a), such that

- a tree connected to the $j'th$ node of the path (b), where $j \in \{\frac{3}{4}(k+1)i + 6k(i-1) + \alpha | 1 \leq i \leq n-1; \alpha \in \{k, 3k, 5k\}\}$ is mapped to the nodes $\{2(k+1)+j-k, \ldots, 2(k+1)+j-1\}$.
- a tree connected to the $j'th$ node of the path (b), where $j \in \{\frac{3}{4}(k+1)i + 6k(i-1) + \alpha | 1 \leq i \leq n-1; \alpha \in \{k+1, 3k+1, 5k+1\}\}$ is mapped to the nodes $\{2(k+1)+j-1, \ldots, 2(k+1)+j+k-2\}$.
- every node in H has at most $c-(K+4)$ such nodes mapped upon it.

d) Now notice that we have n "slots" $S_i = \{\frac{3}{4}(k+1)i + 6k(i-1) + (\alpha-1) | 1 \leq \alpha \leq \frac{3}{4}(k+1)\}, (0 \leq i \leq n-1)$. Each node in these slots has $c-1$ free places. We now map each T_x to the slot $S_{f(x)}$. This is done in a manner, similar to (a), but leaving in each node in the slot at least $k+6$ free places. Note that $|T_x| = 2^k - 1 = \frac{1}{2}(k+1)c - 1$, so we have in a slot in total $\frac{3}{4}(k+1)c - (\frac{1}{2}(k+1)c - 1) - \frac{3}{4}(k+1) = \frac{1}{4}(k+1)(c-3) + 1$ free places left.

e) Note that if $(x, y) \in E_G$, then the trees T_x and T_y are mapped at most K slots apart, so the leaves of these trees have distance at most $\frac{3}{4}(k+1)(K+1) + 6kK$. It follows that we can lay out the paths between T_x and T_y of (e). In some cases the path must be folded. This can be done in the space left in the slot $S_{\max(f(x), f(y))}$, such that each node in a slot has at least 3 free places left. Note that each slot contains $\frac{3}{4}(k+1)$ nodes of the path of (b), $2^k - 1$ nodes of a tree T_x, at most $\frac{3}{4}(k+1)K$ nodes of paths that connect trees T_y, T_z that are both mapped to other slots and at most $3\left(\frac{3}{4}(k+1)(K-1) + 6k(K-1)\right)$ nodes of the folded paths to the tree T_x that is mapped to the slot. As $\frac{3}{4}(k+1) + 2^k - 1 + \frac{3}{4}(k+1)K + 3\left(\frac{3}{4}(k+1)(K+1) + 6k(K-1)\right) \leq 2^k + 21(k+1)K - 3\left(\frac{3}{4}(k+1)\right) \leq \frac{3}{4}(k+1)(c-3)$,

it follows that we can map the paths, such that each node in the slot has at least 3 free places left.
f) By folding it once, we can lay out the path of (f), such that both ends connect correctly, and every node in $\{2K+2, \ldots, m-2K-2\}$ has at most 2 nodes of this path mapped upon it.
g) With the path of (g) we can fill every remaining free place. The first 2 nodes are mapped to the 2 free places of node $2K+2$. Suppose $2K+3$ has α free places. Then map the next α nodes to $2K+3$. In this way every free place can be filled, and a uniform emulation of G_1 on P_m results.

Now suppose g is a uniform emulation of G' on P_m. We will construct a linear ordering $f : V_G \rightarrow V_n$ with bandwidth $\leq K$.

First look at the images of the "treeblocks" of (a). Suppose the two roots of the two subtrees of a treeblock are mapped to the same node. Then a contradiction can be derived: every node in the treeblock has distance $\leq k$ to one of the two roots and there are $2 \cdot 2^{k+1} - 2$ such nodes. This means that $2 \cdot 2^{k+1} - 2$ nodes are mapped to the image of the roots and the $2k$ nodes in P_m with distance $\leq k$ to it. This contradicts uniformity. Hence the roots are mapped to neighbouring nodes, and the $2 \cdot 2^{k+1} - 4 = 2(k+1)c - 4$ other nodes are mapped to the $2(k+1)$ nodes with distance $\leq k$ to one of the images of the roots. This means that in these $2(k+1)$ nodes only 2 free places in total are left. It follows that the treeblocks must be mapped to the ends of P_m, i.e. to nodes $\{0, 1, \ldots, 2k+1\}$ and $\{m-(2k+2), \ldots, m-1\}$. As the path of (b) is connected to both treeblocks, it follows that either for all i, the $i'th$ node of the path is mapped to $2k+1+i$, or for all i, the $i'th$ node of the path is mapped to $m-1-(2k+1+i)$. Without loss of generality we assume the former.

Next look at the images of the roots of T_x, for all $x \in V_G$. If there exists such a root r_x, with $g(r_x) \in \{2(k+1) + \frac{3}{4}(k+1)i + 6k(i-1) + \alpha | 0 \leq i \leq n-1, k-1 \leq \alpha \leq 5k\}$, (that is: $g(r_x)$ is more that $(k-1)$ nodes of a "slot" away), then a contradiction arises: $g^{-1}(\{g(r_x) - k + 1, \ldots, g(r_x) + k + 1\})$ contains not only all $2^k - 1$ nodes of T_x, but also at least $(2k-1)(c-K-4) - 2k(K+4) - 2c$ nodes of the trees of (c). Now $2^k - 1 + (2k-1)(c-K-4) - 2k(K+4) - 2c > (2k-1)c$, which contradicts uniformity.

In a similar manner one can prove that for all $i, 0 \leq i \leq n-1$ in an interval $2(k+1) + \frac{3}{4}(k+1)i + 6ki - k + 1, \ldots, 2(k+1) + \frac{3}{4}(k+1)(i+1) + 6ki + k - 1$ (that is : a "slot" plus k extra nodes at each side of the slot), there is at most one $x \in V_G$, with nodes of T_x mapped to nodes in this interval.

So there is a one-to-one correspondence between $x \in V_G$ and slots S_i, such that T_x is mapped to nodes in, or near the slot, i.e. we have for each $x \in V_G$ a unique $i \in \{0, 1, \ldots, n-1\}$, such that $g(r_x) \in \{2(k+1) + \frac{3}{4}(k+1)i + 6ki - k + 1, \ldots, 2(k+1) + \frac{3}{4}(k+1)(i+1) + 6ki + k - 1\} = T_i$. Now define $f(x) = i$, iff $g(r_x) \in T_i$. Clearly, f is a linear ordering of G.

Consider $(x, x') \in E_G$. There is a path with length $\frac{3}{4}(k+1)(K+1) + 6kK$ between a leaf of T_x and a leaf of $T_{x'}$. Hence, $d_{G_1}(r_x, r_{x'}) \leq \frac{3}{4}(k+1)(K+1) + 6kK + 2(k-1)$. It follows that $|g(r_x) - g(r_{x'})| \leq \frac{3}{4}(k+1)(K+1) + 6kK + 2(k-1)$, and hence $|f(x) - f(x')| \leq K$. So f is a linear ordering with bandwidth $\leq K$. □

Finally, note that G_1 can be constructed in time, polynomial in $|V_G|$. Hence theorem 3.1

follows. □

One can easily adapt this proof for the case of uniform emulation on the ring.

Theorem 3.2
The following problem is NP-complete:

> [UNIFORM EMULATION ON A RING for binary trees]
> Instance: An undirected connected graph G, that is a binary tree.
> Question: Is there a uniorm emulation of G on R_n?

4 NP-completeness of finding uniform emulations for directed graphs.

The technique used in section 3 can be modified in order to prove NP-completeness results for the directed versions of the problems. The problem of finding uniform emulations of graphs on directed paths may be less interesting from a practical point of view, as the directed path is not strongly connected. For completeness reasons, results on uniform emulations on directed path graphs are included.

Theorem 4.1
The following problem is NP-complete:

> [UNIFORM EMULATION ON A DIRECTED RING]
> Instance: A directed, strongly connected graph $G = (V, E)$, and a positive integer $n \in \mathbb{N}^+$.
> Question: Is there a uniform emulation of G on $\vec{R_n}$?

Proof.
The proof is similar to, but easier then the proof of theorem 3.1. Clearly the problem is in NP. To prove NP-completeness we transform DIRECTED BANDWIDTH to this problem.

Let a directed graph $G = (V_G, E_G)$, and a positive integer $K \leq |V_G|$ be given. W.l.o.g. suppose $|V_G| \geq 3$. We will construct a strongly connected graph G', that can be uniformly emulated on $R_{|V_G|+1}$, if and only if bandwidth$(G) \leq K$.

Let $n = |V_G|$. Let $c = K(K-1) + 2n + 4$. G' consists of the following parts:

(a) a clique B with c nodes

(b) for each node $x \in V_G$ a clique C_x with $\frac{1}{2}c$ nodes

(c) a path of length n from a node in B to a node in C_x, for every $x \in V_G$

(d) a path of length n from a node in C_x to a node in B, for every $x \in V_G$

(e) for each edge $(x, y) \in E_G$, a path of length K from a node in C_x to a node in C_y

(f) Let the total number of nodes used so far in (a) - (e) be m. Add a path from a node in B to a node in B with $c(n+1) - m$ nodes (i.e. with length $c(n+1) - m + 1$).

The paths in (c) and (d) make that G' is strongly connected.

Claim 4.1.1
Dir Bandwidth $(G) \leq K$, if and only if there is a uniform emulation of G' on $\vec{R}_{n+1}$.

Proof.
Let f be a uniform emulation of G' on $\vec{R}_{n+1}$. Note that for all $v, w \in V_{G'}$: if $(v, w) \in E_{G'}$, and $(w, v) \in E_{G'}$, then $f(v) = f(w)$. it follows that all nodes in a clique are mapped to the same node. Hence, the image of B cannot have other nodes mapped to it. Because every node in V_{n+1} has at least one node from the path of (f) mapped to it, there cannot be two different $C_x, C_{x'}$ that are mapped to the same node.

Suppose $f(B) = \alpha$. Let $g : V_G \rightarrow V_n$ be given by $g(x) = (f(C_x) - \alpha) \bmod n$. Now g is a linear ordering of G, and if $(x, y) \in E_G$, then there is a path with length K from a node in C_x to a node in C_y, that cannot pass over $f(B) = \alpha$, so $g(y) - g(x) \leq K$ and $g(y) \geq g(x)$, hence dir bandwidth $(G) \leq K$. □

Note that G' is strongly connected and can be constructed in polynomial time. Hence theorem 4.1. follows. □

With a small refinement of the techniques we can get a slightly stronger result. We first need the following lemma.

Lemma 4.2
For each $n \in \mathbb{N}+, n \geq 2$, there exists a directed graph $G^n = (V^n, E^n)$, with the following properties:

- G^n is strongly connected
- $|V^n| = n$
- each node of G^n is involved in at most 3 edges
- at least 2 nodes of G^n are involved in exactly 2 edges
- For every graph G, that contains G^n as a subgraph, every integer $m \geq 5$, and every emulation f of G on $\vec{R}_m$, f maps every node of G^n on the same node of $\vec{R}_m$, i.e. $|f(V^n)| = 1$.

Proof.
We use induction to n. For $n = 2, 3, 4$ and 5, see figure 4.1. For $n \geq 6$, let $G_A = G^{\lfloor \frac{n}{2} \rfloor - 1}$, and $G_B = G^{\lceil \frac{n}{2} \rceil - 1}$ be given. Let $v_{A,1}, v_{A,2}$ be the nodes, involved with 2 edges in G_A, and $v_{B,1}, v_{B,2}$ these nodes in G_B. Then one easily verifies that the construction of figure 4.2. satisfies the properties. □

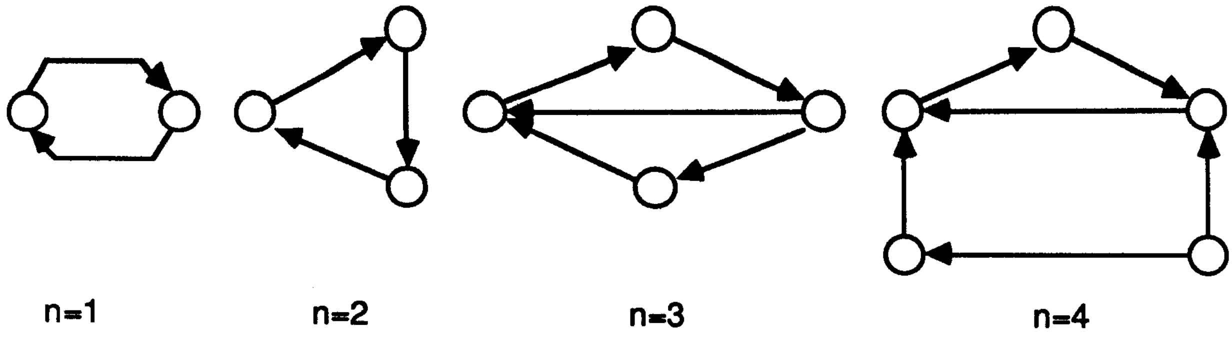

Figure 4.1

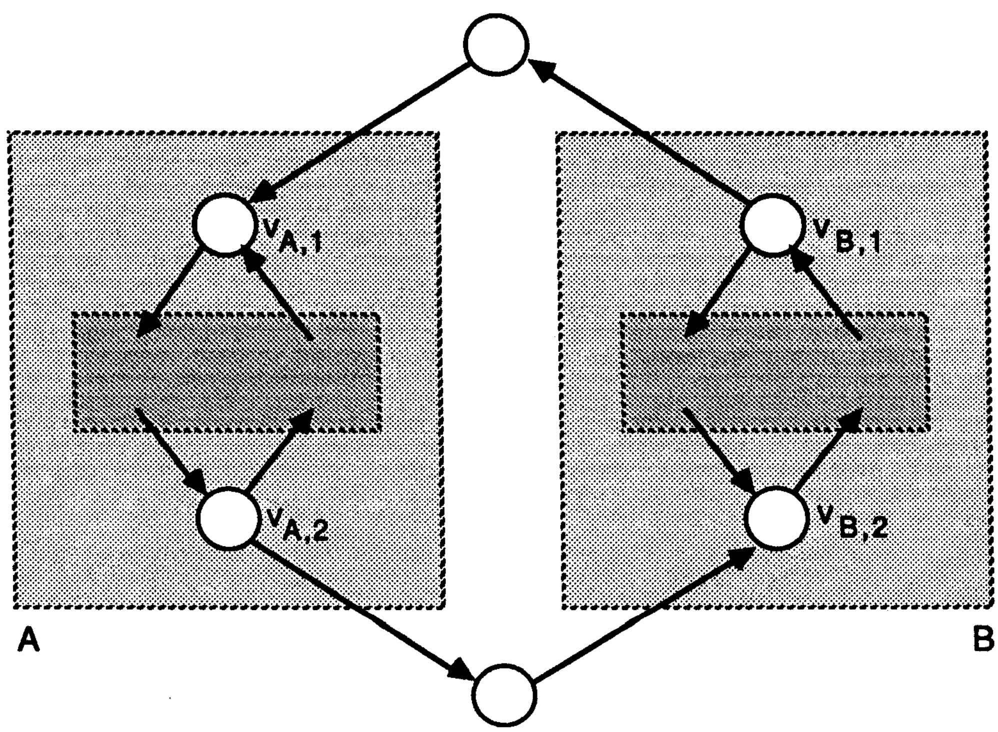

Figure 4.2

Theorem 4.3
The following problem is NP-complete:

> Instance: A directed, strongly connected graph $G = (V_G, E_G)$ with each node of G involved in at most 3 edges, and an $n \in \mathbb{N}^+$.
> Question: Is there a uniform emulation of G on $\vec{R}_n$?

Proof.
Clearly the problem is in NP. To prove NP-completeness we will again use a transformation from DIRECTED BANDWIDTH. Let $G = (V_G, E_G)$ be a directed graph. We give a construction of a graph G', such that G' can be uniformly emulated on $\vec{R}_{n+3}$, if and only if Dir Bandwidth $(G) \leq K$. Let $n = |V_G|$ and let $c = K(K-1) + 2n + 4$. We only stress the differences with the construction of theorem 4.1.

The clique B is replaced by
(a1) a cycle with c nodes, with an edge from this cycle to
(a2) a copy of G^c, with an edge to
(a3) another cycle with c nodes.
Every node of (a2) must be mapped to the same node x in $\vec{R}_{n+3}$, w.l.o.g. suppose $x = 1$. It follows that every node of (a1) must be mapped to 0, and every node of (a3) must be mapped to 2. The cliques C_x are replaced by cycles of $\frac{1}{2}c$ nodes: these cycles must also be mapped on the same node $\tilde{g}\ (x)$, because no node of the cycle can be mapped to 0,1 or 2. The paths of (c), (d), (e) and (f) can be added in such a manner, that all relevant connections are made, but no node becomes involved in more than 3 edges. The remainder of the proof is similar to 4.1. □

With a similar proof one can show:

Theorem 4.4
The following problem is NP-complete:

> Instance: A directed, connected graph $G = (V, E)$, with each node involved in at most 3 edges, and a positive integer $n \in \mathbb{N}^+$.
> Question: Is there a uniform emulation of G on $\vec{P}_n$?

5 Polynomial time algorithms for fixed computation factors.

For the NP-completeness proofs in the preceding sections, we needed that the computation factor $c = |V_G|/n$ could grow with the size of the problem. If we fix the computation factor, then the problems become solvable in polynomial time. In 1980 Saxe [9] has shown that for each constant K there exists an algorithm that uses $\mathcal{O}(f(K)n^{K+1})$ time to determine whether a given graph $G = (V, E), |V| = n$, has bandwidth K or less. Gurari and Sudbourough [7] improved the running time of the algorithm to $\mathcal{O}(f(K)n^K)$. We will further let K and c be constants, and do not mention factors like $f(K)$.

For each constant c, we will give an algorithm that determines in $\mathcal{O}(n^{2c-1})$ time whether a *connected* graph $G = (V, E)$ with $|V| = n$ can be uniformly emulated on $P_{\frac{n}{c}}$, and a similar algorithm, that uses $\mathcal{O}(n^{3c-1})$ time, and determines whether a connected G can be uniformly emulated on $R_{\frac{n}{c}}$. The algorithms are modifications of Saxe's bandwidth algorithm.

First note that if there is a node v in G with degree $\geq 3c$, then G cannot possibbly be emulated on $P_{\frac{n}{c}}$ or $R_{\frac{n}{c}}$ with computation factor c; there are $\geq 3c+1$ nodes which must be mapped upon $f(v)-1, f(v)$ or $f(v)+1$. So we can assume that every node of G has degree $\leq 3c-1$. We now introduce the notion of a partial uniform emulation.

Definition.
A partial uniform emulation of $G=(V,E)$ on $P_{\frac{n}{c}}$ is a function f from some subset $V' \subseteq V$ onto $\{0,\ldots,M-1\}$, for some M such that $1 \leq M \leq \frac{n}{c}$, such that:

(i) $\forall i \in \{0,\ldots,M-1\} : |f^{-1}(i)| = c$,

(ii) if $u,v \in V'$ and $(u,v) \in E$ then $f(u) = f(v)$ or $(f(u), f(v)) \in E_n^P$ (i.e. $|f(u)-f(v)| \leq 1$), and

(iii) if $u \in V'$ and $f(u) \leq M-2$ and $(u,v) \in E$ then $v \in V'$.

We say that f is feasible, if it can be extended to a (total) uniform emulation of G on $P_{\frac{n}{c}}$. For a set $S \subseteq V$, define $Adj(S) = \{w \in V | w \text{ is adjacent to a node } v \in S, \text{ and } w \notin S\}$. The active region of a partial uniform emulation f is the set $f^{-1}(M-1)$, together with the set $Adj(f^{-1}(M-1)) \backslash f^{-1}(M-2)$. I.e., the active region of a partial uniform emulation is the set of nodes that are mapped upon the last-so-far used node, and the set of nodes that have to be mapped upon the next node.

Theorem 5.1
Let f and g be two partial uniform emulations of G on $P_{\frac{n}{c}}$ with identical active regions. Then

(i) f and g have identical domains, and

(ii) f is feasible, if g is feasible.

Proof.
(i) Let the active region of f and g be (S, A). G is connected, so a node $v \in V$ is in the domain of f, or of g, if and only if it is path-connected to a node in S by a path, that does not use a node in A. So the domains of f and g are the same.
(ii) Any mapping of the remaining vertices to $\{M,\ldots,\frac{n}{c}\}$, which extends either f or g to a uniform emulation must also extend the other to a uniform emulation. □

Two partial uniform emulations are said to be equivalent, if they have identical active regions. Note that the number of different active regions is bounded: each node has degree $\leq 3c-1$, so $|Adj(f^{-1}(M-1))| \leq (3c-1)c$, so there are at most $2^{(3c-1)c} \cdot \binom{n}{c} = \mathcal{O}(n^c)$ different active regions, or equivalence classes. The algorithm is essentially a breadth-first search over the space of all equivalence classes of partial uniform emulations. The algorithm uses the following datastructures:

1. A queue Q, whose elements are active regions
2. An array A, which contains one element with two fields for each active region. For each active region $r, A[r]$.examined is a boolean which is set to true, when r is examined for the first time, and $A[r]$.unplaced is a list of the nodes, not in the domain of each partial uniform emulation with active region r.

We now give the algorithm:
<u>Algorithm P</u>

1. Set all $A[r]$.examined FALSE
2. Let Q be the empty queue.
3. (Set all the active regions of partial uniform emulations with $M = 1$ in the queue). For each set $S \subseteq V$ with $|S| = c$, perform the following steps:
   (a) Let s be the active region $s = (S, Adj(S))$.
   (b) Set $A[s]$.examined to TRUE.
   (c) Set $A[s]$.unplaced to $V \backslash S$.
   (d) Insert s at the end of Q.
4. Extract an active region r from the head of Q.
5. Suppose $r = (R, A)$.
   If $|A| > c$ then go to step 7. (r cannot be extended to a uniform emulation, so does not have to be considered.)
6. For each set $S \subseteq A[r]$.unplaced, with $A \cap S = \emptyset$, and $|A| + |S| = c$, perform the following steps.
   (a) Compute $B = Adj(A \cup S) \backslash R$
   Let s be the active region $s = (A \cup S, B)$.
   (b) If $A[s]$.examined $=$ FALSE then perform the following steps:
      i. Set $A[s]$.examined to TRUE
      ii. Let $A[s]$.unplaced be $A[r]$.unplaced $\backslash (S \cup A)$
      iii. If $A[s]$.unplaced is the empty set, then halt: there exists a uniform emulation of G on $R_{\frac{n}{c}}$.
      iv. Insert s at the end of Q.
7. If Q is empty, then halt: there does not exist a uniform emulation of G on $R_{\frac{n}{c}}$. Otherwise go to step 4.

In step 3 every active region of a partial uniform emulation with $M = 1$ (i.e. with the first c nodes mapped) is put on the queue. In step 6 the partial uniform emulation(s), whose active region is (are) considered, is (are) extended by mapping c nodes on the first node in $P_{\frac{n}{c}}$ that still has no nodes mapped upon. Note that for such extension $\overline{f}$ of a partial uniform emulation with active region $(f^{-1}(M-1), Adj(f^{-1}(M-1)) \backslash f^{-1}(M-2))$, it must hold that $\overline{f}(Adj(f^{-1}(M-1))\backslash f^{-1}(M-2)) = \{M\}$. The space required by this algorithm is $\mathcal{O}(n^{c+1})$. Now estimate the (worst-case) running time of the algorithm. Note that we first have to test whether there exists a node with degree at least $3c$. This can be done in time $\mathcal{O}(n)$. Step 1 costs $\mathcal{O}(n^c)$ time, step 2 $\mathcal{O}(1)$, step 3 costs $\mathcal{O}(n^{c+1})$ time. For each active region r, step 4-7 are performed at most once. So the cost for step 4,5 and 7 are at most $\mathcal{O}(n^c)$. To calculate the costs of step 6, note that A must contain at least one node (G is connected), so step 6a and the test of step 6b are performed at most $\mathcal{O}(n^{2c-1})$ times. For every active region, steps 6b(i) - 6b(iv) are performed at most once. The amount of computation needed for one execution of steps 6b(i) - 6b(iv) is $\mathcal{O}(n)$ so the total cost of steps 6b(i) - 6b(iv) becomes $\mathcal{O}(n^{c+1})$. So the total time the algorithm costs is $\mathcal{O}(n^{2c-1})$ (if $c \geq 2$).

Theorem 5.2
Let $c \in N^+$ be a constant. There exists an $\mathcal{O}(n^{2c-1})$ time and $\mathcal{O}(n^{c+1})$ space algorithm, which determines whether a connected graph $G = (V, E)$ with $|V| = n$ and $c|n$ can be uniformly emulated on $P_{\frac{n}{c}}$.

Proof.
If $c = 1$ then test in linear time whether G is isomorphic to $P_{\frac{n}{c}}$. If $c \geq 2$ then test whether there exists a node with degree $\geq 3c$. If such a node does not exist, then use Algorithm P. □

Let c again be a constant. We can modify algorithm P to an algorithm that tests whether a connected graph $G = (V, E)$, with $|V| = n$ and $c|n$ can be uniformly emulated on $R_{\frac{n}{c}}$. The class of partial uniform emulations is defined similarly as before, but condition (iii) is replaced by: if $u \in V'$ and $1 \leq f(u) \leq M-2$, and $(u, v) \in E$, then $v \in V'$. Again a partial uniform emulation is feasible if it can be extended to a (total) uniform emulation of G on $R_{\frac{n}{c}}$. We will not consider partial uniform emulations with $M < 2$. For $M \geq 2$, the active region of partial uniform emulation now is the 4-tupple $(f^{-1}(M-1), Adj(f^{-1}(M-1))\backslash f^{-1}(M-2), f^{-1}(0), Adj(f^{-1}(0))\backslash f^{-1}(1))$, i.e. we include now the set of nodes, mapped to 0, and the set of nodes that have to be mapped to $\frac{n}{c} - 1$. Again, if f and g are partial uniform emulations of G on $R_{\frac{n}{c}}$ with identical active region, then f and g have identical domains, and f is feasible, iff g is feasible. (Let (S, A, T, B) be the active region of f and g. A node v is in the domain of f (or of g), if it is path connected to a node in S, or a node in T, not using any nodes in A and B. So f and g have identical domains.)

First we can test whether G has a node of degree $\geq 3c$. So, again, we may suppose such a node does not exist. Choose an arbitrary node $v^* \in V$. For symmetry reasons we may suppose that $f(v^*) = 0$. So we only have to consider active regions (S, A, T, B) with $v^* \in T$. The number of such active regions is $\mathcal{O}(n^{2c-1})$.

The algorithm is further more or less similar to Algorithm P. We sketch the main differences. In step 3 we compute all active sets of partial uniform emulations $V' \rightarrow \{0, 1\}$: for each pair of sets S, S' with $|S| = |S'| = c$ and $v \in S'$, take the 4-tupple $(S, Adj(S)\backslash S', S', Adj(S')\backslash S)$. In step 6 we cannot immediately conclude that an active region $= (S, A, T, B)$ with $A[S]$.unplaced $= \emptyset$ corresponds with a uniform emulation of G on $R_{\frac{n}{c}}$, but we have to check that "the ends of the ring fit together", i.e. that $B \subseteq S$. The remaining details are easily provided, and left to the reader.

To estimate time and space, note that $v^* \in T$, and no longer necessarily $A \neq \emptyset$, for active regions (S, A, T, B). It follows that the algorithm uses $\mathcal{O}(n^{2c})$ space and $\mathcal{O}(n^{3c-1})$

time.

Theorem 5.3
Let $c \in \mathbb{N}^+$ be a constant. There exists an $\mathcal{O}(n^{3c-1})$ time and $\mathcal{O}(n^{2c})$ space algorithm that determines whether a connected graph $G = (V, E)$ with $|V| = n$ and $c|n$ can be uniformly emulated on $R_{\frac{n}{c}}$.

6 The case of disconnected graphs.

Interestingly, if the requirement that G is connected is dropped, then the problems to determine whether $G = (V, E)$ can be uniformly emulated on $P_{\frac{n}{c}}$ or $R_{\frac{n}{c}}, (|V| = n, c|n)$ become NP-complete, even for fixed computation factors c.

Theorem 6.1
For every $c \in \mathbb{N}^+, c \leq 4$, the following problem is NP-complete:

> Instance: An undirected graph $G = (V, E)$, with $n = |V|, c|n$.
> Question: Is there a uniform emulation G on $P_{\frac{n}{c}}$?

Proof.
Clearly the problem is in NP. To prove NP-completeness we transform 3-PARTITION to it. 3-PARTITION is the following problem:

> [3-PARTITION]
> Instance: A set A of $3m$ elements, a bound $B \in \mathbb{N}^+$, and a size $s(a) \in \mathbb{N}^+$ for each $a \in A$, such that $B/4 < s(a) < B/2$ and $\sum_{a \in A} s(a) = mB$.
> Question: Can A be partitioned into m disjoint sets $A_1, \ldots, A_m$ such that for each $i, 1 \leq i \leq m \sum_{a \in A_i} s(a) = B$?

3-PARTITION is NP-complete in the strong sense [6]. (Note that necessarily $|A_i| = 3$ for each $i, 1 \leq i \leq m$.)

Let an instance of 3-PARTITION be given, i.e. a set $A, |A| = 3m, B \in \mathbb{N}^+$, and a size $s(a) \in \mathbb{N}^+$ for each $a \in A$, with $\frac{B}{4} < s(a) < \frac{B}{2}$, and $\sum_{a \in A} s(a) = mB$.

Let $W \subseteq N$ be the set $\{0, 1, B+2, 2B+3, 3B+4, \ldots, mB+B+1, mB+B+2\}$. We will construct a graph G^o, that, when emulated on P_{mB+B+3} such that each node in P_{mB+B+3} has at most c nodes mapped upon it, forces that each node in W has c nodes mapped upon it, and each node in $\{0, \ldots, mB + B + 2\} \backslash W$ has $c - 1$ nodes mapped upon it. To this graph G^o we add for each $a \in A$ a copy of $P_{s(a)}$ (a path with length $s(a)$). The resulting graph G can be uniformly emulated on P_{mB+B+3}, if there is a solution to the 3-PARTITION problem; the paths must be mapped upon the remaining free places after G^o is mapped, and there are m parts of exactly B consequtive nodes with one free place each. The construction of G^o and G takes polynomial time.

Definition.
$G^o = (V^o, E^o)$ is defined as follows: $V^o = \{v_{i,j} | 0 \leq i \leq mB + B + 2,\ 1 \leq j \leq c$ and

$(j \neq c \text{ or } i \in W)\}, E^o = \{(v_{i_1,j_1}, v_{i_2,j_2}) | v_{i_1,j_1} \neq v_{i_2,j_2} \in V^o \text{ and } |i_1 - i_2| \leq 1 \text{ and } (i_1 \neq i_2 \text{ or } j_1 \neq j_2)\}$.

Lemma 6.1.1
For every emulation f of G^o on P_{mB+B+3}, such that $|f^{-1}(i)| \leq c$ for every $i \in V_{mB+B+3}$:

(i) Either $f(v_{i,j}) = i$ for all $v_{i,j} \in V_o$ or $f(v_{i,j}) = mB + B + 2 - i$ for all $v_{i,j} \in V_o$ and

(ii) If $i \in W$ then $|f^{-1}(i)| = c$.
If $i \in V_{mB+B+3} \backslash W$ then $|f^{-1}(i)| = c - 1$.

Proof.

(i) Suppose for some $i \in V_{mB+B+3} - \{0, mB + B + 2\}$ there are j_1, j_2 with $f(v_{i,j_1}) \neq f(v_{i,j_2})$. It follows that $|f(v_{i,j_1}) - f(v_{i,j_2})| = 1$. Now $f(\{v_{i_1,j} \in V^o | i_1 \in \{i-1, i, i+1\}\}) = \{f(v_{i,j_1}), f(v_{i,j_2})\}$, hence $\exists i_1 : |f^{-1}(i_1)| > c$. Contradiction. It follows that there is a bijection $\varphi : V_{mB+B+3} \to V_{mB+B+3}$, such that $f(v_{i,j}) = \varphi(i)$. By observing that $|\varphi(i) - \varphi(i+1)| \leq 1$, the result now easily follows.

(ii) This follows directly from (i).

□

Now let $G = (V, E)$ be defined by $V = V^o \cup \{(a)_i | a \in A \wedge 0 \leq i \leq s(a) - 1\}$ and $E = E^o \cup \{((a)_i, (a)_j) | (a)_i, (a)_j \in V \backslash V^o \wedge |i - j| = 1\}$. Note that $|V| = |V_o| + \sum_{a \in A} s(A) = c.(mB + B + 3)$.

Lemma 6.1.2
G can be uniformly emulated on P_{mB+B+3} if and only if A can be partitioned in m disjoint sets $A_1, \ldots, A_m$ such that for each $i, 1 \leq i \leq m \sum_{a \in A_i} s(a) = B$.

Proof.
Suppose an emulation f of G on P_{mB+B+3} exists. For each $i, 1 \leq i \leq m$, look at $R = f^{-1}(\{(i-1).(B+1) + 2, \ldots, i(B+1)\})$. Necessarily this set exists of $(c-1).B$ nodes $\in V^o$, and B nodes $\in V \backslash V^o$. There must be 3 elements $(a^1), (a^2), (a^3) \in A$ with $(a^i)_j \in R$ for $0 \leq j \leq s(a^i) - 1, 1 \leq i \leq 3$. Now let A_i consist of $(a^1), (a^2), (a^3)$. In this way a correct partition is obtained.

Now suppose a partition of A in $A_1 \ldots A_m$ with $\sum_{a \in A_i} s(a) = B$ $(1 \leq i \leq m)$ is given. Let $f(v_{i,j}) = i$. For each $i, 1 \leq i \leq m$ we can map the nodes in $\{(a)_j | a \in A_i, 1 \leq j \leq s(a)\}$ on the nodes $(i-1)(B+1) + 2, \ldots, i(B+1)$. (Each of these nodes has exactly one free place left after mapping of G^0). In this way a uniform emulation of G on P_{mB+B+3} can be obtained. □

The constraint $c \geq 4$ is essential for the proof. If $c \leq 3$ then nodes, that we want to have mapped on consequtive nodes can be mapped on the same node. (See figure 6.1.).

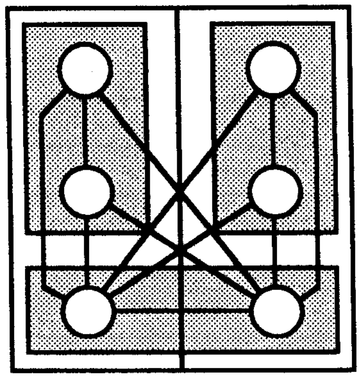

Figure 6.1

In fact, we conjecture that there exist polynomial time algorithms for the considered problem for $c = 2$ and $c = 3$. (For $c = 1$ it is trivial that a polynomial algorithm exists.) With a similar technique one can prove:

Theorem 6.2
For every $c \in \mathbb{N}^+, c \geq 2$, the following problem is NP-complete:

> Instance: An undirected graph $G = (V, E)$ with $|V| = n$ and $c|n$.
> Question: Is there a uniform emulation of G on $R_{\frac{n}{c}}$?

The same results hold for the directed versions of the problems.

Acknowledgement.

I would like to thank Jan van Leeuwen for his excellent guidance during this research.