mathematics

MDPI

Article

A Topological Characterization to Arbitrary Resilient Asynchronous Complexity

Yunguang Yue [1], Xingwu Liu [1,2,*], Fengchun Lei [1,*] and Jie Wu [3]

[1] School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China; guangyy@dlut.edu.cn or gyyue1990@163.com
[2] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3] Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Beijing 101408, China; wujie@bimsa.cn
* Correspondence: liuxingwu@dlut.edu.cn (X.L.); fclei@dlut.edu.cn (F.L.)

Abstract: In this work, we extend the topology-based framework and method for the quantification and classification of general resilient asynchronous complexity. We present the *arbitrary resilient asynchronous complexity theorem*, applied to decision tasks in an iterated delayed model which is based on a series of communicating objects, each of which mainly consists of the delayed algorithm. In order to do this, we first introduce two topological structures, delayed complex and reduced delayed complex, and build the topological computability model, and then investigate some properties of those structures and the computing power of that model. Our theorem states that the time complexity of any arbitrary resilient asynchronous algorithm is proportional to the level of a reduced delayed complex necessary to allow a simplicial map from a task's input complex to its output complex. As an application, we use it to derive the bounds on time complexity to approximate agreement with $n+1$ processes.

Keywords: distributed computation; asynchronous computation; combinatorial topology; computability; complexity; resilience; task-solvability

MSC: 68Q05; 68Q22; 68W15; 68M14

check for updates

Citation: Yue, Y.; Liu, X.; Lei, F.; Wu, J. A Topological Characterization to Arbitrary Resilient Asynchronous Complexity. *Mathematics* **2022**, *10*, 2720. https://doi.org/10.3390/math10152720

Academic Editor: Pedro A. Castillo Valdivieso

Received: 4 July 2022
Accepted: 26 July 2022
Published: 1 August 2022

1. Introduction

Since the in-depth applications of computers to all aspects of modern life, computers are progressively and mainly being used as coordination devices in asynchronous distributed systems. However, the proof of the FLP theorem [1], which says that the consensus task cannot be solved in an asynchronous message passing system even though only one process may fail by halting, implies that distributed computing is different from the standard Turing computing.

A distributed computing system consists of finitely many sequential processes communicating via shared-memory, message-passing and other mechanisms [2]. The communicating mechanisms include communication channels, synchronizing primitives and general services [3]. The processes are asynchronous, which may make the message and communication delay leading to a great effect in Information Exchange, such as a hyper hexa-cell interconnection network [4]. In addition, they may also fail by stopping, so it is indistinguishable whether an irresponsive process has failed or is only running slowly in a distributed system.

Fortunately, as a revolutionary development, a new framework of modeling and analysis based on classical algebraic topology was introduced by Herlihy and Shavit [5] for understanding and reasoning of computability problems in an asynchronous distributed system in 1993. In that work, they presented a topological characterization of the asynchronous computability of general tasks with $t \geq 1$ crash failures in a share-memory model which is equivalent to a message-passing model showed by Attiya et al. [6]. Furthermore, they then extended to a complete characterization of wait-free solvability of distributed tasks in shared-memory systems, namely, a task is solvable if and only if its specification is topologically compatible in some sense [7,8]. Later, that technique was further generalized in three directions. The first direction is generalization to systems with arbitrary communication objects [9–13], to arbitrary resilience (rather than one or n failures) [9,14], to arbitrary synchrony [9,15], or to Byzantine failures [16,17]. The second direction is to explore the way for the classification of distributed tasks in asynchronous shared-memory systems: two tasks are reducible to each other if and only if they are equivalent in a topological manner [18–21]. The last is to explore of the complexity of decision tasks in some communicating model, that is to say, one can give the upper bounds and/or lower bounds [22–24], or just give a theoretical estimation of the cost of time or space and so on [25–32]. We feel the time is ripe to extend these techniques to arbitrary resilient asynchronous complexity.

This paper studies asynchronous shared memory solutions to the decision tasks in a distributed asynchronous system with $n+1$ processes. We focus on the *iterated delayed model* which is based on a delayed snapshot algorithm introduced by Saraph et al. in [33]. The model can guarantee that the scan operations of each process return a view that contain more than $n-t$ elements of sets of the participating processes' inputs, where t means the number of crashing processes it tolerates by t $(1 \leq t \leq n)$ or fewer for that system.

Our contribution lies in three aspects. Firstly, we introduce a topological structure, N-fold delayed complex, which is a sub-complex of the $2N$-fold standard chromatic subdivision, where N is an arbitrary non-negative integer. Using it can give a precise description of the protocol complex of full-information protocol leading to the acquisition of computing power, which is a modification of the result showed by Saraph et al. [33]. Secondly, we introduce a novel computational model, the iterated delayed model, which has a particularly nice geometric representation and hence easily lends itself to topological analysis. Furthermore, by a new topological structure, the reduced complex is constructed from a delayed complex, and we will give a theoretical measure of computational complexity of an arbitrary resilient asynchronous model, which may lead some applications in practical applications. That is, the time (or round) complexity in that model is equal to the level of the chromatic subdivisions necessary to allow a simplicial map from a task's input simplex corresponding to the worst case to its output complex. As an application, we derive the tight bound on the time to achieve an $n+1$ process approximate agreement in an iterated delayed model: $\lfloor log_m \frac{max\{input\ value\}-min\{input\ value\}}{\epsilon} \rfloor$ on any input n-simplex I, where $m=9$ if $n=1$ and $m=4$ if $n \geq 2$.

2. Preliminaries

In this subsection, we give an overview of the basic definitions and concepts of combinatorial topology that we will use to formulate our model. The complete definitions on algebraic topology can be taken from the classical textbooks [34,35].

2.1. Basic Concepts of Combinatorial Topology

An $(abstract)$ $simplicial$ $complex$ $\mathcal{K}$, or complex for short, is a finite set V together with a collection of subsets of V closed under containment, which means that, if $\alpha \in \mathcal{K}$, any subset of α is also in $\mathcal{K}$. An element v of V is called a vertex and an element α of $\mathcal{K}$ is called a simplex, and $\beta(\in \mathcal{K})$ is called a face of α if $\beta \subseteq \alpha$. Subset $\mathcal{K}$ of $\mathcal{K}$ is called a sub-complex if itself is closed under containment. The dimension of simplex α is defined as $||\alpha||-1$ (here $||\alpha||$ means the number of the vertices of α), denoted $dim(\alpha)$, and the dimension of $\mathcal{K}$ is defined as the highest dimension among its simplexes. Call simplex $\alpha \in \mathcal{K}$ a facet of $\mathcal{K}$ if $dim(\alpha) = dim(\mathcal{K})$. Usually, use n-simplex (complex) as shorthand for an n-dimensional simplex (complex). By default, any complex in this paper is pure in the sense that any simplex is a face of a facet in that complex. In addition, call $\mathscr{G}(X)$ the complex generated by X if it is a collection of X and all its faces, where X is a simplex in $\mathcal{K}$.

To ease understanding, one can equivalently view simplicial complex $\mathcal{K}$ through a geometric lens. Bijectively map the set $V(\alpha) = \{v_i\}_{i=0}^{k}$ of vertices of a k-simplex α of $\mathcal{K}$ to an arbitrary set $V'(\alpha) = \{v_i'\}_{i=0}^{k}$ of affinely independent points in a Euclidean space with an appropriate dimension. Then, there is natural convex hull $\bar{V}'(\alpha) = \{x \in \sum_{i=0}^{k} \lambda_i v_i' | \sum_{i=0}^{k} \lambda_i = 1, \lambda_i \geq 0, \lambda_i \in R\}$ spanned by $V'(\alpha)$. Call $\bar{V}'(\alpha)$ a geometric simplex or the geometric realization of α, and call α the vertex scheme of $\bar{V}'(\alpha)$. Put all geometric simplexes together such that the collection is closed under containment and every pair of distinct simplexes has disjoint interiors if they have intersections, by which we can obtain a geometric simplicial complex corresponding to $\mathcal{K}$. For the details, one can follow Munkres [35]. Nevertheless, hereunder we still adopt the abstract simplex complex definition.

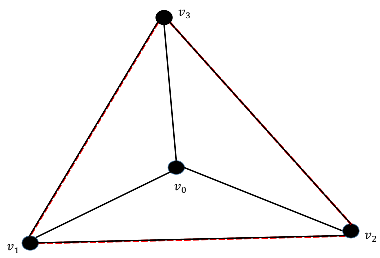


Figure 1. The star of v_0 in complex $\mathcal{K}$.

Let α and β be two simplexes of complex $\mathcal{K}$ with no intersection; then, the join of them is $\alpha * \beta$ with the vertex scheme being $\alpha \cup \beta$. The star of α is the collection of simplexes $\{\beta \in \mathcal{K} | \alpha \subseteq \beta\}$ of $\mathcal{K}$, denoted $St(\alpha, \mathcal{K})$. See Figure 1; consider a complex $\mathcal{K}$ consisting of all vertices $\{v_i\}_{i=0}^{3}$, and all segments $\{v_i, v_j\}_{0 \leq i \neq j \leq 3}$ and all triangles $\{v_i, v_j, v_r\}_{0 \leq i \neq j \neq r \leq 3}$; the star of 0-simplex v_0, $St(v_0, \mathcal{K})$ consists of all simplexes of $\mathcal{K}$ except the red dotted segments and its faces.

Given two complexes $\mathcal{K}$ and $\mathcal{L}$, a vertex map $f : V(\mathcal{K}) \longrightarrow V(\mathcal{L})$ is a simplicial map if f carries each simplex of $\mathcal{K}$ to a simplex of $\mathcal{L}$, and f is called non-collapsed if, for every simplex $\alpha \in \mathcal{K}$, there is $dim(f(\alpha)) = dim(\alpha)$, and f is called equivalent if there is a simplicial map $g : \mathcal{L} \longrightarrow \mathcal{K}$ such that $g \circ f : \mathcal{K} \longrightarrow \mathcal{K}$ and $f \circ g : \mathcal{L} \longrightarrow \mathcal{L}$ are all identity maps. A map $\Phi : \mathcal{K} \longrightarrow \mathcal{L}$ is called a carrier map if $\Phi(\alpha) \subseteq \mathcal{L}$ and $\Phi(\alpha \cap \beta) \subseteq \Phi(\alpha) \cap \Phi(\beta)$ for any $\alpha, \beta \in \mathcal{K}$. In addition, f is said to be carried by Φ if $f(\alpha) \in \Phi(\alpha)$ for any α. $\mathcal{K}$ is said to be a chromatic complex with colors C if there is a non-collapsed simplicial map χ from $\mathcal{K}$ to $\mathscr{C}$; call χ the coloring of $\mathcal{K}$. A map $\phi : \mathcal{K} \longrightarrow \mathcal{L}$ is said to be a color-preserving map if, for each vertex v of $\mathcal{K}$, there is $\chi_{\mathcal{K}}(v) = \chi_{\mathcal{L}}(\phi(v))$, where $\chi_{\mathcal{K}}$ and $\chi_{\mathcal{L}}$ are the colorings of complexes $\mathcal{K}$ and $\mathcal{L}$. In addition, a map is called a color-preserving simplicial map if it is both a color-preserving map and a simplicial map.

Suppose χ is a coloring of n-simplex X with colors $C = \{p_i\}_{i=0}^{n}$; call $Ch(X)$ once standard chromatic subdivision of simplex X if any m-simplex of $Ch(X)$ $(m \leq n)$ can be given by the form of $\{(p_{i_t}, S_{r_t})\}_{t=0}^{m}$ satisfying the following conditions:

- there are $2^n(n+1)$ vertices in total in $Ch(X)$;
- for all $t, s \in [m]$, $p_{i_t} \neq p_{i_s}$ if $s \neq t$, and either $S_{r_t} \subseteq S_{r_s}$ or $S_{r_s} \subseteq S_{r_t}$;
- for all $t, s \in [m]$, if $p_{i_t} \in \chi(S_{r_s})$, then $S_{r_t} \subseteq S_{r_s}$;

where S_{r_x} is a face of X, and $[m]$ is equal to $\{0, 1, 2, \ldots, m\}$.

As showed in Figure 2, on the left side of this figure is a chromatic 2-simplex $\alpha = \{v_0, v_1, v_2\}$ with colors $\{0, 1, 2\}$, and in the middle of it is a once standard chromatic subdivision $Ch(\alpha)$ of α, and on the right is a general chromatic subdivision $Div(\alpha)$ of α, since it is easy to check that it does not satisfy the second condition.

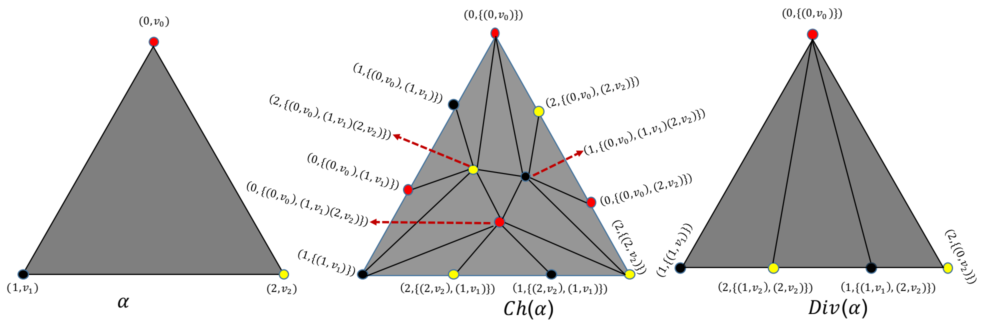


Figure 2. Once standard chromatic and general chromatic subdivision of 2-simplex α.

2.2. Distributed Computing Model

For the sake of the discussion, we first construct a series datatype for an arbitrary given datatype D. The datatype $\mathfrak{V}^k(D)$ (k is a non-negative integer) can be defined inductively as follows:

- $\mathfrak{V}^0(D) \triangleq \{(p_i, x_i) : x_i \in D \cup \{\perp\}, i \in [n]\}$;
- $\mathfrak{V}^k(D) \triangleq \{(p_i, s_i) : s_i \subset \mathfrak{V}^{k-1}(D) \cup \{\perp\}, i \in [n]\}$, where p_i can be regarded as the location for some data, $\perp$ means nothing but a placeholder, and (p_i, s_i) is a pair.

Following the model by Moses and Rajsbaum [15], there are $n+1$ processes, up to t of which may fail by crashing. The processes execute a round-by-round protocol in an asynchronous manner. A decision task is a specification of eligible outputs with regard to the inputs, which intuitively models coordination problems. A protocol is a distributed program consisting of the processes. We say that a protocol solves a decision task if the outputs of any execution sequence (which consists of a series of round executions) conform with the specification of the task.

Our distributed computing model is based on the delayed object, denoted $DO_{n,t}$, a delayed algorithm proposed by Saraph et al. [33], and has been showed to be a useful building block for analysis of a t-resilient distributed asynchronous system; see Algorithm A1. Although Delporte et al. [36] showed that a t-resilient immediate snapshot is impossible, two snapshots each in delayed objects are wait-free immediate snapshots which can be implemented in a t-resilient asynchronous distributed system [37].

Intuitively, $DO_{n,t}$ consists of three phases, the first and the third phases each are wait-free immediate snapshot (IS) operations, and the middle phase is a waiting operation. Before each process executes the second IS, it may need to wait until the view of some process contains at least $n-t+1$ pairs after the first IS. Formally, we can specify $DO_{n,t}$ as an I/O automata for $(n+1)$-processes with at most t processes crashing and datatype D, denoted by $DO_{n,t}[0]_{\mathfrak{V}^0(D)}$. Here, we only take $1 \le t < n$, since $t = n$ means the model is wait-free, and the complexity of that situation had been investigated by Hoest and shavit [30].

Our memory model is an iterated delayed model abbreviated as $\mathcal{IDM}$, in which each process communicates with the other processes only by delayed object $DO_{n,t}$, and every process accesses that object at most once in every round. It assumes an unbounded sequence of delayed objects $DO_{n,t}[0]_{\mathfrak{V}^{0\times 2}(D)}$, $DO_{n,t}[1]_{\mathfrak{V}^{1\times 2}(D)}$, $DO_{n,t}[2]_{\mathfrak{V}^{2\times 2}(D)}$ $\ldots$ with initial datatype D, abbreviated as $DO_{n,t}[0], DO_{n,t}[1], DO_{n,t}[2], \ldots$, if the datatype is explicit in the context. This model has each participating process proceed in ascending order in the sequence.

Suppose $P_{(n,t,\tau,\delta)}$, as in Algorithm 1, is a protocol in $\mathcal{IDM}$, and each process starts with a value coming from the datatype $\mathfrak{V}^m(D)$ in that model, where m is a non-negative integer and D is some given datatype. Before communicating with other processes in round r_i by object $DO_{n,t}[r_i]$, process p_i needs to check whether or not its local state ls_i can be decidable by a evaluated function $\tau : \bigcup_{i=0}^{\infty} \mathfrak{V}^{m+2i}(D) \to \{false, true\}$, and a decision map $\delta : \bigcup_{i=0}^{\infty} \mathfrak{V}^{m+2i}(D) \to \mathfrak{V}^0(D_O)$, where D_O is called the output datatype. If $\tau(ls_i)$ is *true*, the process can decide and execute a decision operation; otherwise, do nothing. In addition, *Sig* is taken as the signal that records whether or not a process has output before the current round, and it admits only two possible values 0 and 1, where $Sig = 0$ means the process does not have an output before the current round; otherwise, it means it already has an output. If $\tau_i(ls_i) = true$ and $Sig_i = 0$, the process p_i can make a decision but has no output; then, it decides and outputs $\delta_i(ls_i)$; Otherwise, it needs to access the object $DO_{n,t}[r_i]$ to accumulate information and then it accesses the next round with object $DO_{n,t}[r_i + 1]$.

Algorithm 1: An execution of a protocol $P_{(n,t,\tau,\delta)}$ for process p_i in $\mathcal{IDM}$

(1) $ls_i \leftarrow$ input value;
(2) $Sig_i \leftarrow 0$;
(3) $r_i \leftarrow 0$;
(4) **forever do**
(5) $\quad r_i \leftarrow r_i + 1$;
(6) $\quad$ **if** $\tau_i(ls_i) = true$ **and** $Sig_i = 0$
(7) $\quad\quad$ **then** output $\delta_i(ls_i)$ and $Sig_i \leftarrow 1$;
(8) $\quad ls_i \leftarrow DO_{n,t}[r_i].communicate(i, ls_i)$;
(9) **od**;

2.3. Topological Task Specification

In this subsection, we set some notions to make our statement straightforward. Let D_I be the set of input values and D_O the set of output values; then, any local state of the process p_i can be regarded as a pair $s_i = (p_i, v_i)$ such that $p_i = Ids(s_i)$, $v_i = val(s_i)$ and $s_i \in \mathfrak{V}^m(D_I)$, where m is a non-negative integer.

The input configurations for $n+1$ processes with input D_I are a chromatic n-complex, in which each k-simplex, $0 \leq k \leq n$, has form $\{s_i\}_{i=0}^k$, $s_i \in \mathfrak{V}^0(D_I)$—likewise for output configurations except $s_i \in \mathfrak{V}^0(D_O)$. Formally, a topological specification of a decision task with $n+1$ processes is a triple $T = (\mathcal{I}, \mathcal{O}, \Delta)$, where $\mathcal{I} = \mathfrak{V}^0(D_I)$ and $\mathcal{O} = \mathfrak{V}^0(D_O)$ are input n-complex and output n-complex, respectively, and $\Delta : \mathcal{I} \longrightarrow 2^{\mathcal{O}}$ is a name (color)-preserving carrier map [17,38].

3. The Topological Description of $\mathcal{IDM}$

This section introduces a topological structure, k-fold delayed complex, which can be showed a sub-complex of $2k$ times standard chromatic subdivision, and investigate their properties. By this, we give a characterization of computing power of $\mathcal{IDM}$, which is a modification of the results showed by Saraph et al. in [33].

3.1. Delayed Complexes

Suppose I^n is a chromatic n-simplex with the coloring map χ and the colors $C = \{p_i\}_{i=0}^n$, and $\mathcal{I}^n = \mathscr{G}(I^n)$ is the chromatic n-complex generated by I^n. $|I^n|$ and $||I^n||$ mean the geometric realization and the number of vertices of I^n, respectively.

Definition 1. *A $\mathscr{P}$-partition of a given finite set S is a partition $\pi_0, \pi_1, \ldots, \pi_m$ such that $S = \cup_{i=0}^m \pi_i$, $\pi_i \cap \pi_j = \emptyset$ for any $i \neq j \in [m]$, and $\pi_i \succ \pi_j$ if $0 \leq i < j \leq m$, where " $\succ$ " means some "priority" to the property $\mathscr{P}$.*

Lemma 1. *There is a one-to-one correspondence between the set of the facets of $Ch(I^n)$ and the collection of all $\mathscr{P}$-partitions of the set $C = \chi(I^n)$.*

Proof of Lemma 1. Suppose $\Pi = \pi_0, \pi_1, \ldots, \pi_m$ is a $\mathscr{P}$-partition of C. Let $\mathscr{P}$ be the containment in the complex $\mathcal{I}^n$, that is, if $\alpha \subset \beta$ for any simplexes $\alpha, \beta \in \mathcal{I}^n$, $\chi(\alpha) \succ \chi(\beta)$. Consider set $V = \{(p_i, S_i) | \ p_i \in \chi(S_i) \subseteq C, S_i \in \mathcal{I}^n\}$; there must be a subset V_Π of V with $||V_\Pi|| = n+1$, such that, for any two elements $(p_i, S_i), (p_j, S_j) \in V_\Pi$, there are $S_i = S_j$ if $p_i, p_j \in \pi_s$ for $s \in [m]$, and $S_i \subset S_j$ if $p_i \in \pi_s, p_j \in \pi_k$ for $0 \leq s < k \leq m$. In fact, it only needs to take the subset $\{S_i\}$ of $\mathcal{I}^n$, in which each simplex satisfies $\chi(S_i) = \cup_{t=0}^{i}\pi_t$. By Kozlov [39], V_π can span a facet of $Ch(I^n)$.

On the contrary, let $\alpha = \{(p_i, S_i)\}_{i=0}^{n}$ be a facet of $Ch(I^n)$, by the definition of standard chromatic subdivision of I^n. There is naturally a partition of the colorings set C under the containment in the set $\{S_i \subseteq I^n | (p_i, S_i) \in \alpha\}$. $\square$

By this lemma, any facet of a standard chromatic subdivision of a chromatic n-simplex can be represented by a $\mathscr{P}$-partition of its colors, which provides a simple way to describe a sub-complex structure of the standard chromatic subdivision. Suppose X is any facet of standard chromatic subdivision $Ch(I^n)$ corresponding to a $\mathscr{P}$-partition $\pi_0, \pi_1, \cdots, \pi_s$.

Definition 2. *A n-simplex is said to be a $\mathcal{C}$-extended of the facet X in $Ch(I)$, denoted by $Flip_i(X)$, if it can be given as follows:*

$$Flip_i(X) = \begin{cases} \{p_i\} \cup \pi_1, \pi_2, \ldots, \pi_s & \text{if } \pi_0 = \{p_i\}, \\ \{p_i\}, \pi_0 - \{p_i\}, \pi_1, \pi_2 \ldots, \pi_s & \text{if } ||\pi_0|| > 1 \text{ and } p_i \in \pi_0. \end{cases} \tag{1}$$

Call the collection $\{Flip_i(X) | p_i \in \pi_0\}$ the $\mathcal{C}$-extended of X, denoted by $Flip(X)$. Furthermore, call $Flip(X) \cup X$ the $\mathcal{C}$-neighborhood of X, denoted by $\widehat{X}$.

It is not hard to see that the $\mathcal{C}$-extended is symmetrical, that is, if X is a $\mathcal{C}$-extended of Y, Y is also a $\mathcal{C}$-extended of X. In addition, a $\mathcal{C}$-extended of a facet is usually not unique unless the cardinal number of the first component in its $\mathscr{P}$-partition is equal to one.

The $\mathcal{C}$-neighborhood describes a combinatorial adjacent relation of two facets in $Ch(I^n)$, which is not a neighborhood in the usual sense in topology or geometry. In general, a neighborhood of a facet in the triangulation of an n-manifold is equal to the union of facets in itself and its collar, which is indeed not its $\mathcal{C}$-neighborhood. Seeing Figure 3, let X be the black fields, the collection of all the facets of the star of $Ch^2(skel^0(I^2))$ in $Ch^2(I^2)$, the green areas are the $\mathcal{C}$-extended of X, and the $\mathcal{C}$-neighborhood is the collection of all the 2-simplexes colored black and green.

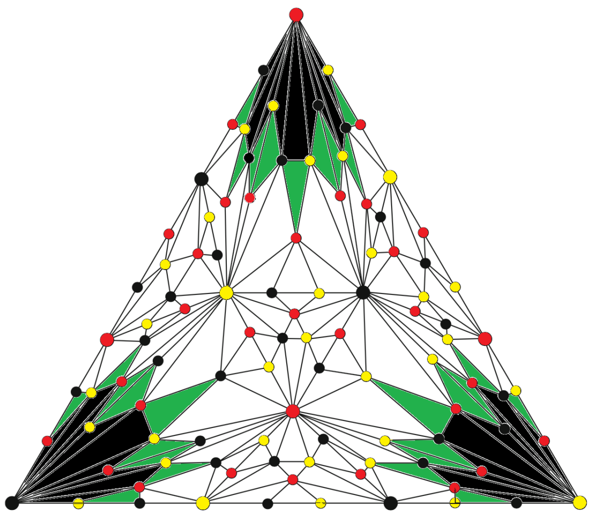


Figure 3. $\mathcal{C}$-extended and $\mathcal{C}$-neighborhood.

Lemma 2. *Suppose α is a facet of $Ch(I^2)$; then, a $\mathcal{C}$-extended $Flip_i(\alpha)$ is also a facet in $Ch(I^2)$.*

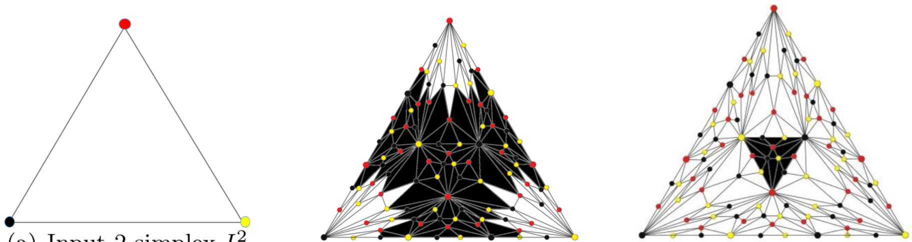
Proof of Lemma 2. This is a direct result of the Definition 2 and the Lemma 1. $\square$

For general discussion, let $\mathcal{K}$ be any pure chromatic n-complex with the coloring map χ and colors $C = \{p_i\}_{i\in[n]}$, and let $\mathbb{S}_{\mathcal{K}}$ be the collection of the facets of $St(Ch^2(skel^{n-t-1}(\mathcal{K}))$,

$Ch^2(\mathcal{K}))$, where t is an integer and $0 \le t < n$. The $\mathcal{C}$-neighborhood $\widehat{\mathbb{S}}_{\mathcal{K}}$ of $\mathbb{S}_{\mathcal{K}}$ is equal to $(\bigcup_{\alpha \in \mathbb{S}_{\mathcal{K}}} Flip(\alpha)) \bigcup \mathbb{S}_{\mathcal{K}}$. Set $C_{\mathbb{S}_{\mathcal{K}}} = \{\alpha \in Ch^2(\mathcal{K}) | \alpha \notin \widehat{\mathbb{S}}_{\mathcal{K}}, dim(\alpha) = n\}$.

Definition 3. *A subcomplex, denoted $\widehat{Ch}_{n,t}(\mathcal{K})$, of $Ch^2(\mathcal{K})$ is called the delayed complex about complex $\mathcal{K}$ if $\widehat{Ch}_{n,t}(\mathcal{K}) = \bigcup_{\alpha \in C_{\mathbb{S}_{\mathcal{K}}}} \mathscr{G}(\alpha)$, where $\mathscr{G}(\alpha)$ is the complex generated by simplex α. Inductively, the k-fold delayed complex about complex $\mathcal{K}$ can be defined as $\widehat{Ch}^k_{n,t}(\mathcal{K}) = \widehat{Ch}_{n,t}(\widehat{Ch}^{k-1}_{n,t}(\mathcal{K}))$.*

Example 1. *Consider an execution in delayed object $DO_{2,t}$ with an input 2-simplex $I^2 = \{(p_i, v_i)\}_{i=0}^2$, where the colors red, black, and yellow are corresponding to p_0, p_1, and p_2, and v_i $(i \in [2])$ is the input value of the process p_i, seeing the left side of Figure 4. Each 2-simplex colored black in the middle and the right of Figure 4 corresponds to a possible execution for $t = 1$ and $t = 0$, respectively. Comparing Figure 3 and the middle of Figure 4, it can be seen that each black 2-simplex in the middle of Figure 4 is a 2-simplex of $\widehat{Ch}_{n,1}(I^2)$.*



(a) Input 2-simplex I^2 (b) Executions in $DO_{2,1}$ (c) Executions in $DO_{2,0}$

Figure 4. Execution for three processes with input simplex I^2.

It is not hard to see that any two vertices can be linked by a path (a sequence of 1-simplexes) in $\widehat{Ch}_{2,1}(I^2)$, that is to say, it is 0-connected. In fact, by the similar arguments as Theorem 5 and Theorem 6 in [33], we can also show that $\widehat{Ch}^k_{n,t}(\mathcal{K})$ is $(t-1)$-connected, which means that any continuous map f from $(t-1)$-sphere S^{t-1} to the geometric realization $|\widehat{Ch}^k_{n,t}(\mathcal{K})|$ of $\widehat{Ch}^k_{n,t}(\mathcal{K})$ can be extended to a continuous map $\bar{f}$ from t-disk D^t to $|\widehat{Ch}^k_{n,t}(\mathcal{K})|$. Here, we do not re-show those topology properties of delayed complexes any more, while we focus on the relation between executions in a delayed object and delayed complex. Saraph et al. introduced a sub-complex of $Ch^2(I^n)$, denoted by $Ch_t(I^n)$, which is in fact the complementary of the star of $Ch^2(skel^{n-t-1}(I^n))$ in $Ch^2(I^n))$. In addition, they implied that there is one-to-one correspondence between a simplex in $Ch_t(I^n)$ and an execution of that object, which is actually inaccurate. There actually exists some simplex in $Ch_t(I^n)$ that does not correspond to any execution of object $DO_{n,t}$. The next results present some properties of $\mathcal{C}$-extended and $a\mathcal{C}$-neighborhood of the complex $\mathcal{K}$, by which will give an accurate geometric description of object $DO_{n,t}$.

Lemma 3. *Suppose $\alpha = \{(p_i, S_i)\}_{i \in [n]}$ is any facet of $\mathbb{S}_k$ with the $\mathscr{P}$-partition $\Pi_\alpha = \pi_0, \pi_1, \ldots, \pi_t$; then, there exists an integer m such that $||S_i|| \le n - t$ and $||S_i'|| \le n - t$ for any $x \in [m]$, any $p_i \in \pi_x$ and any $(p_i, S_i') \in S_i$, where $(p_i, S_i) \in \mathfrak{V}^2(val(V(\mathcal{K})))$, $(p_j, S_j') \in S_i \subseteq \mathfrak{V}^1(val(V(\mathcal{K})))$ and $m \le t$.*

Proof of Lemma 3. Let $\mathcal{A}_\alpha = \{\beta \in Ch^2(\mathcal{K}) \mid \beta \in Ch^2(skel^{n-t-1}(I^n)),\ \beta \subseteq \alpha \in \mathbb{S}_{\mathcal{K}}, I^n = carrier(\alpha, K)\}$; then, for any vertex $(p_i, S_i) \in V(\alpha)$, there are $(p_i, S_i) \in \mathfrak{V}^2(val(V(I^n)))$ and $(p_j, S_j') \in S_i \subset \mathfrak{V}^1(val(V(I^n)))$. Take a simplex, denoted $\mathcal{B}(\alpha)$, from $\mathcal{A}_\alpha$ with the maximal dimension, and it is obvious that $\mathcal{B}(\alpha)$ is unique. In addition, then there exists a $\mathscr{P}$-partition $\bar{\pi}_0, \bar{\pi}_i, \ldots, \bar{\pi}_s$ of $\mathcal{B}(\alpha)$, denoted $\Pi_{\mathcal{B}(\alpha)}$.

We claim that $\Pi_{\mathcal{B}(\alpha)}$ is the front of $s+1$ components of Π_α. If not, $\Pi_{\mathcal{B}(\alpha)}$ must be a successive truncation in the interior of Π_α, for $\mathcal{B}(\alpha)$ is a face of α. Hence, there exists at least one component π_x in Π_α but not in $\Pi_{\mathcal{B}(\alpha)}$, such that, for any element p_i in π_x with vertex

$(p_i, S_i) \in V(\alpha)$, there is $S_i \subset S_j$, where p_j is any element of an arbitrary component of $\Pi_{\mathcal{B}(\alpha)}$ with vertex $(p_j, S_j) \in V(\alpha)$. Therefore, $||S_i|| < ||S_j||$. Note that $dim(skel^{n-t-1}(I^n)) = n - t - 1$; then, $dim(\mathcal{B}(\alpha)) \leq n - t - 1$, and then $||S_j|| \leq ||\Pi_{\mathcal{B}}(\alpha)|| = \sum_{i \in [s]} ||\bar{\pi}_i|| \leq n - t$. Since $(p_i, S_i) \in \alpha$ but $(p_i, S_i) \notin \mathcal{B}(\alpha)$, there is $||S_i|| \geq n - t + 1$; then, $||S_i|| > ||S_j||$, which is an obvious contradiction. It follows that there is always $\bar{\pi}_i = \pi_i$ for any $0 \leq i \leq s$. Next, it needs to show that $m = s$ is a satisfied integer.

In fact, it has showed that $||val((p_i, S_i))|| = ||S_i|| \leq n - t$ for any element p_i in any component π_x with $0 \leq x \leq s$. Hence, it only needs to show that $||S'_j|| \leq n - t$ for any element $(p_j, S'_j) \in S_i$. If it is not true, there is a vertex $(p_i, S_i) \in V(\mathcal{B}(\alpha))$ such that there exists a vertex $(p_j, S'_j) \in S_i$ with $||S'_j|| \geq n - t + 1$. Since $\mathcal{B}(\alpha) \in Ch^2(skel^{n-t-1}(I^n))$, there must be a simplex $\beta \in Ch(skel^{n-t-1}(I^n))$ with $\mathcal{B}(\alpha) \in Ch(\beta)$. Note that $(p_i, S_i) \in V(\mathcal{B}(\alpha))$, and it follows that $(p_j, S'_j) \in V(\beta)$; then, $p_j \in \chi(\beta)$. By the construction of $\mathcal{B}(\alpha)$, there is $\chi(\beta) = \chi(\mathcal{B}(\alpha)) \subseteq \cup_{r \in [s]} \bar{\pi}_r$. Note that $dim(skel^{n-t-1}(I^n)) = n - t - 1$ and $S'_j \in skel^{n-t-1}(I^n)$, hence $||S'_j|| \leq dim(skel^{n-t-1}(I^n)) + 1 = n - t$, which is a contradiction. □

If we take I^n as an input n-simplex in an execution of $DO_{n,t}$, it is obvious that α corresponds to an impossible execution. Any process $p_i \in \bar{\pi}_0$ executing the second IS does not need to wait after the first IS leading to $||S'_i|| \geq n - t + 1$, which contradicts with Lemma 3. The next lemma implies that, if a facet is in $\mathcal{C}$-extended of the $\mathbb{S}_{\mathcal{K}}$, it corresponds to an impossible execution of object $DO_{n,t}$.

Lemma 4. *Suppose α is an arbitrary facet of $\mathbb{S}_{\mathcal{K}}$, and α' is any facet in $Flip(\alpha)$ with $\mathscr{P}$-partition $\Pi' = \bar{\pi}_0, \bar{\pi}_1, \ldots, \bar{\pi}_m$. Then, there exists an integer q such that, for any integer $x \in [q]$ and any element $p_i \in \bar{\pi}_x$, there is either $||S_i|| \leq n - t$ and $||S'_j|| \leq n - t$ for any vertex $(p_j, S'_j) \in S_i$, or $||S_i|| \geq n - t + 1$, and there exists at least one vertex $(p_j, S'_j) \in S_i$ with $||S'_j|| \leq n - t$, where (p_i, S_i) is a vertex of α'.*

Proof of Lemma 4. Assume the $\mathscr{P}$-partition of α is $\Pi = \pi_0, \pi_0, \ldots, \pi_p$. If $||\pi_0|| = 1$, there is only one element p_i in π_0, and then the $\mathscr{P}$-partition of α' can be represented as $\Pi' = \pi_0 \cup \pi_1, \pi_2, \ldots, \pi_p$ by Definition 2. Otherwise, $||\pi_0|| > 1$, and the $\mathscr{P}$-partition of α' is $\Pi' = \{p_i\}, \pi_0 - \{p_i\}, \pi_1, \ldots, \pi_p$, where p_i is an arbitrary element of π_0.

Set $\mathcal{A}_\alpha = \{\beta \in Ch^2(\mathcal{K}) \mid \beta \in Ch^2(skel^{n-t-1}(I^n)), \beta \subseteq \alpha \in \mathbb{S}_{\mathcal{K}}, I^n = carrier(\alpha, Ch(\mathcal{K}))\}$. Let $\mathcal{B}(\alpha)$ be a simplex of $\mathcal{A}_\alpha$ with the maximal dimension; then, $\mathscr{P}$-partition of $\mathcal{B}(\alpha)$ is the previous $s + 1$ components of Π by the same argument of the proof of Lemma 3, denoted $\Pi_{\mathcal{B}(\alpha)} = \pi_0, \pi_1, \ldots, \pi_s$, $0 \leq s < p$. Since $0 \leq dim(\mathcal{B}(\alpha)) \leq n - t - 1$, there is $1 \leq || \cup_{i \in [s]} \pi_i|| \leq n - t$.

If $|| \cup_{i=0}^{s} \pi_i|| = 1$, then $s = 0$, and there is only one element in π_0. Suppose it is p_0; then, $\mathscr{P}$-partition of $\mathcal{B}(\alpha)$ is $\pi_0 = p_0$. By Lemma 3, for any non-negative integer j, if $j = 0$, then $||S_0|| = 1 \leq n - t$ and $||S'_0|| \leq n - t$, where $(p_0, S'_0) \in S_0$. Otherwise, $j \neq 0$, and there is $||S_j|| \geq n - t + 1$ for the choice of α, where (p_j, S_j) is a vertex of α but not for $\mathcal{B}(\alpha)$, and $(p_0, S'_0) \in S_j$. By Definition 2, $\Pi' = \{p_0\} \cup \pi_1, \pi_2, \ldots, \pi_p$, then $p = m + 1$, $\bar{\pi}_i = \pi_{i+1}$ with $0 \leq i \leq m$, which implies that it only changes S_0 of the vertex (p_0, S_0) of α into $\bar{S}_0$ of the vertex $(p_0, \bar{S}_0)$ of α' with $\bar{S}_0 = S_x$, where (p_x, S_x) is any vertex of α with $p_x \in \pi_1$. It follows that $||S_j|| \geq n - t + 1$ and at least one vertex $(p_0, S'_0) \in S_j$ with $||S'_0|| \leq n - t$ for any $0 \leq q \leq m$, any component $\bar{\pi}_x$ with $0 \leq x \leq q$, and any element $p_j \in \bar{\pi}_x$, where (p_j, S_j) is a vertex of α'.

If $|| \cup_{i=0}^{s} \pi_i|| \geq 2$, then $s = 0$ or $s \geq 1$.

For the former case, the $\mathscr{P}$-partition of $\mathcal{B}(\alpha)$ is $\{p_0\}$. By Lemma 3, for any $p_i \in \pi_0$, there are $||S_i|| \leq n - t$ and $||S'_j|| \leq n - t$, where (p_j, S'_j) is an arbitrary element of S_i and $(p_i, andS_i)$ is a vertex of α. Hence, $\Pi' = \{p_i\}, \pi_0 - \{p_i\}, \pi_1, \ldots, \pi_p$, where $p_i \in \pi_0$; then, $\bar{\pi}_0 = \{p_i\}$, $\bar{\pi}_1 = \pi_0 - \{p_i\}$, $\bar{\pi}_2 = \pi_1, \ldots, \bar{\pi}_m = \pi_p$. Note that the transformation from α to α' is only changing S_i into $\bar{S}_i$, which contains only one element, where $(p_i, \bar{S}_i)$ is a vertex of α'. Let $q = 1$; then, for any $x \in [q]$ and any $p_r \in \bar{\pi}_x$, $p_r = p_i$ or $p_r \in \pi_0 - \{p_i\}$. It follows that $||\bar{S}_r|| = 1 \leq n-t$ or $||\bar{S}_r|| \leq n-t$, and $||\bar{S}'_j|| \leq n-t$, where $(p_r, \bar{S}_r)$ is a vertex of α' and $(p_j, \bar{S}'_j)$ is any element of $\bar{S}_r$.

For the latter case, the $\mathscr{P}$-partition of $\mathcal{B}(\alpha)$ is $\pi_0, \pi_1, \ldots, \pi_s$; then, either $\Pi' = \{p_i\}, \pi_0 - \{p_i\}, \pi_1, \ldots, \pi_p$ for $p_i \in \pi_0$ and $||\pi_0|| \geq 2$, or $\Pi' = \pi_0 \cup \pi_1, \pi_2, \ldots, \pi_p$ for $||\pi_0|| = 1$. For the former situation, let $q = s+1$, and, for the latter situation, let $q = s-1$. By the similar arguments as the situation $s = 0$, it can always obtain that, for any $0 \leq x \leq q$ and any $p_i \in \bar{\pi}_x$, there are $||S_i|| \leq n-t$ and $||S'_j|| \leq n-t$, where S_i is the value of the corresponding vertex of p_i in α', and (p_j, S'_j) is an arbitrary element of S_i. □

Through the discussion above, we can obtain the following result directly.

Corollary 1. *Suppose α is an n-simplex of $\widehat{\mathbb{S}}_{\mathcal{K}}$ with the $\mathscr{P}$-partition $\Pi = \pi_0, \pi_1, \ldots, \pi_m$; then, there is a non-negative integer q such that, for any $x \in [q]$ and any $p_i \in \pi_x$, there exists one vertex $(p_j, S'_j) \in S_i$ with $||S'_j|| \leq n-t$ and $(p_i, S_i) \in V(\alpha)$.*

3.2. The Characterization of $\mathcal{IDM}$

In the previous section, we have given the concepts of the k-fold delayed complex and $\mathcal{C}$-neighborhood, and then discussed about their properties. In this section, we will characterize $\mathcal{IDM}$ from a topological point of view.

Suppose ε is an execution in a protocol $\mathcal{P}$ with a given input n-simplex I^n; we say a simplex X is reachable by execution ε if it can take $V(X)$ as a possible return after executing ε.

Lemma 5. *Suppose $\mathcal{P}$ is the delayed algorithm in Algorithm A1 and I^n an input n-simplex, then the protocol complex $\mathcal{P}(I^n)$ is equivalent to delayed complex $\widehat{Ch}_{n,t}(I^n)$.*

Proof of Lemma 5. We only need to show that there is a one-to-one correspondence between the facets of $\mathcal{P}(I^n)$ and the facets of $\widehat{Ch}_{n,t}(I^n)$.

Suppose α is an arbitrary facet in $\mathcal{P}(I^n)$; then, there exists an execution ε_α in $\mathcal{P}$ such that α is reachable. Every process in ε_α just accesses object $DO_{n,t}$ once. That is, the process runs the first IS (denoted by IS_1), waits or not, and then runs the second IS (denoted by IS_2). Hence, it can model ε_α as a composite of ε^1_α and ε^2_α, each of which having a wait-free IS. Following Kozlov [39], every IS can be modeled as an ordered partition $\pi_0 \succ \pi_1 \cdots \succ \pi_m$ along the processes' running, such that processes in the same component run concurrently, and processes in π_i run after the processes in π_j for $j < i$. It follows that there are ordered partitions for ε^1_α and ε^2_α, denoted $\pi^1_0 \succ \pi^1_1 \cdots \succ \pi^1_{m_1}$ and $\pi^2_0 \succ \pi^2_1 \cdots \succ \pi^2_{m_2}$, respectively. Assuming $\{(p_i, ls^1_i)\}_{i\in[n]}$ is a return after executing IS_1, then ls^1_i is a face of I^n for any $i \in [n]$, and $ls^1_i \subset ls^1_j$ for $p_i \in \pi^1_s, p_j \in \pi^1_t$ and $s < t$, and $ls^1_i = ls^1_j$ for $p_i, p_j \in \pi^1_x$, where $s, t, x \in [m_1]$. Likewise, for a return $\alpha = \{(p_i, ls_i)\}_{i\in[n]}$ except after executing IS_2. Let $S_i = ls_i$ and $\bar{\alpha} = \{(p_i, S_i)\}_{i\in[n]}$; then, $\bar{\alpha}$ is a facet in $Ch^2(I^2)$. In the next, it only needs to show that $\bar{\alpha}$ is not a facet in a $\mathcal{C}$-neighborhood of an arbitrary facet of $St(Ch^2(skel^{n-t-1}(I^n)), Ch^2(I^n))$.

By steps 5 and step 6 in a delayed algorithm in Algorithm A1, we know that, if the number of elements of the view of a process is at most $n-t$ after finishing ε^1_α, it needs to wait until *done* is *true*. Assume $-1 \leq s < m_1$ is the largest integer such that all processes need to wait in π^1_x, $x \leq s$ (here, $s = -1$ means that all processes are wait-free). Then, any process in π^1_x should wait until at least one process in π^1_{s+1} has finished ε_α. Meanwhile, processes in π^1_y do not need to wait and immediately run IS_2 after IS_1, where $x \leq s$, $y \geq s+1$. It follows that, for any process p_i in π^2_0, it must appear in π^1_x with $s+1 \leq x \leq m_1$. Assume ls^1_i is the view of p_i after executing IS_1; then, $||ls^1_i|| \geq n-t+1$. Let $S'_i = ls^1_i$; then, $\alpha' = \{(p_i, S'_i)\}_{i\in[n]}$ is an n-simplex in $Ch(I^n)$ with $||S'_i|| \geq n-t+1$ for $p_i \in \pi^1_y, y \geq s+1$. Since $\pi^2_0, \pi^2_1, \ldots, \pi^2_{m_2}$ is actual, the $\mathscr{P}$-partition of α and $\alpha \in Ch(\alpha') \subset Ch^2(I^n)$, it follows that $\bar{\alpha} \notin (Ch^2(I^n) - \widehat{Ch}_{n,t}(I^n))$ by Corollary 1. As a result, $\bar{\alpha} \in \widehat{Ch}_{n,t}(I^n)$. Since α is arbitrary, $\mathcal{P}(I^n) \subseteq \widehat{Ch}_{n,t}(I^n)$ up to equivalence.

On the other hand, assume $\beta = \{(p_i, S_i)\}_{i\in[n]}$ is an arbitrary facet of $\widehat{Ch}_{n,t}(I^n)$. We will construct an execution ε in object $DO_{n,t}$, such that β is a reachable n-simplex for ε.

Since $\widehat{Ch}_{n,t}(I^n) \subset Ch^2(I^n)$, there exists only one facet $\beta' \in Ch(I^n)$ admitting form $\{(p_i, S'_i)\}_{i\in[n]}$ such that β is a facet in $Ch(\beta')$, where S'_i is a face of I^n for any $i \in [n]$. By Lemma 1, β' has a $\mathscr{P}$-partition under containment of the values of its vertices, denoted $\Pi' = \pi'_0, \pi'_1, \ldots, \pi'_m$, such that, for any vertex (p_i, S'_i), (p_j, S'_j), $S'_i \subset S'_j$ if $p_i \in \pi'_s$, $p_j \in \pi'_q$ and $s < q$, and $S_i = S_j$ if $p_i, p_j \in \pi'_x$, $x \in [m]$. Note that this partition corresponds to an execution IS, hence we can divide the first IS into a sequence of executions, denoted $\varepsilon' = \varepsilon'_0 \circ \varepsilon'_1 \cdots \varepsilon'_{m'}$, such that processes run ε'_i concurrently if they are in the same component π'_i, and processes in π'_i execute ε'_i after processes in π'_j running ε'_j for $i > j$. It not hard to see that β' is a reachable n-simplex after executing ε'. Likewise, the argument for the $\mathscr{P}$-partition of β except a series of executions $\varepsilon = \varepsilon_0 \circ \varepsilon_1 \cdots \varepsilon_q$ and β is a reachable n-simplex after executing ε. Let $\varepsilon_\beta = \varepsilon' \circ \varepsilon$. It only needs to show that ε_β is possible executing in object $DO_{n,t}$ with input I^n. Equivalently, it only needs to show that, for any vertex $(p_i, S'_i) \in \beta'$, if $|S'_i| \leq n - t$, p_i does not appear in π_0. That is, process p_i does not execute ε immediately after running ε', and it needs to wait until "done" is *true*.

Assume there exists a vertex (p_i, S'_i) in β' such that $p_i \in \pi'_0$ when $||S'_i|| \leq n - t$. By Corollary 1, β is an n-simplex of the $\mathcal{C}$-neighborhood of an facet in $St(Ch^2(skel^{n-t-1}(\mathcal{I}^n), Ch^2(\mathcal{I}^n))$, which is an obvious contradiction for $\beta \in \widehat{Ch}_{n,t}(I^n)$. As a result, ϵ_β is indeed an execution in $DO_{n,t}$ with input I^n.

Let ls_i be the view of process p_i after executing ε_β, then $ls_i = S_i$. It follows that β is indeed a reachable n-simplex after executing ε_β. Since β is arbitrary in $\widehat{Ch}_{n,t}(I^n)$, $\widehat{Ch}_{n,t}(I^n) \subseteq \mathcal{P}(I^n)$ up to equivalence. By the upper arguments, $\mathcal{P}(I^n)$ is equivalent to $\widehat{Ch}_{n,t}(I^n)$. □

Corollary 2. *Suppose I^n is an input n-simplex and $\mathcal{P}$ a protocol with k rounds in which each executes a delayed algorithm in Algorithm A1; then, the protocol complex $\mathcal{P}(I^n)$ is equivalent to k-fold delayed complex $\widehat{Ch}_{n,t}(I^n)$.*

Proof of Corollary 2. Execution ε in P can be modeled as a sequence of executions, denoted by $\varepsilon_1 \circ \varepsilon_2 \circ \cdots \circ \varepsilon_k$, in which each execution ε_i is an one round execution in delayed object $DO_{(n,t)}[i]$, and each return of a process after executing ε_i is as an input in execution ε_{i+1} for $0 \leq i < k$. By Lemma 5 and Definition 3, we know that $\mathcal{P}(I^n)$ is equivalent to $\widehat{Ch}^k_{n,t}(I^n)$. □

4. Measure Complexity

In this section, we introduce some concepts about the measure of complexity of arbitrary resilient $\mathcal{IDM}$, and then we give a topological characterization of the complexity for that model.

4.1. Complexity of the Delayed Model

Suppose $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is a decision task and $\mathcal{P}_{(n,t,\tau,\delta)}$ is a general algorithm solving T in $\mathcal{IDM}$. Let ε be an arbitrary execution of $\mathcal{P}_{(n,t,\tau,\delta)}$ with a legal input simplex I (i.e., $dim(I) \geq n - t$), and t_ε the maximum number among the effective access to object $DO_{n,t}$ for all participating processes in the execution ε with input I. Here, "effective" means the number of outputs is less than or equal to $n - t$ in an execution in a distributed system. In addition, from now on, once we refer to a protocol $\mathcal{P}_{(n,t,\tau,\delta)}$, it means that the protocol can aways solve some implied decision task.

Definition 4. *The time complexity of a protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ on a given input simplex I is the supremum of the set $\{t_\varepsilon | \varepsilon$ is any execution of the $\mathcal{P}_{(n,t,\tau,\delta)}$ with input $I\}$, denoted t_I.*

Remark 1. *Set $t_I = -1$ if an input simplex is illegal, since the dimension of that input simplex is less than $n - t$ and this time any participating process could not make a decision or output.*

Definition 5. *The time complexity of a protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ on a given input complex $\mathcal{I}$, denoted $t_{\mathcal{P}_{(n,t,\tau,\delta)}}$, is the supremum of the set $\{t_I | I \in \mathcal{I}^n\}$.*

Before we explore the complexity of a protocol in a delayed model, we introduce some properties of any execution in $DO_{n,t}$, which may be useful for the next discussing.

Lemma 6. *Any execution ε' in object $DO_{n,t}$ is finite.*

Proof of Lemma 6. Since any execution in $DO_{n,t}$ can be modeled as an execution of two-round executions of wait-free IS, even though there is a barrier layer between the two-round IS, it does not contain other actions except for waiting. Note that any execution of IS is finite by Hoest [30], then any execution of two-round IS is also finite. Hence, ε' is finite. □

Lemma 7. *Let ε be any execution of protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ for a task T in $\mathcal{IDM}$, then ε is finite.*

Proof of Lemma 7. Suppose there is no process stopping by crashing; then, ε contains at most $n+1$ starting actions and $n+1$ decision actions for there are at most $n+1$ processes and each of them has at most one initial state and one final state. Consider an integer $\bar{t}_\varepsilon$ which is the time complexity of ε; then, ε can be modeled as an ordered executing sequence with at most $\bar{t}_\varepsilon$ one-round execution of delayed object $DO_{n,t}$, denoted $\varepsilon_1, \varepsilon_2, \ldots,$ $\varepsilon_{\bar{t}_\varepsilon}$. Despite the fact that some processes may make a decision and output a value among those executions, there is no essential effect on the finiteness. By Lemma 6, each ε_i is finite. Note that $\bar{t}_\varepsilon$ is finite, hence ε is finite. □

4.2. Reduced Delayed Complex

Suppose $\mathcal{I}$ is a pure chromatic n-complex with coloring map χ and colors $C = \{p_i\}_{i\in[n]}$. Let X be an m-simplex in $\mathcal{I}$ and $\mathscr{G}(X)$ the complex generated by X; then, there is a facet $\bar{X}$ in $\mathcal{I}$ such that $X \subseteq \bar{X}$, $\chi(X) \subseteq \chi(\bar{X})$ and $\mathscr{G}(X) \subseteq \mathscr{G}(\bar{X})$. Let (S_X, A_X) be any partition of the set of vertices $V(X)$ of X with $A_X \neq \emptyset$, such that $V(X) = S_X \cup A_X$, $\chi(S_X) \cap \chi(A_X) = \emptyset$.

Definition 6. *Call $\mathcal{R}(X)$ a reduced delayed complex of X about $\chi(X)$ if*
(a) each simplex has form $C * T$, where $C \in \mathscr{G}(S_X) \bigcup \emptyset$, $T \in \widehat{Ch}_{n,t}(\bar{X}) \bigcup \emptyset$ and $\chi(T) \subseteq \chi(A_X)\}$;
(b) there exists one simplex $carrier(T, \bar{X})$ in $\bar{X}$ such that $C * carrier(T, \bar{X})$ is a face of X.
Furthermore, call $\mathcal{R}(\mathcal{I})$ a reduced delayed complex of $\mathcal{I}$ about C if $\mathcal{R}(\mathcal{I}) = \bigcup_{X\in\mathcal{I}} \mathcal{R}(X)$.

Remark 2.

- *C and T can not be empty-set at the same time.*
- *If $S_X = \emptyset$, then $\mathcal{R}(X)$ is a sub-complex of $\widehat{Ch}_{n,t}(\bar{X})$.*
- *$\mathcal{R}(X)$ is also a pure chromatic m-complex with the same set of colors as X.*
- *Suppose $(S_{\bar{X}}, A_{\bar{X}})$ is a partition of $\bar{X}$; then, there must be $S_X \subseteq S_{\bar{X}}$ and $A_X \subseteq A_{\bar{X}}$.*
- *Suppose Y is another simplex in $\mathcal{I}$ with a partition (S_Y, A_Y), and assume $X \cap Y = H \neq \emptyset$. If $v \in H$ and $v \in S_X$, then $v \in S_Y$, and vice versa; if $v \in H$ and $v \in A_X$, then $v \in A_Y$, and vice versa.*
- *In fact, it needs only to consider all the facets of complex $\mathcal{I}$ to construct the reduced delayed complex $\mathcal{R}(\mathcal{I})$; that is, $\bigcup_{X\in\mathcal{I}} \mathcal{R}(X) = \bigcup_{X\in\mathcal{I}, dim(X)=n} \mathcal{R}(X)$.*

Example 2. *Let $I^2 = \{v_0, v_1, v_2\}$ be a chromatic 2-simplex with coloring map χ, where $\chi(v_0) =$ red, $\chi(v_1) =$ black and $\chi(v_2) =$ yellow, seeing the left in Figure 5. Suppose $S_{I^2} = \{v_0\}$ and $A_{I^2} = \{v_1, v_2\}$; then, $V(\mathcal{R}(I^2))$ is a collection of v_0 and all the vertices coloring black and yellow in the black area of the middle figure in Figure 5. Reduced delayed complex $\mathcal{R}(I^2)$ can be obtained* *by pulling all vertices coloring red in the black area of the middle figure in Figure 5 over the flat and then pasting them together as just one vertex, seeing the right in Figure 5.*

Inductively, we can define a k-fold reduced delayed complex for n-complex $\mathcal{I}$, denoted $\mathcal{R}^k(\mathcal{I})$. For $k = 0$, it is equal to $\mathcal{I}$. For $k \geq 1$, it can be given by the following procedure: the vertices of $\mathcal{I}$ can be divided into two disjoint parts, denoted S_0 and A_0 with $A_0 \neq \emptyset$, respectively. Let $\bar{I} = \{X \in \mathcal{I} \mid dim(X) = n\}$; then, for any element $X \in \bar{I}$, there exists a partition $(S_{0,X}, A_{0,X})$ of X such that $S_{0,X} \subseteq S_0$, $A_{0,X} \subseteq A_0$ and $\chi(A_{0,X}) \subseteq \chi(X) = C$. Hence,

$$S_0 = \bigcup_{X \in \bar{I}} S_{0,X}, \; A_0 = \bigcup_{X \in \bar{I}} A_{0,X}.$$

Let $\mathcal{S}_0 = \bigcup_{X \in \bar{I}} \mathscr{G}(S_{0,X})$ be a sub-complex of complex $\mathcal{I}$, and $C_{A_0} = \chi(A_0)$, $C_{A_{0,X}} = \chi(A_{0,X})$. Any simplex in $\mathcal{R}^k(\mathcal{I})$ has form $C' * T'$, where $C' \in \mathcal{S}_0$, $T' \in \mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(\mathcal{I})_{C_{A_0}})$, $C' * carrier(T', \mathcal{I})$ is an n-simplex of $\mathcal{I}$, and $\widehat{Ch}_{n,t}(\mathcal{I})_{C_{A_0}} = \bigcup_{X \in \bar{I}} \widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}$. Here, $\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}$ represents a complex, in which each simplex α is a face of a simplex $\beta \in \widehat{Ch}_{n,t}(X)$ with $\chi(\alpha) \subseteq C_{A_{0,X}}$. It is not hard to show that $\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}$ is a pure sub-complex of the complex $\widehat{Ch}_{n,t}(X)$ with dimension $||C_{A_{0,X}}|| - 1$. By Definition 6, we can construct $\mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}})$ inductively.

Intuitively, each vertex of the reduced delayed complex $\mathcal{R}(X)$ with input simplex X records a local state, in which the process can decide and has no output. Without loss of generality, assume $X \in \mathcal{I}$ is a facet. For each process p_x, it needs to check whether its initial state can decide by the predicted function before accessing $DO_{n,t}[1]$. If it can decide and there is no output, then it will make a decision and give an output, and then the process p_x may continue executing in the following round or just stop for crashing; anyway, it has nothing to do with output of the process p_x. If it can not, it needs to accumulate more information by object $DO_{n,t}[1]$, which does have an effect on its output. Putting all initial states in which each process can decide together as a set S_X; it not hard to see that S_X can span a face of X. Since, for a process that is not in $\chi(S_X)$, it needs to access object $DO_{n,t}$ to obtain more information to decide, each local state of a process p_j with on output after executing object $DO_{n,t}[1]$ corresponds to a vertex in $V(\widehat{Ch}_{n,t}(X)_{C_{A_X}})$, where $A_X = V(X) - S_X$. By the Definition 6, $\mathcal{R}(X)$ indeed describes the one-round execution of $\mathcal{P}_{(n,t,\tau,\delta)}$. The next two lemmas show the general execution (i.e., there are many rounds in it) of $\mathcal{P}_{(n,t,\tau,\delta)}$.

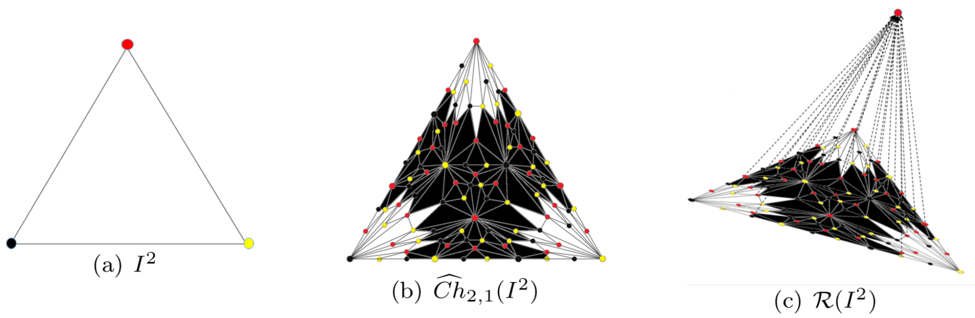


Figure 5. An intuitive procedure of 2-simplex I^2 to some reduced delayed complex.

Lemma 8. *Suppose $\mathcal{P}_{(n,t,\tau,\delta)}$ is a protocol in $\mathcal{IDM}$ with input n-complex $\mathcal{I}$. If the time complexity of $\mathcal{P}_{(n,t,\tau,\delta)}$ about $\mathcal{I}$ is k, then protocol complex $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I})$ is equivalent to a k-fold reduced delayed complex $\mathcal{R}^k(\mathcal{I})$ of $\mathcal{I}$, where k is non-negative integer.*

Proof of Lemma 8. Use induction to the complexity. Assume $t_{\mathcal{P}_{(n,t,\tau,\delta)}} = 0$, it is obvious that $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I}) \cong \mathcal{I}$. Note that $\mathcal{R}^0(\mathcal{I}) = \mathcal{I}$, hence $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I}) \cong \mathcal{R}^0(\mathcal{I})$. Suppose the conclusion is established for any non-negative integer $t_{\mathcal{P}_{(n,t,\tau,\delta)}}$, which is less than k. Consider $t_{\mathcal{P}_{(n,t,\tau,\delta)}} = k$.

Let $v_i = (p_i, x_i)$ be any vertex in $\mathcal{I}$, and χ and val the coloring map and assignment function for $\mathcal{I}$ such that $\chi(v_i) = p_i$ and $val(v_i) = x_i$. Note that whether the process p_i can make a decision and output a value in its initial state before accessing object $DO_{n,t}[1]$ or not depends on the value of predicted function τ on v_i. If $\tau(v_i)$ is $true$ and p_i does not output, p_i is decidable and outputs $\delta(v_i)$ before proceeding. Otherwise, process p_i can not make a decision, and it needs to accumulate information by accessing object $DO_{n,t}[1]$. It follows that τ divides $V(\mathcal{I})$ into two disjoint parts, denoted S_0 and A_0, respectively, such that $\tau(S_0) = true$ and $\tau(A_0) = false$. Since $k = t_{\mathcal{P}_{(n,t,\tau,\delta)}} \geq 1$, there is at least one vertex v_x in $V(\mathcal{I})$ such that $\tau(v_x) = false$, and then $A_0 \neq \emptyset$.

Consider an arbitrary facet $X \in \mathcal{I}$, such that the time complexity of the protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ about X is k. By the former argument, there must exist a partition of $V(X)$, denoted by $(S_{0,X}, A_{0,X})$, such that $S_{0,X} \subseteq S_0, A_{0,X} \subseteq A_0$ and $X = S_{0,X} * A_{0,X}$. Any execution of $\mathcal{P}_{(n,t,\tau,\delta)}$ can be modeled as a composite of an ordered sequence of one-round executions, in which each participating process accesses object $DO_{n,t}[r]$ at most once in round r, and the returns of concurrent round can be as the inputs to the next round. By Lemma 5, each return after executing round 1 is a simplex in $\widehat{Ch}_{n,t}(X)$; then, the time complexity of $\mathcal{P}_{(n,t,\tau,\delta)}$ on each simplex in $\widehat{Ch}_{n,t}(X)$ is at most $k-1$. By hypothesis,

$$\mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X)) \cong \mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(X)).$$

If $\chi(\bar{v}) \in \chi(S_{0,X})$ for a vertex $\bar{v} \in V(\widehat{Ch}_{n,t}(X))$, the process $\chi(\bar{v})$ has obtained an output before accessing object $DO_{n,t}[1]$. Hence, it only needs to focus on the sub-complex $\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}$ of complex $\widehat{Ch}_{n,t}(X)$, and there is

$$\mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}) \cong \mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}).$$

Let $\mathscr{G}(S_X) * \mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X))$ be the collection of all simplexes with form $\alpha * \beta$, where $\alpha \in \mathscr{G}(S_{0,X}) \cup \emptyset$, $\beta \in \widehat{Ch}_{n,t}(X_{C_{A_{0,X}}}) \cup \emptyset$, and α and β can not be empty sets at the same time. Since $\mathscr{G}(S_{0,X})$ and $\widehat{Ch}_{n,t}(X_{C_{A_{0,X}}})$ all are chromatic complexes and $\chi(\mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}})) \cap \chi(S_{0,X}) = \emptyset$, $\mathscr{G}(S_X) * \mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X))$ is a chromatic complex. Likewise, $\mathscr{G}(S_{0,X}) * \mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}})$ is also a chromatic complex.

It follows that

$$\mathscr{G}(S_X) * \mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X)) \cong \mathscr{G}(S_{0,X}) * \mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}}).$$

Note that $\mathcal{P}_{(n,t,\tau,\delta)}(X) = \mathscr{G}(S_{0,X}) * \mathcal{P}_{(n,t,\tau,\delta)}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}})$ and $\mathcal{R}^k(X) = \mathscr{G}(S_{0,X}) * \mathcal{R}^{k-1}(\widehat{Ch}_{n,t}(X)_{C_{A_{0,X}}})$. Hence, $\mathcal{P}_{(n,t,\tau,\delta)}(X) \cong \mathcal{R}^k(X)$. Since X is arbitrary facet in $\mathcal{I}$, there is

$$\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I}) = \bigcup_{X \in \mathcal{I},\, dim(X)=n} \mathcal{P}_{(n,t,\tau,\delta)}(X).$$

Therefore,

$$\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I}) \cong \bigcup_{X \in \mathcal{I},\, dim(X)=n} \mathcal{R}^k(X).$$

By Definition 6, there is

$$\bigcup_{X \in \mathcal{I},\, dim(X)=n} \mathcal{R}^k(X) = \mathcal{R}^k(\mathcal{I}).$$

It follows that $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I}) \cong \mathcal{R}^k(\mathcal{I})$. $\square$

Lemma 9. *Suppose $\mathcal{I}$ is a pure chromatic n-complex with coloring map χ, and $\mathcal{R}^k(\mathcal{I})$ is a k-fold reduced delayed complex about $\mathcal{I}$, where k is a non-negative integer. Then, there exists a protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ in $\mathcal{IDM}$ with time complexity k about complex $\mathcal{I}$, such that its protocol complex is equivalent to $\mathcal{R}^k(\mathcal{I})$.*

Proof of Lemma 9. Assume we have obtained the predicted function τ and the decision map δ in the delayed system with $n+1$ processes, then we can construct a protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ in that model in the way shown as in Algorithm A1. By Lemma 8, the protocol complex $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I})$ is indeed equivalent to complex $\mathcal{R}^k(\mathcal{I})$. Next, we mainly focus on constructing the predicted function τ and the decision map δ.

Consider an arbitrary facet X in n-complex $\mathcal{I}$. By Definition 6 and the fundamental configuration process of k-fold reduced delayed complex $\mathcal{R}^k(X)$, there is a sequence of complexes, denoted $X, \mathcal{R}(X), \mathcal{R}^2(X), \ldots, \mathcal{R}^k(X)$, respectively, and two corresponding complex sequences, denoted $\mathcal{S}_{0,X}, \mathcal{S}_{1,X}, \mathcal{S}_{2,X}, \ldots, \mathcal{S}_{k-1,X}$ and $\mathcal{A}_{0,X}, \mathcal{A}_{1,X}, \mathcal{A}_{2,X}, \ldots, \mathcal{A}_{k-1,X}$, respectively. Such that, for any $i, j \in [k-1]$, if $i \neq j$, there are $\chi(\mathcal{S}_{i,X}) \cap \chi(\mathcal{S}_{j,X}) = \emptyset$ and $\chi(\mathcal{S}_{i,X}) \cap \chi(\mathcal{A}_{i,X}) = \emptyset$, and, for any $r \in [k-2]$, there is $\chi(\mathcal{A}_{r,X}) = \chi(\mathcal{S}_{r+1,X}) \cup \chi(\mathcal{S}_{r+1,X})$.

Note that, for any $i \in [k-1]$, $V(\mathcal{S}_{i,X})$ and $V(\mathcal{A}_{i,X})$ are the subsets of $\mathfrak{V}^{2i}(D_{val(X)})$. Define function

$$\tau_X : \bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(X)}) \longrightarrow \{true,\ false\},$$

satisfying if $v \in V(\mathcal{S}_{i,X}), i \in [k-1]$, then $\tau_X(v) = true$, otherwise $\tau_X(v) = false$. Since $V(\mathcal{S}_{i,X}) \cap V(\mathcal{S}_{j,X}) = \emptyset$ for any $i \neq j \in [k-1]$, τ_X is well-defined. By the similar way, we can define a color-preserving simplicial map

$$\delta_X : \bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(X)}) \longrightarrow \mathfrak{V}^0(D_{val(\mathcal{O})}),$$

such that, for any $v \in \bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(X)})$, if $\tau_X(v) = true$, then $\delta_X(v) = (\chi(v), x)$, otherwise $\delta_X(v) = (\chi(v), \perp)$, where x is a value in $val(\mathcal{O})$ and $\perp$ is just a placeholder. It is obvious that δ_X is also well-defined.

Note that $\mathcal{I}$ is pure and X is arbitrary in complex $\mathcal{I}$, and let

$$\tau = \sum_{X \in \mathcal{I}, dim(X)=n} \tau_X,\ \delta = \sum_{X \in \mathcal{I},\ dim(X)=n} \delta_X.$$

Next, we only need to check that τ and δ are well-defined. Assume Y is another facet in complex $\mathcal{I}$ and $X \cap Y = C \neq \emptyset$. By Definition 6 and the construction of τ_X and τ_Y, there is

$$\tau_X(\bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(C)})) = \tau_Y(\bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(C)})),$$

which implies that τ is well-defined. Therefore, τ is exactly a satisfied predicted function from $\bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(\mathcal{I})})$ to $\{true,\ false\}$. In a similar way, it can also show that δ is well-defined and a satisfied decision map from $\bigcup_{i \in [k-1]} \mathfrak{V}^{2i}(D_{val(\mathcal{I})})$ to $\mathfrak{V}^0(D_{\mathcal{O}})$. $\square$

4.3. Arbitrary Resilient Asynchronous Complexity Theorem

The strength and usefulness of $\mathcal{IDM}$ of computation comes from the fact that each of its associated protocol complexes has a slightly nice, recursive structure. In fact, it turns out that any protocol complex of $\mathcal{IDM}$ is equivalent to a reduced delayed complex which can be constructed by iterated chromatic subdivision of the input complex, and vice versa. This is the essence of our main theorem, which we state and prove in this section. The level of subdivision necessary for the existence of a simplicial map from the input to the output complex of a decision task that agrees with the task specification can be interpreted as a topological measure of the task's time complexity.

Let $T = (\mathcal{I}, \mathcal{O}, \Delta)$ be a decision task and k a non-negative integer. We say $\mathcal{R}^k(\mathcal{I})$ is a mappable reduced subdivision of the input complex and k is a mappable level of reduced subdivision if there exists a color-preserving simplicial map μ from $\mathcal{R}^k(\mathcal{I})$ to $\mathcal{O}$ such that, for all $X \in \mathcal{R}^k(\mathcal{I})$, $\mu(X) \in \Delta(carrier(T, \mathcal{I}))$.

Theorem 1. *A decision task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ has a t-resilient solvable protocol in $\mathcal{IDM}$ with worst case time complexity k_X on legal inputs $X \in \mathcal{I}$ if and only if there is a mappable reduced subdivision $\mathcal{R}^k(\mathcal{I})$ with level k_X on X.*

Proof of Theorem 1. Herlihy and Shavit in [40] imply that decision task T has a t-resilient solvable protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ in $\mathcal{IDM}$ if and only if the following triangle is commutative, where μ is a color and carrier preserving simplicial map from $\mathcal{P}_{(n,t,\tau,\delta)}$ to $\mathcal{O}$.

$$\begin{tikzcd}
& \mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I}) \arrow[d, "\delta"] \\
\mathcal{I} \arrow[ur, "\mathcal{P}_{(n,t,\tau,\delta)}"] \arrow[r, "\Delta"] & \mathcal{O}
\end{tikzcd}$$

By Lemma 8, any protocol complex $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I})$ is equivalent to a k-fold reduced delayed complex $\mathcal{R}^k(\mathcal{I})$. The protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ solves task T in $\mathcal{IDM}$ with worst case time complexity k_X on X. It follows that $k \geq k_X$. Let k be equal to k_X and μ' a color-preserving isomorphism from $\mathcal{R}^k(\mathcal{I})$ to $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I})$, and let $\mu = \delta \circ \mu'$. Then, μ is a color and carrier preserving simplicial map from $\mathcal{R}^k(\mathcal{I})$ to $\mathcal{O}$. It follows that $\mathcal{R}^k(\mathcal{I})$ is mappable.

On the other hand, any mappable reduced subdivision $\mathcal{R}^k(\mathcal{I})$ with level k_X on X is equivalent to the protocol complex $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I})$ of a protocol in the delayed model with worst case complexity k_X on legal input X by Lemma 9. Assume γ is a color-preserving simplicial map from $\mathcal{P}_{(n,t,\tau,\delta)}(\mathcal{I})$ to $\mathcal{R}^k(\mathcal{I})$. If there is a color and carrier preserving simplicial map μ from $\mathcal{R}^k(\mathcal{I})$ to $\mathcal{O}$, by setting $\delta = \mu \circ \gamma$, then protocol $\mathcal{P}_{(n,t,\tau,\delta)}$ can solve T in a delayed model with worst case time complexity k_X on legal inputs $X \in \mathcal{I}$. □

Remark 3. *When $t = n$, $\mathcal{IDM}$ is actually a wait-free non-standard iterated immediate snapshot model(NIIS) proposed by Hoest and Shavit [30]. As a corollary of Theorem 1, it can easily obtain the wait-free asynchronous complexity theorem.*

5. Application

In this section, we will analyze the time complexity of the well-known *Approximate Agreement* task as an application of Theorem 1. An approximate agreement $T = (\mathcal{I}, \mathcal{O}, \Delta)$ in a distributed system with $n+1$ processes can be specified as follows: each n-simplex $I \in \mathcal{I}$ has form $\{(p_i, v_i)\}_{i \in [n]}$, where each v_i is in a finite set S of real numbers, and p_i is the ID of the process. In addition, each n-simplex $O \in \mathcal{O}$ has the form $\{(p_i, w_i)\}_{i \in [n]}$, such that, for any $i \in [n]$, w_i is also in S; and, for any $i, j \in [n]$, there is $|w_i - w_j| < \epsilon$, where ϵ is the same predetermined number that is more than 0, and $(I, O) \in \Delta$ if $val(O) \subseteq val(I) \cup \{\perp\}$.

For a wait-free model, Hoest et al. [30] have showed that the time complexity of approximate agreement is $\lfloor log_d \frac{max\{val(I)\} - min\{val(I)\}}{\epsilon} \rfloor$ for $d = 3$ if $n = 1$ and $d = 2$ if $n \geq 2$ with any input simplex I. Here, we investigate the time complexity of approximate agreement in $\mathcal{IDM}$. Since any legal input simplex α_I, there is always a facet $\alpha \in \mathcal{I}$ such that α_I is a face of α, and they admit the same time complexity. Therefore, it only needs to consider all the facets in the input complex $\mathcal{I}$.

In order to make the next discussion brief, let $D(\alpha) = max\{val(\alpha)\} - min\{val(\alpha)\}$ for any chromatic simplex α, and $D(\mathcal{K}) = max_{\alpha \in \mathcal{K}} D(\alpha)$ for a chromatic complex $\mathcal{K}$. In addition, let χ be the coloring map with the collection of all the IDs of the processes.

Theorem 2. *Suppose $\epsilon > 0$ is a predetermined number and $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is an approximate agreement, then there is a protocol $\mathcal{P}$ solving T in $\mathcal{IDM}$ with time complexity $\lfloor log_m \frac{D(I)}{\epsilon} \rfloor$ on any input n-simplex I, where $m = 9$ if $n = 1$ and $m = 4$ if $n \geq 2$.*

Proof of Theorem 2. Assume task T has protocol $\mathcal{P}$ with the time complexity k_I on input n-simplex $I \in \mathcal{I}$. By Theorem 1, there is a color-preserving simplicial map

$$\psi : \mathcal{R}^k(\mathcal{I}) \longrightarrow \mathcal{O},$$

carried by Δ, such that $k = k_I$ if ψ acts on the sub-complex $\mathcal{R}^k(I)$ of the complex $\mathcal{R}^k(\mathcal{I})$.

Relabel all of the vertices of $\mathcal{R}^k(\mathcal{I})$ by associating map ψ in the following way. For each vertex $v \in \mathcal{R}^k(\mathcal{I})$, label v with $val(\psi(v))$ while retaining its coloring. It is obvious that this labeling satisfies the task specification, since, for any n-simplex $\alpha \in \mathcal{I}$ that contains v, the labeling $val(\psi(v))$ must be in $val(\alpha)$. Note that ψ is a simplicial map from $\mathcal{R}^k(\mathcal{I})$ to $\mathcal{O}$; then,

$$D(\psi(\alpha)) < \epsilon$$

for any $\alpha \in \mathcal{R}^k(\mathcal{I})$. Since $D(\alpha) = D(\psi(\alpha))$ and α is chosen arbitrary, then

$$D(\mathcal{R}^k(\mathcal{I})) < \epsilon.$$

Consider an arbitrary n-simplex $I \in \mathcal{I}$. By the Definition 6, $(l+1)$-fold reduced delayed complex $\mathcal{R}^{l+1}(I)$ can be obtained from an l-fold reduced delayed complex $\mathcal{R}^l(I)$. That is, $V(\mathcal{R}^{l+1}(I))$ can be divided into two disjoint sets of vertices $S_{l,I}$ and $A_{l,I}$, such that $S_{l,I} \subseteq V(\mathcal{R}^l(I)) \subseteq V(\mathcal{R}^k(I))$ and $A_{l,I} \subseteq V(\widehat{Ch}^{2l}_{n,t}(I))$ and each simplex of $\mathcal{R}^l(I)$ can be spanned by a subset of $S_{l,I}$ and a subset of $A_{l,I}$. Hence, $D(\mathcal{R}^l(I)) = D(\mathscr{G}'(A_{l,I}))$, where $\mathscr{G}'(A_{l,I})$ is the complex generated by $A_{l,I}$ in which each simplex in $\widehat{Ch}^{2l}_{n,t}(I)$. By the similar argument of the proof of Theorem 6.1 [30], there is

$$D(\mathcal{R}^k(I)) \geq \frac{D(I)}{d^{2k_I}},$$

where k_I is the complexity on I, and $d = 3$ if $n = 1$ and $d = 2$ if $n \geq 2$. Then,

$$\frac{D(I)}{d^{2k_I}} < \epsilon.$$

Let $m = d^2$, then

$$\lfloor log_m \frac{D(I)}{\epsilon} \rfloor \leq k_I,$$

where $n = 1$ if $m = 9$ and $m = 4$ if $n \geq 2$. Therefore, $\lfloor log_m \frac{D(I)}{\epsilon} \rfloor$ is a lower bound of the time complexity on simplex I.

On the other hand, suppose there is a color and carrier preserving simplicial map ψ from k-fold reduced delayed complex $\mathcal{R}^k(\mathcal{I})$ to $\mathcal{O}$, such that $k = \lfloor log_m \frac{D(I)}{\epsilon} \rfloor$ if ψ acts on the sub-complex $\mathcal{R}^k(I)$ of the complex $\mathcal{R}^k(\mathcal{I})$ for any input facet $I \in \mathcal{I}$, where $m = 9$ if $n = 1$ and $m = 4$ if $n \geq 2$. By Theorem 1, there is a protocol that solves approximate agreement T with the worst case time complexity $\lfloor log_m \frac{D(I)}{\epsilon} \rfloor$ on input n-simplex I, which implies that $\lfloor log_m \frac{D(I)}{\epsilon} \rfloor$ is also an upper bound of the time complexity on input I.

In fact, the existence of that map ψ is equivalent to that there exists a labeling of the vertices of $\mathcal{R}^k(\mathcal{I})$ which agrees with the initial values of input complex $\mathcal{I}$ with $D(\mathcal{R}^k(\mathcal{I})) < \epsilon$. By definition 6, we know that k-fold reduced delayed complex $\mathcal{R}^k(\mathcal{I})$ is recursive, hence we can label each vertex of $\mathcal{R}^k(\mathcal{I})$ at each fold as follows. Consider any facet I of $\mathcal{I}$; for each fold $1 \leq l \leq k$, a vertex v is in $A_{l-1,I}$ if there exists a facet α in $\widehat{Ch}^{l}_{n,t}(I)$ taking v as one of its vertices, such that $|val(v) - val(w)| > \epsilon$; otherwise, v is in $S_{l-1,I}$, where w is a vertex in α. Assume it has relabelled all vertices of $\widehat{Ch}^{l}_{n,t}(I)$ but with colors in $\chi(A_{l-1,I})$. Before entering the next fold, it needs to relabel all vertices of $\widehat{Ch}^{l}_{n,t}(I)_{C_{A_{l-1,I}}}$ but in $S_{l,I}$. Consider any facet α in $\widehat{Ch}^{l-1}_{n,t}(I)$. At first, relabel all vertices of $Ch^2(\alpha)$ as follows: If $dim(\alpha) = 1$, label the new vertices with values $\frac{M_l+8\cdot m_l}{9}, \frac{2\cdot M_l+7\cdot m_l}{9}, \cdots, \frac{8\cdot M_l+m_l}{9}$, such that $D(Ch^2(\alpha)) = \frac{M_l-m_l}{9}$; otherwise, label new vertices with $\frac{M_l+3\cdot m_l}{4}, \frac{2\cdot M_l+2\cdot m_l}{4}, \frac{3\cdot M_l+m_l}{4}$, such that $D(Ch^2(\alpha)) = \frac{M_l-m_l}{4}$, where $M_l = max\{val(\alpha)\}, m_l = min\{val(\alpha)\}$. It follows that $D(\widehat{Ch}_{n,t}(\alpha)) \leq D(Ch^2(\alpha)$ for the su-bcomplex $\widehat{Ch}_{n,t}(\alpha)$ of $Ch^2(\alpha)$. By this construction, we can see that, if $D(\mathcal{R}^l(I)) > \epsilon$, either $D(\mathcal{R}^{l+1}(I)) < \epsilon$ or $D(\mathcal{R}^{l+1}(I)) = \frac{D(\mathcal{R}^l(I))}{m}$, where $m = 9$ *or* 4, which implies that, for any facet $I \in \mathcal{I}$, $\mathcal{R}^k(I)$ is as a $\lfloor log_m \frac{D(I)}{\epsilon} \rfloor$-fold reduced sub-complex of $\mathcal{R}^k(\mathcal{I})$ for $m = 9$ if $dim(I) = 1$ and $m = 4$ if $dim(I) \geq 2$. □

Remark 4. *We only consider the input simplex which is a facet, since, for any legal input simplex α, there always exists a facet $\alpha_0(\supseteq \alpha)$ such that they have the same time complexity for it only needs to assign the processes that are in $\chi(\alpha_0)$ but not in $\chi(\alpha)$ with the same value as any value in $val(\alpha)$.*

6. Conclusions and Future Work

This paper modifies the description of a protocol complex of a delayed algorithm proposed by Saraph, Herlihy, and Gafni in [33], in which they thought the protocol complex was equivalent to $Ch_t(I^n)$. However, we give a precisely topological characterization of that algorithm and show that the protocol complex is equivalent to $\widehat{Ch}_{n,t}(I^n)$, which is a proper subset of complex $Ch_t(I^n)$, which implies there exists a simplex in complex $Ch_t(I^n)$ corresponding to an impossible execution in $DO_{n,t}$.

Take $DO_{n,t}$ as a black-box and then construct an iterated delayed model ($\mathcal{IDM}$), which turns out to be an arbitrary resilient algorithm in an asynchronous distributed system. Even though Delporte et al. [36] have showed that a t-resilient immediate snapshot is impossible in a t-resilient asynchronous distributed system by showing the equivalence of consensus ($2t - n + 1$-set agreement) and a t-resilient immediate snapshot for $t < (n+1)/2$ ($t \geq (n+1)/2$), there exist some tasks that admit weaker computing power than consensus and set agreement. In addition, each immediate snapshot in $\mathcal{IDM}$ is actually a wait-free immediate snapshot from a local point of view, which turns out to be implemented in a t-resilient asynchronous distributed system [37] but not for a t-immediate snapshot. Hence, it makes sense for the construction of $\mathcal{IDM}$.

This paper gives a topological characterization of the time complexity of an arbitrary resilient distributed asynchronous model theoretically by the number of subdivisions of the worst input case. Although the topological structures are very complex and difficult to construct, it turns the dynamic analysis of that system into a static geometric structure analysis, which may lead to a great significance in practical applications. Using that characterization gives a time complexity of approximate agreement in t-resilient $\mathcal{IDM}$, in which it can obtain the time complexity of approximate agreement in a wait-free asynchronous model by taking $t = n$, showed by Hoest et al. [30].

Unfortunately, any model involved in this paper refers to only a crashing failure model, which is an ideal failure model. Meanwhile, the most common failure model is the Byzantine failure model, in which the behavior of a byzantine process is out of control, leading to difficulty in analyzing its complexity in a distributed asynchronous system. In addition, in order to obtain the characterization of the complexity of the Byzantine failure model, it may be necessary to obtain more information during an execution, not just the initial states and terminal states. However, the main technique of our work is based on the method and framework proposed by Herlihy et al. [8,18,40,41], in which it is not necessary to care about the intermediate process during an execution but the initial states and terminal states.

In the future, we will construct a new geometric structure, the directed topological model, which is proposed mainly to be used to study concurrent computation [42], and from which it may catch up not only the initial states and terminal states but also almost all information during an execution. Investigating the topological characterization, such as the invariant of directed homotopy or directed homology, and then exploring the relationship between the computational complexity and those topological characterizations, may give us a feasible way to study Byzantine complexity.

Author Contributions: Conceptualization, Y.Y. and X.L.; Formal analysis, Y.Y., X.L., and F.L.; Methodology, Y.Y. and J.W.; Writing—original draft, Y.Y.; Writing—review and editing, X.L. and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work is partially supported by the National Natural Science Foundation of China (Grant No. 12071051, 62072433), partially supported by National Natural Science Foundation of China 647 (Grant No. 11971144), and partially supported the Fundamental Research Funds for the Central Universities (Grant No. DUT21RC(3)102) and partially supported High-level Scientific Research Foundation of Hebei Province and the start-up research fund from BIMSA.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Symbol	Corresponding Meaning
$\mathcal{K}, \mathcal{I}$	(abstract) simplicial complexes
X, α, β, γ	(abstract) simplexes
$V(X)/V(\mathcal{K})$	the set of vertices of simplex X / simplicial complexes $\mathcal{K}$
$\|\|X\|\|$	the number of vertices (or elements) of simplex (or set) X
$\|X\|$	the geometric realization of simplex X
$\mathscr{G}(X)$	the complex generated by simplex X
$DO_{n,t}$	delayed object consists of $n+1$ processes with t-resilience
$DO_{n,t}[r]$	invoking delayed object $DO_{n,t}$ in r-th round execution
$\mathfrak{V}^k(D)$	the datatype generated by k-fold iterations with data D
$Flip(X)$	$\mathcal{C}$-extended of simplex X
$\widehat{X}$	$\mathcal{C}$-neighborhood of simplex X
$\widehat{Ch}^k_{n,t}(\mathcal{I})$	k-fold delayed complex about n-simplicial complex $\mathcal{I}$
$\mathcal{R}^k(\mathcal{K})$	k-fold reduced delayed complex about n-simplicial complex $\mathcal{I}$
$\mathcal{IDM}$	iterated delayed model

Appendix A. Delayed Algorithm

Algorithm A1: Delayed algorithm [33]

(1) **shared** $mem0[n+1]$, $mem1[n+1]$, done;
(2) done $\leftarrow$ false;
(3) **immediate**
(4) $mem0[id] \leftarrow input_{id}$;
(5) $snap0[id] \leftarrow snapshot(mem0)$;
(6) **if** $|snap0[id]| \leq n - t$ **then**
(7)   **while** *not done*
(8)   skip
(9) **immediate**
(10) $men1[id] \leftarrow snap0[id]$;
(11) $snap1 \leftarrow snapshot(mem1)$;
(12) $done \leftarrow ture$;
(13) **return** $snap1$;

References

1. Fischer, M.J.; Lynch, N.A.; Paterson, M.S. Impossibility of distributed consensus with one faulty process. In Proceedings of the Second ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, Atlanta, GA, USA, 21–23 March 1983; pp. 1–7.
2. Lynch, N. *Distributed Algorithms*; Morgan Kaufmann: San Francisco, CA, USA, 1996.
3. Herlihy, M. Wait-free synchronization *ACM Trans. Program. Lang. Syst.* **1991**, *13*, 124–149. [CrossRef]
4. Mahafzah, B.A.; Al-Zoubi, I. Broadcast communication operations for hyper hexa-cell interconnection network. *Telecommun. Syst.* **2018**, *67*, 73–93. [CrossRef]
5. Herlihy, M.; Shavit, N. The asynchronous computability theorem for t-resilient tasks. In Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 16–18 May 1993; pp. 111–120.
6. Attiya, H.; Amotz, B.N.; Dolev, D. Sharing memory robustly in message-passing systems. *J. ACM* **1995**, *42*, 124–142. [CrossRef]
7. Herlihy, M.; Rajsbaum, S. Set consensus using arbitrary objects (preliminary version). In Proceedings of the Thirteenth Annual ACM Symposium on Principles of Distributed Computing, Los Angeles, CA, USA, 14–17 August 1994; pp. 324–333.
8. Herlihy, M.; Rajsbaum, S. Algebraic spans (preliminary version). In Proceedings of the Fourteenth Annual ACM Symposium on Principles of Distributed Computing, Ottawa, ON, Canada, 20–23 August 1995; pp. 90–99.
9. Herlihy, M.; Rajsbaum, S. The topology of distributed adversaries. *Distrib. Comput.* **2013**, *26*, 173–192. [CrossRef]
10. Herlihy, M.; Rajsbaum, S.; Raynal, M.; Stainer, J. From wait-free to arbitrary concurrent solo executions in colorless distributed computing. *Theor. Comput. Sci.* **2017**, *683*, 1–21. [CrossRef]
11. Herlihy, M.; Rajsbaum, S.; Tuttle, M.R. Unifying synchronous and asynchronous message-passing models. In Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing, PODC '98, Puerto Vallarta, Mexico, 28 June–2 July 1998; ACM: New York, NY, USA, 1998; pp. 133–142.
12. Kuznetsov, P.; Rieutord, T.; He, Y. An asynchronous computability theorem for fair adversaries. *CoRR* **2020**, abs/2004.08348.
13. Yue, Y.G.; Lei, F.C.; Liu, X.W.; Wu, J. Asynchronous computability theorem in arbitrary solo models. *Mathematics* **2020**, *5*, 757. [CrossRef]
14. Gafni, E.; Kuznetsov, P.; Manolescu, C. A generalized asynchronous computability theorem. In Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC'14, Paris, France, 15–18 July 2014; pp. 222–231.
15. Moses, Y.; Rajsbaum, S. Toward a topological characterization of asynchronous complexity. *SIAM J. Comput.* **2002**, *31*, 989–1021. [CrossRef]
16. Nicolas, B.S.; Guerraoui, R.; Florian, H. Fast byzantine agreement. In Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC'13, Montreal, QC, Canada, 22–24 July 2013; pp. 57–64.
17. Mendes, H.; Tasson, C.; Herlihy, M. Distributed computability in byzantine asynchronous systems. *CoRR* **2013**, abs/1302.6224.
18. Herlihy, M.; Rajsbaum, S. The decidability of distributed decision tasks (extended abstract). In Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, TX, USA, 4–6 May 1997; pp. 589–598.
19. Herlihy, M.; Rajsbaum, S. A classification of wait-free loop agreement tasks. *Theor. Comput. Sci.* **2003**, *291*, 55–77. [CrossRef]
20. Liu, Y.W.; Xu, Z.W.; Pan, J.Z. Classifying rendezvous tasks of arbitrary dimension. *Theor. Comput. Sci.* **2009**, *410*, 2162–2173. [CrossRef]
21. Yue, Y.G.; Lei, F.C.; Wu, J. The evolution of non-degenerate and degenerate rendezvous tasks. *Topol. Its Appl.* **2019**, *264*, 187–200. [CrossRef]
22. Biran, O.; Moran, S.; Zaks, S. Tight bounds on the round complexity of distributed 1-solvable tasks. *Theor. Comput. Sci.* **1995**, *145*, 271–290. [CrossRef]

23. Castañeda, A.; Rajsbaum, S. New combinatorial topology bounds for renaming: The lower bound. *Distributed Comput.* **2010**, *22*, 287–301. [CrossRef]
24. Castañeda, A.; Rajsbaum, S. New combinatorial topology bounds for renaming: The upper bound. *J. ACM* **2012**, *59*, 3. [CrossRef]
25. Abraham, I.; Dolev, D. Byzantine agreement with optimal early stopping, optimal resilience and polynomial complexity. In Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, Portland, OR, USA, 14–17 June 2015; pp. 605–614.
26. Attiya, H.; Guerraoui, R.; Hendler, D.; Kuznetsov, P. The complexity of obstruction-free implementations. *J. ACM* **2009**, *56*, 24:1–24:33. [CrossRef]
27. Galil, Z.; Mayer, A.J.; Yung, M. Resolving message complexity of byzantine agreement and beyond. In Proceedings of the 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WN, USA, 23–25 October 1995; pp. 724–733.
28. Georgiou, C.; Russell, A.; Shvartsman, A.A. The complexity of synchronous iterative do-all with crashes. *Distributed Comput.* **2004**, *17*, 47–63. [CrossRef]
29. Helmi, M.; Higham, L.; Pacheco, E.; Woelfel, P. The space complexity of long-lived and one-shot timestamp implementations. *J. ACM* **2014**, *61*, 7. [CrossRef]
30. Hoest, G.; Shavit, N. Toward a topological characterization of asynchronous complexity. *SIAM J. Comput.* **2006**, *36*, 457–497. [CrossRef]
31. Chordia, S.; Rajamani, S.K.; Rajan, K.; Ramalingam, G.; Vaswani, K. Asynchronous Resilient Linearizability. In *Distributed Computing, Proceedings of the 27th International Symposium, DISC 2013, Jerusalem, Israel, 14–18 October 2013*; Lecture Notes in Computer Science; Afek, Y., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8205, pp. 164–178.
32. Choudhury, A.; Patra, A. Optimally Resilient Asynchronous MPC with Linear Communication Complexit. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking, ICDCN 2015, Goa, India, 4–7 January 2015*; Das, S.K., Krishnaswamy, D., Karkar, S., Korman, A., Kumar, M.J., Portmann, M., Sastry, S., Eds.; ACM: New York, NY, USA, 2015; p. 5.
33. Vikram, S.; Herlihy, M.; Gafni, E. Asynchronous Computability Theorems for t-Resilient Systems. In *Distributed Computing, Proceedings of the 30th International Symposium, DISC 2016, Paris, France, 27–29 September 2016; Lecture Notes in Computer Science*, Cyril, G.; David, I., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9888; pp. 428–441.
34. Kozlov, D.N. *Combinatorial Algebraic Topology*; Algorithms and Computation in Mathematics; Springer: Berlin/Heidelberg, Germany, 2008, Volume 21.
35. Munkres, J.R. *Elements of Algebraic Topology*; Addision-Wesley: Reading, MA, USA, 1984.
36. Delporte, C.; Fauconnier, H.; Rajsbaum, S.; Raynal, M. t-resilient immediate snapshot is impossible. In *Structural Information and Communication Complexity, Proceedings of the 23rd International Colloquium, SIROCCO 2016, Helsinki, Finland, 19–21 July 2016*; Revised Selected Papers; Lecture Notes in Computer Science; Jukka, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 177–191.
37. Borowsky, E.; Gafni, E. A simple algorithmically reasoned characterization of wait-free computations (extended abstract). In Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara, CA, USA, 21–24 August 1997; pp. 189–198.
38. Herlihy, M.; Rajsbaum, S. Concurrent computing and shellable complexes. In Proceedings of the Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, 13–15 September 2010; pp. 109–123.
39. Kozlov, D.N.; Shavit, N. Combinatorial topology of the standard chromatic subdivision and weak symmetry breaking for 6 processes. *CoRR* **2015**, abs/1506.03944.
40. Herlihy, M.; Shavit, N. The topological structure of asynchronous computability. *J. ACM* **1999**, *46*, 858–923. [CrossRef]
41. Herlihy, M.; Kozlov, D.N.; Rajsbaum, S. *Distributed Computing through Combinatorial Topology*, 1st ed.; Morgan Kaufmann: San Francisco, CA, USA, 2013.
42. Fajstrup, L.; Raußen, M.; Goubault, E. Algebraic topology and concurrency. *Theor. Comput. Sci.* **2006**, *357*, 241–278. [CrossRef]