Survey on Topological Methods in Distributed Computing

Muntasir Raihan Rahman

Abstract

One of the most exciting developments in the theory of distributed computing in recent years has been the application of powerful concepts from topology to prove results about computability in resilient distributed systems. Topology is a branch of mathematics that deals with connectivity and convergence of certain types of objects. As it turns out, the higher dimensional connectivity properties of these special objects are related to the solvability of certain distributed computing tasks. In this survey, we thoroughly investigate the topological approach to distributed computing, especially how to use techniques from combinatorial and algebraic topology to prove impossibility and lower bound results for important problems in distributed systems.

Contents

1 Introduction

In recent years, techniques borrowed from combinatorial and algebraic topology [24, 17, 16, 18] have brought about significant progress in characterizing synchronous and asynchronous distributed computing tasks and their solvability. Topology is a branch of pure mathematics that deals with higher dimensional connectivity properties of certain topological objects. As it turns out, these objects are generalizations of graphs, and there connectivity properties are closely related to the computability of important distributed problems. This approach of exploiting certain topological properties of higher dimensional geometric objects to prove structural results for distributed computing tasks has been coined as the *the topological approach to distributed computing* [18]. Our goal in this survey is to investigate this topological approach to distributed computing and its applications to prove impossibility results for various distributed computing tasks, including consensus, k-set consensus, and renaming. For some of these problems, the topological approach is the only way known so far to resolve impossibility and lower bound results.

2 History

Topology entered distributed computing with three independent teams of researchers. Borowsky and Gafni [5], Saks and Zaharoglou [26], and Herlihy and Shavit [19], worked on deriving lower bounds for solving the set agreement problem using powerful topological methods. In particular, the paper by Borowsky and Gafni introduced a powerful simulation method (BG simulation) for proving possibility and impossibility results in distributed systems.

In a landmark paper, Herlihy and Shavit [19] introduced a new paradigm based on algebraic topology to reason about asynchronous computations, where at-most one process can fail. There framework consisted of modeling tasks and protocols using *simplicial complexes*, and then applying *homology theory* [27] to reason about them. A key feature in this framework is that the exponential number of possible executions can be compactly represented using a static topological object in a model independent manner. In case of the *immediate snapshot (IS) model* [5], the simplicial complexes turn out to be *manifolds*, which are a special class of simplicial complexes that are easy to deal with. Herlihy and Shavit further extended this framework by proving the *Asynchronous Computability Theorem* which states necessary and sufficient conditions for task solvability by a wait-free protocol in shared memory [21]. They also showed how to apply this theorem to other problems like set agreement and renaming. Borowsky, generalized this theorem to a model of regular shared memory with set-consensus objects under more general failure conditions [4].

3 Background on Topology

In this section, we review some basic notions from topology that will be used later to characterize distributed computing tasks. We refer the readers to standard textbooks for more coverage of the topic [24].

3.1 Simplicial Complexes

Simplicial complexes are the main building blocks of the topological formulation of distributed computing tasks. They are higher dimensional analogs of graphs. Graph connectivity can only model distributed computations with a single failure. For more than one failure, the corresponding graph structure can become extremely complex and hard to reason with. As it turns out, simplicial complexes can model computations with more than one failure quite nicely. We now review some elementary notions from the topology of simplicial complexes.

Simplicial complexes can be represented both combinatorially and geometrically. A n-dimensional simplex is a set of $n + 1$ independent vertices. Geometrically, the vertices can be imagined as affinely independent points in Euclidean space. For example, a 0-simplex is a single point, a 1-simplex is a line segment, and a 2-simplex is represented by a filled in triangle. A *simplicial complex* is a finite set of simplexes closed under inclusion and intersection. The *dimension* of a complex is the maximum dimension of any simplex it contains. Some examples of complexes are shown in Figure 1.

3.2 Simplicial Maps

A *simplicial map* $\delta : \mathcal{C}_1 \rightarrow \mathcal{C}_2$ is a vertex-to-vertex map that sends simplexes of $\mathcal{C}_1$ to simplexes of $\mathcal{C}_2$. Geometrically, it is a piece-wise linear map of geometric simplexes. A *carrier map* $\mathcal{M}$ maps simplexes to sub-complexes and preserves intersections, that is, the map of the intersection of two simplexes is equal to the intersection of their maps: $\mathcal{M}(\sigma \cup \tau) = \mathcal{M}(\sigma) \cap \mathcal{M}(\tau)$.

3.3 Manifolds

An n-dimensional complex is called a *mainfold* if each $(n-1)$ simplex is contained in exactly two n-simplexes. Geometrically, the neighborhood of a point on a manifold looks locally like n-dimensional Euclidean space $\mathbb{R}^n$. A complex is a manifold with boundary if each $(n-1)$-simplex is in either one or two simplexes. An $(n-1)$ simplex is internal if it is in two simplexes, and external otherwise. The sub-complex generated by all the $(n-1)$-dimensional external simplexes is called the *boundary* of the manifold. Manifolds have nice combinatorial structures which will be exploited later.

3.4 Connectivity

Since simplicial complexes are higher dimensional generalizations of graphs, it is natural to view connectivity of simplicial complexes as generalizations of graph-connectivity. The concept of higher dimensional connectivity is crucial to the topological approach, since the connectivity properties of complexes representing tasks can be related to their solvability. Mathematically, a complex $\mathcal{C}$ is *n-connected*, if for all $m \leq n$, every continuous map of the m-sphere $f : S^m \rightarrow \mathcal{C}$ can be extended to a continuous map of the $m+1$ disk $f : D^{m+1} \rightarrow S$. Intuitively, this means that we can shrink a loop to a point on the complex without any obstruction. When this is not possible, the complex is said to have *holes* in the corresponding dimension. As an example, the 0-sphere can be extended to the 1-disk on a torus, so the torus is 0-connected. This is the same as graph connectivity. But the 1-sphere cannot be always continuously extended to the 2-disk, so the torus is not 2-connected. As a degenerate case, a complex is -1-connected when it is non-empty.

A geometric complex is *sub-divided* by partitioning each of its simplexes into smaller simplexes without changing the complex's polyhedron.

3.5 Important Topological Results

3.5.1 Sperner's Lemma

Sperner's lemma is a combinatorial equivalent of the Brouwer fixed point theorem. Sperner's lemma states that every sperner coloring of a triangulation of an n-dimensional simplex contains a simplex colored with all $n+1$ colors, where $n+1$ is the number of vertices. This surprisingly simple lemma plays a pivotal role in the topological framework for distributed computation. The lemma is usually stated and proved in the two dimensional case.

Lemma 1 *Given a triangle ABC, and a triangulation T of the triangle, the set of vertices of the T can be colored in three ways: (a) the end point vertices must have 3 distinct colors, (b) Any vertex on an edge must be colored with either of the two colors at the endpoints of the edge, and (c) any vertex interior to the triangle can be colored with any of the three colors at the vertices of ABC. If this is true, then there exists an odd number of triangles in T whose vertices are colored with three different colors.*

3.5.2 Brouwer's Fixed Point Theorem

Sperner's lemma is discrete in nature, and as usual it has a continuous counterpart, known as the *Brouwer Fixed Point Theorem.*

Theorem 2 *Given any continuous function $f : \mathbb{B}^n \to \mathbb{B}^n$ from the n-dimensional ball to itself, there exists a point $x \in \mathbb{B}$ with $f(x) = x$, that is, any continuous function f on the n-dimensional ball must have a fixed point.*

4 Topological Representation of Tasks

A single vertex can be used to represent the state of a single process. A d-simplex consisting of $d + 1$ vertices representing distinct processes can be used to represent compatible states of $d + 1$ processes. As an example, consider binary consensus for three processes P, Q, R. There are eight possible input configurations for this problem, which are represented as the eight faces (or 2-simplexes) of the octahedron on the left side of Figure 1, the corresponding output complex is shown on the right. In the upper 2-complex, all processes output 0, in the lower complex they all output 1. Every input simplex has a set of valid output simplexes. By the validity condition of consensus, if all processes start with 0, they must be mapped to the all 0 output complex.

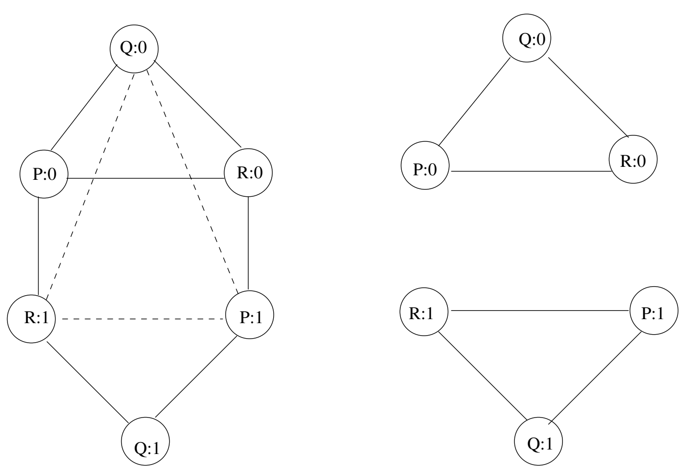

Figure 1: Input and output complexes for 3-process binary consensus.

In general, any decision task can be modeled in this way. The input complex $\mathcal{I}$ contains one $(n-1)$ simplex for every input configuration. Similarly, the output complex $\mathcal{O}$ contains a simplex for each valid output configuration. A map Δ that maps each input simplex to a set of output simplexes defines which output states are legal for each input state according to the problem definition. So a task can be fully characterized by the triple $(\mathcal{I}, \mathcal{O}, \Delta)$.

In this context, simplicial complexes are used to describe whether processes can distinguish different configurations from one another. A situation where two output configurations differ in only one output, is modeled in the complex by having the two corresponding simplexes share $n-1$ common vertices. Complexes can capture more information about the degree to which to configurations are similar. Specifically, two simplexes that have d common vertices are indistinguishable to exactly d processes. This makes it easier to study f-resilient algorithms, where graphs can usually only handle a single failure. We need complexes to move from the $f = 1$ case to the $f > 1$ case.

4.1 Protocol Complexes

So far, we have talked about representing input and out configurations as simplicial complexes. In order to solve a task, we need a protocol. Here we will be concerned with *full information protocols*, where processes keep track of everything they have seen during the entire execution. As it turns out such a protocol can also be represented as a *protocol complex*. Consider a wait-free protocol[1] for n processes that solves some task. One can define a corresponding $(n-1)$-dimensional complex , where each vertex is labeled by a process id and state of that process when it terminates in some execution. Given an input configuration, and a schedule of the processes, the final state of every process will be completely determined. This final configuration is represented by a simplex in the protocol complex.

4.2 Snapshot Models

In order to resolve the k-set agreement impossibility, Borowsky and Gafni [6] introduced the *immediate snapshot (IS) model*, where processes communicate using a single writer snapshot object. In this model, we can imagine that each execution is divided into a sequence of phases. In each phase, the adversary selects a set of processes that have not yet taken a step. All processes in that set write simultaneously, and then they all take a snapshot of the entire memory. Phases proceed until every process has been scheduled exactly once. It should be clear why this is called an immediate snapshot model, each process takes a snapshot immediately after it writes. A great advantage of this simplified model is that the corresponding protocol complex turns out to be a manifold, which has nice structure. As we will see later, this greatly simplifies the impossibility proof for k-set agreement. We have the following theorem which we will not prove [18].

[1] A protocol is *wait-free* if it can tolerate upto $n-1$ failures for an n process system.

Theorem 3 *If the input complex $\mathcal{I}$ is a manifold, so is the immediate snapshot protocol complex $\mathcal{IS}(\mathcal{I})$. Moreover, if $\mathcal{I}$ is a manifold, then it preserves boundaries, that is $\mathcal{M}(\partial\mathcal{I}) = \partial\mathcal{M}(\mathcal{I})$.*

5 Problems

In this section, we describe some canonical problems that have benefitted from the topological approach. We have chosen four such problems. Among these consensus and symmetry breaking problems don't really need the topological results, in a sense that the results for these problems were already settled using non-topological approaches. On the other hand, for problems like set consensus and renaming, the topological approach is the only viable way to resolve impossibility and lower bound results. However, there have been some research that have tried to discover non-topological results for these problems [4].

5.1 Consensus

Consensus is probably the most rigorously investigated problem in the theory of distributed systems [13, 12, 25]. In the simplest form of the problem, each process starts with a private binary input, and they have to agree on a single output within the following three constraints:

- *Termination*: Every non-faulty process eventually chooses an output value.
- *Agreement*: All non-faulty processes agree on the same output value.
- *Validity*: The output value chosen should be the input of some process.

The validity condition implies that if all processes start with the same input value, then all the non-faulty processes should decide on that value.

5.2 Set Consensus

In 1990, Chaudhuri [10] introduced a generalization of consensus where more than one distinct output value can be chosen. The topological approach entered the distributed computing arena through this problem. In this problem, the termination and validity conditions are the same as consensus, only the agreement condition (*k-set agreement*)is changed to ensure that all non-faulty processes decide on at most k distinct values (for k-set consensus). Formally, in the *k-set consensus* problem for $k < n$, each process gets a value from the set $\{0, 1, \ldots, n-1\}$, and has to agree on an output value satisfying termination, validity, and k-set agreement. The problem is trivial for $k > n$. From the FLP result [13], we already know that set consensus is unsolvable for $k = 1$. To prove, that k-set consensus is not solvable if $k < f + 1$, we need to exploit higher dimensional connectivity of simplicial complexes, that is, we need to use topology.

Problem(s)	Result	Topological Tool
k-Set Agreement	Impossibility [21], [19]	Algebraic Topology, Nerve Lemma, Sperner's Lemma
k-Set Agreement	Impossibility [26]	Point-set Topology, Brouewer's Fixed Point Theorem, Sperner's Lemma
Consensus	Impossibility [21]	Algebraic Topology
Renaming	Lower Bounds [9]	Combinatorial Topology
k-Set Consensus and WSB	Separation [18]	Algebraic Topology, Sperner's Lemma
$(2n-1)$-Renaming and WSB	Equivalence [9]	Combinatorial Topology
Approximate Agreemenent	Complexity Bound [22]	Algebraic Topology

Table 1: Summary of Results

5.3 Renaming

In the *K-renaming* task [2] $n+1$ processes start with unique input names from a large name-space, and must choose unique output names from a much smaller name-space $\{0, 1, \ldots, K\}$. To rule out trivial solutions, any protocol to solve this problem has to be *anonymous*, that is, the value chosen by a process cannot depend on its specific process id.

5.4 Weak Symmetry Breaking (WSB)

The *weak symmetry breaking (WSB)* task was introduced in [15]. In this task, processes have no input value, but need to select output values 0 or 1. It is required that in every execution at-least one process decide 1, and at-least one decide 0. Intuitively, we need the processes to break up into two (possibly uneven sized) groups. Later, we will see that WSB and renaming are related, more precisely, a solution for $(2n-1)$-renaming solves WSB. This shows that the topological approach can not only resolve impossibility results, but also reduction results.

6 Applications of the Topological Approach

In this section, we show some applications of the topological approach to prove impossibility, lower bound, separation and equivalence results for distributed computing problems. A summary of the results is shown in Table 6. We will not prove the results rigorously, rather we will try to provide intuitive proof sketches. We refer to the original papers for detailed proofs.

6.1 General Strategy

The basic method of proving impossibility results using the topological method is as follows. To prove that a certain problem is unsolvable, one can use model specific information (for example, the model could be IIS wait-free synchronous) to show that the associated protocol complex has a certain topological property (for example, connectivity) which is preserved by the simplicial map δ. The task specification can then be used to show that the image of δ cannot have the specified property (for example, the image could be disconnected), implying that the map δ cannot exist.

At this point we should point out a limitation of the topological approach. Although this paradigm give us model independent results, and promotes static combinatorial reasoning rather than dynamic operational reasoning, it has an important drawback. Despite making impossibility results more intuitive, algorithm design is at best messy and cumbersome in this approach. We use the topological approach to exploit the structure of protocol complexes, so a *topological algorithm* would also have to exploit such structure. However, we have not seen any such algorithms in the literature.

6.2 Impossibility of Consensus

Let us use this strategy to show that wait-free binary consensus is impossible. It has been shown in [21] that in the asycnhronous shared memory read-write model, any protocol complex that starts with a connected input complex remains connected. The connectivity property is preserved by the map δ, since it is a simplicial map. As shown in Figure 1, the input complex for three process binary consensus is connected. But the output complex is disconnected, and the image of δ must include simplexes from both the upper and lower output simplexes. As a result, wait-free three-process binary consensus is impossible in this model. $\mathcal{Q.E.D.}$

6.3 Impossibility of Set Consensus

In this section, we will give a proof sketch of the impossibility of set consensus for the IIS model. The reason for this is that the protocol complex for the IIS model turns out to be a manifold. This makes the proof more amenable and intuitive. We will briefly describe the results for non-manifold protocol complexes, but we won't go into details. We refer to the papers for detailed proofs [21, 19, 26].

Assume, we have $n+1$ processes. We consider the wait-free case, where the number of failures $f = n$, and $k < n+1$. This easier case captures the main structural results, and the general case can be obtained via reduction using a method called BG-simulation [8][2].

Consider an input 2-simplex, $\mathcal{S}^2$, where processes p_0, p_1, p_2 start with values $0, 1, 2$ respectively. For the IS model, the protocol complex is a manifold, so $\mathcal{IS}(\mathcal{S}^2)$ is a subdivision of $\mathcal{S}^2$. If p_i finishes its computation before seeing any operation by the other processes, then it must decide i. Thus the i-th corner of $\mathcal{IS}(\mathcal{S}^2)$ is colored with value i, for each i. Similarly in any execution where p_i, p_j see only each other's input, they have to decide either i or j. So all vertices along the edges ij must be colored with either i or j. All internal vertices must be colored with $0, 1$ or 3. It follows that this coloring scheme satisfies the conditions of

[2]BG-simulation is an elegant technique that allows a set of $f+1$ processes to wait-free simulate a larger system of n processes, that may also exhibit up to f stopping failures.

Sperner's lemma. So there must be a simplex in $\mathcal{IS}(\mathcal{S}^2)$ colored with all three colors, hence there is an execution where the all three processes decide on different values. This shows that set agreement is impossible for three processes. $\mathcal{Q.E.D}$.

6.4 Reduction between Renaming and Weak Symmetry Breaking

We now show an example of a topological reduction result. We state the following lemma without a proof [9]. Our goal is to show that WSB is equivalent to a special case of renaming.

Lemma 4 *Any subdivision of a simplex with a binary coloring on its vertices and with symmetric coloring on the boundary edges, must have a positive number of mono-chromatic (all* 0 *or all* 1*) in its interior.*

Notice that Lemma 4 is closely related to Sperner's lemma.

Consider a protocol complex that solves K-renaming. Each vertex of this complex is labeled with a process local state at the end of the execution, and hence has an associated output name. Consider relabeling each vertex of this complex with the parity of the output name (odd or even parity). The anonymity requirement implies that the parities of the renaming protocol induces a binary coloring that is symmetric on the boundary edges. So according to Lemma 4, the protocol complex has at-least one mono-chromatic triangle. Let $K = 2n - 1$. Then there cannot be any mono-chromatic simplex since there are a total of n output names with odd parity, and n with even parity. So this contradiction shows that $(2n - 1)$-renaming is impossible. Now, notice that a solution for $(2n - 1)$-renaming gives a solution to WSB. It can be shown that a solution to WSB also solves $(2n - 1)$-renaming, so these two tasks are equivalent. So, if WSB is not wait-free solvable, neither is $(2n - 1)$-renaming.

6.5 Separation between Weak Symmetry Breaking and Set Consensus

A *separation* result says that for two tasks T_1, T_2, task T_1 can be solved by a protocol for T_2, but not vice versa. The key topological idea here is to find a "magic" blackbox protocol complex that can solve one task, but not the other. Using this approach, we will show that set agreement is strictly stronger than WSB. The goal is to construct a manifold protocol complex that solves weak symmetry breaking. From the set consensus result, we know that manifolds cannot solve set consensus. So this will show that set agreement is harder than WSB.

Let us now construct the protocol complex. Take a standard chromatic subdivision, and two inverted copies of it. Join the two inverted complexes along any two edges of the first one, and then flip one of the boundary edges before merging both boundary edges. Since we are combining manifolds, the new complex is still a manifold. The resulting manifold is a *moebius* protocol complex. We can see that if we color the moebius complex with 0 and 1, we can always find a simplex with both 0 and 1 colors. So the moebius complex can solve WSB. However it cannot solve set consensus since it is a manifold. $\mathcal{Q.E.D}$.

6.6 General Results

6.6.1 The Asynchronous Computability Theorem

So far, we have seen characterizations specific to certain problems. We now turn our attention to more general results. Specifically, Herlihy and Shavit [20, 21, 19] proposed an elegant way to characterize wait-free computable tasks in 1993. The following theorem gives necessary and sufficient conditions for a decision task to be wait-free solvable in the read-write memory model.

Theorem 5 *A decision task $(\mathcal{I}, \mathcal{O}, \Delta)$, $\Delta \subset \mathcal{I} \times \mathcal{O}$, is wait-free solvable using read-write memory if and only if there exists a chromatic sub-division σ of $\mathcal{I}$ and a color-preserving map $\mu : \sigma(\mathcal{I}) \to \mathcal{O}$ such that for each simplex $S \in \sigma(\mathcal{I})$, $\mu(S) \in carrier(S, \mathcal{I})$.*

Notice that this is a purely topological criterion for wait-free solvability, independent of any model specific parameters. However, it is also an existence result, and does not provide us with an algorithm to solve the task. Unfortunately, it has been showed that wait-free computability is undecidable [14]. This led Borowsky and Gafni to propose an alternative criterion that could be convenient for specific tasks in distributed computing [7]:

Theorem 6 *A decision task $(\mathcal{I}, \mathcal{O}, \Delta)$, $\Delta \subset \mathcal{I} \times \mathcal{O}$, is wait-free solvable using read-write memory if and only if there exists an iterated standard chromatic sub-division χ^K of $\mathcal{I}$ and a color-preserving map $\mu : \chi^K(\mathcal{I}) \to \mathcal{O}$ such that for each simplex $S \in \chi^K(\mathcal{I})$, $\mu(S) \in carrier(S, \mathcal{I})$.*

Theorem 6 is more restrictive than Theorem 5, since the chromatic sub-division is replaced with the iterated standard chromatic subdivision. This makes Theorem 6 more convenient for proving impossibility results, especially the impossibility of set consensus turns out to be an easy consequence of it.

6.6.2 The Asynchronous Complexity Theorem

Hoest and Shavit [22] used algebraic topological techniques to derive time complexity bounds for approximate agreement for a generalization of the iterated immediate snapshot model. They related the time complexity of the task to the degree to which the input complex needs to be sub-divided before it can be mapped to the output complex. They also provide a general theorem for the time-complexity of wait-free tasks in the non-uniform iterated immediate snapshot (NIIS) model:

Theorem 7 *A decision task $(\mathcal{I}, \mathcal{O}, \Delta)$, $\Delta \subset \mathcal{I} \times \mathcal{O}$, is wait-free solvable in the NIIS model with worst-case time complexity k_S for a simplex $S \in \mathcal{I}$, if and only if there is a* mappable *non-uniform iterated chromatic sub-division with level k_S on S.*

We refer to [22] for detailed explanation and proof of the theorem. But it should be noticed that this complexity results only holds for the NIIS model, and does not carry over to the read-write model like Theorem 5.

7 Discussion and Open Research Problems

In this section we list some open research problems in this area. To the best of our knowledge, these issues have not been addressed in the existing literature.

1. So far we have only seen applications of topology for proving impossibility results. It could be interesting to see whether topology could be used to derive distributed algorithms. Such *topological algorithms* would have to efficiently extracted from protocol complexes, which tend to be pretty complicated for most canonical distributed systems problems.

2. Another important goal is better understanding of protocol complexes. Although protocol complexes can be complicated, usually we don't need to completely characterize a protocol complex to obtain a lower bound. This is because, most of the time all we need is to show that the protocol complex has a certain property. So partial characterizations of protocol complexes could be an interesting research problem [11]. Also, it might be possible to simplify the protocol complex by reducing the power of the adversarial scheduler using practical assumptions or by introducing randomness into the protocol complex. In this regard, it might be interesting to investigate the protocol complex structure for randomized consensus protocols [1].

3. It has been shown that multi-part equality (MEQ) computation is related to distance 2-edge coloring of bipartite graphs [23]. If we can find a connection between edge coloring and sperner's lemma, then we can investigate MEQ using the topological framework.

4. Applying topological methods to mutual exclusion lower bounds seems to be an interesting avenue for future research.

5. We can try to obtain alternative topological lower bound proofs for some of the existing impossibility results, for example, (1) The lower bound on the size of vector clocks, and (2) Lower bounds associated with hardware clock synchronization. Since these results are already proved using classical techniques, they have not been addressed using topological methods.

6. The hardness of reduction using topological methods seems to be an open research problem. More specifically, is there a topological way to show the hardness (or impossibility) of reducing one problem (say consensus) to another problem (say hardware clock synchronization)? So far, we have not seen any papers dealing with this. we know that although byzantine consensus and byzantine clock synchronization are related, their $3f + 1$ lower bound proofs are very different [3]. We can attempt to find topological characterizations of byzantine consensus ($\mathcal{T}_1$), and byzantine clock synchronization ($\mathcal{T}_2$), and try to discover some *topological obstruction* [18] between the protocol complexes associated with $\mathcal{T}_1$, and $\mathcal{T}_2$.

7. Although the time complexity results in [22] are derived for the iterated immediate snapshot model which is equivalent to the standard asynchronous model, the time complexity only holds for the snapshot model, and not for the equivalent asynchronous model. This gap provides an opportunity for future work.

8 Conclusion

In this survey paper, we have attempted to provide an overview of the topological approach to distributed computing. This paradigm makes it possible to exploit the extensive mathematical machinery that has accumulated since Poincare and others invented modern topology and its various branches to approach distributed systems problems.

References

[1] J. Aspnes, H. Attiya, and K. Censor. Randomized consensus in expected o(n log n) individual work. In *PODC*, 2008.

[2] H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, and R. Reischuk. Renaming in an asynchronous environment. *J. ACM*, 37, 1990.

[3] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. John Wiley & Sons, 2004.

[4] E. Borowsky. Capturing the power of resiliency and set consensus in distributed systems, 1995.

[5] E. Borowsky and E. Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC, 1993.

[6] E. Borowsky and E. Gafni. Immediate atomic snapshots and fast renaming. In *Proceedings of the twelfth annual ACM symposium on Principles of distributed computing*, PODC, 1993.

[7] E. Borowsky and E. Gafni. A simple algorithmically reasoned characterization of wait-free computation (extended abstract). In *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, PODC '97, 1997.

[8] E. Borowsky, E. Gafni, N. A. Lynch, and S. Rajsbaum. The bg distributed simulation algorithm. *Distributed Computing*, 14(3), 2001.

[9] A. Castañeda and S. Rajsbaum. New combinatorial topology bounds for renaming: the lower bound. *Distributed Computing*, 22(5-6), 2010.

[10] S. Chaudhuri. Agreement is harder than consensus: Set consensus problems in totally asynchronous systems. In *PODC*, 1990.

[11] F. E. Fich and E. Ruppert. Hundreds of impossibility results for distributed computing. *Distributed Computing*, 2003.

[12] M. J. Fischer. The consensus problem in unreliable distributed systems (a brief survey). In *FCT*, 1983.

[13] M. J. Fischer, N. A. Lynch, and M. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2), 1985.

[14] E. Gafni and E. Koutsoupias. 3-processor tasks are undecidable. In *Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing*, PODC '95, 1995.

[15] E. Gafni, S. Rajsbaum, and M. Herlihy. Subconsensus tasks: Renaming is weaker than set agreement. In *DISC*, 2006.

[16] M. Herlihy and S. Rajsbaum. Algebraic topology and distributed computing a primer. *Computer science today*, 1995.

[17] M. Herlihy and S. Rajsbaum. New perspectives in distributed computing. In *Proceedings of the 24th International Symposium on Mathematical Foundations of Computer Science*, MFCS, 1999.

[18] M. Herlihy, S. Rajsbaum, and D. Feichtner-Kozlov. Combinatorial topology and distributed computing, 2010.

[19] M. Herlihy and N. Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC, 1993.

[20] M. Herlihy and N. Shavit. A simple constructive computability theorem for wait-free computation. In *STOC*, 1994.

[21] M. Herlihy and N. Shavit. The topological structure of asynchronous computability. *J. ACM*, 46, 1999.

[22] G. Hoest and N. Shavit. Toward a topological characterization of asynchronous complexity. *SIAM J. Comput.*, 36(2), 2006.

[23] G. Liang and N. H. Vaidya. Multiparty equality function computation in networks with point-to-point links. In *SIROCCO*, 2011.

[24] J. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, 1984.

[25] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27, April 1980.

[26] M. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible: the topology of public knowledge. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC, 1993.

[27] A. J. Zomorodian, M. J. Ablowitz, S. H. Davis, E. J. Hinch, A. Iserles, J. Ockendon, and P. J. Olver. *Topology for Computing (Cambridge Monographs on Applied and Computational Mathematics).* Cambridge University Press, New York, NY, USA, 2005.