Symmetry and Computability in Anonymous Networks: A Brief Survey*

Evangelos Kranakis**

Carleton University, School of Computer Science, Ottawa, Ontario, K1S 5B6, Canada. E-mail kranakis@scs.carleton.ca

Abstract. Processors in anonymous networks are as identical to each other as possible and possess "little" knowledge about the network. Anonymous networks are very useful in theoretical studies for testing "true distributivity". In this paper we give a brief survey of results illuminating how symmetry influences computability in anonymous networks. Problems and issues considered include leader election, spanning tree construction, orientations, randomization, processor views, and computability problems on arbitrary as well as symmetric functions. Results mentioned are applicable to several topologies ranging from the rings, tori, hypercubes, and Cayley networks to arbitrary networks. We also propose several related open problems.

1980 Mathematics Subject Classification: 68Q99
CR Categories: C.2.1
Key Words and Phrases: Anonymous network, boolean function, cayley graphs, computability, gap theorem, hypercube, leader election, ring, symmetry.
SCS Technical report: TR-96-18

1 Introduction

Ideal distributed networks maximize effective use of commonly used resources by distributing efficiently communication tasks. One of the main goals of distributed computing is the design and analysis of efficient algorithms for computing "interesting network functions" of the input in a given topology of interconnected processors (i.e., network). A basic assumption is that the network consists of a set of interconnected processors (i.e., computers) transmitting along the links of the network and that the efficiency of the algorithms designed depends on several factors, including topology, synchronicity, interprocessor communication, as well as other technical characteristics, like existence of a center, spanning tree, etc.

* Dedicated to the memory of my beloved father Konstantionos who passed away unexpectedly on June 17, 1996.

** Research supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

Anonymous networks form a special but natural class of distributed networks; their processors have either no identities or are at least as similar to each other as possible. This is particularly important in view of the high cost of the software that must be used by the processors.

The first serious study of anonymous networks is in the work of Dana Angluin [2] who attempted to illuminate and find something interesting in those questions that "may ultimately contribute to the goal of a correct and fruitful theory of distributed computing". In her paper the following question is posed.

> "How much does each processor in a network of processors need to know about its own identity, the identity of other processors and the underlying connection network in order for the network to be able to carry out useful functions?"

During a computation step messages may be transmitted by any processor of the network along a link. By function one understands one of several important network utilities, like setting up a spanning tree, leader election, recognition of graph properties, etc. Leader election is an important problem, whose solution leads to easy solutions of other network problems, like constructing a spanning tree, etc. Among other things, it is shown in this paper [2] that in the complete network there exists a line of processors that deterministically establishes a center, while this is impossible in the four node graph connected in a cycle.

1.1 Goals of the survey

The similarity of processors implicit in the definition of anonymous networks gives rise to interesting notions of symmetry which can be exploited to optimize the computability of network functions. The purpose of this paper is to outline results and problems about the use of symmetry in the design of efficient computability algorithms in anonymous networks. Issues considered include leader election, spanning tree construction, and computability of boolean functions in arbitrary as well as special classes of networks. In particular, this survey will argue that anonymous networks will continue to play a significant role in the evolution of "fair and good principles" for network distributivity.

1.2 Note on notation and definitions

Throughtout this paper by network, usually but not always denoted by $\mathcal{N}$, we understand a simple connected graph. The processors are located at nodes of the graph and the edges are the communication links. N is the number of nodes of the network.[3]

[3] For the sake of brevity several of the concepts are introduced only in an intuitive manner. The reader is requested to accept "liberally" the terms and definitions introduced throughout the paper. In particular, it will be advisable to seek precise definitions and proofs in the extensive literature cited.

2 On with Anonymity

This section begins with eight of the most commonly used assumptions on anonymity in the literature. We then discuss the importance of anonymity and give some simple rules of thumb which are useful in research practice.

2.1 Assumptions

In the study of anonymity a combination of the following eight assumptions (or their **negation**) appear prominently.

1. The processors know the network topology and the exact (or upper bound on) number of processors in the network.
2. The network is anonymous (i.e., the processors do not know either the identities of themselves or of the other processors).
3. The processors are identical (i.e., they all run the same algorithm).
4. The processors are deterministic.
5. The network is asynchronous.
6. The network is labeled (by a labeling we mean a global, consistent labeling of the network links).
7. The network links are FIFO.
8. Each processor port has two buffers, one for storing incoming and the other for storing outgoing data.

In theory, this gives rise to a total of at least 2^8 possible combinations of assumptions, which also explains the "plethora of variations of the notion of anonymity" that occur in the literature.

2.2 Importance of anonymity

Anonymity has played a significant role in analyzing ideal distributed environments consisting of "similar" processors. Three important reasons motivating the study of anonymous networks are the following.

- The more anonymity the more widely applicable the algorithm is.
- The more anonymous the network is the more distributed it is.
- The more anonymity the closer to parallelism.

Added to these reasons one should also not neglect to mention the importance of anonymous networks for the study of the "extremal behavior" of network properties and functions (i.e. upper and lower bounds) in distributed computing.

2.3 Paradigms and rules of thumb

Every field of research has its own paradigms and rules of thumb which are used in everyday research practice. Anonymous networks are no exception. In this case there are three such rules that can be used to guide your nvestigations.

1. Try to break symmetries (deterministically or otherwise).
2. If you cannot break symmetries,...,try to take advantage of them.
3. Symmetric functions usually have the highest bit complexity.

To these rules one may also add that computations in anonymous networks can be suitably formulated through stabilization, in the sense that all processors in the network eventually "converge" to the desired common value. The significance and applicability of these rules will become apparent in the sequel.

3 Symmetry and Computability

The idea of symmetry has been playing a significant role in all aspects of scientific and social life. In the words of Herman Weyl [35]:

> "Symmetry,..., is one idea by which man through the ages has tried to comprehend and create order, beauty and perfection".

Similarly, symmetry has played a significant role in facilitating the study of anonymous networks. The "heart" of computability in anonymous networks is based on collecting locally the inputs available to all the processors of the network. This process of collecting is called "input collection". Intuitively, the amount of symmetry available affects "proportionately" the bit complexity of the algorithm constructed. Symmetry may be expressed in one of several forms.

1. Degree of anonymity of processors.
2. Topological symmetry in the network.
3. Symmetry in the function to be computed.

The first property refers to the similarity of the processors, the second depends on the graph theoretic parameters of the underlying topology, while the third is is related to the degree of symmetry of what is being computed.

In the case of anonymous networks we distinguish two important notions that have played a significant role for the development of efficient algorithms.

- First is the notion of processor views that was explicitely developed by Yamashita and Kameda [36, 36]. This notion was also implicitely defined by Angluin [2] and others. Its main feature is the notion of "similarity of views".
- Second, is the notion of invariance group developed by Clote and Kranakis [12] for the study of circuit complexity of Boolean functions. It is used in conjunction with the automorphism group of the network and is suitable for "highly symmetric" networks (e.g., Cayley networks).

In the sequel we make precise these notions of symmetry.

3.1 Processor views

Let $\mathcal{L}$ be an priori arbitrary numbering of the links. The view of processor p with respect to the input configuration $I =< b_p : p \in V >$ and labeling $\mathcal{L}$, denoted by $T^{\infty}_{\mathcal{L},I}(p)$, is an infinite, labeled rooted tree defined recursively. The label of the root of $T^{\infty}_{\mathcal{L},I}(p)$ is the input bit b_p; for each processor $p_i, i = 1, \ldots, \deg(p)$, adjacent to p the tree $T^{\infty}_{\mathcal{L},I}(p)$ has a vertex p_i and an edge from the root to p_i labeled $l(p, p_i) := \{\mathcal{L}(p, p_i), \mathcal{L}(p_i, p)\}$. Moreover, at the vertex p_i of the tree $T^{\infty}_{\mathcal{L},I}(p)$ is rooted the tree $T^{\infty}_{\mathcal{L},I}(p_i)$.

Two views are called similar if they are isomorphic including edge and vertex labels. It is clear that this similarity is an equivalence relation all of whose equivalence classes have the same size. We also denote by $T^{h}_{\mathcal{L},I}(p)$ the tree obtained from $T^{\infty}_{\mathcal{L},I}(p)$ by removing all levels of height $> h$. The following results are proved in the literature.

- A Boolean function f is computable in a network $\mathcal{N}$ if and only if for any inputs I, I' such that there exist labelings $\mathcal{L}, \mathcal{L}'$ so that the trees $T^{\infty}_{\mathcal{L},I}, T^{\infty}_{\mathcal{L}',I'}$ are similar we must have that $f(I) = f(I')$ (Yamashita and Kameda, [36, 37])
- Isomorphism up to depth N^2 implies isomorphism up to all depths (Yamashita and Kameda, [36, 37])
- Isomorphism up to depth $N-1$ implies isomorphism up to all depths (Norris, [31]), i.e If $T^{N-1}_{\mathcal{L},I}(p), T^{N-1}_{\mathcal{L},I}(p')$ are similar so are $T^{\infty}_{\mathcal{L},I}(p), T^{\infty}_{\mathcal{L},I}(p')$.

Intuitively, a processor view is an infinite tree that "collects" all the information accumulated by the nodes. The significance of the results above for complexity measurements is that in performing "input collection" processors need only iterate for $N-1$ steps before they have collected all information available in the network.

3.2 Invariance groups

The motivation for the next notion comes from circuit complexity but it has proved quite useful in the study of anonymous networks. A boolean function $f : \{0,1\}^N \rightarrow \{0,1\}$ is invariant under a permutation $\sigma \in S_N$ if for all $x_1, \ldots, x_N \in \{0,1\}$, $f(x_1, \ldots, x_N) = f(x_{\sigma(1)}, \ldots, x_{\sigma(N)})$. We denote by $S(f)$ the set of permutations in S_N that leave f invariant. Not every subgroup of the symmetric group S_N is of the form $S(f)$, for some f. (For example, try to prove that the alternating group A_N is not of this form.) The problem of recognizing invariance groups and related complexity results have been studied in the literature. It can be shown that

- Symmetric functions on N variables can be realized with circuit size $O(N)$ and depth $O(\log N)$ (Lupanov, [27]).
- If $S(f)$ has polynomial index (as a subgroup of the symmetric group S_N on N letters), i.e., $|S_N : S(f)| = N^{O(1)}$ then f is in NC^1 (Clote and Kranakis, [12]) (Notice that the proof of this theorem uses the classification of finite simple groups).

- If $S(f)$ has polynomially many orbits in its action on $\{0,1\}^N$ then f is in NC^1 (Babai, Beals and Takácsi-Nagy, [5]).

If we view $S(f)$ as a measure of the size of symmetry of f, then the larger its invariance group the easier it is to compute the boolean function f, in the circuit model. In the sequel we will see how the invariance group of a function f is related to the anonymous network it is being computed.

4 The Ring as an Anonymous Network

The ring is the simplest and most versatile interconnection network and many results are initially verified in this topology. In this section we introduce anonymity by studying the "simple" case of the ring. Subsequently, we consider the case of arbitrary anonymous networks.

4.1 Computing in rings

The first work on anonymity for a specific topology was that of Attiya, Snir and Warmuth, [3], who proposed the study of the computational capabilities of a system of N indistinguishable (i.e. anonymous) processors arranged on a ring in the synchronous and asynchronous models of distributed computing. A precise characterization of the functions that can be computed in this setting were given. More specifically, it is shown that no nonconstant function $f:\{0,1\}^N \rightarrow \{0,1\}$ can be computed on an oriented ring of N processors without knowledge of N. It is also shown that

- f is computable on an oriented ring $\Leftrightarrow$ f is invariant under cyclic shifts of inputs. (To see this, each processor sends its bit to the left neighbor; now for $N-1$ steps each processor sends the bit it receives from its right to left neighbor.)
- f is computable on an unoriented ring $\Leftrightarrow$ f is invariant under cyclic shifts and reversals of inputs. (To see this, each processor sends its bit to both neighbors; now for $\lfloor N/2 \rfloor$ steps each processor sends the bit it receives from a neighbor to its other neighbor.)

Moreover, it is easy to see that any such function can be computed in $O(N^2)$ bits. In addition, it is shown that there is no algorithm that solves the orientation problem for rings of arbitrarily large size. Also, the authors give upper and lower bounds for computing on a ring, both synchronous and asynchronous, as well as lower bounds for specific boolean functions, like XOR.

It is interesting to note that this simple paper gave a significant research impetus by providing a framework of questions whose study proved fruitful in other topologies as well.

4.2 Computability gaps

A significant momentum to the study of anonymity was given by the study of the "gap" phenomenon, which signifies a computability gap between constant and non-constant computable functions. Lower bounds for some specific boolean functions have been known for some time. For example, it is easy to see that computing the OR function on N input bits requires $\Omega(N^2)$ bits. (To see this take an input configuration with all processors but one having the input bit 0. Then take the processor, say p, diametrically opposed to the processor whose inpout bit is 1. Then for $\lceil N/4 \rceil$ steps all processors $p \pm i$, with $i = 0, 1, \ldots, \lceil N/4 \rceil$, execute identical programs. This easily implies the $\Omega(N^2)$ lower bound.)

Moran and Warmuth [30] establish a gap theorem for asynchronous distributed computation on the ring which says that either the function computed is constant and no messages need to be sent, or, in case of an arbitrary non-constant function $\Omega(N \log N)$ bits are required. Moreover, it is not difficult to construct non-constant functions whose bit complexity is $O(N \log N)$. If the processors are allowed to have distinct identities from a set of size $N + O(1)$ then it is easy to compute non-constant functions in $O(N)$ bits. However, the bound is still valid even if the N processors are allowed to have distinct identities chosen from a "larger" set of size at least $N^{1+\epsilon}$, for some $\epsilon > 0$ (Bodlaender, Moran and Warmuth, [7]).

The situation becomes even more interesting if the processors do not know the exact size of the network but rather an upper bound, say M. In this case, for any non-constant function there is an integer k depending only on the function such that computing the function requires $\Omega(NM)$ bits if k divides N, and $\Omega(N^2)$ bits if k does not N [18, 29]. It is interesting to note that no such gap theorem is possible in the torus if the processors know the exact size of the network. Beame and Bodlaender [6] construct a non-constant function that can be computed in $O(N)$ bits. However, if only un upper bound M is known then Monti and Roncato, [29], prove an $\Omega(N\sqrt{M})$ for all non-constant functions on the torus.

For the case of rings with a leader Mansour and Zaks [28] prove the following result. The language of input configurations recognized by an asynchronous ring algorithm **with a leader** has bit complexity $O(N)$ if and only if it is regular. Furthermore, non-regular languages require $\Omega(N \log N)$ bits (**processors do not know ring size**).

5 Arbitrary networks

A general study of anonymity on arbitrary networks did not appear until the seminal paper of Yamashita and Kameda [36, 37]. In this paper the authors identify four fundamental network problems. Leader Election Problem (LEP), Edge Election Problem (EEP), Spanning Tree Construction Problem (STP), and Topology Recognition Problem (TRP). They consider the following four possible: assumptions

– No network attribute information at all is available.
– Knowledge of exact number of processors.
– Knowledge of upper bound on number of processors.
– Knowledge of topology.

This gives rise to a total of sixteen possible problems. In their paper, Yamashita and Kameda offer graph theoretic characterizations of networks for which combinations of problems/assumptions above are solvable.

In the sequel we discuss complexity bounds, first for arbitrary and then for Cayley networks.

5.1 Complexity

Let $\mathcal{N}$ be an anonymous network with maximal node valency d and diameter δ. The following complexity results have been proved.

– The message complexity of computing a Boolean function on an arbitrary anonymous network is $O(N^2 \cdot m)$, where m is the number of links of the network (Yamashita and Kameda, [36, 37]),
– Let $\diamondsuit$ be a commutative, associative and idempotent binary operation. There is an algorithm for computing $\diamondsuit(I)$ for any input $I =< i_p : p \in V >\in A^N$ with bit complexity $O(N \cdot \alpha \cdot \delta \cdot d)$, where α denotes the number of bits necessary to represent an element of A (Kranakis, Krizanc and van der Berg, [21]), The algorithm implied by this bound was idependently discovered by G. Tel [34] in the context of infimum computations in directed networks.
– There is an algorithm that computes any Boolean function which is computable on the network with bit complexity $O(N^3 \cdot \delta \cdot d^2 \cdot \log N)$ (Kranakis, Krizanc and van der Berg, [21]).

In all the algorithms above the idea is to use the processor views in order to perform input collection. It is interesting to note that no better upper bound is known for arbitrary networks.

5.2 Cayley networks

The first case study of anonymity on Cayley graphs was for the ring by Attiya, Snir and Warmuth, [3]; this was extended to tori of constant dimension by Beame and Bondlaender, [6], and to hypercubes by Kranakis and Krizanc, [22]. Algorithms for the tori and hypercubes with bit-complexities $O(N^2)$ are easy to construct. (For example, an $O(N^2)$ algorithm on the hypercube is as follows. Processors exchange their inputs along dimension 1; and for $i = 2, \ldots, \log N$ processors exchange the view colloected so far up to step $i - 1$ along dimension i.)

However, it is expected that lower communication complexities are achievable on topologies other than the ring, especially if there are more links, as is the case in tori and hypercubes. (For hypercubes, an improved algorithm can be given along the following lines [22]. (1) We introduce a leader election mechanism which for each $i \leq \log N$ elects leaders among the processors with lexicographically maximal view at the ith step of the algorithm. (2) We use elementary results from the theory of finite permutation groups in order to introduce a coding mechanism of the views; leaders at the $(i-1)$st step exchange the encoded versions of their views; upon receipt of the encoded view they recover the original view sent and elect new leaders for the ith step. (3) The leader election and coding mechanisms help keep low the number of bits transmitted during the ith step of the algorithm to $O(N \cdot i^3)$ bits.)

The more general study of Cayley networks appeared in (Kranakis and Krizanc, [23]). Let G be a set of generators for the group $\mathcal{G}$. A boolean function f on $|\mathcal{G}|$ variables is computable on the anonymous network $\mathcal{N}_G[\mathcal{L}_G]$ if and only if $S(f) \geq Aut(\mathcal{N}_G[\mathcal{L}_G])$ and the bit complexity of computing any such function is $O\left(|\mathcal{G}| \cdot \log^2 |\mathcal{G}| \cdot \delta(G) \cdot \mathrm{dep}_G(\mathcal{G}) \cdot \sum_{g \in G} |g|^2\right)$, where $\delta(G)$ denotes the diameter of the network, $|g|$ the group order of the element g, and $\mathrm{dep}_G(\mathcal{G})$ is the maximum number of iterations of elements of G in a representation of a group element of $\mathcal{G}$ (see [23] for a precise definition). Moreover, there exists a set G of generators for $\mathcal{G}$ for which the above bit complexity is $O\left(|\mathcal{G}| \cdot \log^4 |\mathcal{G}| \cdot \sum_{g \in G} |g|^2\right)$ (Kranakis and Krizanc, [22, 23]). Such a set G is a set of Erdös-Renyi generators for the group $\mathcal{G}$ [17]. Here is a table of complexities for specific Cayley networks.

Network	Bit Complexity
R_N (Ring)	$O(N^2)$
$T_{m,n}$ (Torus)	$O(N^{1+2/n} \cdot \log^4 N / \log^3 m)$
Q_n (Hypercube)	$O(N \cdot \log^4 N)$
n-Star	$O(N \cdot \log^5 N / \log\log^3 N)$
n-Bubble-Sort	$O(N \cdot \log^7 N / \log\log^5 N)$
n-Pancake-Sort	$O(N \cdot \log^7 N / \log\log^5 N)$

It is interesting to point out here that all the bit complexities listed above refer to the network endowed with its "canonical" orientation, as this is specified by the given set of generators. Although it is expected that the absence of this orientation should lead to algorithms with worse bit-complexity, no mathematically formal statement of this fact is known in the literature; the only algorithms known for unoriented Cayley networks are only those implied by the general results of [21], which are also valid for arbitrary topologies. Another interesting open problem is to find a quantitative relationship between the bit complexity of computing a boolean function f and the size of its invariance group $S(f)$.

5.3 Symmetric functions

The output of symmetric functions depends only on the number of ones in the input. This makes it possible to give algorithms with improved bit complexity. Let $\mathcal{N}$ be an anonymous N-node network with maximal node valency d and diameter δ. There are algorithms with the following bit complexities.

Network	Bit Complexity	
Arbitrary	$O(N^2 \cdot \delta \cdot d^2 \cdot \log^2 N)$	[21]
Distance Regular	$O(N \cdot \delta \cdot d \cdot \log N)$	[21]
Hypercubes	$O(N \log^2 N)$	[22]

One of the algorithms for computing symmetric boolean functions on arbitrary networks is deterministic but uses the technique of convergence of a Markov chain for termination. (Namely, each processor sends its initial input value to all its neighbors. After receiving the values from all its neighbors the processor updates the value it already has based on the average of the values it receives. The number of iterations required depends on the second largest eigenvalue of the incidence matrix of the network. For more details see [21].)

6 Improving the Computability

Several ideas have been proposed and implemented for improving the computability of anonymous networks. In the sequel we discuss three important ones: randomization, symmetry breaking and orientations.

6.1 Randomization in rings

Average complexity and/or Probabilistic algorithms for Boolean functions have been studied by Attiya and Snir, [4]. For the asynchronous case they prove an $O(N \cdot \log N)$ bound, and for the synchronous case an $O(N)$ bound on the number of messages for computing boolean functions.

Probabilistic algorithms for Boolean functions were studied by Abramson, Adler, Higham and Kirkpatrick, [1]. They prove that all boolean functions require $\Omega(N \cdot \sqrt{\log N})$ bits, and there is a non-constant boolean function that can be computed in $O(N \cdot \sqrt{\log N})$ bits.

6.2 Deterministic symmetry breaking

Symmetry breaking occurs frequently in the research practice of anonymous networks because it can be used in simplifying algorithms, as well as for reducing the communication complexity. In a sense, randomization is a form of symmetry breaking. However, here we are interested in deterministic forms of symmetry braeking.

Electing a leader had significant influence on anonymity. Dana Angluin [2] gives sufficient conditions for the impossibility of electing a leader by using the notion of "covering" from topological graph theory. (A graph H is a covering of a graph G if there is a way to label the nodes of H with names of the nodes in G in such a way that if a node x of H is labeled "v" then the labels of the neighbors of x in H are precisely the neighbors of x in G. If the given network is a covering of a smaller network then there cannot exist a deterministic election algorithm for the network.) In a recent paper (Boldi, Shammah, Vigna, Gemmell, Codenotti, and Simon, [9]) characterize exactly when an anonymous network can elect of leader by a deterministic algorithm, using the theory of graph filtrations, which generalizes the notion of covering introduced by Angluin. Their algorithm can be made self-stabilizing in the sense of Dijkstra [13].

Deterministic algorithms for recognizing strings (a special case of the boolean function computability problem) of length N are studied in (Kranakis, Krizanc and Luccio, [25]). Here, the processors are given a string of length N, instead of a boolean function, and are asked to compare this string to their inputi, up to rotation. The complexities of the given algorithms are as follows.

String	Complexity
$x = w^{\lfloor N/k \rfloor} v$	$O(Nk + \frac{N^2}{k})$ if $v = \emptyset$
$x = w^{\lfloor N/k \rfloor} v$	$O(Nk + N \log n)$ if $v \neq \emptyset$
Kolmogorov random	$\Theta(N \log N)$

The algorithm and its analysis is particularly interesting for the case where the given string is Kolmogorov incompressible. (If the given string is Kolmogorov random, then it can be shown that $O(\log N)$ iterations of input collection are sufficient to discern whether or not the given string is identical to the string computed. For more details see [25].) It should be mentioned here that no thorough randomization studies for symmetry breaking on arbitrary anonymous networks are known.

6.3 Network orientations

The communication topology of a distributed network can be represented by an edge-labeled undirected graph where each node has a local label (or port number) associated to each of its ports. Without going into a formal definition of orientation for arbitrary networks, we point out that a "good labeling of the links" is expected to produce a consistent orientation of the network. However, it may not always be possible to produce such an orientation. (For example, in the ring this will be either clockwise or counter-clockwise. However, in an anonymous ring this is impossible to achieve. Intuitively, the reason is that such an orientation is a rather "asymmetric" situation and no deterministic algorithm could be used to break a symmetric configuration. For more details see [3].) By using marking of the links to break symmetries it is possible to give a deterministic orientation algorithm for the ring having communication complexity $O(N \log^2 N)$; also a Las Vegas algorithm (i.e. terminates with probability 1) with the the same communication complexity is possible by using coin-tossing to break symmetries (see Cidon and Shavitt, [11]).

In general, $\mathcal{L}$ is a labeling of the network $\mathcal{N}$ if for each vertex u, $\mathcal{L}_u : E(u) \rightarrow \{1, 2, \ldots, \deg(u)\}$ is one-to-one. It is well-known that if $\mathcal{L}$ satisfies a set of consistency constraints then it becomes a "sense of direction". The importance of labeling networks for achieving more efficient computations was recognized early by (N. Santoro [33]). For interesting ideas on the notion of sense of direction the reader is referred to the article of (Flocchini, Mans and Santoro, [20]). For the case of Cayley networks, the following results are known in the literature.

– Certain Cayley networks oriented with their canonical labeling are more powerful than unlabeled networks. This can be made precise by comparing the corresponding classes of computable functions, as e.g. (Kranakis and Krizanc, [24])
– Cayley graphs are precisely the regular graphs with minimal, symmetrical sense of direction (Flochini, Roncato and Santoro, [19])

We note that Cayley networks have natural orientations induced by their set of group generators. For example, in a torus such an orientation is North, South, East, West.

Research studies on this interesting topic of constructing suitable and efficient algorithms for "sense of direction" in arbitrary networks are expected to continue with fruitful results.

7 Related Issues

In this section two related but important issues are studied. The first concerns the number of buffers used by the local processors, and the second fault tolerance.

7.1 Reducing the number of buffers

Traditional modeling of anonymous networks requires the existence of a buffer at each port of a processor for handling processed bits. For example, in rings each processor is endowed with two buffers one corresponding to each neighbor. Let the "single buffer" ring consist of the same topology but each processor has only one buffer, the same one for both neighbors. This raises the following interesting question.

> What network computation can be performed and at what bit complexities, if the number of buffers at a node is less than the degree of the node?

It can be shown that a boolean function $f : \{0,1\}^N \rightarrow \{0,1\}$ is computable in the single-buffer ring of size N if and only if f is invariant under cyclic shifts and reflections of the inputs. In both the synchronous and asynchronous model the computation uses $O(N^2)$ one-bit messages. Moreover, in the synchronous case such a function can be computed in $O(N)$ time (Diks, Kranakis, Malinowski and Pelc, [14, 15]). (An algorithm whose complexity is exponential is as follows in the synchronous ring. Before transmission at the i-th step, each processor encodes its input in unary: "a sequence of ones followed by zeroes". It is now easy to see that a processor receiving bits from its two neighbors can discern the two values transmitted, without necessarily being able to identify where each value comes from. For details see [14, 15].)

It is interesting that no other topology, like tori, hypercubes, etc., has been studied. For example, what functions are computable and at what bit complexity on a two-dimensional torus in which each processor has less than four buffers (in the standard model one has access to four buffers: North, South, East, and West)? Although characterizations of the corresponding classes of computable functions are possible (see Norris [32], and Boldi and Vigna [8]) along the lines of the idea of processor views, as previously defined, these have not so far produced any interesting bit-complexity bounds, as is the case for the ring.

7.2 Fault tolerance

An algorithm for computing Boolean functions on anonymous hypercubes with bit complexity $O(N\delta_n(\gamma)^2\lambda^2 \log\log N)$, where γ is the number of faulty components (i.e. links plus processors), λ is the number of links which are either faulty, or non-faulty but adjacent to faulty processors, and $\delta_n(\gamma)$ is the diameter of the resulting "faulty hypercube" (Kranakis and Santoro, [26]). (The idea of the algorithm is as follows. Let f be a given Boolean function. Each processor p is given an input bit b_p and the Boolean function f. Let $Input$ be the input sequence of bits. Each processor p concerned executes the following algorithm: (1) determines whether or not the hypercube has a faulty link, (2) uses a "path-generation" algorithm in order to determine the location of the faulty links relative to itself, (3) uses an input collection mechanism in order to determine the entire input configuration $Input_p$, where $Input_p$ denotes p's view of $Input$, (4) determines whether or not the given function is computable on the given input by checking an invariance condition on the given function f, (5) if f is computable then processor p outputs $f(Input_p)$.)

Very few studies have been initiated for arbitrary, faulty, anonymous networks. It is a "peculiarity" of anonymity that fault-tolerance can be viewed as a form of symmetry breaking. Hence, do not be surprised that when only few errors occur then the complexity of computing in "faulty hypercubes" is less than in "non-faulty hypercubes". Faults considered in [26] may only occur at the start of the computation. It would be interesting to consider other kinds of malicious faults, like Byzantine.

8 Is the Future Anonymous?

We have given a survey of the influence of symmetry in anonymous distributed networks. Problems considered included leader election, spanning tree construction, orientations, randomization, processor views, and computatbility problems on arbitrary as well as symmetric functions. Results mentioned are applicable to several topologies ranging from the rings, tori, hypercubes, and Cayley networks to arbitrary networks. The basic model of anonymity relies on two simple principles: (1) keep the processors as similar to each other as possible, and (2) maintain locally the minimum amount of information about the network. In view of the more general applicability of algorithms designed for anonymous networks, anonymity is expected to continue to play a significant role as a "guide" in the evolution of distributed networks.

In contrast to anonymous networks, it is also worth mentioning the recent growth of high-speed networking. It appears that some efficient algorithms suitable for anonymous environments, especially those for which efficiency is achieved at the cost of higher local bit-processing, may run contrary to the "new" trends in networking. In high-speed networks one needs to reduce to minimum or even eliminate processing at the local level in order to accommodate the high speeds of local switching elements. (For more details see the papers of Cidon and Gopal, [10], and the more recent study by Dolev, Kranakis, Krizanc and Peleg, [16].)

Acknowledgements

Many thanks to Danny Krizanc for numerous enjoyable conversations on the topics of this survey. Many of the results presented and open problems suggested originate in our frequent discussions. The fruitful comments of the participants of SIROCCO'96 are greatfully acknowledged.

References

1. K. Abrahamson, A. Adler, L. Higham, and D. Kirkpatrick, "Randomized Evaluation on a Ring", in proceedings of 2nd International Workshop on Distributed Algorithms, held in Amsterdam, The Netherlands, July 1987, Jan van Leeuwen, ed., Springer Verlag Lecture Notes in Computer Science, vol 312, pp. 324 - 331, 1988.
2. D. Angluin, "Local and Global Properties in Networks of Processors", 12th Annual ACM Symposium on Theory of Computing, 1980, pp. 82 - 93.
3. H. Attiya, M. Snir and M. Warmuth, "Computing on an Anonymous Ring", Journal of the ACM, 35 (4), 1988. Preniminary version has appeared in proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computation, 1985, pp. 845 - 875.
4. H. Attiya and M. Snir, "Better Computing on the Anonymous Ring", Journal of Algorithms, vol. 12(2), pp. 204 - 238, 1991.
5. L. Babai, R. Beals and P. Takácsi-Nagy, "Symmetry and Complexity", in proceedings of 24th annual ACM Symposium on the Theory of Computing, pp. 438 - 449, 1992.
6. P. W. Beame and H. L. Bodlaender, "Distributed Computing on Transitive Networks: The Torus", 6th Annual Symposium on Theoretical Aspects of Computer Science, STACS, 1989, B. Monien and R. Cori, eds., Springer Verlag Lecture Notes in Computer Science. pp. 294-303.
7. H. Bodlaender, S. Moran and M. Warmuth, "The Distributed Bit-Complexity of the Ring: from the Anonymous to the Non-anonymous Case", Inf. and Comp., 118, pp. 34 - 50, 1994.
8. P. Boldi and S. Vigna, "Computing Vector Functions on Anonymous Networks", preprint, Department of Computer Science, University of Milan, 12 pages, 1990.
9. P. Boldi, S. Shammah, S. Vigna, P. Gemmell, B. Codenotti, and J. Simon, "Symmetry Breaking in Anonymous Networks: Characterizations", in Proceedings of 4th Israel Symposium on Theory of Computing and Systems, June 10 - 12, 1996, Jerusalem, Israel, pp. 16 - 26, IEEE press.

10. I. Cidon and I. Gopal. "PARIS: An approach to private integrated networks," *Journal of Analog and Digital Cabled Systems* 1(2), pp. 77-86, 1988.
11. I. Cidon and Y. Shavitt, "Message Terminate Algorithms for Anonymous Rings of Unknown Size", in proceedings of 6th International Workshop on Distributed Algorithms, held in Haifa, Israel, November 1992, A. Segall and S. Zaks, eds., Springer Verlag Lecture Notes in Computer Science, vol. 647, pp. 264 - 276, 1992.
12. P. Clote and E. Kranakis, Boolean Functions Invariance Groups and Parallel Complexity, SIAM Journal on Computing, Vol 20, No 3, pp. 553-590, 1991. Preliminary version has appeared in proceedings of 4th IEEE Conference on Structure in Complexity Theory.
13. E. Dijkstra, "Self-Stabilizing Systems in spite of Distributed Control", Comm. ACM, 17(11), pp. 643 - 644, 1974.
14. K. Diks, E. Kranakis, A. Malinowski, and A. Pelc, "The Buffer Potential of a Network", Proceedings of 1st International Conference on Structural Information and Communication Complexity, Ottawa, May 1994, P. Flocchini, B. Mans, and N. Santoro eds., pp. 149-150, 1995, Carleton University Press.
15. K. Diks, E. Kranakis, A. Malinowski, and A. Pelc, "Anonymous Wireless Rings", Theoretical Computer Science, 145/1-2, pp. 95-109, 1995.
16. S. Dolev, E. Kranakis, D. Krizanc and D. Peleg, "Bubbles: Adaptive Routing Scheme for High-Speed Dynamic Networks", in Proceedings of 27th ACM Symposium on Theory of Computing, pp. 528-537.
17. P. Erdös and A. Renyi, "Probabilistic Methods in Group Theory", J. d'Analyse Math., 14, pp. 127 - 138, 1965.
18. P. Ferragina, A. Monti, and A. Roncato, "Trade-off between Computational Power and Common Knowledge in Anonymous Rings", Proceedings of 1st International Conference on Structural Information and Communication Complexity, Ottawa, May 1994, P. Flocchini, B. Mans, and N. Santoro eds., pp. 35 - 48, 1995, Carleton University Press.
19. P. Flocchini, A. Roncato and N. Santoru, "Symmetries and Sense of Direction in Labeled Graphs", paper presented to the 27th South-Eastern Conference on Combinatorics, Graph Theory and Computing, 1996.
20. P. Flocchini, B. Mans and N. Santoro, "Sense of Direction: Formal Definitions and Properties", Proceedings of 1st International Conference on Structural Information and Communication Complexity, Ottawa, May 1994, P. Flocchini, B. Mans, and N. Santoro eds., pp. 9 - 34, 1995, Carleton University Press.
21. E. Kranakis, D. Krizanc and J. van der Berg, "Computing Boolean Functions on Anonymous Networks", Journal of Information and Computation, Vol. 114, No. 2, pp. 214-236, 1994. Preliminary version has appeared in the proceedings of International Conference on Automata Languages and Programming, ICALP 1990, Vol. 443, Springer Verlag Lecture Notes in Computer Science, pp. 254-267.
22. E. Kranakis and D. Krizanc, "Distributed Computing on Anonymous Hypercube Networks", Journal of Algorithms, accepted Feb. 1996. Preliminary version has appeared in proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing, Dallas, Dec. 2-5, pp. 722 - 729, 1991.
23. E. Kranakis and D. Krizanc, "Computing Boolean Functions on Cayley Networks", Proceedings of the 4th IEEE Symposium on Parallel and Distributed Processing, Arlington, Texas, Dec. 1-4, 1992, pp. 222-229.
24. E. Kranakis and D. Krizanc, "Labeled versus Unlabeled Distributed Cayley Networks", Discrete Applied Mathematics, 63/3 (1995) 223-236. Preliminary version

has appeared in proceedings of 1st International Conference on Structural Information and Communication Complexity, Ottawa, May 1994, P. Flocchini, B. Mans, and N. Santoro eds., pp. 71-82, 1995, Carleton University Press, 1995.
25. E. Kranakis, D. Krizanc, and F. Luccio, "String Recognition on Anonymous Rings", in proceedings of 20th International Symposium on Mathematical Foundations of Computer Science, Prague, Chech Republic, July/Sep 1995, SVLNCS, J. Wiedermann, and Peter Hajek, eds., pp. 392-401.
26. E. Kranakis and N. Santoro, "Distributed Computing on Anonymous Hypercubes with Faulty Components", Proceedings of 6th International Workshop on Distributed Algorithms, Haifa, November 2-4, 1992 Springer Verlag Lecture Notes in Computer Science, Vol. 647, A. Segall and S. Z. Saks, eds., pp. 253 - 263, 1992.
27. O. Lupanov, "On the Principle of Local Coding and the Realization of Functions of Certain Classes of Networks Composed by Functional Elements", Ser. Phys. Dokl., 6, pp. 750-752, 1962.
28. Y. Mansour and S. Zaks, "On the Bit Complexity of Distributed Computation in a Ring with a Leader", Inf. and Comp. 75(2), pp. 162 - 177, 1987.
29. A. Monti and A. Roncato, "A Gap Theorem for the Anonymous Torus", Inf. Proc. Let. 57, pp. 279 - 285, 1996.
30. S. Moran and M. K. Warmuth, "Gap Theorems for Distributed Computation", *Proceedings of the 5th ACM Symposium on Principles of Distributed Computing*, pp. 131-140, (1986).
31. N. Norris, "Universal Covers of Graphs: Isomorphism to depth $n-1$ Implies Isomorphism to all Depths", Discrete Applied Math., 56/1, pp. 61 - 74, 1995.
32. N. Norris, "Computing Functions on Partially Wireless Networks", Proceedings of 1st International Conference on Structural Information and Communication Complexity, Ottawa, May 1994, P. Flocchini, B. Mans, and N. Santoro, eds., pp. 83 - 98, 1995, Carleton University Press.
33. N. Santoro, "Sense of Direction, Topological Awareness, and Communication Complexity", ACM SIGACT News, 16, pp. 50 - 56, 1984.
34. G. Tel, "Directed Network Protocols", in proceedings of 2nd International Workshop on Distributed Algorithms, held in Amsterdam, The Netherlands, July 1987, Jan van Leeuwen, ed., Springer Verlag Lecture Notes in Computer Science, vol 312, pp. 12 - 29, 1988.
35. H. Weyl, "Symmetry", Princeton University Press, 1952
36. M. Yamashita and T. Kameda, "Computing on an Anonymous Network, in 7th Annual ACM Symposium on Principles of Distributed Computing, pp. 117 - 130, 1988,
37. M. Yamashita and T. Kameda, "Computing Functions on an Anonymous Network, Laboratory for Computer and Communication Research, Simon Fraser University, TR 87-16, 1987, 27 pp..

This article was processed using the LaTeX macro package with CUP style