On the Size of Permutation Networks and Consequences for Efficient Simulation of Hypercube Algorithms on Bounded-Degree Networks

Juraj Hromkovič, Przemyslawa Kanarek, Ralf Klasing, Krzysztof Lorys, Walter Unger, Hubert Wagener

ON THE SIZE OF PERMUTATION NETWORKS AND CONSEQUENCES FOR EFFICIENT SIMULATION OF HYPERCUBE ALGORITHMS ON BOUNDED-DEGREE NETWORKS [†‡]

JURAJ HROMKOVIČ [§], PRZEMYSŁAWA KANAREK [¶], RALF KLASING [‖], KRZYSZTOF LORYŚ [¶], WALTER UNGER [**], AND HUBERT WAGENER [††]

Abstract. The sizes of permutation networks and planar permutation networks for special sets of permutations are investigated. Several asymptotically optimal estimations for distinct subsets of the set of all permutations are established here.

The two main results are:

(i) an asymptotically optimal switching network of size $O(N \log\log N)$ for shifts of power 2.
(ii) an asymptotically optimal planar permutation network of size $\Theta(N^2 \cdot (\log\log N/\log N)^2)$ for shifts of power 2.

A consequence of our results is the construction of a 4-degree network which can simulate each communication step of any hypercube algorithm using edges from at most a constant number of different dimensions in one communication step in $O(\log\log N)$ communication steps. An essential improvement of gossiping in vertex-disjoint path mode in bounded-degree networks follows.

Key words. Network design and communication, Communication networks, Permutation networks, Switching networks, Parallel algorithms

AMS subject classifications. 68M07, 68M10, 90B18, 94C15

1. Introduction and Definitions. The study and the comparison of the computational power of distinct interconnection networks as candidates for the use as parallel architectures for existing parallel computers is an intensively investigated research branch of current computation theory. One of the fundamental approaches helping to search for the best (most effective) structures of interconnection networks is the study of the communication facilities of networks (i.e., of the complexity (effectivity) of solving fundamental communication tasks).

The basic communication tasks are routing (for an extensive survey see [31]), broadcasting, and gossiping (for an overview see [12, 19, 22, 23]). Our paper is devoted to the search for effective, realistic 1-to-1 routing (permutation) networks. The effectivity in our paper is mainly interpreted as the minimal size of a network realizing a given subset of permutations (one-to-one routing tasks). By realistic we mean that we consider only networks with bounded degree (usually with degree 4).

Our study of the sizes of permutation networks for distinct subclasses of permutations has been motivated not only by searching for efficient bounded-degree networks for special routing tasks, but mainly by looking for realistic networks in the class of bounded-degree networks with very high (or even with the highest possible) communication facilities supporting different fundamental computing and communicational tasks. For instance, the study of permutation networks for shifts 2^i is well-motivated by the simulation of hypercube algorithms in networks of bounded degree.

The study of permutation networks is of importance in several research contexts. Starting with the initial work by Beneš [3, 4] and Waksman [41] in the context of telephone networks, permutation networks have

[†]This paper is an essential extension of the extended abstract presented at the 12th Symposium on Theoretical Aspects of Computer Science (STACS '95).

[‡]This work was partially supported by SNF grant 200021-109252/1, by the ANR projects "ALADDIN" and "IDEA", the project "CEPAGE" of INRIA, and by the European projects COST Action 293 "Graphs and Algorithms in Communication Networks" (GRAAL) and COST Action 295 "Dynamic Communication Networks" (DYNAMO).

[§]Information Technology and Education, ETH Zürich, Switzerland. E-Mail: juraj.hromkovic@inf.ethz.ch

[¶]University of Wrocław, Institute of Computer Science, PL-51-151 Wrocław, Poland. E-Mail: {pka, lorys}@ii.uni.wroc.pl

[‖]LaBRI - Université Bordeaux 1 - CNRS, 351 cours de la Libération, 33405 Talence cedex, France. E-Mail: Ralf.Klasing@labri.fr

[**]Lehrstuhl für Informatik I, RWTH Aachen, D-52056 Aachen, Germany. E-Mail: quax@I1.Informatik.RWTH-Aachen.DE

[††]Department of Mathematics and Computer Science, University of Magdeburg, D-39016 Magdeburg, Germany

been investigated over the past 40 years in different contexts (e.g. telephone networks, interconnection networks for parallel computer architectures, packet routing, VLSI design, optical switching networks, multistage networks, rearrangeable networks), and constitute still nowadays an active field of research. (Some recent publications and surveys in this area are e.g. [1, 2, 5, 6, 7, 8, 9, 11, 14, 15, 16, 33, 34, 35], and the references contained therein.) Our study reconsiders the classical result that any permutation can be realized by a permutation network of size $\Theta(N \log N)$ [3, 4, 38, 41]. We are interested in the question whether for important subclasses of permutations it is possible to beat the lower bound $\Omega(N \log N)$ and derive improved realizations. Efficient planar permutation networks for these subclasses of permutations are also derived. The subclasses of permutations we consider are all permutations of N elements, all shifts, shifts of power two, and shifts by Fibonacci numbers. We will discuss further below the importance of these subclasses of permutations, and the implications that our results have in various applications.

This paper is organized as follows. In this section, we give the basic definitions and the overview of the achieved results. Section 2 contains the exact formulation of the results, and further references to related work. We also give the proofs of theorems which do not have long technical proofs. Section 3 presents the main technical proofs of the paper.

Section 2 is divided into three subsections. Subsection 2.1 is devoted to permutation networks for all permutations of N elements, all shifts, shifts of power two, and shifts by Fibonacci numbers. The main results of this subsection are the constructions of a switching networks of size $O(N \log\log N)$ for the shifts 2^i for $i = 1, 2, \ldots, \log_2 N$, and the construction of a switching networks of size $O\left(N \cdot 2^{2\sqrt{\log\log N}}\right)$ for Fibonacci shifts. (Switching networks are very regular permutation networks of degree 4.) The result for shifts of power of two is asymptotically optimal. Previously, switching networks of size $O(N \log N)$ were known for the shifts of power two and for Fibonacci shifts [3, 4, 41], and it was known that size $\Omega(N \log\log N)$ is necessary for both of these sets of permutations [38].

Subsection 2.2 is devoted to planar permutation networks. Here, one needs to distinguish between vertex-disjoint and edge-disjoint permutation networks. For instance, for the vertex-disjoint path mode one needs networks of size $\Theta(N^3)$ to realize all permutations [29], but we show that the size $O(N^2)$ is necessary and sufficient for the realization in the edge-disjoint path mode. Moreover, five random permutations are as hard as the set of all permutations for planar realization in the edge-disjoint path mode. Finally, the main result of this subsection shows that the optimal planar permutation network for shifts of power two has the size in $\Theta(N^2 \cdot (\log\log N/\log N)^2)$. To the best of our knowledge, our paper is the first to investigate planar permutation networks for shifts of power two.

The last Subsection 2.3 shows various applications of the results of the previous sections. The smallest known delay to simulate hypercubes of N nodes on bounded-degree networks is $\log_2 N$ [31]. Constant delay is possible if the communication steps of the hypercube algorithm consist only of communication via edges of one dimension and additionally any two consecutive communication steps correspond to communication in two consecutive dimensions [31, 39, 40]. Here, using the permutation networks for shifts 2^i, we obtain the delay $6d \cdot \log\log N$ for the simulation of hypercube algorithms communicating via edges of d arbitrary dimensions in one communication step. Note that this simulation may be even considered to be an improvement of the $O(1)$-delay simulation from [39] because our simulation is static while the simulation of [39] is dynamic. This means that during the whole simulation every processor of our network simulates the work of exactly one processor of the hypercube. In [39] after each simulation step the processors change their roles in simulating the hypercube processors. This means that each processor has to submit the whole content of its memory to some of its neighbours after each simulation step. A consequence beyond the simulation are quick gossip algorithms for bounded-degree networks in vertex-disjoint path mode. This is an essential improvement of the results established in [24, 25, 26, 28] for gossiping in disjoint-path modes.

1.1. Definitions and Simple Observations. Now, we give the fundamental definitions. Let Π_N denote the set of all permutations of N elements. For any set A, $|A|$ denotes the cardinality of A. F_i denotes the i-th Fibonacci number for any positive integer i, i.e. $F_0 = 0$, $F_1 = 1$, $F_{i+1} = F_i + F_{i-1}$ for $i \geq 1$. Let $\text{shift}_N(i) : \{0, \ldots, N-1\} \to \{0, \ldots, N-1\}$ be the permutation $x \mapsto (x + i) \bmod N$.

DEFINITION 1.1. *Let A be a subset of Π_N, and k be a positive integer. We say that a graph $G = (V, E)$ is a* **vertex-disjoint (edge-disjoint) k-permutation network for A** *if the following conditions hold:*

(i) *the degree of G is bounded by k,*
(ii) *V contains $2N$ special vertices $x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N$, ($x_1, x_2, \ldots, x_N$ are called* **inputs**, *$y_1, y_2, \ldots, y_N$ are called* **outputs** *of G)*
(iii) *for every permutation $\pi = (i_1, i_2, \ldots, i_N) \in A$ there exist N vertex-disjoint (edge-disjoint) paths $x_1, \ldots, y_{i_1}$; $x_2, \ldots, y_{i_2}$; $\ldots$; $x_N, \ldots, y_{i_N}$ in G.*

The size of G, denoted **size(G)**, *is $|V|$. The depth of G (according to A), denoted* **depth(G)**, *is* $\max\{$*distance between x_i and y_j* $\mid i, j \in \{1, \ldots, N\}, \pi(i) = j$ *for some* $\pi \in A\} - 1$, *where the distance between two nodes is the length of the shortest path that connects two nodes.*

A k-permutation network $G = (V, E)$ for A is **leveled** *if V can be partitioned into $r = depth(G) + 2$ nonempty subsets $V_0, V_1, \ldots, V_{r+1}$ (called* **levels** *of G or* **layers** *of G) such that*

(a) *$V_0 = \{x_1, \ldots, x_N\}$, $V_{r+1} = \{y_1, \ldots, y_N\}$;*
(b) *$V_i \cap V_j = \emptyset$ for $i \neq j$, $i, j \in \{0, \ldots, r+1\}$;*
(c) *for every edge $e \in E$, there exists $i \in \{0, \ldots, r\}$ such that e connects one vertex from V_i with one vertex from V_{i+1};*
(d) *for every permutation $\pi = (i_1, i_2, \ldots, i_N) \in A$ there exist N vertex-disjoint (edge-disjoint) paths $x_1, u_{1,1}, \ldots, u_{1,r}, y_{i_1}$; $\ldots$; $x_N, u_{N,1}, \ldots, u_{N,r}, y_{i_N}$ in G with $u_{i,j}$ in V_j for every $j = 1, \ldots, r$.*

Let G be a leveled edge-disjoint 4-permutation network for A with some levels $V_0, V_1, \ldots, V_{r+1}$ for an $r \geq 0$. We say that G is a **switching network for A** *if*

(1) *for every node $u \in V_0$, u has exactly one edge to a node in V_1;*
(2) *for every node $u \in V_{r+1}$, u has exactly one edge to a node in V_r;*
(3) *for every node $u \in V_i$, $i = 1, \ldots, r$, one of the following conditions holds:*
  (a) *u has exactly two edges to nodes in V_{i-1} and exactly two edges to nodes in V_{i+1} (then u is called a* **switch**);
  (b) *u has exactly one edge to nodes in V_{i-1} and exactly one edge to nodes in V_{i+1} (then u is called an* **idle node**).

Let G be a leveled vertex-disjoint 4-permutation network for A with some levels $V_0, V_1, \ldots, V_{r+1}$ for an $r \geq 0$. We say that G is a **regular permutation network for A** *if*

(1) *for every node $u \in V_0$, u has one or two edges to nodes in V_1;*
(2) *for every node $u \in V_{r+1}$, u has one or two edges to nodes in V_r;*
(3) *for every node $u \in V_i$, $i = 1, \ldots, r$, u has one or two edges to nodes in V_{i-1} and one or two edges to nodes in V_{i+1};*
(4) *$|V_i| = N$ for every $i \in \{0, 1, \ldots, r+1\}$;*
(5) *for every $0 \leq i \leq r+1$, the nodes in V_i can be labeled by $(i, 0), (i, 1), \ldots, (i, N-1)$ in such a way that for every $0 \leq i \leq r$, $0 \leq j \leq N-1$, there is an edge between the nodes (i, j) and $(i+1, j)$.*

In Figure 1 we give an example of a switching network of depth 2 and 9 inputs. Note that the model of switching networks appears very frequently in the literature (see e.g. [3, 4, 38, 41]). In Section 3, we will introduce the more commonly used **switch-wire** form of the switching-network model.

The next observation relates the depth of a switching networks to its size.

OBSERVATION 1.2. *Let A be a subset of Π_N. Let G be a switching network for A. Then $depth(G) \cdot \frac{N}{2} + 2N \leq size(G) \leq depth(G) \cdot N + 2N$.*

The next observation shows that switching networks can be efficiently simulated by regular permutation networks.

OBSERVATION 1.3. *Let A be a subset of Π_N. Let G be a switching network for A of depth d. Then there is a regular permutation network G' for A of depth $d-1$ and size $N(d+1)$.*

Proof. Let $G = (V, E)$ with levels $V_0, V_1, \ldots, V_{r+1}$ for some $r \geq 0$. Define $G' = (V', E')$ as follows: $V' = E$ and $(e_1, e_2) \in E'$ iff e_1 is an edge between levels V_i and V_{i+1}, e_2 is an edge between levels V_{i+1} and V_{i+2} for some $0 \leq i \leq r-1$, and e_1 and e_2 are incident to each other. This is the well-known line-digraph construction [18] if one interprets G as being directed from the inputs to the outputs. □

Figure 2 depicts how the switching network of Figure 1 is transformed to a regular permutation network as described in Observation 1.3.

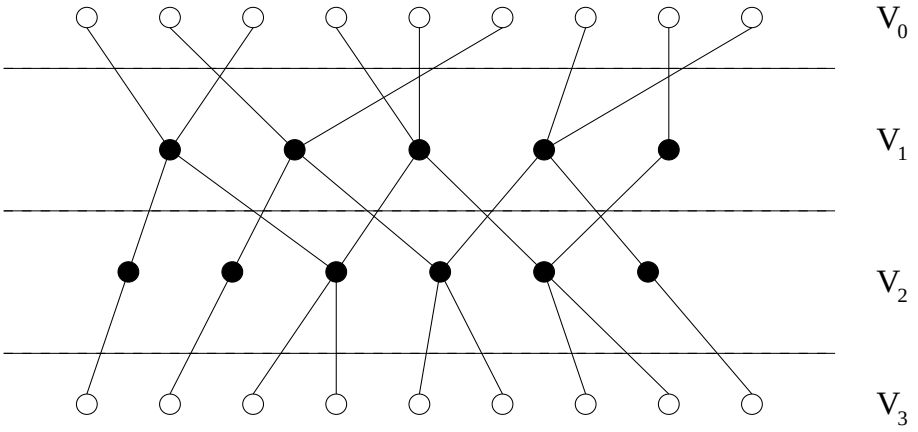

FIG. 1. *A switching network*

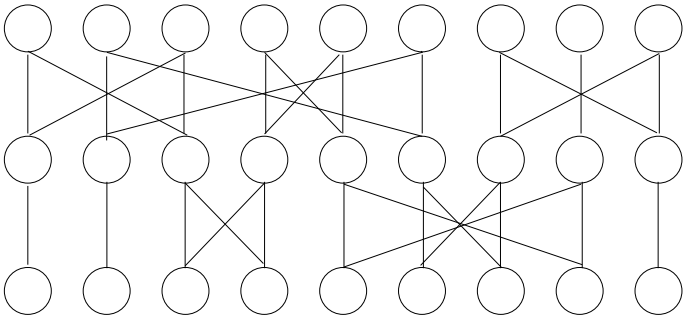

FIG. 2. *A regular permutation network*

Finally, note that condition (d) of the leveled permutation networks defined in Definition 1.1 assures that the edges of the network are used in one direction only (from x_i's to y_i's). We call attention to the fact that almost all permutation networks constructed here are leveled and of degree at most 4. On the other hand, the lower bound proofs are general, i.e, they work for unrestricted constant-degree networks.

DEFINITION 1.4. *Let A be a subset of Π_N, and k be a positive integer. We define*

$$\begin{aligned}
\textbf{Size-vd}_\mathbf{k}(\mathbf{A}) &= \min\{size(G) \mid G \text{ is a vertex-disjoint } k\text{-permutation network for } A\},\\
\textbf{Size-ed}_\mathbf{k}(\mathbf{A}) &= \min\{size(G) \mid G \text{ is an edge-disjoint } k\text{-permutation network for } A\},\\
\textbf{Plsize-vd}_\mathbf{k}(\mathbf{A}) &= \min\{size(G) \mid G \text{ is planar and } G \text{ is a}\\
&\qquad \text{vertex-disjoint } k\text{-permutation network for } A\},\\
\textbf{Plsize-ed}_\mathbf{k}(\mathbf{A}) &= \min\{size(G) \mid G \text{ is planar and } G \text{ is an}\\
&\qquad \text{edge-disjoint } k\text{-permutation network for } A\}.
\end{aligned}$$

OBSERVATION 1.5. *Let A be a subset of Π_N, and k be a positive integer. Then Size-vd$_k(A) = \Theta($Size-ed$_k(A))$.*

Proof. Each edge-disjoint k-permutation network G_1 can be simulated by some vertex-disjoint k-permutation network G_2 in such a way that size$(G_2) \leq k \cdot$ size(G_1), as follows.

In order to construct G_2 from G_1, each vertex v (except for inputs and outputs) of degree $d(v)$ in G_1 is replaced by a complete graph $K_{d(v)}(v)$ (of $d(v)$ nodes) in G_2. An edge (v, w) in G_1 is added as an edge between (some vertex in) $K_{d(v)}(v)$ and (some vertex in) $K_{d(w)}(w)$ in G_2 such that each vertex in G_2 receives only one such edge. A path (of the realization of A) in G_1 routed through a vertex v (i.e. using edges $(u, v), (v, w)$) is now routed through $K_{d(v)}(v)$ in G_2 accordingly (i.e. using the edge between $K_{d(u)}(u)$ and $K_{d(v)}(v)$, the edge inside $K_{d(v)}(v)$, and the edge between $K_{d(v)}(v)$ and $K_{d(w)}(w)$). □

OBSERVATION 1.6. *Let A be a subset of Π_N. Then Size-ed$_4(A) = O(N \cdot |A|)$.*

Proof. Let $A = \{A_0, A_1, \ldots, A_q - 1\} \subseteq \Pi_N$. Consider the network G consisting of the $q+1$ layers $V_0, V_1, \ldots, V_q$ (of N nodes each) in which the connections between layer V_i and layer V_{i+1} realize the permutation A_i $(0 \leq i \leq q-1)$. (More precisely, for every $0 \leq i \leq q$, let the nodes in V_i be labeled by $(i,0), (i,1), \ldots, (i, N-1)$. Then, for every $0 \leq i \leq q-1$, the vertex (i,j) in V_i is connected with vertex $(i+1, j)$ and with vertex $(i+1, A_i(j))$ in V_{i+1}.) Then G is an edge-disjoint 4-permutation network for A of size $O(N \cdot |A|)$. □

In what follows, we use the following notation.

$S_N \subseteq \Pi_N$	...	the set of all shifts, i.e. $\{\text{shift}_N(i) \mid 0 \leq i < N\}$,
$\text{Pow}\,2_N \subseteq S_N$	...	the set of all shifts of the power of 2, i.e. $\{\text{shift}_N(2^i) \mid 0 \leq i < \lceil \log N \rceil\}$,
$\text{Fib}_N \subseteq S_N$	...	the set of all shifts by Fibonacci numbers, i.e. $\{\text{shift}_N(F_i) \mid 0 \leq i \leq k\}$ where k is the largest integer with the property $F_k \leq N$.

DEFINITION 1.7. *For each positive integer N, let A_N, B_N be subsets of Π_N. We say that $B = \{B_N \mid N = 1, 2, \ldots\}$ is a* **kernel** *of $A = \{A_N \mid N = 1, 2, \ldots\}$ if $B_N \subseteq A_N$ for each $N = 1, 2, \ldots$ and Size-ed$_k(B_N) = \Omega($Size-ed$_k(A_N))$ for any constant $k \geq 4$. We say that B is a* **planar e-kernel** *[**v-kernel**] of A if $B_N \subseteq A_N$ for each $N = 1, 2, \ldots$ and for any constant $k \geq 4$ Plsize-ed$_k(B_N) = \Omega($Plsize-ed$_k(A_N))$ [Plsize-vd$_k(B_N) = \Omega($Plsize-vd$_k(A_N))$]. A kernel (planar e-kernel, planar v-kernel) B of A is called* **(asymptotically) minimal** *if, for every kernel (planar e-kernel, planar v-kernel) $C = \{C_N \mid N = 1, 2, \ldots\}$ of A, $|C_N| = \Omega(|B_N|)$.*

2. Results.

2.1. On the Size of Permutation Networks for Special Subclasses of Permutations. Since there is no asymptotical difference between Size-vd$_k(A)$ and Size-ed$_k(A)$ for any permutation set A (see Observation 1.5), we shall consider only the edge-disjoint path mode in this subsection. It is well-known that

$$\text{Size-ed}_k(\Pi_N) = \Theta(N \log N) = \text{Size-ed}_k(S_N).$$

To see the upper bound Size-ed$_k(\Pi_N) = O(N \log N)$, it is sufficient to take the well-known permutation network [3, 4, 41]. To see the lower bound Size-ed$_k(S_N) = \Omega(N \log N)$, one has to apply the following lower bound method of [38] to S_N.

LEMMA 2.1 ([38], Theorem 2.2.1). *Let A be any subset of Π_N. If A fulfils the following property*

(i) Each input is assigned to $|A|$ different outputs by the $|A|$ permutations.

then Size-ed$_k(A) = \Omega(N \cdot \log_2 |A|)$.

Thus, we obtain that S_N is a kernel of Π_N. But S_N is not a minimal kernel of Π_N because a random set $A \subseteq \Pi_N$ with $|A| \leq \log_2 N$ has Size-ed$_k(A) = \Omega(N \cdot |A|)$ with probability tending to 1 with growing N and $|A|$ [43]. (A summary of the argument of [43] is provided in the Appendix.) Since each $A \subseteq \Pi_N$ has Size-ed$_4(A) = O(N \cdot |A|)$ (see Observation 1.6), the random sets of $\log_2 N$ permutations are **minimal kernels** of Π_N.

In [24], it has been conjectured that $\text{Pow}\,2_N$ and Fib_N are kernels of S_N. We now show that this is not the case.

THEOREM 2.2. *The set of permutations Pow$\,2_N$ can be realized by a (edge-disjoint) switching network of depth $6 \log\log N + 3$ and size $6N \log\log N + 5N$.*

Proof. The proof of this theorem is given in Section 3. □

THEOREM 2.3. *Size-ed$_4(Fib_N) = O(N \cdot 2^{2\sqrt{\log\log N}})$.*

Proof. The proof of this theorem is given in Section 3. □

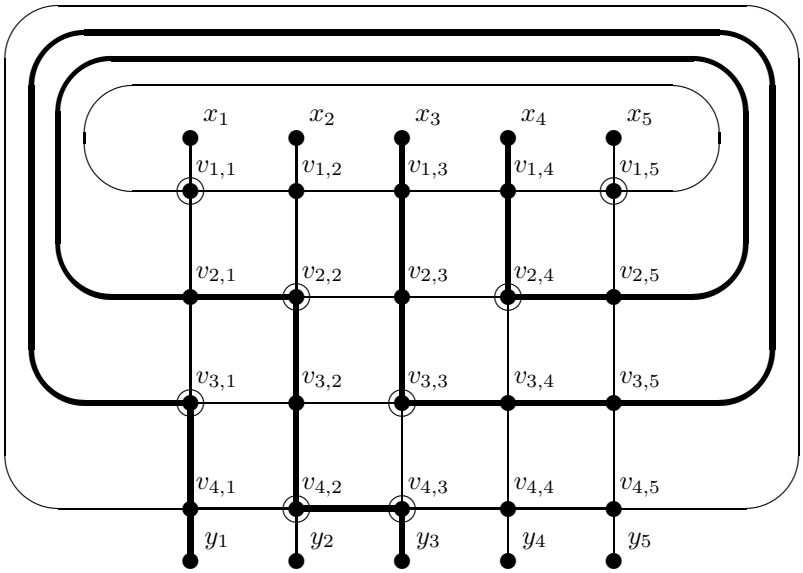

FIG. 3. *PE_5 and the realization of the permutation (5,4,1,3,2)*

COROLLARY 2.4. *The set of permutations $Pow2_N$ can be realized by a regular (vertex-disjoint) permutation network of depth $6\log\log N + 2$ and size $6N\log\log N + 4N$.*

COROLLARY 2.5. *Size-ed$_k$($Pow2_N$) $= \Theta(N \cdot \log\log N)$ for any constant $k \geq 4$ independent of N.*

Proof. The upper bound is in Theorem 2.2 and the lower bounds follow from the fact that Pow 2_N fulfils Property (i) of Lemma 2.1. □

Note that especially Theorem 2.2 brings a crucial contribution for the simulation of hypercube algorithms in bounded-degree networks of degree 4 for several fundamental computing problems. More about this can be found in Subsection 2.3.

2.2. On the Size of Planar Permutation Networks. It is well-known that Plsize-vd$_k(\Pi_N) = \Theta(N^3)$ [10, 29]. Our first result shows that there is an essential difference between edge-disjoint paths mode and vertex-disjoint paths mode for planar permutation networks.

THEOREM 2.6. *For any constant $k \geq 4$ independent of N*

$$\textit{Plsize-ed}_k(\Pi_N) = \Theta(N^2) = \textit{Plsize-ed}_k(S_N).$$

Proof.

UPPER BOUND

Consider the following network $PE_N = (V, E)$ defined by $V = \{ v_{i,j} \mid 1 \leq i < N \ , \ 1 \leq j \leq N\} \cup \{ x_i, y_i \mid 1 \leq i \leq N\}$ and $\{v_{i,j}, v_{k,l}\} \in E$ iff $(i + 1 = k$ and $j = l)$ or $(i = k$ and $j \bmod N = l - 1)$. Furthermore $\{x_i, v_{1,i}\} \in E$ for $1 \leq i \leq N$ and $\{v_{N-1,i}, y_i\} \in E$ for $1 \leq i \leq N$. Observe that PE_N is planar (see Figure 3).

Furthermore, it is well-known that any permutation $\pi \in \Pi_N$ can be built from t transpositions with $t \leq N - 1$. Assume the permutation π is built from t transpositions τ_j $(1 \leq j \leq t)$, where the positions a_j and b_j with $a_j < b_j$ are exchanged in τ_j, i.e., $\tau_j = (1, 2, ..., a_j - 1, b_j, a_j + 1, ..., b_j - 1, a_j, b_j + 1, ..., N)$. Let π_j $(0 \leq j \leq t)$ be the permutation defined by the first j transpositions i.e. $\pi_j = \tau_1 \cdot \tau_2 \cdot ... \cdot \tau_j$. Thus $\pi_0 = id$ and $\pi_t = \pi$ holds (where id is the identity permutation, i.e. $id = (1, 2, \ldots, N)$).

We will now define the N edge-disjoint paths P_i (connecting x_i with $y_{\pi(i)}$) for $1 \leq i \leq N$ step by step. In step j a path P_i^j (connecting x_i with $v_{j+1,\pi_j(i)}$) will be defined using the path P_i^{j-1}. In the final step the path P_i will be defined using the path P_i^t.

Note that for $1 \leq j \leq t$ and $1 \leq i \leq N$:

$$\begin{array}{rcl} \pi_{j-1}(i) \notin \{a_j, b_j\} & \Leftrightarrow & \pi_j(i) = \pi_{j-1}(i) \\ \pi_{j-1}(i) = a_j & \Leftrightarrow & \pi_j(i) = b_j > a_j \\ \pi_{j-1}(i) = b_j & \Leftrightarrow & \pi_j(i) = a_j < b_j \end{array}$$

We define for $1 \leq j \leq t$ and $1 \leq i \leq N$:

$$\begin{array}{rcll} P_i^0 & = & (x_i, v_{1,i}) & \\ P_i^j & = & P_i^{j-1}, v_{j+1,\pi_j(i)} & \text{if } \pi_{j-1}(i) \notin \{a_j, b_j\} \\ P_i^j & = & P_i^{j-1}, v_{j,a_j+1}, v_{j,a_j+2}, ..., v_{j,b_j}, v_{j+1,b_j} & \text{if } \pi_{j-1}(i) = a_j \\ P_i^j & = & P_i^{j-1}, v_{j,b_j+1}, v_{j,b_j+2}, ..., v_{j,n}, v_{j,1}, ..., v_{j,a_j}, v_{j+1,a_j} & \text{if } \pi_{j-1}(i) = b_j \\ P_i & = & P_i^t, v_{t+1,\pi(i)}, v_{t+2,\pi(i)}, ..., v_{N-1,\pi(i)}, y_{\pi(i)} & \end{array}$$

This is illustrated in the example of Figure 3: The nodes marked by a circle are the nodes v_{j,a_j}, v_{j,b_j} corresponding to a_j, b_j $(1 \leq j \leq t)$. They send the message from the top edge to the right hand side edge and the message from the left hand side to the bottom edge. The other nodes send the message from the top edge to the bottom edge and the message from the left hand side to the right hand side edge. In Figure 3 the permutation $(5, 4, 1, 3, 2)$ is carried out:

$$\begin{array}{llllll} \pi_0 = (1,2,3,4,5) & \pi_1 = (5,2,3,4,1) & \tau_1 = (5,2,3,4,1) & a_1 = 1 & b_1 = 5 \\ & \pi_2 = (5,4,3,2,1) & \tau_2 = (1,4,3,2,5) & a_2 = 2 & b_2 = 4 \\ & \pi_3 = (5,4,1,2,3) & \tau_3 = (3,2,1,4,5) & a_3 = 1 & b_3 = 3 \\ \pi = (5,4,1,3,2) & \pi_4 = (5,4,1,3,2) & \tau_4 = (1,3,2,4,5) & a_4 = 2 & b_4 = 3 \end{array}$$

The two paths from x_3 to $x_{\pi(3)} = y_1$ and from x_4 to $x_{\pi(4)} = y_3$ in the realization of the permutation are depicted in the figure.

It is now easy to check that:

1. An edge of the form $(v_{j,\pi_j(i)}, v_{j+1,\pi_j(i)})$ $(1 \leq i \leq N, 1 \leq j \leq t)$ is only used in the path P_i.
2. An edge of the form $(v_{j,\pi(i)}, v_{j+1,\pi(i)})$ $(1 \leq i \leq N, t \leq j < N-1)$ is only used in the path P_i.
3. An edge of the form $(v_{j,i}, v_{j,i'})$ is only used by one of the paths $P_1, P_2, ..., P_N$.

It follows from 1.–3. that the paths $P_1, P_2, ..., P_N$ are edge-disjoint. From the construction of $P_1, P_2, ..., P_N$ it also follows that P_i connects x_i with $y_{\pi(i)}$ for all $1 \leq i \leq N$. Thus we conclude $\text{Plsize-ed}_4(\Pi_N) \leq N \cdot (N+1)$.

Lower Bound

We first introduce some additional definitions. Let $A \subseteq \Pi_N$. The **permutation graph of A** is $\mathbf{G(A)} = (V, E)$, where $V = \{x_1, \ldots, x_N, y_1, \ldots, y_N\}$ and, for every $\pi = (i_1, i_2, \ldots, i_N) \in A$, E contains the edges $(x_1, y_{i_1}), \ldots, (x_N, y_{i_N})$. Let (V_1, V_2), $V_1 \cup V_2 = V$, $|V_1| = |V_2| = N$, be a bisection of $G(A)$. The bisection width of $G(A)$ according to (V_1, V_2) and π, $\mathbf{bw(G(A), (V_1, V_2), \pi)}$, is the number of edges defined by π and leading between V_1 and V_2. $\mathbf{bw(G(A), (V_1, V_2))} = \max\{\text{bw}(G(A), (V_1, V_2), \pi) \mid \pi \in A\}$. The **balanced bisection width of $G(A)$** is $\mathbf{bw(G(A))} = \min\{\text{bw}(G(A), (V_1, V_2)) \mid (V_1, V_2) \text{ is a bisection of } G(A)\}$.

Now, we present the lower bound proof. It is based on an extended planar separator theorem [29]. The precise construction is as follows. Let $A \subseteq \Pi_N$. Let $G = (\bar{V}, \bar{E})$ be a planar, edge-disjoint k-permutation network for A. Let $a(G) \subseteq \bar{V}$ be the input and output nodes of G. Let all nodes in $a(G)$ be coloured by red, and all other nodes in $\bar{V} - a(G)$ be coloured by blue. (Note that the nodes of $a(G)$ are exactly the nodes of $G(A)$.) Let $n := \text{size}(G)$. Then, according to an extended planar separator theorem [29], there exists a constant c (independent of n) and $c \cdot k\sqrt{n}$ edges of G such that their removal from G bisects G into two equal-sized components G^1 and G^2, where each G^i $(i = 1, 2)$ contains N red nodes and at least $\lfloor n/2 \rfloor - N$ blue nodes. This bisection of G determines the bisection of $G(A)$ into H^1 and H^2, where $V(H^1) = a(G) \cap V(G^1), V(H^2) = a(G) \cap V(G^2)$. For $\pi \in A$, let E_π be the set of edges of $G(A)$ between $V(H^1)$ and $V(H^2)$ defined by π. According to the definition of $\text{bw}(G(A))$, there exists $\pi_0 \in A$ such that $|E_{\pi_0}| \geq \text{bw}(G(A))$. To realize π_0 in G, there must exist $\text{bw}(G(A))$ edge-disjoint paths between the red nodes of G^1 on one side and the red nodes of G^2 on the other side. Clearly, this is only possible if

$$c \cdot k\sqrt{n} \geq \text{bw}(G(A)).$$

Thus, we obtain

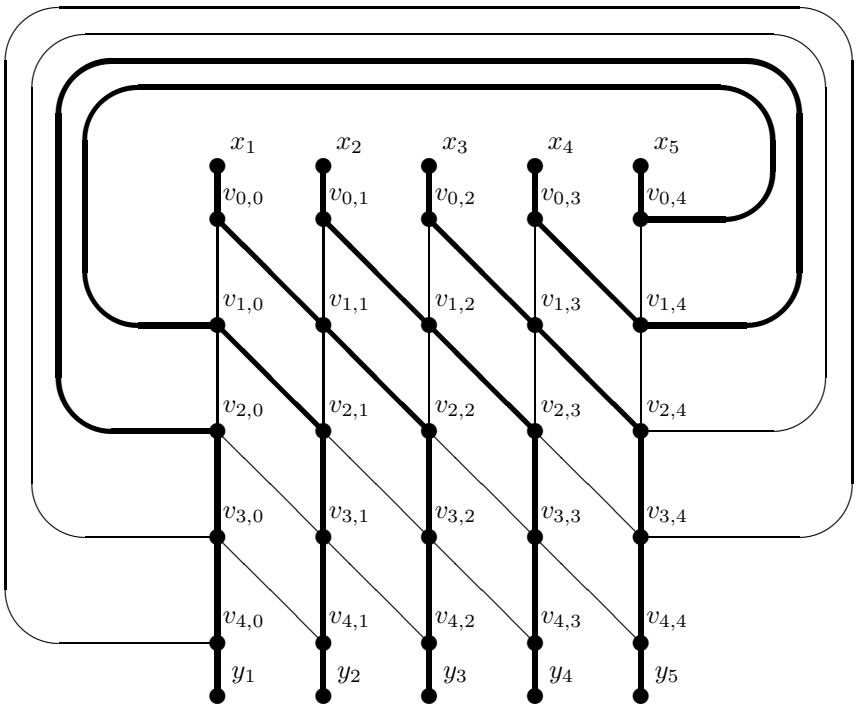

FIG. 4. *A shift by 2 realized by* PS_5

$$\text{size}(G) = n \ \geq \ (\text{bw}(G(A)))^2 \ / \ (c \cdot k)^2. \qquad (*)$$

It remains to determine a lower bound on $\text{bw}(G(A))$ for $A = S_N$. For this purpose, we apply the standard technique described in [31], i.e., we first embed the complete graph of $2N$ nodes, K_{2N}, into S_N such that the congestion of each edge of S_N is small. Consider an embedding $f : V(K_{2N}) \to V(S_N)$. We specify a path $\text{path}(x, y)$ between two arbitrary nodes x, y in S_N as follows:

a) $\text{path}(x_i, y_j) = (x_i, y_j)$ for all $i, j \in \{1, \ldots, N\}$,
b) $\text{path}(x_i, x_j) = (x_i, y_k, x_j),\ k = (j-i)/2,$ for all $i, j \in \{1, \ldots, N\}$, $i < j$,
c) $\text{path}(y_i, y_j) = (y_i, x_k, y_j),\ k = (j-i)/2,$ for all $i, j \in \{1, \ldots, N\}$, $i < j$.

a) contributes 1 to the congestion of an edge $e \in E(S_N)$, b) and c) each contribute at most 2. Hence, the congestion of each edge of S_N is at most 5. Each bisection of S_N defines a bisection of K_{2N}. As the bisection width of K_{2N} is N^2 and the embedding f has congestion 5, the bisection width of S_N is at least $N^2/5$. As $|S_N| = N$, it follows that $\text{bw}(G(S_N)) \geq N/5$. Combining this result with $(*)$ yields

$$n \ \geq N^2/(25 \cdot c^2 k^2) \ .$$

Thus, we obtain $\text{Plsize-ed}_k(S_N) = \Omega(N^2)$. □

Again, we see that S_N is a planar e-kernel of Π_N. But the next result showing that S_N is no planar v-kernel of Π_N underlines the difference between these two communication modes.

THEOREM 2.7. *Plsize-vd*$_4(S_N) \leq N \cdot (N+2)$.

Proof. We define the network $PS_N = (V, E)$ by $V = \{ v_{i,j} ; 0 \leq i, j < N\} \cup \{ x_i, y_i ; 1 \leq i \leq N\}$ and $\{v_{i,j}, v_{k,l}\} \in E$ iff $i + 1 = k$ and $l \in \{j, (j+1) \bmod N\}$. Furthermore $\{x_i, v_{0,i-1}\} \in E$ for $1 \leq i \leq N$ and $\{v_{N-1,i-1}, y_i\} \in E$ for $1 \leq i \leq N$. Observe that PS_N is planar (see Figure 4). The messages from the nodes $\{ x_i ; 1 \leq i \leq N\}$ are shifted to the nodes $\{ y_i ; 1 \leq i \leq N\}$. The $i-th$ message $(1 \leq i \leq N)$ uses the path P_i^s when a shift by s $(0 \leq s < N)$ is carried out, where $P_i^s = (x_i, v_{0,i-1}, v_{1,(i+\min(1,s)-1) \bmod N}, v_{2,(i+\min(2,s)-1) \bmod N}, \ldots, v_{j,(i+\min(j,s)-1) \bmod N}, \ldots, v_{N-1,(i+\min(N-1,s)-1) \bmod N}, y_{(i+s-1) \bmod N+1})$. Again it is easy to observe that these paths are vertex-disjoint (see Figure 4). □

COROLLARY 2.8. *For any constant $k \geq 4$ independent of N*

$$\textit{Plsize-vd}_k(S_N) = \Theta(N^2).$$

Proof. The result is a direct consequence of Theorem 2.6 and Theorem 2.7 (and the fact that Plsize-$\mathrm{vd}_k(A) \geq$ Plsize-$\mathrm{ed}_k(A)$ for any $A \subseteq \Pi_N$). □

Again, as in the general case (Subsection 2.1), S_N is no minimal e-kernel of Π_N. In the following, we show that there are e-kernels of Π_N of size 5.

THEOREM 2.9. *Let $N = m^2$, $A_m = \{0, 1, \ldots, m-1\} \times \{0, 1, \ldots, m-1\}$, and let $B_N = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5\} \subseteq \Pi_N$ be the following permutations on A_m:*

$$\begin{aligned}
&\pi_1(x, y) = (x, y),\\
&\pi_2(x, y) = (x, x + y),\\
&\pi_3(x, y) = (x, x + y + 1),\\
&\pi_4(x, y) = (x + y, y),\\
&\pi_5(x, y) = (x + y + 1, y),
\end{aligned}$$

where the + is modulo m. Then, for any constant k independent of N,

$$\textit{Plsize-ed}_k(B_N) = \Theta(N^2).$$

THEOREM 2.10. *Let R_N be a set of 5 random permutations from Π_N. Then, with probability tending to 1 with growing N*

$$\textit{Plsize-ed}_k(R_N) = \Theta(N^2),$$

for any constant k independent of N.

Proofs of Theorem 2.9 and 2.10. Let $A = B_N$ [$A = R_N$]. Applying the upper bound of Theorem 2.6, we obtain Plsize-$\mathrm{ed}_k(A) = O(N^2)$. The lower-bound proof uses the same technique as the lower-bound proof of Theorem 2.6. Let G be a planar, edge-disjoint k-permutation network for A. Using the same notation as in Theorem 2.6, we have

$$\mathrm{size}(G) \;\geq\; (\mathrm{bw}(G(A)))^2 \;/\; (c \cdot k)^2 \qquad (*)$$

for some constant c (independent of size(G)) [cf. the proof of Theorem 2.6]. Since $G(A)$ is an expander [17, 36], $G(A)$ has bisection width linear in N. As $|A| = 5$, bw($G(A)$) is also linear in N. It follows from $(*)$ that Plsize-$\mathrm{ed}_k(A) = \Omega(N^2)$. □

Finally, we consider the set $\mathrm{Pow}\,2_N$. Despite the fact that $\mathrm{Pow}\,2_N$ is no planar e-kernel of Π_N, the following result shows that $\mathrm{Pow}\,2_N$ requires almost the same size for planar realization as Π_N.

THEOREM 2.11. *For any constant k independent of N*

$$\textit{Plsize-ed}_k(\textit{Pow}\,2_N) \quad = \quad \Theta(N^2 \cdot (\log\log N)^2/(\log N)^2)\,.$$

Proof. The proof of this theorem is given in Section 3. □

2.3. Application of Permutation Networks to Hypercube Simulation and Gossiping. The aim of this section is to show the consequences of Theorem 2.2 for other tasks than routing of messages. The main idea of the use of Theorem 2.2 is in the simulation of hypercube algorithms in degree-bounded networks.

Given a network G, a **network communication algorithm** in G [23, 31] is a synchronized parallel algorithm executing alternately communication steps and computing steps. In each **computing step**, each processor of the network G executes some computation on its local data. In each **communication step**, each processor of G can exchange some message with one of its neighbours via an adjacent edge (i.e. each processor may communicate with at most one neighbour in one communication step).

A **hypercube algorithm** is a network communication algorithm where the network G is the (binary) hypercube of dimension m. The **(binary) hypercube** of dimension m, denoted by H_m, is the network whose nodes are all binary strings of length m and whose edges connect those binary strings which differ in exactly one position. For each i, $1 \leq i \leq m$, an edge $(a_1a_2 \ldots a_{i-1}0a_{i+1} \ldots a_m, a_1a_2 \ldots a_{i-1}1a_{i+1} \ldots a_m)$,

$a_1, a_2, \ldots, a_{i-1}, a_{i+1}, \ldots, a_m \in \{0,1\}$, is said to be in **dimension** i. An illustration of H_3 is shown in Figure 5.

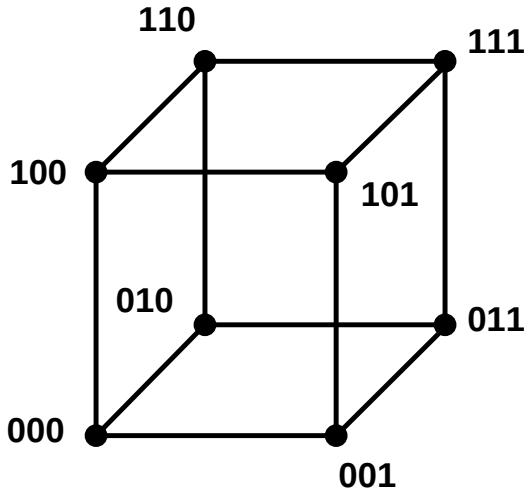

FIG. 5. *The hypercube* H_3

We distinguish three classes of hypercube algorithms:

(1) **general hypercube algorithms**, as described above
(2) **leveled hypercube algorithms** [31, 44], which communicate in every communication step via edges of one fixed dimension of the hypercube
(3) **normal hypercube algorithms** [31, 39, 40] are leveled ones with the additional condition that every two consecutive communication steps use two consecutive dimensions of the hypercube.

Additionally, we define ***d*-leveled hypercube algorithms** as hypercube algorithms using edges of at most d different dimensions in each communication step.

Any communication step of a normal hypercube algorithm running on a hypercube of size N can be simulated by two communication steps by the Shuffle-Exchange network of size N. For general algorithms, the best simulation of one communication step of the hypercube by bounded-degree networks uses $\log_2 N$ steps. Using the network from Theorem 2.2 we can construct a 4-degree network simulating one communication step of any leveled hypercube algorithm in $6 \cdot \log_2 \log_2 N$ steps.

THEOREM 2.12. *There is a 4-degree network of size* $6 \cdot N \cdot \log\log N$ *which can simulate any communication step of any* d*-leveled hypercube algorithm on a hypercube of* N *nodes in* $6 \cdot d \cdot \log\log N$ *communication steps.*

Because almost all common hypercube algorithms are leveled, we obtain several efficient parallel algorithms running on bounded-degree networks as a consequence of Theorem 2.12. Note that our simulation beats the previous ones in the delay. All the previous simulations of the general and the d-leveled hypercube algorithms by degree-bounded networks had delay $O(\log N)$ [31], while we have delay $O(\log\log N)$. Another positive property of our network is its size $6 \cdot N \log\log N$, i.e., we do not need to pay too much with the increase of the size for this small delay. Moreover, Theorem 2.12 can be considered as an improvement of the $O(1)$-delay simulation of normal hypercube algorithms [39] because the simulation given by Theorem 2.12 is static, which means that during the whole simulation every processor of our permutation network simulates the work of at most one processor of the hypercube. The simulation of normal hypercube algorithms [39] is dynamic. For this particular simulation it is even forced that each processor has to submit the whole content of its memory to some of its neighbours after each simulation step. If this content is essentially larger than the communication messages exchanged, then communicating the content may cause a delay larger than $6 \cdot \log\log N$.

Another application of the previous theorems is related to gossiping (all to all broadcasting) in edge-disjoint (vertex-disjoint) paths mode.

Gossiping (all to all broadcasting) is one of the basic communication tasks in network communication. It can be described as follows. Assume that each vertex (processor) in a graph (network) has some piece of information. The **cumulative message** of G is the set of all pieces of information originally distributed in all vertices of G. To solve the **gossip** problem for a given graph G, a communication strategy such that all vertices in G learn the cumulative message of G must be found.

The meaning of a "communication strategy" depends on the communication mode. A communication strategy is realized by a **communication algorithm** consisting of a number of **communication steps (rounds)**. The rules describing what can happen in one communication step (round) are defined exactly by the communication mode. Here, we consider the following two modes:

- **One-way [Two-way] vertex-disjoint paths mode** (1VDP mode [2VDP mode])
  One round can be described as a set $\{P_1, \ldots, P_k\}$ for some $k \in I\!N$, where $P_i = x_{i,1}, \ldots, x_{i,\ell_i}$ is a simple path of length $\ell_i - 1$, $i = 1, \ldots, k$, and the paths are vertex-disjoint. The executed communication of this round in one-way mode consists of the submission of the whole actual knowledge of $x_{i,1}$ to x_{i,ℓ_i} via path P_i for any $i = 1, \ldots, k$. [The executed communication of this round in two-way mode consists of the complete exchange of the actual knowledge between $x_{i,1}$ and x_{i,ℓ_i} for any $i = 1, \ldots, k$]. The inner nodes of path P_i (nodes different from the end points $x_{i,1}$ and x_{i,ℓ_i}) do not learn the message submitted from $x_{i,1}$ to x_{i,ℓ_i} [exchanged between $x_{i,1}$ and x_{i,ℓ_i}] they are only used to realize the connection from $x_{i,1}$ to x_{i,ℓ_i}.

The complexity of a communication algorithm A is measured as the number of rounds of A. For any graph G, the **one-way (two-way) vertex-disjoint gossip complexity** of G is the number of rounds (complexity) of the optimal gossip algorithm for G in the 1VDP (2VDP) mode.

Let C_M denote the complete graph of M nodes. In [24] it is shown that there are bounded-degree networks of size M whose

(i) two-way vertex-disjoint gossip complexity is $r_2(C_M) + O(\log\log M)$, where $r_2(C_M) = \log_2 M$ is the two-way gossip complexity of the complete graph C_M on M nodes.
(ii) one-way vertex-disjoint gossip complexity is $r_1(C_M) + O(\log\log M)$, where $r_1(C_M) \approx 1.44 \log_2 M$ is the one-way gossip complexity of C_M.

Thus, one can gossip in bounded-degree networks almost as quickly as in complete graphs using the vertex-disjoint paths mode. (Note that there are no bounded-degree networks in which gossiping in the standard communication modes –in which communication is only allowed via single edges and not via paths– can be performed only about $O(\log_2 \log_2 n)$ rounds slower than the optimal gossip algorithms on complete graphs [23].) In [24] it has been conjectured that the additional $\log\log M$ steps are necessary to gossip in bounded-degree networks. Theorems 2.2 and 2.3 surprisingly provide networks which can gossip even faster.

THEOREM 2.13. *There is a 4-degree network of size $M = 12N \log\log N + 7N$ for any positive integer $N \geq 16$ with two-way vertex-disjoint paths gossip complexity smaller than $r_2(C_M) + \log\log\log M + const$.*

THEOREM 2.14. *There is a 4-degree network of size $M = O\left(N \cdot 2^{\,2\sqrt{\log\log N}}\right)$ for every positive integer $N \geq 16$ with one-way vertex-disjoint paths gossip complexity smaller than $r_1(C_M) + 1.13\sqrt{\log\log M} + const$.*

Proofs of Theorems 2.13 and 2.14. To see the connection between Theorems 2.2 and 2.3 and gossiping in the two-way mode and one-way mode resp., we first recall the concept of three-phase gossip algorithms introduced in [24, 25].

Let G be any graph. Let $a(G)$ be any subset of nodes of G. We call $a(G)$ the **set of accumulation nodes**, and every node in $a(G)$ is called an **accumulation node**. A **three-phase gossip algorithm for G according to $a(G)$** works in the following three phases:

1. **Accumulation Phase**
   Divide G into $|a(G)|$ connected components, each component containing exactly one accumulation node of $a(G)$. These components are called **accumulation components**. Each $v \in a(G)$ accumulates the information from the nodes lying in its component.
   {After the first phase, the nodes in $a(G)$ together know the cumulative message of G.}

2. **Gossip Phase**
   Perform a gossip algorithm among the nodes in $a(G)$ in the given (one-way or two-way) vertex-disjoint paths mode (i.e., all nodes in $V(G) - a(G)$ are considered to have no information, and they are only used to build disjoint paths between receivers and senders from $a(G)$).
   {After the second phase, every node in $a(G)$ knows the cumulative message of G.}
3. **Broadcast Phase**
   Every node in $a(G)$ broadcasts the cumulative message in its component.
   {After this, all nodes of G know the cumulative message of G.}

In order to construct a really effective gossip algorithm, we shall search for an $a(G)$ in G such that

a) Phase 2 can be performed (almost) as quickly as gossiping in a complete graph of $a(G)$ nodes.
b) The maximal size of a component is as small as possible, which minimizes the time for the first and third phase.

Obviously, every second phase of a three-phase algorithm A corresponds unambiguously to a gossip algorithm C in a complete graph of $|a(G)|$ nodes. We say that C **is implemented in the second phase of** A. Note that all algorithms designed so far for disjoint-path modes are three-phase algorithms with second phases implementing an optimal (or almost optimal) gossip algorithm on graphs of $|a(G)|$ nodes.

We now start with the proof of Theorem 2.13. Corollary 2.4 yields a regular vertex-disjoint 4-permutation network $G = (V, E)$ for $\mathrm{Pow}\,2_N$ of size $6N \log\log N + 4N$ and depth $d = 6 \log\log N + 2$. Let $r = d + 2$. Let $V_1, V_2, \ldots, V_r$ be the levels of G. For $1 \leq i \leq r$, let $V_i = \{(i, 0), (i, 1), \ldots, (i, N-1)\}$ such that for every $1 \leq i \leq r-1$, $0 \leq j \leq N-1$, there is an edge between the nodes (i, j) and $(i+1, j)$. Define G' as follows:

$$
\begin{aligned}
&G' = (V', E') \text{ with}\\
&V' = \{(i,j) \mid i \in \{1, 2, \ldots, r-1, r, -(r-1), -(r-2), \ldots, -1\}, 0 \leq j \leq N-1\}\\
&E'\colon\ 1 \leq i \leq r-1\colon\ \{(i,j),(i+1,j')\} \in E' \text{ iff } \{(i,j),(i+1,j')\} \in E,\\
&\qquad \{(r,j),(-(r-1),j')\} \in E' \text{ iff } \{(r-1,j),(r,j')\} \in E,\\
&\qquad 1 \leq i \leq r-2\colon\ \{(-i,j),(-(i+1),j')\} \in E' \text{ iff } \{(i,j),(i+1,j')\} \in E.
\end{aligned}
$$

G' can be viewed as two copies of G put back-to-back to each other. Let the levels of G' be denoted by $V'_1, V'_2, \ldots, V'_{r-1}, V'_r, V'_{-(r-1)}, V'_{-(r-2)}, \ldots, V'_{-1}$. The purpose of using G' rather than G in the following construction is that all the permutations which can be routed in G from V_1 to V_r can be routed in G' from V_1 to V_{-1} and from V_{-1} to V_1.

Now, we construct a three-phase algorithm for G' by choosing as accumulation nodes the vertices $(1, i), (-1, i), 0 \leq i \leq N-1$, at level V'_1 and V'_{-1} of G'. As accumulation components we take the paths $\{(1, i), (2, i), \ldots, (r, i)\}$ and $\{(-(r-1), i), (-(r-2), i), \ldots, (-1, i)\}$ for all $0 \leq i \leq N-1$. Then Phase 1 and Phase 3 of the three-phase algorithm take $\lceil \log_2 r \rceil$ rounds each. In Phase 2 of the three-phase algorithm, we simulate the Knödel algorithm [22] on all accumulation nodes as follows: Instead of having a direct communication between i and $\mathrm{shift}_N(2^i)$ in round i of the Knödel algorithm, we communicate via the vertex-disjoint paths given by the realization of $\mathrm{Pow}\,2_N$ on G'. Hence, Phase 2 takes $\lceil \log N \rceil + 1$ rounds. Therefore, the whole 3-phase algorithm takes

$$
\begin{aligned}
&\lceil \log N \rceil + 1 + 2 \cdot \lceil \log r \rceil\\
&\quad \leq\ \log N + 2 \cdot \log\log\log N + O(1)\\
&\quad \leq\ r_2(C_M) + \log\log\log M + O(1)
\end{aligned}
$$

rounds. This completes the proof of Theorem 2.13.

The proof of Theorem 2.14 is completely analogous to the proof of Theorem 2.13. Instead of using the regular vertex-disjoint 4-permutation network for $\mathrm{Pow}\,2_N$ from Theorem 2.2, we use the regular vertex-disjoint 4-permutation network G for Fib_N from Theorem 2.3. Now, we construct a three-phase algorithm like above, implementing the well-known Fibonacci algorithm [22] in the gossip phase. □

3. Proofs.

3.1. The Proof of Theorems 2.2 and 2.3. In what follows we slightly alter the formalism of the description of switching networks to make tracing its performance easier (this is the so-called **switch-wire form** of the switching-network model): we enumerate inputs by $0, 1, \ldots, N-1$ and give the same labels to the edges coming out of these nodes. Then we spread the enumeration to the rest of edges maintaining the following property: if an incoming edge of a switch s has a label p, then also one outcoming edge of s is labelled p. When we reach the last layer we give the labels of edges to the adjacent output nodes. So the output vertices receive labels $0, 1, \ldots, N-1$. Notice that the edges with a label p $(0 \leq p \leq N-1)$ form a path from the input p to the output p consisting of exactly $r+1$ edges. We call this path a **wire** p. In Figure 6 we present the example from Figure 1 with some node labels and wires 3 and 6 marked with dashed lines.

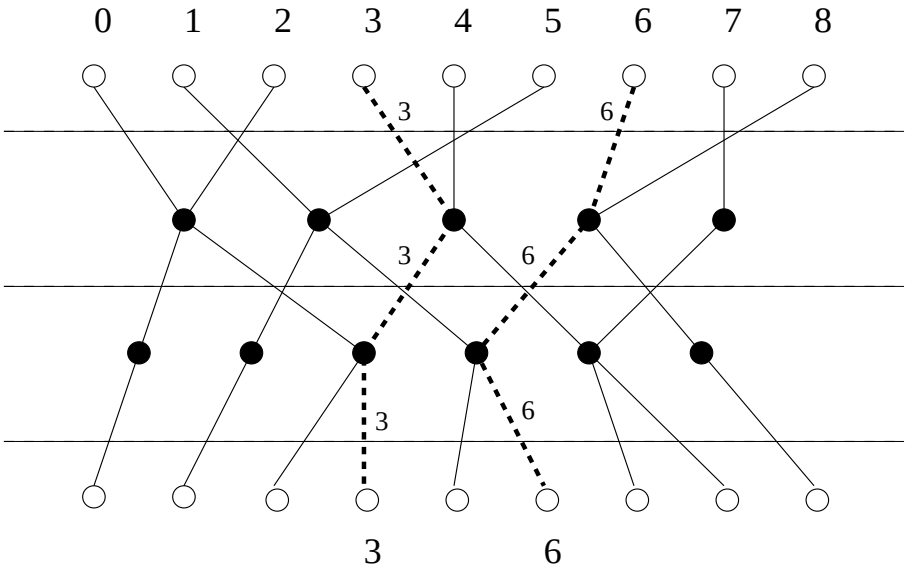

FIG. 6. *The labeling of the network*

Now we can present a graph G in a bit different way. We draw wires as vertical parallel lines ordered according to the numbering. Then consecutive layers are drawn from top to bottom. Every switch, whose edges were named p and q is marked as a horizontal bar connecting wires p and q. An idle node is marked as a dot on the appropriate wire (see Figure 7). Assuming the layer is given, let $\langle p, q\rangle$ denote the switch joining wires p and q. To represent a switching network, it also suffices to list all switches in each layer (Figure 8).

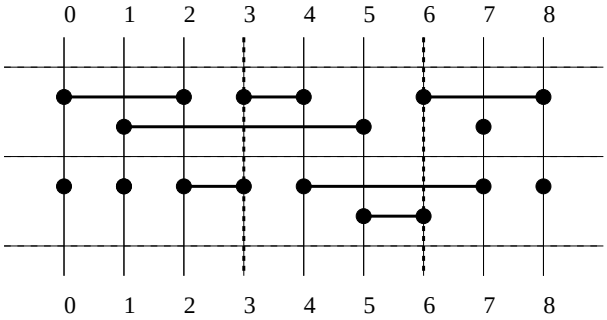

FIG. 7. *A switching network in the switch-wire form*

$V_1 = \{\langle 0,2\rangle, \langle 1,5\rangle, \langle 3,4\rangle, \langle 6,8\rangle\}$
$V_2 = \{\langle 2,3\rangle, \langle 4,7\rangle, \langle 5,6\rangle\}$.

FIG. 8. *Layers of the network*

The switching network works as follows. Initially we set every switch in the network: the switch $s = \langle p, q\rangle$ is **active** if it exchanges p with q and vice versa, otherwise s is **inactive**. After the network is set, we

assign items $v_0, v_1, \ldots, v_{N-1}$ (in what follows, we usually take $v_0 = 0, v_1 = 1, \ldots, v_{N-1} = N-1$) to the inputs and then send this sequence, step by step, through the layers $V_1, V_2, \ldots, V_r$. If a switch $s = \langle p, q \rangle$ is active and the wires p and q bring items v_i and v_j, respectively, then s puts v_i on the wire q and v_j on the wire p and sends them to further layers of the network. If the switch is inactive, then the items follow the wires they came. When the items $v_0, v_1, \ldots, v_{N-1}$ reach the output layer V_{r+1}, the process is finished. Hence everything G does is to permute the positions of the input items. (See Figure 9 for an example performance of a switching network; in the figure the active switches are drawn in a solid line while the inactive ones are drawn in a dashed line.)

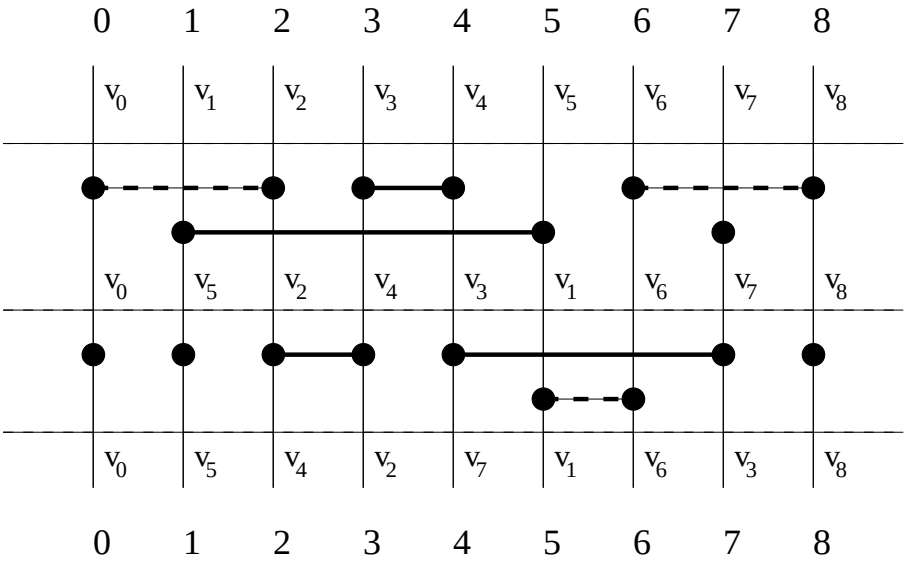

FIG. 9. *An example performance of the network*

Let $Perm(G)$ be the set of permutations that can be obtained as the output of the network G by some settings of the switches of G. A set A of permutations is **realizable** by G if $A \subseteq Perm(G)$. Let $Shift_k$ denote the permutation $\text{shift}_N(k)$.

In order to present some simple but useful (in our construction) network, let us define

$$Sym_t = \{\langle i, (t-i) \bmod N \rangle \mid 0 \leq i < N\},$$

for $t = 0, 1, \ldots, N-1$. Regarding $\langle i, j \rangle$ and $\langle j, i \rangle$ as the same switch we obtain that Sym_t is a network of depth 1. We can consider Sym_t as a cyclic symmetry on the interval $\{0, 1, \ldots, N-1\}$ with the center at $t/2$. As a shift is a cyclic equivalent of a translation, we can reformulate the classical fact saying that each translation is a composition of two symmetries in the following way:

LEMMA 3.1. *Every shift can be performed by a network of depth* 2.

Proof. Suppose we have to shift items $\langle 0, 1, \ldots, N-1 \rangle$ by k $(0 \leq k < N)$. Notice that for any $0 \leq i < N$:

$$Sym_k \circ Sym_0(i) = Sym_k(-i \bmod N) = (i+k) \bmod N,$$

where $\circ$ denotes the composition of permutations. (For two permutations π_1, π_2, the composition $\pi_2 \circ \pi_1$ is the permutation that first performs π_1 and then π_2, i.e. $\pi_2 \circ \pi_1(i) = \pi_2(\pi_1(i))$.) Consider the network consisting of the two layers: Sym_0 (the first layer) and Sym_k (the second layer). Then this network (with all switches set active) performs the permutation $Sym_k \circ Sym_0$, i.e. the cyclic shift by k. □

A **cycle** is the permutation $Shift_1$ of its elements. Since every permutation is a composition of disjoint cycles, we get easily the next Corollary:

COROLLARY 3.2. *Every permutation can be performed by a network of depth* 2.

EXAMPLE 3.3. *A sample construction of a network performing a permutation.* Let us consider a

permutation

$$P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 3 & 7 & 2 & 1 & 5 & 8 & 0 & 6 & 9 & 10 & 11 & 4 \end{pmatrix}.$$

P is a composition of disjoint cycles:

$$P = (0, 3, 1, 7, 6)(2)(4, 5, 8, 9, 10, 11).$$

(The three disjoint cycles of P are as follows: $0 \mapsto 3 \mapsto 1 \mapsto 7 \mapsto 6 \mapsto 0$, $2 \mapsto 2$, $4 \mapsto 5 \mapsto 8 \mapsto 9 \mapsto 10 \mapsto 11 \mapsto 4$, where $i \mapsto j$ stands for $P(i) = j$. The cycle notation for permutations used in this example is a standard notation, see e.g. in [30].) We build a separate $Shift_1$ network for each cycle (see Figure 10).

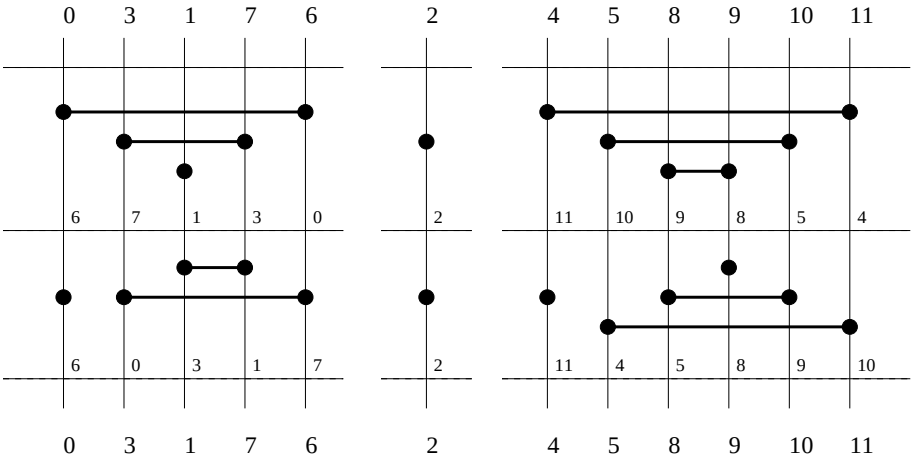

FIG. 10. *The separate networks for each cycle*

Then, rearranging the wires according to their numbering, we obtain the network given in Figure 11. □of Example

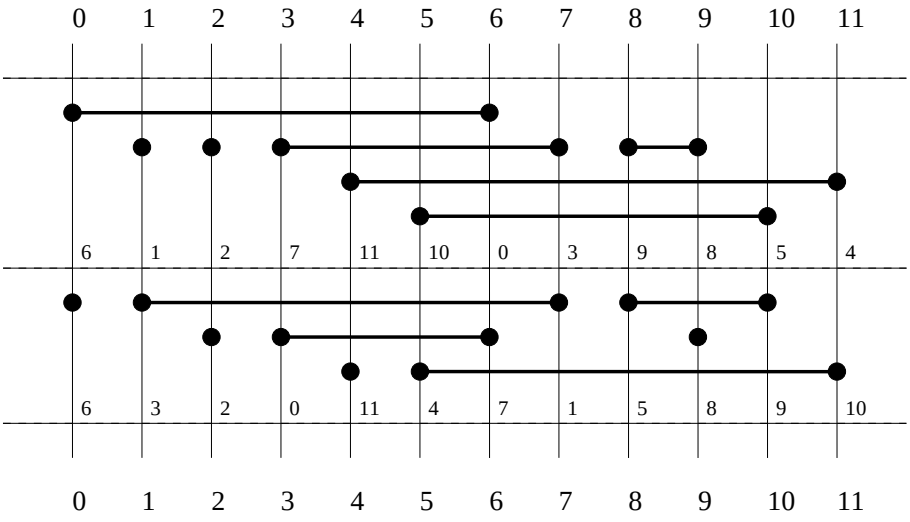

FIG. 11. *The final network for permutation*

The next problem we consider is to perform all shifts of the input $\langle 0, 1, \ldots, N-1 \rangle$, namely $\{Shift_t \mid 0 \leq t < N\}$. Evidently, this can be done by Waksman-Beneš network, of depth $2\lceil \log N \rceil - 1$. However, it can be improved.

LEMMA 3.4. *There is a network of depth* $\lceil \log N \rceil + 1$ *performing cyclic shifts by* $0, 1, \ldots, N-1$.

Proof. Let $q = \lfloor \log N \rfloor$. Consider the network consisting of $q+2$ layers: Sym_1, Sym_2, Sym_4,..., $Sym_{2^{q+1}}$. We claim that this network may perform all required cyclic shifts. Let $0 \leq t \leq N-1$ and $t = t_q t_{q-1} \cdots t_1 t_0$ be a binary representation of t. Thus

$$Shift_t = \circ_{i=0}^{q} (Shift_{2^i})^{t_i},$$

where $Shift_s^k$ denotes $Shift_s$ applied k times. Notice that for any $0 \leq i < N$:

$$\begin{aligned} Sym_{2^{i+1}} \circ Sym_{2^i}(i) &= Sym_{2^{i+1}}((2^i - i) \bmod N) = (2^{i+1} - (2^i - i)) \bmod N \\ &= (i + 2^i) \bmod N = Shift_{2^i}(i). \end{aligned}$$

Hence

$$\begin{aligned} Shift_t &= \circ_{i=0}^{q} (Sym_{2^{i+1}} \circ Sym_{2^i})^{t_i} \\ &= (Sym_{2^{q+1}})^{t_q} \circ (\circ_{i=1}^{q} (Sym_{2^i}^{t_i} \circ Sym_{2^i}^{t_{i-1}})) \circ Sym_1^{t_0} \\ &= (Sym_{2^{q+1}})^{t_q} \circ (\circ_{i=1}^{q} (Sym_{2^i})^{t_i + t_{i-1}}) \circ Sym_1^{t_0}. \end{aligned}$$

Because $Sym_k^2 = Sym_k \circ Sym_k$ is an identity, it follows that $Shift_t$ is a composition of different symmetries from the set $\{Sym_{2^i} \mid 0 \leq i \leq q+1\}$. To perform $Shift_t$ it is enough to activate only layers corresponding to these symmetries. It is easy to check that $\lceil \log N \rceil + 1$ symmetries suffice to perform all required shifts (one has to improve the construction slightly for N being a power of 2, otherwise obviously $q+2 = \lceil \log N \rceil + 1$). □

For later use we mention one more simple but useful construction.

LEMMA 3.5. *Every set of shifts S can be performed by a network of depth $|S| + 1$.*

Proof. Let $S = \{Shift_{t_i} \mid 1 \leq i \leq m\}$. We build a network consisting of layers: Sym_0, Sym_{t_1}, Sym_{t_2}, ..., Sym_{t_m}. As $Shift_{t_i} = Sym_{t_i} \circ Sym_0$, to realize $Shift_{t_i}$ it suffices to set the switches of the layers 0 and t_i active and the remaining switches inactive. □

3.1.1. Shifts by powers of 2. The first step towards our final result for shifts by powers of 2 is the following construction. Let G' be an N-wire network performing shifts by $a_1, a_2, \ldots, a_k$. For any integer $m = 1, \ldots, N$, we build a network $G(N, m, G')$ which performs shifts by a_i, for $i = 1, \ldots, k$, and by $m \cdot a_i$, for all $a_i \leq \lfloor \frac{N}{m} \rfloor$.

To make the construction of $G(N, m, G')$ more clear we arrange its wires in an array $A(N, m) = \{a_{ij} \mid 0 \leq i < m,\ 0 \leq j < r_i\}$, where $r_i = \lfloor N/m \rfloor + 1$ if $0 \leq i < (N \bmod m)$, and $r_i = \lfloor N/m \rfloor$ otherwise. It may be convenient to imagine that we are looking at the wires from the top, so each of them is seen as a dot. The wires are placed according to the row-major order, i.e. the first r_0 wires are placed in the first row, the next r_1 wires are placed in the second row, etc. For example, for $N = 21$ and $m = 4$ see the arrangement of wires in Figure 12.

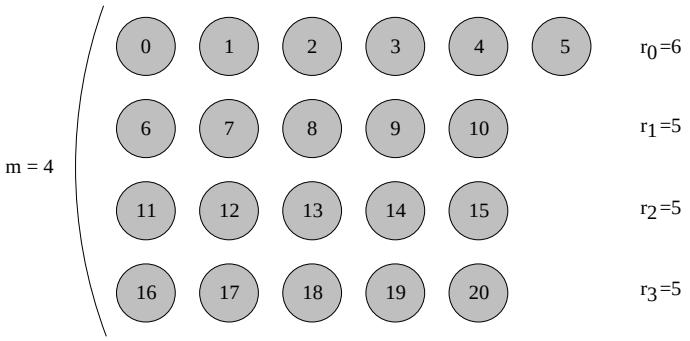

FIG. 12. *The top view of the arrangement of the wires of the network*

Notice that all columns have length m, except the last one in the case when m does not divide N. Then this column has length $N \bmod m$.

For the construction of $G(N, m, G')$ we need a block performing the following permutation π:

DEFINITION 3.6. *For all* $0 \leq i < N$, *let*

$$\pi(i) = (i \bmod m) \cdot \left\lfloor \frac{N}{m} \right\rfloor + \left\lfloor \frac{i}{m} \right\rfloor + \min(i \bmod m, N \bmod m).$$

It is not evident that π is a permutation. We omit an easy but a bit tedious proof of this fact, giving instead an intelligible description of how π transforms the domain. Let σ be the sequence of elements $\{0, \ldots, N-1\}$ taken from the table $A(N, m)$ in column-major order. Then π is the permutation which for any i takes the i^{th} element of σ. For the above example $\sigma = \langle 0, 6, 11, 16, 1, \ldots, 15, 20, 5 \rangle$.

A crucial property of π follows immediately:

PROPOSITION 3.7. *Let* G_π *and* $G_{\pi^{-1}}$ *be switching networks, which perform permutations* π *and* π^{-1}, *respectively. Then*

(i) for any $k \in \{0, .., N-1\}$, G_π *moves the item from the wire* k *to the wire placed at* a_{ij}, *with* $i = k \bmod m$ *and* $j = \lfloor \frac{k}{m} \rfloor$,
(ii) for any $i \in \{0, .., m-1\}$ *and* $j \in \{0, .., r_i\}$, $G_{\pi^{-1}}$ *moves the item from the wire placed at* a_{ij} *to the wire* $i + jm$.

Now we define $G(N, m, G')$. It consists of the following four blocks; each block comprises of a group of layers performing a certain permutation:

- G_π**- distributive block**
  This block performs permutation π on the wires $0, 1, ..., N-1$.
- G_{rec}**- recursive block**
  This block consists of a single copy of G'.
- G_{cor}**- correction block**
  This block consists of r_{m-1} independent subnetworks performing $Shift_{m-(N \bmod m)-1}$ inside columns of A (except for the last one, if m does not divide N). Let Bsh_i denote the network acting on the i^{th} column ($i = 0, \ldots, r_{m-1} - 1$).
- $G_{\pi^{-1}}$**- redistributive block**
  This block performs permutation π^{-1} on all wires $0, 1, ..., N-1$.

Notice that all blocks, except G_{rec}, perform a single permutation, so by Corollary 3.2 each of them consists of at most two layers.

LEMMA 3.8. *For any* N *of the form* 2^l *there exists an* N*-wire switching network of depth* $4\lceil \log \log N \rceil + 5$ *performing shifts by all powers of* 2 *less than* N.

Proof. In order to show this, we assume first, that $N = 2^{2^r}$ for some natural r, and then we extend the construction to all powers of two. The key fact is that if $N = m^2$ (let us remind that m is the greatest power of 2 not exceeding $\sqrt{N}$, so this is the case for $N = 2^{2^r}$), then the permutation π consists of disjoint transpositions (actually, in this case, A is a square matrix and π is its transposition) and therefore both π and π^{-1} can be performed by one-layer networks. Suppose that we can construct an m-wire network H performing shifts by some $a_1, \ldots, a_k$. Now, let the blocks of the network G be as follows: G_π transposes A, G_{rec} consists of m copies of H acting on separate rows of A, G_{cor} performs $Shift_1$ on each column of A independently and $G_{\pi^{-1}} = G_\pi$. Thereby we obtain a network G which can perform any shift a_i and $m \cdot a_i$. The idea of the network performance is visualised by Figures 13 and 14. As one can see in the case of short shifts (see Figure 13) only the blocks G_{rec} and G_{cor} are set active, and to obtain a long shift (see Figure 14) we activate only the blocks G_π, G_{rec} and $G_{\pi^{-1}}$. By this construction we obtain an N-wire network G of $\text{depth}(G) = \text{depth}(H) + 4$ thus its iteration results in the N-wire network H_r of the depth $4(r-1)+3$, for any N, of the form 2^{2^r} ($r \geq 1$).

The above method can be easily extended to any power of two. Let $N = 2^s$ and $s = 2^r + p$, for some $p < 2^r$. Then we chose $m = 2^p$ and put the wires into an array A of size $2^p \times 2^{2^r}$. Afterwards, we repeat the above construction with $H = H_r$ in each row of the recursive block. Since now π is no longer a composition of disjoint transpositions we need 2 layers for π and 2 layers for π^{-1}. Thus we have proved the lemma. □

LEMMA 3.9. *If* $m < N$ *and* G' *is an* N*-wire network performing shifts by* $a_1, a_2, \ldots, a_k$, *then* $G(N, m, G')$

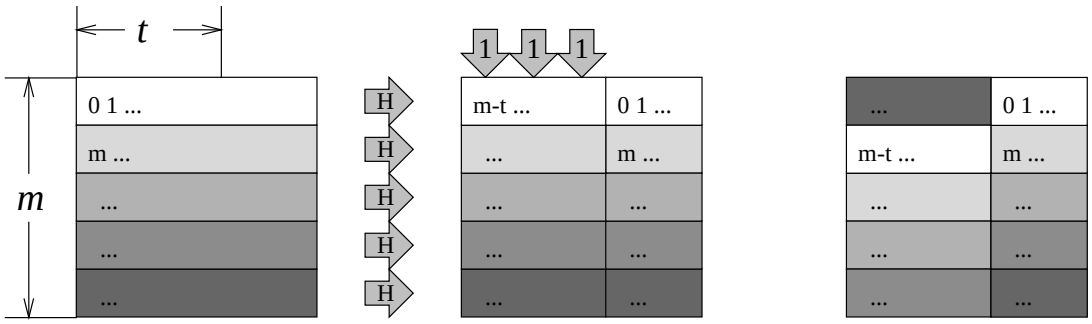

FIG. 13. *Short shifts, i.e.* $t \in \{a_1, a_2, \ldots, a_k\}$.

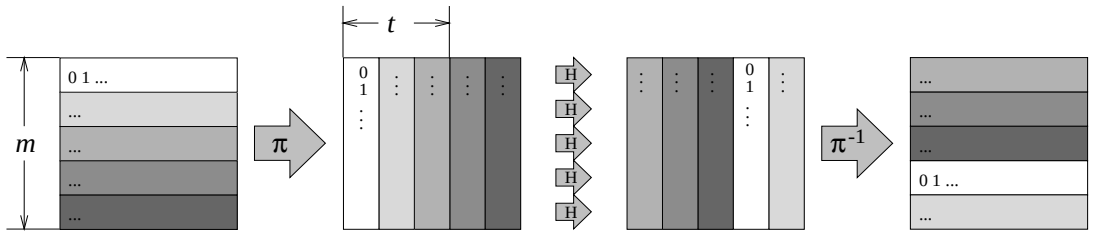

FIG. 14. *Long shifts, i.e. by* $m \cdot t$, *where* $t \in \{a_1, a_2, \ldots, a_k\}$.

can perform shifts by $a_1, a_2, \ldots, a_k$ *and by* $m \cdot a_i$ *for all* $a_i \leq \lfloor \frac{N}{m} \rfloor$.

Proof. It is obvious that if we set all switches in blocks G_π, G_{cor} and $G_{\pi^{-1}}$ inactive, then $G(N, m, G')$ will perform exactly the same shifts as G'. Therefore it can perform shifts by all a_i's and it remains to show how to perform $Shift_{m \cdot t}$ for $m \geq 2$, $t \in \{a_1, a_2, ..., a_k\}$ and $t \leq \lfloor \frac{N}{m} \rfloor$. We set the switches in $G(N, m, G')$ as follows:

- G_π is set to perform π,
- G' is set to perform $Shift_t$,
- inside G_{cor}, networks $Bsh_0, Bsh_1, ..., Bsh_{t-1}$ are set to perform $Shift_{m-(N \bmod m)-1}$ of their columns; $Bsh_t, ..., Bsh_{r_{m-1}-1}$ are inactive,
- $G_{\pi^{-1}}$ is set to perform π^{-1}.

We check that in this state the network performs $Shift_{m \cdot t}$. Let x be the input item on a wire s ($s = 0, \ldots, N-1$), $j = \lfloor \frac{s}{m} \rfloor$ and $i = s \bmod m$. By Proposition 3.7, the distributive block G_π moves x to the wire a_{ij}. We show that the next blocks send it to the wire $(s + tm) \bmod N$. We consider the following cases:

Case 1. $j + t < r_i$.
G' sends x to $a_{i,j+t}$, where it stays until the block $G_{\pi^{-1}}$, because the subnetworks Bsh_l are not active for $l \geq t$. Finally $G_{\pi^{-1}}$ sends it to $i + (j+t)m = i + jm + tm = s + tm$.

Case 2. $j + t \geq r_i$.
Notice that now $s + tm = i + (j+t)m \geq i + r_i m \geq N$ and G' moves the element to the next row and some positions horizontally while G_{Bsh} moves it only vertically.
Let $a_{i'j'}$ be the wire where x is sent. Note that $j' = j + t - r_i$ and $i' = ((i+1) + m - (N \bmod m) - 1) \bmod m$.

Subcase 2a. If $i < N \bmod m$, then
$i' = i + m - (N \bmod m)$ and $r_i = \lfloor \frac{N}{m} \rfloor + 1$. So $i' + j'm$ (the label of the wire where x is sent by

$G_{\pi^{-1}}$) is equal to $i+m-(N \bmod m)+(j+t-\lfloor \frac{N}{m} \rfloor -1)m = i+jm+tm-\lfloor \frac{N}{m} \rfloor m-(N \bmod m) = s+tm-N$.

Subcase 2b. If $i \geq N \bmod m$, then
$i' = i-(N \bmod m)$. Since now $r_i = \lfloor \frac{N}{m} \rfloor$, we have $i'+j'm = i-(N \bmod m)+(j+t-\lfloor \frac{N}{m} \rfloor)m = s+tm-N$.

Thus we have shown that in both cases the contents of the wire s is moved to the wire $(s+tm) \bmod N$. □

To obtain an efficient network accomplishing shifts by powers of 2 we apply the construction of Lemma 3.9 recursively with an appropriate choice of parameters.

LEMMA 3.10. *For any N and $k \leq N$ there exists an N-wire switching network of depth $6 \log\log k + 3$ performing shifts by all powers of 2 less than k. (Notice that the depth of the network is independent of N.)*

Proof. We prove the lemma by induction on k. Using Lemma 3.5, it is easy to construct an N-wire network of depth 3 performing shifts by 1 and 2.

Now assume that $k > 4$ and suppose that for any $k_0 < k$ there is a network $\tilde{G}(k_0)$ of depth $6\lceil \log\log k_0 \rceil + 3$ performing all shifts $\{2^l \mid 2^l \leq k_0\}$. Let $m = \max\{2^l \mid 2^l \leq \sqrt{k}\}$ and let $\tilde{G}(k)$ be the network $G(N, m, \tilde{G}(\lfloor \frac{k}{m} \rfloor))$. We claim that $\tilde{G}(k)$ has the required properties.

By Lemma 3.9, $\tilde{G}(k)$ can be set to perform all shifts from the set

$$\left\{2^l \mid 1 \leq 2^l \leq \left\lfloor \frac{k}{m} \right\rfloor\right\} \cup \left\{m \cdot 2^l \mid 1 \leq 2^l \leq \left\lfloor \frac{k}{m} \right\rfloor\right\} =$$

$$= \left\{2^l \mid 1 \leq 2^l \leq \left\lfloor \frac{k}{m} \right\rfloor \text{ or } m \leq 2^l \leq m \cdot \left\lfloor \frac{k}{m} \right\rfloor\right\}.$$

Directly by the definition of m, we have

$$\sqrt{k}/2 < m \leq \sqrt{k} \quad \text{and} \quad \sqrt{k} \leq \lfloor \tfrac{k}{m} \rfloor < 2\sqrt{k}.$$

Therefore $m \leq \lfloor \frac{k}{m} \rfloor$ and $\tilde{G}(k)$ realizes the shifts by 2^l, for $1 \leq 2^l \leq m \cdot \lfloor \frac{k}{m} \rfloor$.

Since there is no power of 2 between $m \cdot \lfloor \frac{k}{m} \rfloor$ and k, we have proved that $\tilde{G}(k)$ can perform all required shifts.

It follows from the construction that

$$\text{depth}(k) \leq \begin{cases} 3 & \text{for} \quad k \leq 4 \\ 2 + \text{depth}(\lfloor \frac{k}{m} \rfloor) + 2 + 2 & \text{for} \quad k > 4 \end{cases}$$

So in general

$$\text{depth}(k) \leq 6 + \text{depth}(2\sqrt{k}),$$

which can be solved to

$$\text{depth}(k) \leq 6 \log\log(k) + 3.$$

□

3.1.2. Fibonacci shifts. Let us recall the definition of the Fibonacci sequence:

DEFINITION 3.11. $F_0 = 0$, $F_1 = 1$, *and* $F_{k+1} = F_k + F_{k-1}$ for $k \geq 1$.

To apply the construction of Section 3.1.1, we expose the multiplicative nature of Fibonacci numbers.

PROPOSITION 3.12. *Let $r \geq 0$, $a > 0$ be arbitrary natural numbers. Then*

$$F_{r+a} \;=\; F_a \cdot F_{r+1} \;+\; F_{a-1} \cdot F_r.$$

Proof. The proof is a simple induction (see e.g. [30], page 80). □

Below we use the following notation. If S is some set of numbers and t is a number, then $t{\cdot}S = \{t{\cdot}s \mid s \in S\}$ and $S{\downarrow_N} = \{t \mid t \in S \text{ and } t < N\}$. Also for sets of numbers S, T, let $S + T = \{s + t \mid s \in S \text{ and } t \in T\}$.

In Section 3.1.1, we have described a construction that for every N-wire network G, performing shifts S_N, and arbitrary natural $m < N$, yields a network performing shifts $S_N \cup (m \cdot S_N){\downarrow_N}$ at the cost of 6 additional layers. Thus, assuming that we have already constructed a network G' performing shifts $\{F_0, F_1, \ldots, F_a\}$, using the above idea we construct $G(N, F_r, G')$ realizing shifts $(F_r \cdot \{F_0, F_1, \ldots, F_a\}){\downarrow_N}$ and $G(N, F_{r+1}, G')$ realizing shifts $(F_{r+1}{\cdot}\{F_0, F_1, \ldots, F_a\}){\downarrow_N}$. If we put $G(N, F_{r+1}, G')$ after $G(N, F_r, G')$, then together they may realize any shift from the set

$$(F_r \cdot \{F_0, F_1, \ldots, F_a\}){\downarrow_N} + (F_{r+1} \cdot \{F_0, F_1, \ldots, F_a\}){\downarrow_N} \supset$$

$$\supset \{F_1F_{r+1} + F_0F_r, F_2F_{r+1} + F_1F_r, \ldots, F_aF_{r+1} + F_{a-1}F_r\}{\downarrow_N} =$$

$$= \{F_{r+1}, F_{r+2}, \ldots, F_{r+a}\}{\downarrow_N},$$

In order to perform all shifts $\{F_0, F_1, \ldots, F_m\}$, we split this set into a number of smaller intervals

$$\{F_0, F_1, \ldots, F_r\} \cup \{F_{r+1}, F_{r+2}, \ldots, F_{2r}\} \cup \cdots \cup \{F_{\lfloor \frac{m}{r} \rfloor r+1}, \ldots, F_m\}$$

and use the network for $\{F_0, F_1, \ldots, F_r\}$ with six layers extra to perform shifts of each of these intervals.

Now, we describe the above idea more formally and estimate the size of the constructed network. Let a and k be arbitrary integers and let $B(a)$ be a network that may perform shifts by any $x \in \{F_i \mid 0 \leq i \leq a\}$. For any integer $c \geq 2$, let $G^c = G(N, c, B(a))$ and let G^c_π, G^c_{cor}, $G^c_{\pi^{-1}}$ denote respectively its distributive, correction and redistributive blocks. Notice that all G^c's have the same second block, namely $B(a)$. Now, for any sequence $c_1, \ldots, c_k$ of positive integers, we define $G(N, \langle c_1, \ldots, c_k\rangle, B(a))$ to be a network consisting of the following sequence of blocks: $\langle G^{c_1}_\pi,\ G^{c_2}_\pi, \ldots,\ G^{c_k}_\pi,\ B(a),\ G^{c_1}_{cor},\ G^{c_2}_{cor}, \ldots,\ G^{c_k}_{cor}\ , G^{c_1}_{\pi^{-1}},\ G^{c_2}_{\pi^{-1}}, \ldots,\ G^{c_k}_{\pi^{-1}}\rangle$.

LEMMA 3.13. *For any sequence $c_1, \ldots, c_k$ of positive integers, $G(N, \langle c_1, \ldots, c_k\rangle, B(a))$ may perform shift by any element of $\{F_i \mid 0 \leq i \leq a\} \cup \{c_i \cdot F_j \mid 1 \leq i \leq k$ and $0 \leq j \leq a\}{\downarrow_N}$. The depth of $G(N, \langle c_1, \ldots, c_k\rangle, B(a))$ is bounded by $6k + depth(B(a))$.*

Proof. To perform shifts by F_i, $(i = 0, \ldots, a)$, we set all blocks except $B(a)$ inactive, and we set $B(a)$ to perform the required shift.

To perform shift by $c_i \cdot F_j < n$, for some i and j from the permitted intervals, we set $B(a)$ to perform shift by F_j and from among the remaining blocks we activate only $G^{c_i}_\pi$, $G^{c_i}_{cor}$ and $G^{c_i}_{\pi^{-1}}$. Then by Lemma 3.9 we get the required shift.

Because each of the distributive, correction and redistributive blocks is of depth 2,

$$\text{depth}(G(N, \langle c_1, \ldots, c_k\rangle, B(a))) \leq \text{depth}(B(a)) + 6k.$$

□

We define $H(N, \langle k, r\rangle, B(r))$ to be a network consisting of blocks: $G(N, \langle F_{1\cdot r}, \ldots,\ F_{k\cdot r}\rangle,\ B(r))$ and $G(N, \langle F_{1\cdot r+1}, \ldots, F_{k\cdot r+1}\rangle, B(r))$.

LEMMA 3.14. *$H(N, \langle k, r\rangle, B(r))$ performs shifts by $\{F_l \mid 0 \leq l \leq (k+1)r\}{\downarrow_N}$ and its depth is $2(6k +$ $depth(B(r)))$.*

Proof. To perform shift by F_l, for $l \leq r$, we set one copy of $B(r)$ to perform this shift and leave all other blocks inactive.

Let now $r < l \leq (k+1)r$ and $F_l < n$. Let $l = ir + j$, where $1 \leq i \leq k$ and $1 \leq j \leq r$. By Proposition 3.12,

$$F_l = F_{ir+1} \cdot F_j + F_{ir} \cdot F_{j-1}.$$

Since, by Lemma 3.13, block $G(N, \langle F_{1\cdot r}, \ldots, F_{k\cdot r}\rangle, B(r))$ can perform shift by $F_{ir} \cdot F_{j-1}$ and block $G(N, \langle F_{1\cdot r+1}, \ldots, F_{k\cdot r+1}\rangle, B(r))$ can perform shift by $F_{ir+1} \cdot F_j$, the network $H(N, \langle k, r\rangle, B(r))$ can perform shift by F_l.

The estimation of the depth of the network obviously follows from Lemma 3.13. □

LEMMA 3.15. *For any integers n and m, there exists an N-wire switching network of depth $O(2^{2\sqrt{\log m}})$ performing shifts by $\{F_0, F_1, \ldots, F_m\}\downarrow_N$.*

Proof. Let $d, k \in I\!N$ satisfy $d^k \geq m$. We define a sequence of N-wire networks as follows:

- $\tilde{H}(d)$ - a network performing shifts $\{F_0, F_1, \ldots, F_d\}$,
- $\tilde{H}(d^k) = H(N, \langle d-1, d^{k-1}\rangle, \tilde{H}(d^{k-1}))$ for $k > 1$.

By a simple induction on k using Lemma 3.14 one can show that $\tilde{H}(d^k)$ may perform all shifts from the set

$$\{F_l \mid 0 \leq l \leq d \cdot d^{k-1}\}\downarrow_N = \{F_l \mid 0 \leq l \leq d^k\}\downarrow_N$$

for every $k > 0$. The depth of $\tilde{H}(d^k)$ is equal to:

$$\mathrm{depth}(\tilde{H}(d^k)) = \begin{cases} d+1 & \text{for } k = 1 \\ 2(\mathrm{depth}(\tilde{H}(d^{k-1})) + 6(d-1)) & \text{for } k > 1 \end{cases}$$

The first equation comes from Lemma 3.5, the second one follows from Lemma 3.14. Hence

$$\mathrm{depth}(\tilde{H}(d^k)) = 2^{k-1} \cdot (d+1) + (2^k - 2) \cdot 6(d-1).$$

Now, we fix the values of k and m appropriate to be able to perform all required shifts and to minimize the depth of the network. To perform shifts $\{F_0, F_1, \ldots, F_m\}$ we need $d^k \geq m$, so let $k = \lceil \frac{\log m}{\log d} \rceil$ and $d = \lceil 2^{\sqrt{\log m}} \rceil$. Thus $\frac{\log m}{\log d} \leq k < \frac{\log m}{\log d} + 1$ and $2^{\sqrt{\log m}} \leq d < 2^{\sqrt{\log m}} + 1$. So

$$\mathrm{depth}(\tilde{H}(d^k)) \leq 2^{\frac{\log m}{\log d}}(d+1) + (2^{1+\frac{\log m}{\log d}} - 2) \cdot 6(d-1) \leq$$

$$\leq 2^{\frac{\log m}{\sqrt{\log m}}}(d+1) + (2^{1+\frac{\log m}{\sqrt{\log m}}} - 2) \cdot 6(d-1) = 2^{\sqrt{\log m}}(d+1) + (2^{1+\sqrt{\log m}} - 2) \cdot 6(d-1) \leq$$

$$\leq 2^{\sqrt{\log m}}(2^{\sqrt{\log m}} + 2) + 6 \cdot 2^{\sqrt{\log m}}(2^{1+\sqrt{\log m}} - 2) =$$

$$= 2^{2\sqrt{\log m}} + 2 \cdot 2^{\sqrt{\log m}} + 12 \cdot 2^{2\sqrt{\log m}} - 12 \cdot 2^{\sqrt{\log m}} \leq 13 \cdot 2^{2\sqrt{\log m}}.$$

As $d^k \geq m$ we obtain an $O(2^{2\sqrt{\log m}})$ depth network performing shifts $\{F_0, F_1, \ldots, F_m\}$. □

Lemma 3.15 combined with the fact that there are logarithmically many Fibonacci numbers in $\{0, 1, \ldots, N-1\}$ gives us the final result concerning Fibonacci numbers.

3.2. The Proof of Theorem 2.11. We first prove the lower bound of Theorem 2.11. The lower bound uses the same technique as the lower-bound proof of Theorem 2.6. Leighton has mentioned in [31, Exercise 3.8] that the balanced bisection width of the hypercube of N nodes is in $\Omega(N \log\log N / \log N)$. The technical proof of this fact is contained in [27]. As the set of all hypercube shifts [i.e., input node $x_{a_1 \ldots a_{i-1} 0 a_{i+1} \ldots a_d}$ has to be connected to output node $y_{a_1 \ldots a_{i-1} 1 a_{i+1} \ldots a_d}$ for $i = 1, 2, \ldots, d$, $d = \log_2 N$] is a subset of $\mathrm{Pow}\, 2_N$, it follows that $\mathrm{bw}(G(\mathrm{Pow}\, 2_N)) = \Omega(N \log\log N / \log N)$. Let G be a planar, edge-disjoint k-permutation network for $\mathrm{Pow}\, 2_N$. Using the same notation as in Theorem 2.6, we have

$$\text{size}(G) \;\geq\; (\text{bw}(G(\text{Pow}\,2_N)))^2 \;/\; (c \cdot k)^2$$

for some constant c (independent of size(G)) [cf. the proof of Theorem 2.6]. It follows that Plsize-ed$_k(\text{Pow}\,2_N) = \Omega(N^2 \cdot (\log\log N)^2/(\log N)^2)$. This completes the proof of the lower bound of Theorem 2.11. □

To prove the upper bound of Theorem 2.11, we proceed in the following two steps:
Let $Gr(a,b)$ denote the $(a \times b)$-grid.

1. First, we show that, for m being a power of two, m input nodes and m output nodes can be laid out on the row $\{[1,i] \mid 1 \leq i \leq 2m\}$ in $Gr(\left\lfloor 67 \cdot m \cdot \frac{\log_2 \log_2 m+1}{\log_2 m} \right\rfloor + 1, 2m)$ such that all shifts of power 2 can be realized. The layout of the input and output nodes also satisfies the condition that the input nodes are mapped to the nodes $\{[1,2i-1] \mid 1 \leq i \leq m\}$, the output nodes are mapped to the nodes $\{[1,2i] \mid 1 \leq i \leq m\}$, and if the input node x_v, $v \in \{0,1\}^l$, is mapped to $[1,2i-1]$, $1 \leq i \leq m$, then the output node y_v is mapped to $[1,2i]$, and vice versa.
2. Then, we demonstrate how this result can be used to show that
$$\text{Plsize-ed}_k(\text{Pow}\,2_N) \;=\; O(N^2 \cdot (\log\log N)^2/(\log N)^2)\,.$$

We start by verifying the first claim.

LEMMA 3.16. *Let $m = 2^l$ for some $l \in \mathbb{N}$. Then m input nodes and m output nodes can be laid out on the row $\{[1,i] \mid 1 \leq i \leq 2m\}$ in $Gr(\left\lfloor 67 \cdot m \cdot \frac{\log_2 \log_2 m+1}{\log_2 m} \right\rfloor + 1, 2m)$ such that all shifts of power 2 can be realized. The layout of the input and output nodes also satisfies the condition that the input nodes are mapped to the nodes $\{[1,2i-1] \mid 1 \leq i \leq m\}$, the output nodes are mapped to the nodes $\{[1,2i] \mid 1 \leq i \leq m\}$, and if the input node x_v, $v \in \{0,1\}^l$, is mapped to $[1,2i-1]$, $1 \leq i \leq m$, then the output node y_v is mapped to $[1,2i]$, and vice versa.*

Proof. We first define the layout of the input nodes $\{x_v \mid v \in \{0,1\}^l\}$ and the output nodes $\{y_v \mid v \in \{0,1\}^l\}$ on the row $\{[1,i] \mid 1 \leq i \leq 2m\}$ (see Figure 15).

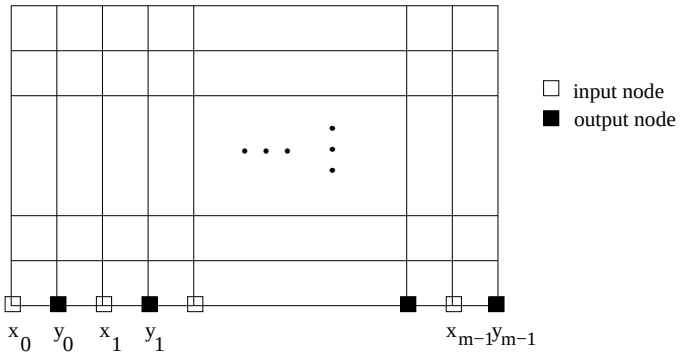

FIG. 15. *The layout of the input and output nodes in the grid*

Let $v = v_{l-1}v_{l-2}\ldots v_0 \in \{0,1\}^l$. Let $b \in \mathbb{N}$ such that $l/2 \leq b \cdot 2^b \leq 2l$. v is divided into $\lfloor l/b \rfloor$ blocks of length b and 1 block of length l mod b if $b \nmid l$, namely

$$\begin{aligned} v \;=\;& B_{\lceil l/b \rceil -1}(v) B_{\lceil l/b \rceil -2}(v) \ldots B_0(v) \\ & \text{where} \;\; B_j(v) = v_{(j+1)b-1} v_{(j+1)b-2} \ldots v_{jb} \in \{0,1\}^b \;\; \text{for } 0 \leq j \leq \lfloor l/b \rfloor - 1 \\ & \text{and} \;\; B_{\lceil l/b \rceil -1}(v) = v_{l-1} v_{l-2} \ldots v_{\lfloor l/b \rfloor \cdot b} \in \{0,1\}^{l \bmod b} \;\; \text{if } b \nmid l\,. \end{aligned}$$

[If v is clear from the context, then $B_j(v)$ is simply referred to as B_j.]

A **blockpattern** is a bitstring $P \in \{0,1\}^b$. For $0 \leq i \leq 2^b - 1$, let bin$(i) \in \{0,1\}^b$ be the binary representation of i. Then, $P_i = \text{bin}(i)$ is called the ***i*-th blockpattern**. Let $v \in \{0,1\}^l$, $0 \leq i \leq 2^b - 1$. If $B_j(v) \neq P_k$ for all $0 \leq j \leq \lfloor l/b \rfloor - 1$, $0 \leq k < i$, and $\exists\, 0 \leq j \leq \lfloor l/b \rfloor - 1 :\; B_j(v) = P_i$, then P_i is called the **leading blockpattern** of v. Let $v \in \{0,1\}^l$. Let P_i, $0 \leq i \leq 2^b - 1$, be the leading blockpattern of v. Then, bd$(n_{\lfloor l/b \rfloor -1}(v), n_{\lfloor l/b \rfloor -2}(v), \ldots, n_0(v)) \in \{0,1\}^{\lfloor l/b \rfloor}$, where

$$n_j(v) \;=\; \begin{cases} 1 & \text{if } \; B_j(v) = P_i \\ 0 & \text{else} \end{cases}$$

for $0 \leq j \leq \lfloor l/b \rfloor - 1$, is called the **block diagram** of v.

For $0 \leq i \leq 2^b - 1$, $1 \leq k \leq \lfloor l/b \rfloor$, let

$$G_i = \{v \in \{0,1\}^l \mid P_i \text{ is the leading blockpattern of } v\},$$
$$G_{i,k} = \{v \in \{0,1\}^l \mid v \in G_i \text{ and there are exactly } k \text{ indices } j_1, j_2, \ldots, j_k \in \{0,1,\ldots,\lfloor l/b \rfloor - 1\} \text{ such that } B_{j_i}(v) = P_i\}.$$

Now, the input nodes $\{x_v \mid v \in \{0,1\}^l\}$ and the output nodes $\{y_v \mid v \in \{0,1\}^l\}$ are laid out on the row $\{[1,i] \mid 1 \leq i \leq 2m\}$ such that:

a) The input nodes are mapped to the nodes $\{[1, 2i-1] \mid 1 \leq i \leq m\}$, the output nodes are mapped to the nodes $\{[1,2i] \mid 1 \leq i \leq m\}$. If the input node x_v, $v \in \{0,1\}^l$, is mapped to $[1, 2i-1]$, $1 \leq i \leq m$, then the output node y_v is mapped to $[1, 2i]$, and vice versa.
b) The node x_{v_1}, $v_1 \in G_i$, is placed before the node x_{v_2}, $v_2 \in G_{i+1}$, for $0 \leq i \leq 2^b - 2$.
c) For $0 \leq i \leq 2^b - 1$:
The node x_{v_1}, $v_1 \in G_{i,k}$, is placed before the node x_{v_2}, $v_2 \in G_{i,k+1}$, for $1 \leq k \leq \lfloor l/b \rfloor - 1$.
d) For $0 \leq i \leq 2^b - 1$, $1 \leq k \leq \lfloor l/b \rfloor$:
The node x_{v_1}, $v_1 \in G_{i,k}$, is placed before the node x_{v_2}, $v_2 \in G_{i,k}$, if $\mathrm{bd}(v_1) < \mathrm{bd}(v_2)$ according to the lexicographical order on $\{0,1\}^{\lfloor l/b \rfloor}$ [i.e., $\mathrm{bin}^{-1}(\mathrm{bd}(v_1)) < \mathrm{bin}^{-1}(\mathrm{bd}(v_2))$].

Let $f : \{0,1\}^l \to \{1,2,\ldots,m\}$ such that $2f(v) - 1$ is the final layout position of the input node x_v [and $2f(v)$ is the final layout position of the output node y_v] on the row $\{[1,i] \mid 1 \leq i \leq 2m\}$ for all $v \in \{0,1\}^l$.

Let $0 \leq T \leq \log_2 m$. We now show how the 2^T-shift can be realized.

Communication between vertices is established by a set of canonical paths. We say that vertices $v = [1,j]$ and $w = [1,k]$ **communicate via the track** r, if the information is passed first vertically from $[1,j]$ to $[r,j]$, then horizontally along the r-th row to $[r,k]$, and then finally vertically from $[r,k]$ to $[1,k]$ (see Figure 16).

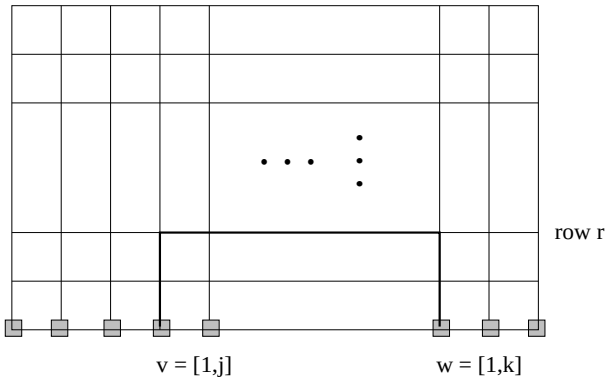

FIG. 16. *Communication via track r*

Consider the graph $G^T = (V, E)$, $V = \{0,1\}^l$ and

$$(v,w) \in E \quad \text{if} \quad (\mathrm{bin}^{-1}(v) + 2^T) \bmod m = \mathrm{bin}^{-1}(w).$$

For $0 \leq i \leq m$, consider the **cut** $C_i = (V_1, V_2)$ of G^T (i.e., $V_1, V_2 \subseteq V$ such that $V = V_1 \dot{\cup} V_2$), where

$$V_1 := \{v \in \{0,1\}^l \mid f(v) \leq i\},$$
$$V_2 := \{v \in \{0,1\}^l \mid f(v) \geq i+1\}.$$

An edge **crossing** the cut (V_1, V_2) [or, a C**-cutting edge**, for short] is an edge $(v,w) \in E$ such that $v \in V_1$, $w \in V_2$.

OBSERVATION 3.17. *The 2^T-shift can be realized using $\max_{0 \leq i \leq m} |\{e \in E \mid e \text{ is crossing the cut } C_i\}|$ tracks.*

The proof of Observation 3.17. We establish the communication paths from left to right in row $\{[1,i] \mid 1 \leq i \leq 2m\}$. A node $[1,i]$ which wants to communicate with the node $[1,j]$ chooses the first "free" track

(i.e., the first track which is not already being used). This completes the proof of Observation 3.17.

The proof of Lemma 3.16 continued. Hence, it suffices to show:

CLAIM 3.18. $|\{e \in E \mid e \text{ is crossing the cut } C_i\}| \leq 67 \cdot m \cdot \frac{\log_2 \log_2 m+1}{\log_2 m}$ *for all* $0 \leq i \leq m$.

The proof of Claim 3.18. Consider a cut $C \in \{C_0, C_1, \ldots, C_m\}$. Let $E_C \subseteq E$ denote the set of C-cutting edges. An edge $(v, w) \in E$ is called **internal** to a set $V' \subseteq V$, if $v, w \in V'$. Let

$$\begin{aligned} E_1 &= \{e \in E_C \mid e \text{ is internal to } G_{i,k} \text{ for some } i \in \{0, 1, \ldots, 2^b - 1\}, \\ &\qquad k \in \{1, 2, \ldots, \lfloor l/b \rfloor\}\}, \\ E_2 &= \{e \in E_C \mid e \notin E_1, e \text{ is internal to } G_i \text{ for some } i \in \{0, 1, \ldots, 2^b - 1\}\}, \\ E_3 &= \{e \in E_C \mid e \notin E_1 \cup E_2\}. \end{aligned}$$

For $1 \leq i \leq 3$, let $g_i := |E_i|$. Then, $E_C = E_1 \dot\cup E_2 \dot\cup E_3$ and $|E_C| = g_1 + g_2 + g_3$ (see Figure 17).

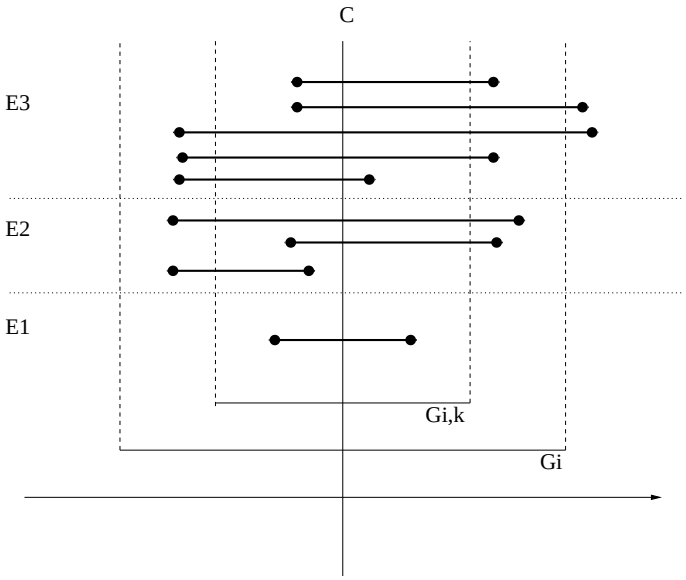

FIG. 17. *C-cutting edges*

We show that $g_1 + g_2 + g_3 \leq 67 \cdot m \cdot \frac{\log_2 \log_2 m+1}{\log_2 m}$. Let t be the index of the block containing the bit with the index T, i.e., $t = \lfloor T/b \rfloor$.

1.) Estimation of g_1:
According to the order f, there is at most one $G_{i,k}$ which is cut by C. If no $G_{i,k}$ is cut, then $g_1 = 0$ and nothing needs to be shown. Let $G_{i,k}$, $i \in \{0, 1, \ldots, 2^b - 1\}$, $k \in \{1, 2, \ldots, \lfloor l/b \rfloor\}$, be cut by C. Let $(v, w) \in E$ be a C-cutting edge.

1a.) $\mathrm{bd}(v) = \mathrm{bd}(w)$:
According to the order f, there is at most one block diagram which occurs left as well as right of C (i.e., all other block diagrams occur only left or right of C). Hence, all the edges (v, w) cutting C and fulfilling condition 1a.) have a fixed given block diagram. This means that (at least) one block (of length p) is fixed, and there are at most 2^{l-b} of these nodes. Hence, there are at most 2^{l-b-1} C-cutting edges of this type (both endnodes in $G_{i,k}$).

1b.) $\mathrm{bd}(v) \neq \mathrm{bd}(w)$:

i) Something is changed in B_r, $r > t + 1$:
Then, the block B_{t+1} in the target node w is the zero block 0^b. Hence, only 2^{l-b} different target nodes are possible.

ii) A leading pattern is changed in B_{t+1}
(i.e., v has a leading pattern in B_{t+1} and w does not, or vice versa):
The block B_{t+1} consists of the leading pattern either in v or w. There are at most $2 \cdot 2^{l-b}$ edges of this type.

iii) Like ii) with block B_t instead of B_{t+1}.

Overall, it follows that $g_1 \leq 5.5 \cdot 2^{l-b}$. (Using a more careful analysis, it can be shown that $g_1 \leq 2.5 \cdot 2^{l-b}$.)

2.) Estimation of g_2:

We first estimate the number of edges which are internal to a group G_i and which cross the cut $(G_{i,k}, G_{i,k+1})$ for given $i \in \{0, 1, \ldots, 2^b - 1\}$, $k \in \{1, 2, \ldots, \lfloor l/b \rfloor - 1\}$.

i) The number of edges which change at least 3 blocks is at most 2^{l-b} [B_{t+1} is the zero block, cf. 1b) i)].

ii) At most two blocks are changed, namely B_t and/or B_{t+1}.
The number of blocks with the leading pattern P_i is changed, i.e., for v or w the leading pattern must occur in B_t or B_{t+1}. There are at most $4 \cdot 2^{l-b}$ edges of this type [cf. 1b) ii) and iii)].

Overall, it follows that there are at most $5 \cdot 2^{l-b}$ edges crossing the cut $(G_{i,k}, G_{i,k+1})$.

We now estimate g_2. According to the order f, there is at most one $G_{i,k}$ which is cut by C. If no $G_{i,k}$ is cut, C is a cut $(G_{i,k}, G_{i,k+1})$ for some $i \in \{0, 1, \ldots, 2^b - 1\}$, $k \in \{1, 2, \ldots, \lfloor l/b \rfloor - 1\}$. If $G_{i,1}$, $i \in \{0, 1, \ldots, 2^b - 1\}$, is cut by C, then all C-cutting edges which are internal to G_i and not internal to $G_{i,1}$ cross the cut $(G_{i,1}, G_{i,2})$. If $G_{i,\lfloor l/b \rfloor}$, $i \in \{0, 1, \ldots, 2^b - 1\}$, is cut by C, then all C-cutting edges which are internal to G_i and not internal to $G_{i,\lfloor l/b \rfloor}$ cross the cut $(G_{i,\lfloor l/b \rfloor - 1}, G_{i,\lfloor l/b \rfloor})$. If $G_{i,k}$, $i \in \{0, 1, \ldots, 2^b - 1\}$, $k \in \{2, 3, \ldots, \lfloor l/b \rfloor - 1\}$, is cut by C, then all C-cutting edges which are internal to G_i and not internal to $G_{i,k}$ cross the cut $(G_{i,k-1}, G_{i,k})$ or $(G_{i,k}, G_{i,k+1})$. In any case, it follows that

$$\begin{aligned} g_2 &\leq 2 \cdot \max_{i,k} |\{e \in E \mid e \text{ is internal to } G_i \text{ and crosses the cut } (G_{i,k}, G_{i,k+1})\}| \\ &\leq 10 \cdot 2^{l-b}. \end{aligned}$$

3.) Estimation of g_3:

We first estimate the number of edges which cross the cut (G_{j-1}, G_j) for given $j \in \{1, 2, \ldots, 2^b - 1\}$.

i) The number of edges which change at least 3 blocks is at most 2^{l-b} [B_{t+1} is the zero block, cf. 1b) i)].

ii) At most two blocks are changed, namely B_t and/or B_{t+1}:
Consider an edge e "starting" in G_i, $i < j$, and "ending" in G_k, $k \geq j$, i.e., $e = (v, w)$ where $v \in G_i$ and $w \in G_k$. For e, it holds that in the "left" node v the leading pattern P_i must occur in B_t or B_{t+1} (or both). As in the "right" node w only patterns P_k, $k \geq j$, occur, there are at most

$$4 \cdot (2^b - j)^{\lfloor l/b \rfloor - 1} \cdot 2^{l \bmod b}$$

such edges.
On the whole, the starting group G_i can be one of the j groups $G_0, G_1, \ldots, G_{j-1}$. Therefore, the overall number of edges crossing the cut (G_{j-1}, G_j) and changing at most two blocks can be bounded above by

$$n(j) := j \cdot 4 \cdot (2^b - j)^{\lfloor l/b \rfloor - 1} \cdot 2^{l \bmod b}.$$

Using standard analytical methods, it can be shown that the function

$$f : \{x \in \mathbb{R} \mid 0 \leq x \leq 2^b\} \to \mathbb{R}, \quad x \mapsto x \cdot (2^b - x)^{\lfloor l/b \rfloor - 1}$$

is maximized for $x = \frac{2^b}{\lfloor l/b \rfloor}$. As $b \cdot 2^b \leq 2l$ and $b \in \mathbb{N}$, $2^{b-1} \leq \lfloor l/b \rfloor$ holds, and we obtain

$$\begin{aligned} f(j) &\leq f\left(\tfrac{2^b}{\lfloor l/b \rfloor}\right) = \tfrac{2^b}{\lfloor l/b \rfloor} \cdot \left(2^b - \tfrac{2^b}{\lfloor l/b \rfloor}\right)^{\lfloor l/b \rfloor - 1} \\ &\leq 2 \cdot (2^b)^{\lfloor l/b \rfloor - 1} = 2 \cdot 2^{\lfloor l/b \rfloor \cdot b - b} \end{aligned}$$

and

$$n(j) = 4 \cdot f(j) \cdot 2^{l \bmod b} \leq 8 \cdot 2^{l-b}.$$

Overall, it follows that there are at most $9 \cdot 2^{l-b}$ edges crossing the cut (G_{j-1}, G_j).

We now estimate g_3. According to the order f, there is at most one G_j which is cut by C. If no G_j is cut, C is a cut (G_{j-1}, G_j) for some $j \in \{1, 2, \ldots, 2^b - 1\}$. If G_0 is cut by C, then all C-cutting edges which are not internal to G_0 cross the cut (G_0, G_1). If G_{2^b-1} is cut by C, then all C-cutting edges which are not internal to G_{2^b-1} cross the cut (G_{2^b-2}, G_{2^b-1}). If G_j, $j \in \{1, 2, \ldots, 2^b - 2\}$, is cut by C, then all C-cutting edges which are not internal to G_j cross the

cut (G_{j-1}, G_j) or (G_j, G_{j+1}). In any case, it follows that

$$g_3 \leq 2 \cdot \max_j |\{e \in E \mid e \text{ crosses the cut } (G_{j-1}, G_j)\}| \leq 18 \cdot 2^{l-b}.$$

Overall, as $l/2 \leq b \cdot 2^b \leq 2l$, it follows that

$$g_1 + g_2 + g_3 \leq 33.5 \cdot 2^{l-b} \leq 33.5 \cdot 2^l \cdot \frac{2b}{l} \leq 67 \cdot 2^l \cdot \frac{\log_2 l + 1}{l} \leq 67 \cdot m \cdot \frac{\log_2 \log_2 m + 1}{\log_2 m}.$$

□

Now, we will see how the result from Lemma 3.16 can be used to show the upper of Theorem 2.11.

LEMMA 3.19.

$$\textit{Plsize-ed}_k(\textit{Pow2}_N) = O(N^2 \cdot (\log\log N)^2/(\log N)^2).$$

Proof. We first consider the case that N is a power of two. Let $N = 2^d$ for some $d \in \mathbb{N}$. Let $0 \leq l_1, l_2 \leq d$ such that $l_1 + l_2 = d$. For $i \in \{1, 2\}$, let $m_i = 2^{l_i}$, $z_i = \left\lfloor 67 \cdot m_i \cdot \frac{\log_2 \log_2 m_i + 1}{\log_2 m_i} \right\rfloor + 2$. We consider the grid $Gr(a, b)$ where $a = m_2 \cdot z_1$ and $b = m_1 \cdot z_2$. We partition $Gr(a, b)$ into $m_2 \cdot m_1$ subgrids Gr_{ij}, $1 \leq i \leq m_2$, $1 \leq j \leq m_1$, of size $z_1 \times z_2$ as displayed in Figure 18.

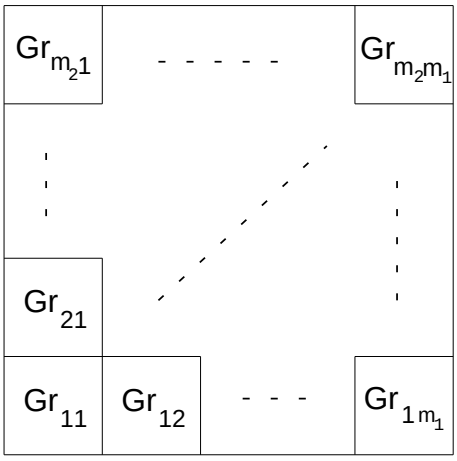

FIG. 18. *The partition of $Gr(a, b)$ into subgrids*

[More precisely, $V(Gr_{ij}) = \{[r, s] \mid (i-1)z_1 + 1 \leq r \leq iz_1,\ (j-1)z_2 + 1 \leq s \leq jz_2\}$.]

We now define the position $\text{pos}(x_v)$ and $\text{pos}(y_v)$ of the input node x_v and the output node y_v for $v \in \{0,1\}^d$. Let $v = v_{d-1} \ldots v_0 \in \{0,1\}^d$. For $i \in \{0,1\}$, let $f_i : \{0,1\}^{l_i} \to \{1, 2, \ldots, m_i\}$ be the function (describing the final position of the input/output nodes) from the construction in Lemma 3.16 applied to $m = m_i = 2^{l_i}$. Then,

$$\text{pos}(x_v) := [(f_2(v_{d-1} \ldots v_{l_1}) - 1)z_1 + 2, (f_1(v_{l_1-1} \ldots v_0) - 1)z_2 + 1],$$
$$\text{pos}(y_v) := [(f_2(v_{d-1} \ldots v_{l_1}) - 1)z_1 + 1, (f_1(v_{l_1-1} \ldots v_0) - 1)z_2 + 2],$$

i.e., the position of x_v is chosen as the left node in the second row and the position of y_v is chosen as the bottom node of the second column of the corresponding subgrid.

It remains to show how the 2^T-shift can be realized in $Gr(a, b)$ for $0 \leq T \leq d$.

The crucial observation will be that a 2^T-shift on $v = v_{d-1} \ldots v_0$ can be decomposed into a 2^{T_1}-shift on $v_{l_1-1} \ldots v_0$ and a 2^{T_2}-shift on $v_{d-1} \ldots v_{l_1}$. The idea then is to realize the 2^{T_1}-shift on $v_{l_1-1} \ldots v_0$ in horizontal direction in $Gr(a, b)$ and the 2^{T_2}-shift on $v_{d-1} \ldots v_{l_1}$ in vertical direction.

Let us first see how, for fixed $v_{d-1} \ldots v_{l_1} \in \{0,1\}^{l_2}$, a 2^{T_1}-shift (modulo 2^{l_1}) on $\{v_{d-1} \ldots v_0 \mid v_{l_1-1} \ldots v_0 \in \{0,1\}^{l_1}\}$ can be realized in horizontal direction in $Gr(a, b)$ and how, for fixed $v_{l_1-1} \ldots v_0 \in \{0,1\}^{l_1}$, a

2^{T_2}-shift (modulo 2^{l_2}) on $\{v_{d-1}\dots v_0 \mid v_{d-1}\dots v_{l_1} \in \{0,1\}^{l_2}\}$ can be realized in vertical direction in $Gr(a,b)$.

Let $0 \le T_1 \le l_1$. Let $v_{d-1}\dots v_{l_1} \in \{0,1\}^{l_2}$. For realizing the 2^{T_1}-shift (modulo 2^{l_1}) between $\{x_{v_{d-1}\dots v_0} \mid v_{l_1-1}\dots v_0 \in \{0,1\}^{l_1}\}$ and $\{y_{v_{d-1}\dots v_0} \mid v_{l_1-1}\dots v_0 \in \{0,1\}^{l_1}\}$, we use the subgrids G_{ij} in which these input and output nodes are placed, i.e., we use the subgrid G_i consisting of $Gr_{i1}, Gr_{i2}, \dots, Gr_{im_1}$ for $i = f_2(v_{d-1}\dots v_{l_1})$. More precisely, we identify the $2m_1$ columns of G_i containing an input or output node with a $(z_1 \times 2m_1)$-grid $\tilde{G}_i$. Lemma 3.16 asserts that the 2^{T_1}-shift (modulo 2^{l_1}) can be realized in $\tilde{G}_i$ (because we have $z_1 - 2 = \left\lfloor 67 \cdot m_1 \cdot \frac{\log_2 \log_2 m_1 + 1}{\log_2 m_1} \right\rfloor$ horizontal tracks available). This is also true for G_i by routing horizontal communications in $\tilde{G}_i$ along the according row in G_i.

Let $0 \le T_2 \le l_2$. Let $v_{l_1-1}\dots v_0 \in \{0,1\}^{l_1}$. For realizing the 2^{T_2}-shift (modulo 2^{l_2}) between $\{x_{v_{d-1}\dots v_0} \mid v_{d-1}\dots v_{l_1} \in \{0,1\}^{l_2}\}$ and $\{y_{v_{d-1}\dots v_0} \mid v_{d-1}\dots v_{l_1} \in \{0,1\}^{l_2}\}$, we use the subgrids G_{ij} in which these input and output nodes are placed, i.e., we use the subgrid G_j consisting of $Gr_{1j}, Gr_{2j}, \dots, Gr_{m_2 j}$ for $j = f_1(v_{l_1-1}\dots v_0)$. This time, we have $z_2 - 2 = \left\lfloor 67 \cdot m_2 \cdot \frac{\log_2 \log_2 m_2 + 1}{\log_2 m_2} \right\rfloor$ vertical tracks available. Hence, we can realize the 2^{T_2}-shift (modulo 2^{l_2}) vertically (by using the same argumentation as in the horizontal case vertically now).

Now, we proceed to implement the 2^T-shift (modulo 2^d) on $Gr(a,b)$. Let $0 \le T \le d$. Let $v = v_{d-1}\dots v_0 \in \{0,1\}^d$. Let $v^l := v_{d-1}\dots v_{l_1}$, $v^r := v_{l_1-1}\dots v_0$. We specify the communication path $P(v)$ from x_v to y_w, $w = w_{d-1}\dots w_0 = \mathrm{bin}((\mathrm{bin}^{-1}(v) + 2^T) \bmod (2^d))$. Let $w^l := w_{d-1}\dots w_{l_1}$, $w^r := w_{l_1-1}\dots w_0$.

a) $\underline{T \ge l_1}$:
Let $T_2 := T - l_1$. Then $w^l = \mathrm{bin}((\mathrm{bin}^{-1}(v^l) + 2^{T_2}) \bmod (2^{l_2}))$, $w^r = v^r$. Consider the realization of the 2^{T_2}-shift (modulo 2^{l_2}) on $\{v_{d-1}\dots v_0 \mid v_{d-1}\dots v_{l_1} \in \{0,1\}^{l_2}\}$ from above. Then, $P(v)$ is taken as the communication path between x_v and y_w in that realization.

b) $\underline{T < l_1 \text{ and } \mathrm{bin}^{-1}(v^r) + 2^T < 2^{l_1}}$:
Let $T_1 := T$. Then $w^l = v^l$, $w^r = \mathrm{bin}((\mathrm{bin}^{-1}(v^r) + 2^{T_1}) \bmod (2^{l_1}))$. Consider the realization of the 2^{T_1}-shift (modulo 2^{l_1}) on $\{v_{d-1}\dots v_0 \mid v_{l_1-1}\dots v_0 \in \{0,1\}^{l_1}\}$ from above. Then, $P(v)$ is taken as the communication path between x_v and y_w in that realization.

c) $\underline{T < l_1 \text{ and } \mathrm{bin}^{-1}(v^r) + 2^T \ge 2^{l_1}}$:
Let $T_1 := T$, $T_2 := 0$. Then $w^l = \mathrm{bin}((\mathrm{bin}^{-1}(v^l) + 2^{T_2}) \bmod (2^{l_2}))$, $w^r = \mathrm{bin}((\mathrm{bin}^{-1}(v^r) + 2^{T_1}) \bmod (2^{l_1}))$. Consider the realization R_1 of the 2^{T_1}-shift (modulo 2^{l_1}) on $\{v_{d-1}\dots v_0 \mid v_{l_1-1}\dots v_0 \in \{0,1\}^{l_1}\}$ and the realization R_2 of the 2^{T_2}-shift (modulo 2^{l_2}) on $\{v_{d-1}\dots v_0 \mid v_{d-1}\dots v_{l_1} \in \{0,1\}^{l_2}\}$ from above. x_v communicates with $y_{v^l w^r}$ in R_1 via the horizontal track r_1. $x_{v^l w^r}$ communicates with y_w in R_2 via the vertical track r_2. Now, x_v can communicate directly with y_w by first using the horizontal track r_1 and then using the vertical track r_2 (i.e., the information is first passed vertically from x_v to r_1, then horizontally along r_1 to r_2, then vertically along r_2 to the row containing y_w, and finally horizontally to y_w).

Hence, we have seen how the 2^T-shift can be realized in $Gr(a,b)$ for $0 \le T \le d$. Choosing $l_1 = \lfloor d/2 \rfloor$, $l_2 = \lceil d/2 \rceil$ yields the desired result

$$\text{Plsize-ed}_k(\text{Pow}\,2_N) = O(N^2 \cdot (\log\log N)^2/(\log N)^2)\,,$$

for N being a power of two. If N is not a power of two, a similar case distinction as above yields the desired result. This completes the proof of Theorem 2.11. □

4. Conclusion. In this paper, we presented asymptotically optimal planar and non-planar permutation networks for shifts of power 2. A new simulation of hypercube algorithms as well as an essential improvement of gossiping in two-way vertex-disjoint paths mode in bounded-degree networks follow. For the set of Fibonacci shifts, Fib_N, which are crucial for the design of efficient gossip algorithms in one-way vertex-disjoint paths mode, we were not able to prove the optimality of our constructions. Thus, the main open problem left is to determine Size-ed$_k(Fib_N)$ and Plsize-ed$_k(Fib_N)$ exactly.

It would be of independent interest to find a difficult set of shifts, i.e. requiring the depth of a network more than logarithmic in the number of performed shifts.

Many networks considered have a regular structure that enables to use them in periodical mode. It means

that a network (presumably of a small depth) is used several times, each time being set independently of previous rounds and fed with the output of the previous run. A constant depth switching network performing all permutations, if run $O(\log N)$ times, is presented in [42]. The investigation of power-two and Fibonacci shifts resulted in periodic constructions of constant depth and $O(\log\log N)$ and respectively $O(2^{2\sqrt{\log\log N}})$ numbers of rounds [21].

Appendix. Summary of the Argument from [43]. Let $P_{N,k,m}$ denote the set of all edge-disjoint k-permutation networks of N input nodes, N output nodes and at most m inner nodes (i.e. neither input nor output nodes). Let Π_N denote the set of all permutations of N elements. Let Π_N^q denote the set of all q-tuples of elements from Π_N. A tuple $(\pi_1, \pi_2, \ldots, \pi_q)$ from Π_N^q is realized by a network G from $P_{N,k,m}$ (by definition) if and only if every π_i is realizable by G (i.e. there exist N edge-disjoint paths in G realizing π_i).

PROPOSITION 4.1 ([43]). *Let $k = O(1)$, $m = Nq/c$, where $c = 4k\log k$ is a constant depending on k. Let also $q = o(\log N)$. Then only an exponentially (in N) small fraction of Π_N^q is realizable by a network from $P_{N,k,m}$.*

Idea of the Proof. For an inner vertex v of a permutation network, let an *in-egde/out-edge assignment* be a matching between the incoming and outgoing edges of the vertex v (while considering an arbitrary orientation of the edges). A network with q *switch presettings* is a permutation network in which for every inner vertex v a tuple of q in-egde/out-edge assignments is given. Let $P'_{N,k,m,q}$ denote the set of all networks with q switch presettings, where the underlying network is from $P_{N,k,m}$.

Some elements from $P'_{N,k,m,q}$ realize in a natural way an element from Π_N^q. Other elements from $P'_{N,k,m,q}$ potentially do not realize any permutation (e.g. paths in the network may stop in the middle of the network). In any case, at most $|P'_{N,k,m,q}|$ many tuples from Π_N^q may be realized by a network from $P_{N,k,m}$.

We have roughly $|P'_{N,k,m,q}| \leq (N+m)^{k(N+m)} \cdot (k^{2k})^{mq}$, and $|\Pi_N^q| \geq N^{qN}$. The first term is exponentially smaller than the second term. □

Acknowledgements. The authors would like to thank Ondrej Sýkora for his careful reading and his helpful comments on parts of the manuscript. Special thanks go to Mirosław Kutyłowski for his contribution in preparing the final version of part of this paper. The authors also wish to thank the anonymous referees for their helpful suggestions.

REFERENCES

[1] S. BANDYOPADHYAY (IN COOPERATION WITH R. KLASING), *Dissemination of Information in Optical Networks: From Technology to Algorithms*, Springer Monograph, Springer-Verlag, 2008.
[2] X. BAO, F.K. HWANG, AND Q. LI, *Rearrangeability of bit permutation networks*, Theoretical Computer Science, vol. 352, nos. 1-3, pp. 197-214, 2006.
[3] V. BENEŠ, *Permutation groups, complexes, and rearrangeable multistage connecting networks*, Bell System Technical Journal, vol. 43, pp. 1619-1640, 1964.
[4] V. BENEŠ, *Mathematical Theory of Connecting Networks and Telephone Traffic*. Academic Press, New York, NY, 1965.
[5] H. ÇAM, *Balanced Permutations and Multistage Interconnection Networks*, International Journal of Computer Mathematics, vol. 73, no. 1, pp. 125-137, Nov. 1999.
[6] H. ÇAM, *Rearrangeability of $(2n-1)$-Stage Shuffle-Exchange Networks*, SIAM Journal on Computing, vol. 32, no. 3, pp. 557-585, 2003.
[7] H. ÇAM AND J.A.B. FORTES, *Frames: a simple characterization of permutations realized by frequently used networks*, IEEE Transactions on Computers, vol. 44, pp. 695-697, May 1995.
[8] H. ÇAM AND J.A.B. FORTES, *Work-Efficient Routing Algorithms for Rearrangeable Symmetrical Networks*, IEEE Transactions on Parallel and Distributed Systems, vol. 10, no. 7, pp. 733-741, July 1999.
[9] H.-B. CHEN AND F.K. HWANG, *On Multicast Rearrangeable 3-stage Clos Networks Without First-Stage Fan-Out*, SIAM Journal on Discrete Mathematics, vol. 20, no. 2, pp. 287-290, 2006.
[10] M. CUTLER AND Y. SHILOACH, *Permutation layout*, Networks, vol. 8, pp. 253-278, 1978.
[11] N. DAS, *More on rearrangeability of combined $(2n-1)$-stage networks*, Journal of Systems Architecture, vol. 51, no. 3, pp. 207-222, 2005.
[12] J. DE RUMEUR, *Communications dans les réseaux de processeurs*, Collection Etudes et Recherches en Informatique, Masson, Paris, 1994.
[13] K. DIKS, H.N. DJIDJEV, O. SÝKORA, AND I. VRŤO, *Edge separators of planar and outerplanar graphs with applications*, Journal of Algorithms, vol. 14, pp. 258-279, 1993.

[14] W. Dou and E. Yao, *On rearrangeable multirate three-stage Clos networks*, Theoretical Computer Science, vol. 372, no. 1, pp. 103-107, 2007.
[15] D.-Z. Du and F.K. Hwang (Eds.), *Advances in Switching Networks*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 42, July 1997.
[16] D.-Z. Du and H.Q. Ngo (Eds.), *Switching Networks: Recent Advances*, Series: Network Theory and Applications, vol. 5. Kluwer Academic Publishers, Dordrecht, June 2001.
[17] O. Gabber and Z. Galil, *Explicit Construction of Linear-Sized Superconcentrators*, Journal of Computer and System Sciences, vol. 22, no. 3, pp. 407-420, 1981.
[18] F. Harary, *Graph Theory*, Addison-Wesley, Reading, MA, 1972.
[19] S.M. Hedetniemi, S.T. Hedetniemi, and A.L. Liestman, *A survey of gossiping and broadcasting in communication networks*, Networks, vol. 18, pp. 319–349, 1988.
[20] J. Hromkovič, P. Kanarek, R. Klasing, K. Loryś, W. Unger, and H. Wagener, *On the sizes of permutation networks and consequences for efficient simulation of hypercube algorithms on bounded-degree networks*, Proc. 12th Symposium on Theoretical Aspects of Computer Science (STACS '95), Springer LNCS 900, pp. 255–266, 1995.
[21] J. Hromkovič, P. Kanarek, and K. Loryś, in preparation.
[22] J. Hromkovič, R. Klasing, B. Monien, and R. Peine, *Dissemination of Information in Interconnection Networks (Broadcasting and Gossiping)*, in: Ding-Zhu Du and D. Frank Hsu (eds.), *Combinatorial Network Theory*, Kluwer Academic Publishers, pp. 125–212, 1996.
[23] J. Hromkovič, R. Klasing, A. Pelc, P. Ružička, and W. Unger, *Dissemination of Information in Communication Networks: Broadcasting, Gossiping, Leader Election, and Fault-Tolerance*, Springer Monograph, Springer-Verlag, 2005.
[24] J. Hromkovič, R. Klasing, and E.A. Stöhr, *Dissemination of information in vertex-disjoint paths mode*, Computers and Artificial Intelligence, vol. 15, no. 4, pp. 295-318, 1996.
[25] J. Hromkovič, R. Klasing, E.A. Stöhr, and H. Wagener, *Gossiping in Vertex-Disjoint Paths Mode in d-Dimensional Grids and Planar Graphs*, Information and Computation, vol. 123, no. 1, pp. 17-28, 1995.
[26] J. Hromkovič, R. Klasing, W. Unger, and H. Wagener, *Optimal Algorithms for Broadcast and Gossip in the Edge-Disjoint Path Modes*, Information and Computation, vol. 133, no. 1, pp. 1-33, 1997.
[27] J. Kahn, G. Kalai, and N. Linial, *The influence of variables on boolean functions (extended abstract)*, Proc. 29th IEEE Symp. on Foundations of Computer Science (FOCS '88), pp. 68–80, 1988.
[28] R. Klasing, *The relationship between the gossip complexity in vertex-disjoint paths mode and the vertex bisection width*, Discrete Applied Mathematics, vol. 83, no. 1-3, pp. 229–246, 1998.
[29] M. Klawe and F.T. Leighton, *A tight lower bound on the size of planar permutation networks*, SIAM J. Disc. Math., vol. 5, no. 4, pp. 558-563, 1992.
[30] D.E. Knuth, *The Art of Computer Programming; Volume 1: Fundamental Algorithms*, Addison-Wesley, 1968; 2nd edition, 1973.
[31] F.T. Leighton, *Introduction to parallel algorithms and architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Publisher, 1992.
[32] R.J. Lipton and R.E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math., vol. 36, no. 2, pp. 177-189, 1979.
[33] H.Q. Ngo, *A new routing algorithm for multirate rearrangeable Clos networks*, Theoretical Computer Science, vol. 290, no. 3, pp. 2157-2167, 2003.
[34] H.Q. Ngo, *WDM Switching Networks, Rearrangeable and Nonblocking* $[w, f]$*-connectors*, SIAM Journal on Computing, vol. 35, no. 3, pp. 766-785, 2005.
[35] H.Q. Ngo and Van H. Vu, *Multirate Rearrangeable Clos Networks and a Generalized Edge-Coloring Problem on Bipartite Graphs*, SIAM Journal on Computing, vol. 32, no. 4, pp. 1040-1049, 2003.
[36] N. Pippenger, *Superconcentrators*, SIAM Journal of Computing, vol. 6, no. 2, pp. 298-304, 1977.
[37] N. Pippenger, *Communication networks*, in: *Handbook of Theoretical Computer Science*, J. van Leeuwen (editor), Elsevier Science Publishers B.V., 1990.
[38] N. Pippenger and L. Valiant, *Shifting graphs and their applications*, Journal of the ACM, vol. 23, no. 3, pp. 423-432, 1976.
[39] F.P. Preparata and J.E. Vuillemin, *The cube-connected cycles: a versatile network for parallel computation*, Communications of the ACM, vol. 24, pp. 300-309, 1981.
[40] J.D. Ullman, *Computational Aspects of VLSI*, Computer Science Press, Rockville, MD, 1984.
[41] A. Waksman, *A permutation network*, Journal of the ACM, vol. 15, no. 1, pp. 159-163, 1968.
[42] R. Wanka, *Paralleles Sortieren auf mehrdimensionalen Gittern*. Ph.D. Dissertation, Universität Paderborn, 1994.
[43] R. Werchner, personal communication.
[44] D.B. Wilson, *Embedding leveled hypercube algorithms into hypercubes*, Proc. 4th ACM Symposium on Parallel Algorithms and Architectures (SPAA '92), pp. 264-270, 1992.