

A General Input-Dependent Colorless Computability Theorem and Applications to Core-Dependent Adversaries

Yannis Coutouly 

Aix-Marseille University, CNRS LIS UMR7020, Marseille, France

Emmanuel Godard 

Aix-Marseille University, CNRS LIS UMR7020, Marseille, France

Abstract

Distributed computing tasks can be presented with a triple $(\mathcal{I}, \mathcal{O}, \Delta)$. The solvability of a colorless task on the Iterated Immediate Snapshot model (IIS) has been characterized by the Colorless Computability Theorem [11, Th.4.3.1]. A recent paper [7] generalizes this theorem for any message adversaries $\mathcal{M} \subseteq IIS$ by geometric methods.

In 2001, Mostéfaoui, Rajsbaum, Raynal, and Roy [19] introduced *condition-based adversaries*. This setting considers a particular adversary that will be applied only to a subset of input configurations. In this setting, they studied the k -set agreement task with condition-based t -resilient adversaries and obtained a sufficient condition on the conditions that make k -Set Agreement solvable.

In this paper we have three contributions:

1. We generalize the characterization of [7] to *input-dependent* adversaries, which means that the adversaries can change depending on the input configuration.
2. We show that core-resilient adversaries of IIS_n have the same computability power as the core-resilient adversaries of IIS_n where crashes only happen at the start.
3. Using the two previous contributions, we provide a necessary and sufficient characterization of the condition-based, core-dependent adversaries that can solve k -Set Agreement.

We also distinguish four settings that may appear when presenting a distributed task as $(\mathcal{I}, \mathcal{O}, \Delta)$. Finally, in a later section, we present structural properties on the carrier map Δ . Such properties allow simpler proof, without changing the computability power of the task. Most of the proofs in this article leverage the topological framework used in distributed computing by using simple geometric constructions.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Distributed algorithms; Theory of computation $\rightarrow$ Computability

Keywords and phrases colorless task, topological methods, geometric simplicial complex, k -set-agreement, t -resilient model, condition-based computability

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2025.13

Funding This work was partially supported by DUCAT (Distributed Network Computing through the Lens of Combinatorial Topology – France ANR-20-CE48-0006).

Acknowledgements The authors wish to sincerely thank Sergio Rajsbaum for suggesting condition-based computability as an application of [7].

1 Introduction

1.1 Topological Methods and Computability Theorems

Since the initial work of Herlihy and Shavit [15], Saks and Zaharoglou [24], and Borowsky and Gafni [4], showing that distributed computability questions are amenable to topological methods, many important applications have been demonstrated. This framework describes



© Yannis Coutouly and Emmanuel Godard;

licensed under Creative Commons License CC-BY 4.0

29th International Conference on Principles of Distributed Systems (OPODIS 2025).

Editors: Andrei Arusoaie, Emanuel Onica, Michael Spear, and Sara Tucci-Piergiovanni; Article No. 13; pp. 13:1–13:21



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

distributed problems as tasks $(\mathcal{I}, \mathcal{O}, \Delta)$. $\mathcal{I}$ is a colored simplicial complex of all the input configurations, $\mathcal{O}$ is a colored simplicial complex with all the output configurations and Δ is a relation that specifies, for any input, which outputs can be accepted. Simplicial complexes proved to be a convenient mathematical tool to represent distributed situations. In this setting, the most used model of communication is the Iterated Immediate Snapshot (*IIS*) model, since one round of computation is simply represented by a Standard Chromatic Subdivision (see [11] for a detailed presentation). One of the biggest computability results around these methods is the Asynchronous Computation Theorem (ACT) which says that a task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable if and only if there is a simplicial map from an iterated Standard Chromatic Subdivision of $\mathcal{I}$ to $\mathcal{O}$. There also exists a simpler, and more powerful, version for colorless tasks (tasks where the specification doesn't need the name of processes), which states [11, Th. 4.3.1]: a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on the *IIS* model if and only if there is a continuous map from $|\mathcal{I}|$ to $|\mathcal{O}|$ respecting Δ . Such computability theorems aim to provide tight characterization for general tasks and allow proofs leveraging simplicial or topological arguments for particular tasks. One major line of research extends this colorless computability theorem to many other models of communication.

1.2 The general message adversary line of work

For this paper, we restrict our attention to the message adversary setting where the communication model has a round structure, and each round corresponds to a communication graph between the processes involved. Moreover, the set of possible graphs can vary between each round, making this a very flexible setting. In particular, the *IIS* model can be expressed in this setting, as well as any subset of executions of the *IIS* model. A lot of work has been done to obtain simplicial understanding of many models, from oblivious message adversaries to core adversaries. The failure pattern has become more refined. For some examples see [14], [12] or [23, 18]. Recently, an extension called *general message adversary* has been proposed (see [9], [21], [6], [3], [7]). This approach can investigate any subset of executions of *IIS*. In particular, “non-compact” sets of executions can be considered. A natural example is the t -resilient adversary: one will eventually get a message from $n - t$ processes, but the time where a process might be silent can get arbitrarily long.

The “non-compact problem” was uncovered with [9] where they obtained a combinatorial and topological characterization of the consensus for 2-processes that observe that, for computability, some executions seem to work in (special) pairs. This was generalized for any number of processes in [6] with the introduction of the *geometrization* topology that interprets special pairs as a non-separable point, in the classical topological meaning. Another direction is [21] that provides a characterization of the solvability of Consensus for any number of processes for the general model of computation using an abstract topology on the set of executions. Then [3] provides a colored version of an ACT-like theorem for general message adversaries, using terminating subdivision and another ACT-like theorem for general models of computation, using the abstract topology. Later, [7] came with a colorless computability theorem for general message adversaries, using again terminating subdivision and the *geometrization* topology.

This article is built upon [7], mostly because this approach to geometrization enables to consider a simple execution space that is mostly like a $\mathbb{R}^N$ space. This makes possible simple geometric reasoning as is demonstrated here. The main Theorem of [7] is the following. Given $(\mathcal{I}, \mathcal{O}, \Delta)$ a colorless task, it is solvable on $M \subseteq IIS_n$ if and only if there is a continuous function $f : geo(skel^n I \times \mathcal{M}) \rightarrow |\mathcal{O}|$ carried by Δ , where *geo* is the geometrization mapping. We can remark that, in this statement, the message adversary is independent of

the input values. In this paper we introduce the *input-dependent* setting of general message adversary of IIS and provide a similar computability theorem for such model in Th. 16. An *input-dependent* adversary can present executions that are different depending on the input configuration, it is a subset $\mathcal{A}$ of $\mathcal{I} \times IIS_n$. The characterization is as follows, a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable under $\mathcal{A}$ if and only if there is a continuous function $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ carried by Δ ,

1.3 Application to Open Problems

This new computability theorem on general message adversaries is an interesting extension in itself, we also provide applications. A contribution of this paper answers an open question from [19] about condition-based adversaries that were introduced by Mostéfaoui, Rajsbaum, Raynal, and Roy. This setting considers the solvability only for a subset of input configurations. In [19], the authors investigated the k -set agreement problem within a t -resilient model. They prove that if $C \subseteq \mathcal{I}$ enables to solve the task, there is a simplicial function from a specific complex, denoted $Kin(C, f, k)$, to $\mathcal{O}$. Also, two conditions are proposed on the set of inputs that make k -Set Agreement solvable for a t -resilient model. Here, we propose an alternate simplicial construction $\mathcal{U}(C)$ (that is actually geometrically related to $Kin(C, f, k)$) such that k -Set Agreement is solvable on a core-dependent model $\mathcal{H}$ if and only if there is a simplicial map from $\mathcal{U}(\mathcal{H}(C))$ to $\mathcal{O}$. Moreover, we show that knowing if a set of inputs makes k -Set Agreement task solvable for a given adversary is computable. While doing so, we show that the computability power of the core-resilient adversary is the same as the situation where crashes happen only before any communication.

2 Models of Computation and Definitions

2.1 The message adversaries framework

Let $n \in \mathbb{N}$, we consider systems with $n + 1$ processes and denote $\Pi_n = [0, \dots, n]$ the set of processes. Sending a message is an asymmetric action, so we use directed graphs with the standard notations: let G , $V(G)$ is the set of vertices, $A(G) \subset V(G) \times V(G)$ is the arcs.

► **Definition 1** (Dynamic Graph and message adversary). *Consider the collection $\mathcal{G}_n$ of directed graphs with vertices as the set Π_n . A dynamic graph $\mathbf{G}$ is a sequence $G_1, G_2, \dots, G_r, \dots$ where G_r is a directed graph in $\mathcal{G}_n$. A message adversary $\mathcal{M}$ is a set of dynamic graphs.*

Since n will be mostly fixed through the paper, we write Π and $\mathcal{G}$ when there is no ambiguity. We use classical vocabulary on infinite words. Let $U \subseteq \mathcal{G}$, U^* is the set of finite sequences in U and U^ω be the set of infinite ones. For a word u , $u_{|r}$ is the prefix of size r and $u(r)$ is the r -th letter of the word. The distributed intuition behind such a graph is that G_r describes whether there will be transmission of some messages between each pair of processes at the round r . To highlight the distributed nature of such a graph, we can use *communication scenario* to describe a word on $\mathcal{G}$ and *instant graph* for a letter in a word.

2.2 Execution of a Distributed Algorithm

Given a message adversary $\mathcal{M}$ and a set of initial configurations $\mathcal{I}$, we define execution as an initialization step and a communication scenario. This corresponds to a (possibly infinite) sequence of rounds of message exchanges and corresponding local state updates. When the initialization is clear from the context, we will use *scenario* and *execution* interchangeably.

With more details, an execution of an algorithm $\mathcal{A}$ under scenario $w \in \mathcal{M}$ and initialization $\iota \in \mathcal{I}$ is denoted as $\iota.w$ and is composed by following steps. First, ι affects the initial state to all processes of Π . Then the system progresses in rounds. A round is decomposed into 3 sub-steps: sending, receiving, and updating the local state. At round $r \in \mathbb{N}$, the processes use the **SendAll**() primitive to send messages. The fact that the corresponding receive actions, using the **Receive**() primitive, will be successful depends on the instant graph G .

Let $p, q \in \Pi$. The message sent by p is received by q on the condition that the arc $(p, q) \in A(G)$. Then, all processes update their state according to the received values and $\mathcal{A}$. Note that it is assumed that p always receives its own value, which is $(p, p) \in A(G)$ for all p and G . However, this might be implicit for clarity and brevity. We denote the local state of a process p in an execution $w = \iota.u$ at the round r of the algorithm as $s_p(\iota.u[r])$. $s_p(\iota.\varepsilon) = \iota(p)$ represents the initial state of p in ι , where ε is the empty word. Note that in our framework, processes have identities (“colors”) and can therefore distinguish identical values sent by different processes. This model is denoted as the colored model of computation.

2.3 Relevant communication model

The message adversary framework allows us to describe a wide array of communication models. Like many other distributed computing papers that adopt a topological approach, the central model in this one is the Iterated Immediate Snapshot (*IIS*) model. It was first introduced as a (shared) memory model, which has been proved equivalent to the message adversary below first as tournaments and iterated tournaments [5, 2], then as this message adversary [11, 13]. See also [22] for a survey of the reductions involved in these layered models. Given a graph G , we denote by $In_G(a) = \{b \in V(G) \mid (b, a) \in A(G)\}$ the set of incoming vertices of a in $V(G)$. A graph G has the *containment Property* if for all $a, b \in V(G)$, $In_G(a) \subset In_G(b)$ or $In_G(b) \subset In_G(a)$. We say that a graph G has the *Immediacy Property* if for all $a, b, c \in V(G)$, $(a, b), (b, c) \in A(G)$ implies that $(a, c) \in A(G)$.

► **Definition 2** (*IIS model* [11]). *We set $ImS_n = \{G \in \mathcal{G}_n \mid G \text{ has the Immediacy and Containment properties}\}$. The Iterated Immediate Snapshot message adversary for $n + 1$ processes is the message adversary $IIS_n = ImS_n^\omega$.*

The setting of this paper is the general sub-message adversaries of the *IIS* model, which is $\mathcal{M} \subseteq IIS_n$. Many studied adversaries in the literature can be represented as such, like *oblivious adversary*, *t-resilient adversary* or *core-resilient adversary* or any ‘situation-specific’ adversary. We also use the terminology *crash* (which is, strictly speaking, irrelevant for messages adversaries) for a process that is not heard of by all other processes for an infinite amount of time.

As an example with two processes $\circ$ and $\bullet$, we can define the message adversary $IIS_1 = \{\circ \leftrightarrow \bullet, \circ \leftarrow \bullet, \circ \rightarrow \bullet\}^\omega$. In the execution $\circ \leftrightarrow \bullet \circ \leftarrow \bullet^\omega$, process $\circ$ is considered crashed starting from the second round. In the execution $\circ \leftrightarrow \bullet (\circ \leftarrow \bullet \circ \rightarrow \bullet)^\omega$, no process is crashed. A message adversary like $\mathcal{M}_1 = \{\circ \leftrightarrow \bullet\}^\omega \cup \{\circ \leftrightarrow \bullet\}^* (\{\circ \leftarrow \bullet^\omega, \circ \rightarrow \bullet^\omega\})$, that represent a system with two synchronized processes, where at most one of the processes may crash is a strict sub-adversary of IIS_1 since $\mathcal{M}_1 \subsetneq IIS_1$.

An iterated t -resilient adversary is a set of executions where at most t processes may “crash” during the execution.

Let $Q(w)$ the set of processes that are seen by all processes an infinite number of times in w

► **Definition 3** (*Iterated t -resilient adversary*). *Let $R_t^n = \{w \in IIS_n \mid \exists Q(w) \in \Pi \text{ such that } \#Q(w) \geq n + 1 - t\}$ be the t -resilient model on IIS_n .*

Core-resilient adversaries are a generalization of t -resilient adversaries.

► **Definition 4** (Core-resilient adversary). *Let P an inclusion-closed collection of sets of processes, a core-resilient adversary on P is the following set of executions $\mathcal{H}_P = \{w \in IIS_n \mid \Pi_n \setminus Q(w) \in P\}$.*

3 Abstract simplicial complexes and colorless tasks

We start by restating some standard definitions of combinatorial topology.

► **Definition 5** (Abstract simplicial complex [11, Def 3.2.1]). *Let V be a set, and C a collection of finite subsets of V . C is an abstract simplicial complex on V if $\forall \sigma \in C, \forall \tau \subseteq \sigma$, we have $\tau \in C$; and $\forall v \in V, \{v\} \in C$.*

An element of V is a *vertex* of C and $V(C)$ denotes the set of vertices of C . A set $\sigma \in C$ is a *simplex* where $\dim \sigma$ is the number of vertices in σ minus one. We say that σ is a *facet* if there is no other simplex that contains σ . If $C_1 \subseteq C_2$ then we say that C_1 is a *subcomplex* of C_2 , a complex is *pure* if all facets have the same dimension.

► **Definition 6** (Simplicial map [11, Def 3.2.2]). *Let C_1, C_2 be two simplicial complexes, a simplicial map is a map $\Phi : V(C_1) \rightarrow V(C_2)$ such that $\forall \sigma \in C_1, \Phi(\sigma) \in C_2$.*

► **Definition 7** (Carrier map [11, Def 3.4.1]). *Let C_1, C_2 be two simplicial complexes, a carrier map $\Phi : C_1 \rightarrow 2^{C_2}$ associates each simplex to a subcomplex of C_2 . The map is monotone, $\forall \sigma, \tau \in C_1, \sigma \subseteq \tau$ implies $\Phi(\sigma) \subseteq \Phi(\tau)$.*

The pair (C_1, χ_{C_1}) is a *chromatic complex* if C_1 is a complex and the function $\chi_{C_1} : V(C_1) \rightarrow \Pi$ has the property that $\forall \sigma \in C_1, \forall v_1, v_2 \in V(\sigma), v_1 \neq v_2 \Leftrightarrow \chi_{C_1}(v_1) \neq \chi_{C_1}(v_2)$.

The *border* of a simplex σ , is $\partial(\sigma) = \{\tau \in C_1 \mid \dim(\tau) = \dim(\sigma) - 1\}$. A ℓ -*skeleton* of C_1 is the collection of the simplices of dimensions equal or less than ℓ , we write $skel^\ell(C_1)$. The *star* of a simplex $\sigma \in C_1$ is $St(\sigma, C_1) = \bigcup_{\tau \in C_1, \sigma \subseteq \tau} \tau$. The *Link* of a simplex σ is $Lk(\sigma, C_1) = \{\tau \in St(\sigma, C_1) \mid \sigma \cap \tau = \emptyset\}$. A simplicial map $\varphi : C_1 \rightarrow C_2$ is *carried* by Φ if $\forall \sigma \in C_1, \varphi(\sigma) \in \Phi(\sigma)$.

This simplicial framework is used to describe *distributed task*, which are distributed problems. Let V_{in} be the domain of input values and V_{out} be the domain of output values. Here we focus on colorless tasks.

► **Definition 8** (Colorless Task [11, Def 4.2.1]). *A colorless task is a triple $(\mathcal{I}, \mathcal{O}, \Delta)$ where:*

- $\mathcal{I}$ is a simplicial complex, with vertices V_{in} ,
- $\mathcal{O}$ is a simplicial complex, with vertices V_{out} ,
- $\Delta : \mathcal{I} \rightarrow 2^{\mathcal{O}}$ is a carrier map.

The complex $\mathcal{I}$ is called the *input complex*, the complex $\mathcal{O}$ the *out complex* and Δ encodes the specification of the task. Colorless tasks correspond to a large family of problems where the number of occurrences of a value is not to take into account. The standard k -set agreement problem is a colorless task (only the number of different values is constrained). The renaming task is not a colorless task (each name should be unique in the output).

4 Colorless tasks into a colored world

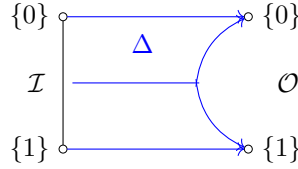
This section explicit the multiple settings that are used in distributed computing with combinatorial topology in particular in the context of colorless tasks.

4.1 Encoding colorless tasks with processes

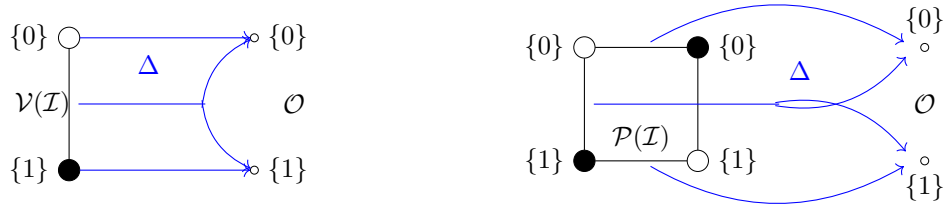
Even though we consider only colorless tasks, our model of computation is colored (see Section 2). In the colored computation model, the input configurations of a given colorless task are usually all the possible assignments of the initial values of the processes, so an initial value could be assigned many times. This means that the colored initial complex is not exactly $\mathcal{I}$. Since we will consider the very general input-dependent model, and a colored model of computation, we need a clear representation of the state of each process.

Consider a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$, and set $V_{in} = V(\mathcal{I})$ and $V_{out} = V(\mathcal{O})$. We assume wlog that there exists a total order $\prec$ on V_{in} and that we can describe V_{in} as $\{v_0, v_1, \dots, v_n\}$, where $v_i \prec v_j$ when $i < j$. The first setting is called *unique-value*. From a set of processes $\Pi = [0, n]$, there is only one process that can have a given initial value, and each process has exactly one value initially. This yields a colored simplicial complex $\mathcal{V}(\mathcal{I})$ with vertices $\{(i, v_i) \mid i \in [n]\}$ and simplices $((i_0, v_{i_0}), \dots, (i_d, v_{i_d}))$ whenever $\{v_{i_0}, \dots, v_{i_d}\}$ is a simplex of $\mathcal{I}$. This colored simplicial complex is actually $\mathcal{I}$, adding colours in the enumeration order $\prec$. The second setting is called *multi-value*, which allows different processes to have the same initial value, including in the same initial configuration. This forms a pseudosphere of $\mathcal{I}$, which is the following colored simplicial complex $\mathcal{P}_n(\mathcal{I})$. The set of vertices is $\Pi \times V(\mathcal{I})$. A set $\{(i_0, v_{i_0}), \dots, (i_d, v_{i_d})\}$, with $i_j < i_{j'}$ for $j < j'$, is a simplex of $\mathcal{P}_n(\mathcal{I})$ whenever $\{v_{i_0}, \dots, v_{i_d}\}$ is a simplex of $\mathcal{I}$. Note that here $\{(i_0, v_{i_0}), \dots, (i_d, v_{i_d})\}$ is a simplex of dimension d whereas $\{v_{i_0}, \dots, v_{i_d}\}$ could be a simplex of dimensions less than d . Fig. 1 illustrates the Binary Consensus task as encoded by a relation between input and output complexes.

Fig. 2 gives two representations of the encoding of Binary Consensus in the colored model of computation (with $\circ$ as color 0 and $\bullet$ as color 1), one in the unique value setting, and the other one in the multi-value setting.



■ **Figure 1** The Binary Consensus task.

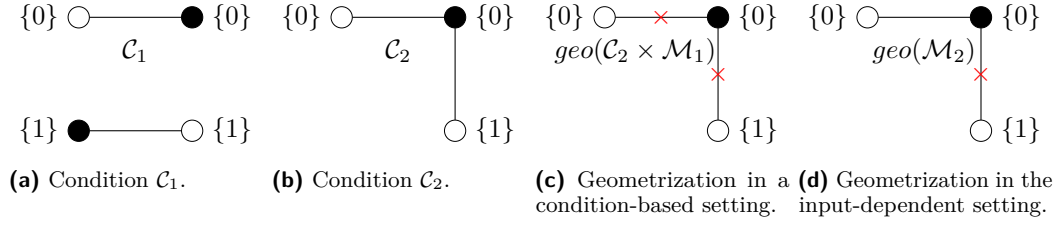


(a) Binary Consensus in the unique value setting.

(b) Binary Consensus in the multi-value setting.

■ **Figure 2** Unique value setting vs Multiple value setting.

Moving from one setting to another can be done using the function $GIV : \mathcal{I}_P \rightarrow \mathcal{I}_V$ (as Get Input Value) to associate a colored simplex to a colorless simplex that contains the same input values: $GIV(\{(p_0, v_0), (p_1, v_1), \dots, (p_n, v_n)\}) = \{v_0, v_1, \dots, v_n\}$. Moreover, the carrier map for the multi-value setting Δ_p needs some adjustments, $\forall \sigma \in \mathcal{P}_n(\mathcal{I}), \Delta_p(\sigma) = \Delta(GIV(\sigma))$. Which makes sense, in a colorless task the value of a simplex decides its possible output.



■ **Figure 3** Examples of the different possible settings.

4.2 Problem Statements

This paper considers 4 classes of adversaries that can be applied to a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$. The *standard setting* has a message adversary $\mathcal{M}_1 = IIS_n$ while the *sub-model setting* has a message adversary $\mathcal{M}_1 \subseteq IIS$. Both can apply to a unique value context ($\mathcal{M} = \mathcal{V}(\mathcal{I}) \times \mathcal{M}_1$) or a multi-value context ($\mathcal{M} = \mathcal{P}_n(\mathcal{I}) \times \mathcal{M}_1$). These two contexts are considered interchangeable in the literature, since in the standard setting, they are simply equivalent, as we will prove on Prop. 10. In the input-dependent submodel setting, one has to be more careful. The results of [7] were presented in the unique-value setting. The *condition-based setting* has a message adversary $\mathcal{M}_1 \subseteq IIS_n$ and $C \subseteq \mathcal{P}_n(\mathcal{I})$ to form executions on $\mathcal{M} = C \times \mathcal{M}_1$. This corresponds to adding a condition of distribution of initial values that are not valid. This is the setting of [19] with $\mathcal{M}_1 = \mathcal{R}_n^t$ (the t -resilient model). Finally, the *input-dependent setting* considers execution in $\mathcal{M} \subseteq \mathcal{I} \times IIS_n$, which adds the possibility of changing the message adversary depending on the input configuration. This setting encompasses all previously described settings.

In Figure 3, we present examples of our various setting for the Binary Consensus. In Fig. 3a, for condition C_1 , we remove the possibility of $\bullet$ and $\circ$ to have different initial values. On Fig. 3b, for condition C_2 , we remove all input configurations where $\bullet$ starts with the value 1. We denote $\mathcal{M}_1 = IIS_1 \setminus \{\circ \leftrightarrow \bullet^\omega\}$. Fig. 3c presents the geometrization of sub-model condition-based adversaries $C_2 \times \mathcal{M}_1$. Intuitively, the red cross represents the missing execution $\circ \leftrightarrow \bullet^\omega$, see later in Section 5.2. We define $\mathcal{M}_2$ to be the input dependent model where the possible inputs are from C_2 , and the possible executions are IIS_1 when the two initial values are 0, and where the possible executions are $\mathcal{M}_1$ when the two initial values are different. The geometrization of $\mathcal{M}_2$ is given in Fig. 3d.

► **Definition 9** (Solvability of a Colorless Task). *Given a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$, it is solvable in the input-dependent setting against executions $\mathcal{M} \subseteq \mathcal{I} \times IIS_n$ if there is a colored algorithm $\mathcal{A}$ such that for any execution $\iota.w \in \mathcal{M}$, there exists u a prefix of w such that the state of the system $\{s_0(\iota.u), \dots, s_n(\iota.u)\} = out$ satisfies the specification of the task, i.e. $out \in \Delta(GIV(\iota))$.*

The unique-value setting and the multi-value setting are equivalent on IIS from a computability point of view.

► **Proposition 10.** *There is an algorithm to solve $(\mathcal{I}, \mathcal{O}, \Delta)$ on IIS_n with condition $\mathcal{V}(\mathcal{I})$ if and only if there is an algorithm to solve $(\mathcal{I}, \mathcal{O}, \Delta)$ on IIS_n with condition $\mathcal{P}_n(\mathcal{I})$.*

Proof. Since $\mathcal{V}(\mathcal{I}) \subset \mathcal{P}_n(\mathcal{I})$, one direction is obvious. Consider that we have an algorithm $Algo$ with input $\mathcal{V}(\mathcal{I})$, then we extend it to input $\mathcal{P}_n(\mathcal{I})$ by letting process p starting with v_i follow the instructions of process i in the given algorithm $Algo$. ◀

5 Geometric Definition of Simplicial Complexes

5.1 Standard Definitions

We present here classical definitions of geometric complexes and provide a link between distributed computability and such geometric setting, as our general computability setting could need infinite complexes. We fix $N \in \mathbb{N}$ and denote $\mathbf{B}(x, r) = \{y \in X \mid d(x, y) \leq r\}$ with $x \in \mathbb{R}^N, r \in \mathbb{R}$ and $d(x, y)$ the Euclidean distance on $\mathbb{R}^N$.

► **Definition 11** (Geometric Simplex). *Let $n \in \mathbb{N}$. A finite set $\sigma = \{x_0, \dots, x_n\} \subset \mathbb{R}^N$ is called a simplex of dimension n if the vectors $\{x_1 - x_0, \dots, x_n - x_0\}$ are linearly independent.*

Let $|\sigma|$ be the convex hull of σ and $\text{Int}(\sigma)$ is the interior of $|\sigma|$. The *open star* of $\sigma \in C_1$: $\text{St}^\circ(\sigma, C_1) = \bigcup_{\tau \in C_1, \sigma \subseteq \tau} \text{Int}(\tau)$. Let $\mathbb{S}^n$ be “the” simplex of dimension n : through this paper we assume a fixed embedding in $\mathbb{R}^N$ for $\mathbb{S}^n = (x_0^*, \dots, x_n^*)$ and a diameter $\text{diam}(\mathbb{S}^n)$ at 1.

► **Definition 12** (Geometric Simplicial Complex [20]). *A simplicial complex is a collection C of simplices such that:*

- (a) *If $\sigma \in C$ and $\sigma' \subset \sigma$, then $\sigma' \in C$,*
- (b) *If $\sigma, \tau \in C$ and $|\sigma| \cap |\tau| \neq \emptyset$ then there exists $\sigma' \in C$ such that*
 - $|\sigma| \cap |\tau| = |\sigma'|$,
 - $\sigma' \subset \sigma, \sigma' \subset \tau$.

For any simplicial complex C , we can associate a set of geometric points $\text{geo}(C) = \bigcup_{\sigma \in C} |\sigma|$. We will use $\text{geo}(C)$ or $|C|$ interchangeably for finite complexes¹. Let A, B be finite simplicial complexes and δ a simplicial map from A to B . It can be extended to $f : |A| \rightarrow |B|$ using *barycentric extension*. Let $\sigma = \{x_0, \dots, x_n\}$ a simplex of A . Since any $y \in |\sigma|$ could be obtained as $y = \sum_{i=0}^d t_i x_i$ with $t_i \in [0, 1]$ and $\sum_{i=0}^d t_i = 1$, we set $f(y) = \sum_{i=0}^d t_i \delta(x_i)$. Let $X \subset \mathbb{R}^N$, a function $f : X \rightarrow |C_2|$ respects a carrier map $\Delta : C_1 \rightarrow 2^{C_2}$ with $X \subseteq |C_1|$, if $\forall \sigma \in C_1, f(|\sigma| \cap X) \subseteq \Delta(\sigma)$.

► **Definition 13** (Subdivision [11, Def. 3.6.1]). *Let C_1, C_2 be two geometric simplicial complexes. We say that C_2 is a subdivision of C_1 if: $\text{geo}(C_1) = \text{geo}(C_2)$, and the geometrization of each simplex of C_1 is the union of the geometrization of finitely many simplices of C_2 .*

In distributed computing there are two subdivisions that are used, the barycentric Subdivision and the Standard Chromatic Subdivision. In this paper, the barycentric is useful for another reason than being the subdivision used to represent colorless computability (see section 7.1). A complete presentation of the second one is in Section A.

► **Definition 14** (Barycentric Subdivision). *Let C an abstract simplicial complex, its barycentric subdivision $\text{Bary}(C)$ is the abstract simplicial complex, whose vertices are the nonempty simplices of C . A $(\ell + 1)$ tuple $\{\sigma_0, \dots, \sigma_\ell\}$ is a simplex of $\text{Bary}(C)$ if and only if the tuple can be indexed by containment.*

¹ As a set of points $\text{geo}(C)$ corresponds to the points of the topological space $|C|$ that is classically called the *geometric realization*. When C is finite, $\text{geo}(C)$ has the same topology (in $\mathbb{R}^N$) as the geometric realization $|C|$. It could not be the case if C is infinite, see [7].

5.2 Geometric Encoding of Iterated Immediate Snapshots

Here, we present the connection between executions of the Iterated Immediate Snapshot and simplicial complexes. To achieve this, we employ an algorithm called the Chromatic Average algorithm. It accepts an execution w in the *IIS* model and produces a geometrical simplex. This has been introduced in [6], for a detailed presentation refer to [6, 7].

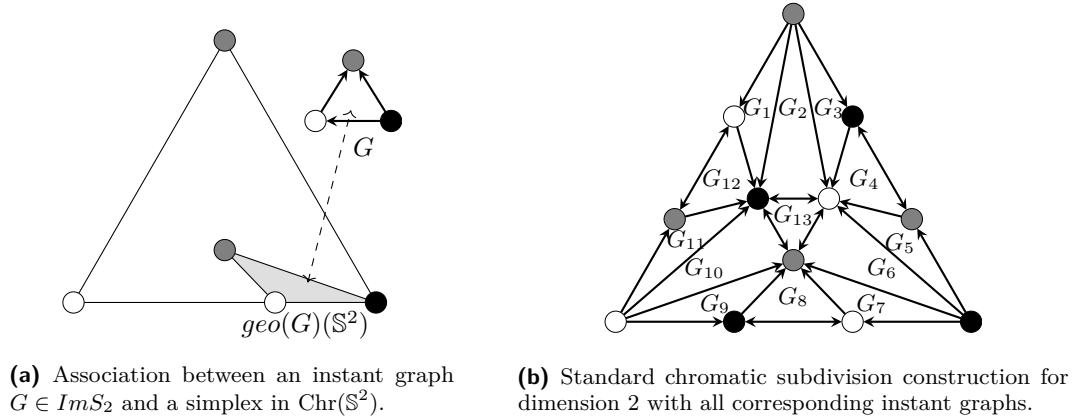
■ **Algorithm 1** The Chromatic Average Algorithm for process i .

```

1  $x \leftarrow x_i^* \in \mathbb{R}^N$ ;
2 Loop forever
3   SendAll(( $i, x$ ));
4    $V \leftarrow \text{Receive}()$  // set of all received messages including its own;
5    $d \leftarrow \text{sizeof}(V) - 1$  //  $i$  received  $d + 1$  messages including its own ;
6    $x = \frac{1}{2d+1}x + \sum_{(j,x_j) \in V, j \neq i} \frac{2}{2d+1}x_j$ ;
7 EndLoop

```

A given loop of this algorithm corresponds to one execution of the Immediate Snapshot protocol. Geometrically, this algorithm associates with every possible view of processes and initial configuration $\sigma \in \mathcal{I}$ in the *ImS* protocol to a given vertex in the Standard Chromatic Subdivision of σ . We present in more detail this subdivision in the Appendix Section A. The equivalence between this subdivision and the *ImS* model can be seen in [17].



■ **Figure 4** Construction of the Standard Chromatic Subdivision.

On Fig. 4a, we represent the mapping of one instant graph G from an execution in *IIS* to a simplex in $\text{Chr}(\mathbb{S}^2)$. On Fig. 4b, these associations are presented for every possible instant graph of ImS_2 . The mapping of a prefix of size t of an execution $w \in \text{IIS}_n$ to $\sigma \in \text{Chr}^t \mathbb{S}^n$ is called the *geometrization* of $w|_t$, denoted as $\text{geo}(w|_t)(\mathbb{S}^n)$.

5.3 Geometrization of Infinite Executions and a Topology for IIS_n

Since [6] the *geometrization* approach has been shown to be a fruitful way to handle (the limit of) iterated executions. From $\text{geo}(w|_t)(\mathbb{S}^n)$ that work on prefixes of execution, we can take the limit on the size of prefixes, $\text{geo}(w) = \lim_{t \rightarrow \infty} \text{geo}(w|_t)(\mathbb{S}^n)$. This operation is well defined, as it makes every execution converge to a geometric point (see [6] for more detail).

The *geometrization topology* is defined on IIS_n by considering as open sets the sets $geo^{-1}(\Omega)$ where Ω is an open set of $\mathbb{R}^N$. A collection of sets can define a topology when any union of sets of the collection is in the collection, and when any finite intersection of sets of the collection is in the collection. This is straightforward for a collection of inverse images of a collection that satisfies these properties. Note this also makes *geo* continuous by definition.

The previous construction took $\mathbb{S}^n$ as input, we extend the previous definition to any input simplices. For $\mathcal{M} \subseteq IIS_n$ and $\sigma \in \mathcal{I}$ with $\dim(\sigma) = n$ we can define $\forall w \in \mathcal{M}, geo(w)(\sigma)$ by using a mapping from $\mathbb{S}^n$ to σ which maps a vertex i of $\mathbb{S}^n$ to v_i with color i in $V(\sigma)$ (this is called the characteristic map of σ). Hence, $geo(\mathcal{I} \times \mathcal{M})$ is defined as $\bigcup_{w \in \mathcal{M}, \sigma \in \mathcal{I}} \varphi_\sigma(geo(w))$. This construction associates to any set of executions $\mathcal{I} \times \mathcal{M}$ a topological subspace of $\mathbb{R}^N$.

Executions in non-compact sub-models of IIS may not terminate at the same round. The well-known correspondence between terminating algorithms and complexes could therefore yield infinite complexes. In this iterated subdivision framework, such (possibly infinite) complexes are called *terminating subdivisions* and were first introduced in [8]. For this article, we use the combinatorial definition of *IIS-terminating subdivision* from [7]. Given a complex C , let $C(T) = \bigcup_{\sigma \in C, V(\sigma) \subseteq T} \sigma$ with $T \subseteq V(C)$ to represent the sub-complex of C formed by the vertices in T . Let $JOIN(C_1, C_2) = \{\sigma \cup \tau \mid \sigma \in C_1, \tau \in C_2\}$. We define *EChr* as the following operator:

$$EChr(T, C) = \left(\bigcup_{\sigma \in C} Chr \sigma(U) \right) \cup \left(\bigcup_{\sigma \in C} JOIN(Chr \sigma(U), \sigma(T)) \right) \quad (1)$$

Intuitively, the vertices marked as terminated are in T . We note $U = V(C) \setminus T$. The operator *EChr* subdivides with the standard chromatic subdivision the facets that are fully in U , does not modify the ones that are fully in T and subdivides in an adequate way the facets containing both.

► **Definition 15** (*IIS-Terminating subdivision* [7]). *Let $\mathcal{I}$ a simplicial complex. The sequences $C_0, C_1, \dots$ (collection of simplices) and $T_0, T_1, \dots$ (collection of increasing set of vertices) form a IIS-terminating subdivision of $\mathcal{I}$, if we have for all $i \in \mathbb{N}$:*

1. $C_0 = \mathcal{I}, T_0 = \emptyset$
2. $C_{i+1} \subseteq EChr(T_i, C_i)$
3. $T_i \subseteq V(C_i)$

We say that $\bigcup C_i(T_i)$ is an *IIS-terminating subdivision complex*. This is indeed an actual geometric simplicial complex, see [7].

6 A General Input-Dependent Colorless Computability Theorem

In the recent characterization of [7] of the computability of colorless tasks, it was presented it in the unique-value setting. In the following, we show it is possible to strengthen these results to the more general input-dependent multi-value setting.

► **Theorem 16.** *A colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on $\mathcal{A} \subseteq \mathcal{P}_n(\mathcal{I}) \times IIS_n$ if and only if there is a continuous function $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ that respects Δ .*

We now prove Theorem 16. The proofs from [7] consider an arbitrary topological space X as input. Since $\mathcal{A} \subseteq \mathcal{P}_n(\mathcal{I}) \times IIS_n$ can define a subspace of $\mathbb{R}^N$ by geometrization, we re-use most of the proof using $X = geo(\mathcal{A})$, making some key adjustments when necessary. In this proof, they extend two concepts from the proof in the *IIS_n* model in Chapter 4 of [11]. The first one is an alternative handling of the “continuity” when the output space is a simplicial complex.

► **Definition 17** (Star Condition for η [7]). *Let $\eta : X \rightarrow]0, +\infty[$ and let $f : X \rightarrow |\mathcal{O}|$, f satisfies the star condition for η if $\forall x \in X, \exists v \in V(\mathcal{O}), f(\mathbf{B}(x, \eta(x)) \cap X) \subseteq St^\circ(v)$.*

The second one is the concept of “semi-simplicial approximation”. It is similar to the classical simplicial approximation [10], except here, the input space is not a simplicial complex, but only a topological space. They use the η -star condition property to construct an IIS-terminating subdivision $\mathcal{K}_\eta$ that approximate X .

► **Definition 18** (semi-simplicial approximation [7]). *Let $f : X \rightarrow |\mathcal{O}|$ a function. The function $\psi : V(\mathcal{K}) \rightarrow V(\mathcal{O})$ is a semi-simplicial approximation for f if $\mathcal{K}$ is a IIS-terminating subdivision that cover X , and ψ is a simplicial map such that $\forall \sigma \in \mathcal{K}, f(St^\circ(\sigma) \cap X) \subseteq St^\circ(\psi(\sigma))$.*

We prove that a continuous function $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ implies a distributed algorithm solving the task, then, conversely, from a distributed algorithm, we can extract a continuous function with Prop. 23.

► **Proposition 19** ([7]). *Let $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ a continuous function. Then there is $\eta : geo(\mathcal{A}) \rightarrow]0, +\infty[$ such that f satisfies the η -star condition.*

► **Proposition 20** ([7]). *Let $\eta : geo(\mathcal{A}) \rightarrow]0, +\infty[$ and let $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ a function that satisfies the η -star condition, then f has a semi-simplicial approximation $\psi_\eta : V(\mathcal{K}_\eta) \rightarrow V(\mathcal{O})$.*

► **Proposition 21** (semi-simplicial approximation and carrier map, [7]). *Let $\eta : geo(\mathcal{A}) \rightarrow]0, +\infty[$ and let $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ a continuous function that respects $\Delta : \mathcal{P}_n(\mathcal{I}) \rightarrow 2^\mathcal{O}$ a carrier map. Then the semi-simplicial approximation $\psi_\eta : \mathcal{K}_\eta \rightarrow \mathcal{O}$ of f also respects Δ .*

► **Proposition 22.** *Let $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ a continuous function which respects a carrier Δ then $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable by an algorithm in model $\mathcal{A}$.*

These four propositions are already proved in [7], hence require no proof here. On the other hand, the following proposition needs to be proved again, since the setting is more general than the unique-value setting. We recall that a normalized algorithm is an algorithm where, when someone observes a value decided by another process, it immediately decides this value. For a colorless task, it does not change the correction of a given algorithm.

► **Proposition 23.** *Let Algo a normalized algorithm for solving $(\mathcal{I}, \mathcal{O}, \Delta)$ against executions $\mathcal{A} \subset \mathcal{P}_n(\mathcal{I}) \times IIS_n$, then there exists a continuous function $f : geo(\mathcal{A}) \rightarrow |\mathcal{O}|$ that respects Δ .*

Proof. An algorithm solving a task $(\mathcal{I}, \mathcal{O}, \Delta)$ on an adversary $\mathcal{A}$ generates a terminating subdivision $\mathcal{K}_\mathcal{A}$ and decision function $\varphi : V(\mathcal{K}_\mathcal{A}) \rightarrow V(\mathcal{O})$ which is simplicial. As emphasized in [9, 3, 7] we cannot directly take the geometric realization of φ to obtain a continuous function if $\mathcal{A}$ is a non-compact message adversary. Consider $v \in V(\mathcal{K}_\mathcal{A})$ that decides at round t , and $u \in V(St(v, \mathcal{K}_\mathcal{A}))$. Using the normalization property: u has decided at round $t + 1$ or before. The subcomplex $St(v, \mathcal{K}_\mathcal{A})$ is a finite complex, which means that $\varphi(St(v, \mathcal{K}_\mathcal{A}))$ we can use the linear extension to get a continuous function on $St(v, \mathcal{K}_\mathcal{A})$.

Since the complex is infinite, we cannot directly deduce the continuity of the whole function. Consider $x \in geo(\mathcal{A})$, and denote σ a simplex of $\mathcal{K}_\mathcal{A}$ that contains x . Since x has a neighborhood inside $St(v, \mathcal{K}_\mathcal{A})$ for some v of $V(\sigma)$, from the previous remark, we get that f is continuous at x . ◀

7 Core-Resilient Adversaries and equivalence of tasks

7.1 Geometrical Representation of Core-Resilient Adversary

When working with a specific model of computation, a natural question may be, “What the geometrization of such model looks like?” The geometrization of t -resilient and core-resilient adversaries is not easy to grasp. Hence, one goal of this paper is to provide computability equivalence with a model that has much nicer geometrization.

Geometrically, we can construct the t -resilient adversary by removing all simplices of dimension $n - t$ for every round of computation. In other words, $geo(R_n^t) = \bigcap_{i \in \mathbb{N}} |\mathcal{I}| \setminus geo(skel^{n-t-1}Chr^i \mathcal{I})$. This yields a fractal-like geometrization of R_n^t . A detailed presentation of the geometrization of R_n^t is in the Appendix, section A.2 for example for 3 processes and one crash. For core-resilient adversaries, any set of the core corresponds to a simplex σ in an input complex $\mathcal{I}$. Then $\forall \tau \subsetneq \sigma$, their images through the iterated subdivisions are removed from $|\mathcal{I}|$, since it corresponds to a subset of processes that can crash altogether starting from that particular initial configuration. As for R_n^t the processes can crash at any moment of the computation; hence, simplices in all steps of the geometrization are to be removed, which also yield a fractal-like space.

These two models have a “quite complicated” geometrization space thanks to the “fractal”-part. In order to have simpler spaces, we will consider similar models where crashes only happened before a given round. The r -restricted t -resilient model is defined as $S_r^t = \{w \in IIS \mid K_r(w) \leq t\}$ with $K_r(w)$ the set of processes that are never seen by some process starting from round r excluded, e.g. $K_0(w)$ is the set of processes that are crashed at the start of the computation. We have simply $geo(S_0^t) = |\mathcal{I}| \setminus |skel^{n-t-1} \mathcal{I}|$.

Similarly, let $\mathcal{H}$ a core-resilient adversary, we denote by $\mathcal{H}_r$ the r -restricted associated core-adversary, where crashes happen at round r . Then, $geo(\mathcal{I} \times \mathcal{H}_0) = \mathcal{I} \setminus \{\sigma \in \mathcal{I} \mid |\sigma| \cap geo(\mathcal{I} \times \mathcal{H}) = \emptyset\}$.

Let $\mathcal{H}$ an input-dependent core adversary (*core-dependent* in short) on simplicial complex $\mathcal{I}$, we denote by $\mathcal{C}(\mathcal{H})$ the *condition for $\mathcal{H}$* , that is $\mathcal{C}(\mathcal{H}) = \{\sigma \in \mathcal{I} \mid |\sigma| \cap geo(\mathcal{H}_0) \neq \emptyset\}$. By extension of notation, we note $\mathcal{U}(\mathcal{H})$ as the part of the barycentric subdivision of $\mathcal{I}$ that intersects in $geo(\mathcal{H}_0)$. More formally: $V(\mathcal{U}(\mathcal{H})) = \{iso(\tau) \mid \tau \in \mathcal{C}(\mathcal{H})\}$, where $iso(\tau)$ is the isobarycenter of simplex $|\tau|$. The simplices of $\mathcal{U}(\mathcal{H})$ corresponds to sets of isobarycenters of simplices that can be ordered by inclusion.

7.2 General Computability Result on core-resilient adversaries

This first theorem shows that models $\mathcal{H}_0$ associated with core-dependent model $\mathcal{H}$, ie when crashes happen before the first round, have the same computability power as $\mathcal{U}(\mathcal{H})$, which corresponds to a subset of the barycentric subdivision of $\mathcal{I}$.

► **Theorem 24.** *Consider a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$, and $\mathcal{H}$ a core-dependent adversary for $\mathcal{P}(\mathcal{I})$. The following statements are equivalent*

1. $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on $\mathcal{H}_0$,
2. $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on $geo^{-1}(\mathcal{U}(\mathcal{H}))$.

Proof. Let $Y = geo(\mathcal{H}_0)$. We have that $|\mathcal{U}(\mathcal{H})| \subseteq Y$ which make the $(\Rightarrow)$ direction done.

For the converse, it is sufficient to prove that there is a continuous function $r : Y \rightarrow |\mathcal{U}(\mathcal{H})|$. For that, we will construct a collection of functions r_σ , that will project Y on $|\mathcal{U}(\mathcal{H})|$. We denote $\mathcal{C} \subseteq \mathcal{I}$ the condition of $\mathcal{H}$. Let $Q = \{\sigma \in \mathcal{C} \mid |\sigma| \cap Y = \emptyset\}$. Then for $\sigma \in Q$, let $r_\sigma : |St(iso(\sigma), Bary(\mathcal{C}))| \setminus \{iso(\sigma)\} \rightarrow |Lk(iso(\sigma), Bary(\mathcal{C}))|$ the projection retract onto

the Link from $iso(\sigma)$. Let $\tau = (\sigma_0, \sigma_1, \dots, \sigma_d) \in Q \cap Bary(\mathcal{C})$. Since by definition of the barycentric subdivision, we have $\sigma_0 \subsetneq \sigma_1 \subsetneq \dots \subsetneq \sigma_d \in Q$, we define $r_\tau = r_{\sigma_d} \circ \dots \circ r_{\sigma_0}$. This is correctly defined since for $\sigma \in Q$, $|\sigma| \cap Y = \emptyset$, so $iso(\sigma_{i+1})$ is never in the image of r_{σ_i} .

We can now make a disjoint sum of the r_τ to construct r , with all τ maximal chains in $Bary(\mathcal{C})$, by remarking that the interiors of $|St(v, Bary(\mathcal{C}))|$, with $v \in V(\mathcal{C})$ form a partition of $Y \setminus |\mathcal{U}(\mathcal{H})|$. $\blacktriangleleft$

This theorem states that, in a core-adversary model, the crashes that happened after the first round of communication don't change the solvability. The proof is in Appendix B.

► **Theorem 25.** *Let $(\mathcal{I}, \mathcal{O}, \Delta)$ a colorless task, $\mathcal{H}$ a core-dependent adversary for $\mathcal{P}(\mathcal{I})$. The task is solvable on $\mathcal{H}$ if and only if it is solvable on $\mathcal{H}_0$.*

Now we have proved the equivalence of message adversaries $\mathcal{H}, \mathcal{H}_r$ and $\mathcal{U}(\mathcal{H})$, we give another interpretation on the result from Th. 24.

► **Theorem 26.** *Consider a colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$, and $\mathcal{H}$ a core-dependent adversary for $\mathcal{P}(\mathcal{I})$. The task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on $geo^{-1}(\mathcal{U}(\mathcal{H}))$ if and only if the task $(\mathcal{U}(\mathcal{H}), \mathcal{O}, \Delta \circ Bary_{\mathcal{I}}^{-1})$ is solvable against IIS.*

7.3 Equivalence of distributed tasks

Presenting a distributed task as triple $(\mathcal{I}, \mathcal{O}, \Delta)$ creates the possibility of having tasks that have equivalent computability, which is, they are solvable on the same message adversaries. In this section, we propose multiple propositions/lemmas showing how to transform a task to another one that is simpler to analyze but has the same computability.

The carrier map of a task describes where some simplex in the input complex can be mapped in the output complex. Then a task is solvable if we are able to construct a simplicial map from a terminating subdivision of the input complex to the output complex. This first lemma leverages the fact that a simplicial function cannot map a simplex to another one of greater dimensions.

Let $T = (\mathcal{I}, \mathcal{O}, \Delta)$ a task is *Non-Expanding* if $\forall \sigma \in \mathcal{I}, \dim(\Delta(\sigma)) \leq \dim(\sigma)$. Lemma 27 implies that any task is equivalent to a non-expanding one. Let $(\mathcal{I}, \mathcal{O}, \Delta)$ a task, let $\sigma \in \mathcal{I}$, we set $\overline{\Delta}(\sigma) = \{\tau \in \Delta(\sigma) | \dim(\tau) \leq \dim(\sigma)\}$.

► **Lemma 27.** $\forall \mathcal{A} \subseteq \mathcal{I} \times IIS$, the task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on $\mathcal{A}$ if and only if $(\mathcal{I}, \mathcal{O}, \overline{\Delta})$ is solvable on $\mathcal{A}$.

Proof. ($\Rightarrow$) If $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable, then there exists an algorithm *Algo* which yields a terminating subdivision $\mathcal{K}$ and a simplicial function $\delta : \mathcal{K} \rightarrow \mathcal{O}$. By simpliciality, $\forall \sigma \dim(\delta(\sigma)) \leq \dim(\sigma)$, hence $\delta(\sigma) \subseteq \overline{\Delta}(\sigma)$. So *Algo* also solves $(\mathcal{I}, \mathcal{O}, \overline{\Delta})$.

($\Leftarrow$) We have that $\forall \sigma \in \mathcal{I}, \overline{\Delta}(\sigma) \subseteq \Delta(\sigma)$, hence solving $(\mathcal{I}, \mathcal{O}, \overline{\Delta})$ implies solving $(\mathcal{I}, \mathcal{O}, \Delta)$. $\blacktriangleleft$

On a distributed task, the carrier map encodes all possible output values. The next proposition moves this non-determinism on the vertices to an equivalence of solvability. From $(\mathcal{I}, \mathcal{O}, \Delta)$ a non-expanding task, then $\forall v \in V(\mathcal{I}), \Delta(v)$ is a collection of vertices from $V(\mathcal{O})$. Given a pair (v, u) with $u \in \Delta(v)$, we set $\Delta_{(v,u)}$ as follows: $\forall \sigma \in \mathcal{I}, \sigma \neq v, \Delta_{(v,u)}(\sigma) = \Delta(\sigma)$ and $\Delta_{(v,u)}(v) = u$. Similarly, given a collection P of such pairs where a vertex of $\mathcal{I}$ appears at most once, we define Δ_P . We denote CP the set of such collections where all vertices in $\mathcal{I}$ appear. A task where all input vertices have one possible value by the carrier map is said to be *vertex-deterministic*. For any collection P in CP , $(\mathcal{I}, \mathcal{O}, \Delta_P)$ is *vertex-determinism*.

► **Proposition 28.** *Let $(\mathcal{I}, \mathcal{O}, \Delta)$ a non-expanding task, $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable on $\mathcal{A}$ if and only if $\exists P \in CP$ such that $(\mathcal{I}, \mathcal{O}, \Delta_P)$ is solvable on $\mathcal{A}$.*

Proof. Since the task is solvable we have an algorithm that yields a terminating subdivision $\mathcal{K}$ and a simplicial map $\delta : \mathcal{K} \rightarrow \mathcal{O}$ that satisfies Δ . The collection $P = \{(v, \delta(v))\}$ satisfies the claim. Conversely, we have that $\forall \sigma \in \mathcal{I}, \Delta_P(\sigma) \subseteq \Delta(\sigma)$ ◀

One way to prove that two submodels are equivalent is to construct a continuous function between the geometrization of one submodel to the one of the other. This means finding a continuous function that preserves the constraints of the carrier map.

► **Definition 29** (Δ -compatible function). *Let $(\mathcal{I}, \mathcal{O}, \Delta)$ a task, let $Y \subseteq |\mathcal{I}|$ then $f : Y \rightarrow Y$ is Δ -compatible if it is continuous and $\forall U \subseteq Y, \Delta(f(U)) \subseteq \Delta(U)$*

8 Condition Based k -set Agreement for core-resilient adversaries

8.1 Characterization of the solvability

In this section we look at the solvability of the k -set Agreement in a condition based context against core-dependent adversaries. This problem was introduced in [19] for t -resilient models, and the answer can be directly inferred for $k + 1 \leq t$ from classical result on t -resilient model [14]. For $k + 1 > t$, from [19], we have some conditions that can solve the k -set Agreement for t -resilient model. In this section, we show that deciding if some condition-based adversary can solve the k -set Agreement for a given t is computable.

Let $\mathcal{C}$ a condition for task $(\mathcal{I}, \mathcal{O}, \Delta)$. If this task is solvable on $\mathcal{C} \times \mathcal{M}$ then, $\forall \mathcal{C}' \subseteq \mathcal{C}$, this is also solvable. Hence, we say that $\mathcal{C}$ is *maximal* for the task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ if $\mathcal{C}$ solves the task T and $\forall \mathcal{C}', \mathcal{C} \subsetneq \mathcal{C}', T$ is not solvable on $\mathcal{C}'$. Given a condition $\mathcal{C}$ on $\mathcal{P}(\mathbb{S}^k)$, we call $\mathcal{C}$ -core-dependent adversary a core-dependent adversary with support $\mathcal{C}$.

► **Proposition 30.** *Let $\mathcal{H}$ be a $\mathcal{C}$ -core-dependent adversary. The k -set Agreement task is solvable on condition $\mathcal{U}(\mathcal{H})$ against the IIS model if and only if there is a simplicial map $\varphi : \mathcal{U}(\mathcal{H}) \rightarrow \partial(\mathbb{S}^k)$ that respects $\Delta \circ \text{Bary}^{-1}$.*

Proof. Using Proposition 28, we get a solvable vertex-deterministic task $(\mathcal{U}(\mathcal{H}), \mathcal{O}, \Delta_P)$ with $P \in CP$. Let φ_P the simplicial mapping such that $\varphi_P(v) = u$ for each pair $(v, u) \in P$. By construction of Δ_P , this respects $\Delta \circ \text{Bary}^{-1}$.

The reverse direction is straightforward. ◀

From Theorem 24 and Proposition 30, we can construct an algorithm that, from any $\mathcal{C}$ -core-dependent adversary $\mathcal{H}$, tests if the k -set Agreement is solvable by enumerating all Δ_P and testing whether φ_P is actually simplicial to $\partial(\mathbb{S}^k)$.

► **Theorem 31.** *The problem of deciding if the k -set Agreement task is solvable on $\mathcal{H}$ a $\mathcal{C}$ -core-dependent adversary is decidable.*

8.2 Going back to the 2002 paper

The [19] paper introduced the condition-based setting against a t -resilient model and asked which conditions solve the k -set agreement.

Given a condition $\mathcal{C} \subseteq \mathcal{P}(\mathcal{I})$, they consider every set of values of size at least $n - t + 1$ that is present in $\mathcal{C}$ as a vector of size $n + 1$ with at most t values that are unset. They consider an associated simplicial complex that is called $\text{Kin}(\mathcal{C}, f, k)$ and an example can be found in the Appendix, section A.1, Figure 6.

With this construction, it is proved in [19] that if there is a simplicial function from $\mathcal{Kin}(C, f, k)$ to $\mathcal{skel}^k \mathcal{I}$, then a condition $\mathcal{C}$ can solve the k -set Agreement for the t -resilient model.

Looking at $\mathcal{Kin}(C, f, k)$, we can see that it is the k -skeleton of $\mathcal{U}(C)$ since the barycentric subdivision corresponds to using inclusion-ordered initial simplexes as new simplices. Hence, Theorem 24 shows that the sufficient condition from [19] is actually necessary.

Theorem 24 provides several improvements: we have a necessary and sufficient characterization that works for a larger class of adversaries, the core-dependent adversaries.

Another contribution of [19] is that they provide two example conditions that generate a set of input configurations that make the k -set agreement solvable.

The first condition (C_1) uses an order on the input value. For any proper simplex $\sigma \in \mathcal{P}(\mathcal{I})$, we write $a_1 \dots a_\ell$ the input value arranged from the biggest to the lowest, $a_1 \geq a_2 \dots \geq a_\ell$. Moreover, $\#a_i$ denotes the number of occurrences of the value a_i . If $\sum_1^k \#a_i > t$, then $\sigma \in C_1$.

The second condition (C_2) works directly on the number of occurrences for each input value in a proper simplex $\sigma \in \mathcal{P}(\mathcal{I})$. Now a_1 denotes the number of occurrences of the most occurring value, a_2 the number of the second most occurring value etc... Then $\sigma \in C_2$ if $\sum_1^k \#a_i - \#a_{k+1} * k > t$.

► **Proposition 32.** *The condition C_1 is a condition for which k -set agreement is solvable and it is a maximal condition.*

Proof. The condition part is from [19], the maximality can be seen from the complex $\mathcal{U}(C_1)$. Every set of $n - t + 1$ processes will select one value, since this corresponds to one vertex using the Lemma 30. In [19], the maximum of this set is the “selection” function. We can see that the condition C_1 will remove all proper simplices, where there are more than k values that were selected. If we try to remove one less proper simplex $\sigma \in \mathcal{P}(\mathcal{I})$, we will have a full simplex in $\mathcal{U}(C) \cup \sigma$ with more than k initial values. By the usual set-Agreement impossibility, k -set agreement is not solvable on $\mathcal{U}(C) \cup \sigma$. ◀

► **Remark 33.** We could generalize the condition C_1 by choosing a different selection for each set of sizes $n - t + 1$ and removing a proper simplex where all of these elections add up to at least $k + 1$ values.

► **Proposition 34.** *Condition C_2 is not maximal.*

Proof. We provide a counter-example, for 3 processes, $t = 1$ and $k = 1$ (Consensus task). Condition C_2 keeps only the simplices where all the processes have the same input value. Using condition C_1 , we know that Consensus is solvable by adding more simplices. For example, we can add simplex $(1, 1, 0)$ and still solve consensus: The simplices with 3 times the same input decide their initial value and simplex $(1, 1, 0)$ decides 1. ◀

9 Conclusion

In this article, we build on the work of [7] to extend their General Colorless Computability Theorem to an even more general setting. Although the proof of this extension uses no new mathematical idea, presenting the different settings for colorless distributed computing in the topological framework was the main difficulty. A colorless tasks can be solved with a colored adversary (or input dependent adversary). We think that this diversity/complexity is a great testimony of the robustness and strength of the topological framework to model distributed systems.

On t -resilient models, the reduction to $\mathcal{U}(\mathcal{C})$ can be interpreted as very close to the classic algorithm that is known on t -resilient models. A group of processes of size $n + 1 - t$ will wait for each other and behave as one process, hence simulating a system with $t + 1$ processes. Moreover, it seems the approach of [7] using fiber bundles may not be easily extended to the input-dependent setting or the core-resilient setting. The multiple results of computability equivalence presented in Section 7.3 may be useful tools for future papers using topological methods. The Δ -compatible functions are relevant to produce reductions between different adversaries on the same input complex. Non-expanding tasks and vertex-deterministic tasks allow us to simplify the description of a task and provide direct proof (as for Proposition 30) and, maybe a better intuition of what a task is really about.

The fact that the solvability of k -set Agreement on a core-resilient model for a particular condition $\mathcal{C}$ is computable leads to an interesting question. The proposed algorithm is a brute force method for finding all possible simplicial maps. Maybe there is a more efficient algorithm that uses the structure provided by the carrier map and the combinatorial structure of complexes. A second complexity question arises: looking at the existence of a simplicial map from $\mathcal{U}(\mathcal{H})$ to $\mathcal{O}$ for k -set Agreement, can we find a class of tasks that has the same property? This research direction may lead to tight round-complexity for distributed tasks.

References

- 1 Jirí Adámek, Horst Herrlich, and George E. Strecker. *Abstract and Concrete Categories - The Joy of Cats*. Dover Publications, 2009.
- 2 Yehuda Afek and Eli Gafni. *Asynchrony from Synchrony*, pages 225–239. Number 7730 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013.
- 3 Hagit Attiya, Armando Castañeda, and Thomas Nowak. Topological characterization of task solvability in general models of computation. In Rotem Oshman, editor, *DISC'23*, 2023.
- 4 Elizabeth Borowsky and Eli Gafni. Generalized flip impossibility result for t -resilient asynchronous computations. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 91–100, New York, NY, USA, 1993. ACM Press. doi:10.1145/167088.167119.
- 5 Elizabeth Borowsky and Eli 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, pages 189–198. ACM, 1997. doi:10.1145/259380.259439.
- 6 Yannis Coutouly and Emmanuel Godard. A topology by geometrization for sub-iterated immediate snapshot message adversaries and applications to set-agreement. In Rotem Oshman, editor, *37th International Symposium on Distributed Computing, DISC 2023, October 10-12, 2023, L'Aquila, Italy*, volume 281 of *LIPICs*, pages 15:1–15:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.DISC.2023.15.
- 7 Yannis Coutouly and Emmanuel Godard. A simple computability theorem for colorless tasks in submodels of the iterated immediate snapshot. In Dan Alistarh, editor, *38th International Symposium on Distributed Computing, DISC 2024, October 28 to November 1, 2024, Madrid, Spain*, volume 319 of *LIPICs*, pages 16:1–16:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.DISC.2024.16.
- 8 Eli Gafni, Petr Kuznetsov, and Ciprian Manolescu. A generalized asynchronous computability theorem. In Magnús M. Halldórsson and Shlomi Dolev, editors, *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*, pages 222–231. ACM, 2014. doi:10.1145/2611462.2611477.
- 9 Emmanuel Godard and Eloi Perdureau. Back to the coordinated attack problem. *Math. Struct. Comput. Sci.*, 30(10):1089–1113, 2020. doi:10.1017/S0960129521000037.
- 10 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

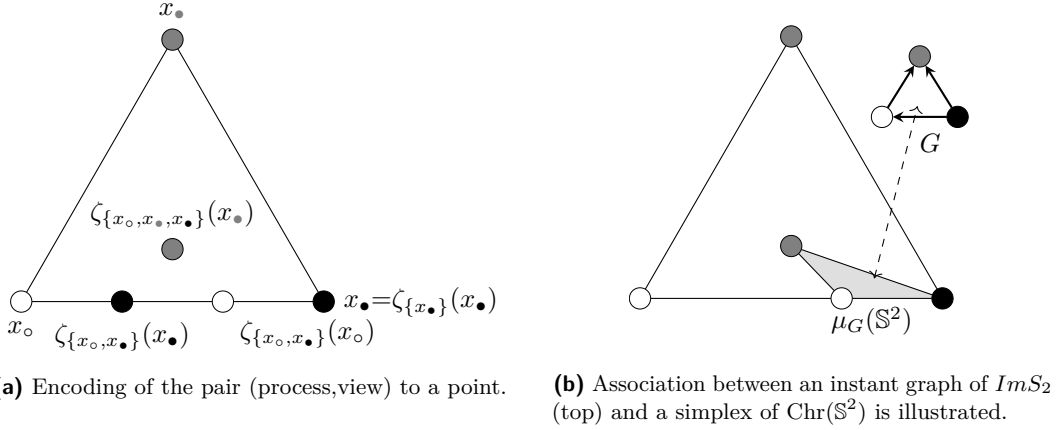
- 11 Maurice Herlihy, Dmitry N. Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 2013.
- 12 Maurice Herlihy and Sergio Rajsbaum. The topology of shared-memory adversaries. In *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '10, pages 105–113. Association for Computing Machinery, 2010. doi:10.1145/1835698.1835724.
- 13 Maurice Herlihy, Sergio Rajsbaum, and Michel Raynal. Computability in distributed computing: A tutorial. *SIGACT News*, 43(3):88–110, 2012. doi:10.1145/2421096.2421118.
- 14 Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 111–120. ACM, 1993. doi:10.1145/167088.167125.
- 15 Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6):858–923, 1999. doi:10.1145/331524.331529.
- 16 Dmitry N. Kozlov. *Combinatorial Algebraic Topology*, volume 21 of *Algorithms and computation in mathematics*. Springer, 2008. doi:10.1007/978-3-540-71962-5.
- 17 Dmitry N. Kozlov. Chromatic subdivision of a simplicial complex. *Homology Homotopy Appl.*, 14(2):197–209, 2012. URL: <http://projecteuclid.org/euclid.hha/1355321488>.
- 18 Petr Kuznetsov, Thibault Rieutord, and Yuan He. An asynchronous computability theorem for fair adversaries. In Calvin Newport and Idit Keidar, editors, *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, pages 387–396. ACM, 2018. URL: <https://dl.acm.org/citation.cfm?id=3212765>.
- 19 Achour Mostéfaoui, Sergio Rajsbaum, Michel Raynal, and Matthieu Roy. Condition-based protocols for set agreement problems. In Dahlia Malkhi, editor, *Distributed Computing, 16th International Conference, DISC 2002, Toulouse, France, October 28-30, 2002 Proceedings*, volume 2508 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2002. doi:10.1007/3-540-36108-1_4.
- 20 James R. Munkres. *Elements Of Algebraic Topology*. Addison Wesley Publishing Company, 1984.
- 21 Thomas Nowak, Ulrich Schmid, and Kyrill Winkler. Topological characterization of consensus under general message adversaries. In *PODC*, pages 218–227. ACM, 2019. doi:10.1145/3293611.3331624.
- 22 Sergio Rajsbaum. Iterated shared memory models. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics*, pages 407–416, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi:10.1007/978-3-642-12200-2_36.
- 23 Thibault Rieutord. *Combinatorial characterization of asynchronous distributed computability. (Caractérisation combinatoire de la calculabilité distribuée asynchrone)*. PhD thesis, University of Paris-Saclay, France, 2018. URL: <https://tel.archives-ouvertes.fr/tel-02938080>.
- 24 M. Saks and F. Zaharoglou. "wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM J. on Computing*, 29:1449–1483, 2000. doi:10.1137/S0097539796307698.

A The Standard Chromatic Subdivision

Here we present the standard chromatic subdivision, [11] and [16], as a geometric complex. We start with chromatic subdivisions.

► **Definition 35** (Chromatic Subdivision). *Given $(S, \mathcal{P})$ a chromatic simplex, a chromatic subdivision of S is a chromatic simplicial complex $(C, \mathcal{P}_C)$ such that*

- C is a subdivision of S (i.e. $|C| = |S|$),
- $\forall x \in V(S), \mathcal{P}_C(x) = \mathcal{P}(x)$.



■ **Figure 5** Construction of $Chr(S^2)$ as a geometric encoding for IIS_2 .

Note that it is not necessary to assume $V(S) \subset V(C)$ here, since the vertices of the simplex S being extremal points, they are necessarily in $V(C)$.

We start by defining some geometric transformations of simplices (here seen as sets of points). The choice of the coefficients will be justified later.

► **Definition 36.** Consider a simplex $V = (y_0, \dots, y_d)$ of size $d + 1$ in $\mathbb{R}^N$. We define the function $\zeta_V : V \rightarrow \mathbb{R}^N$ by, for all $i \in [0, d]$

$$\zeta_V(y_i) = \frac{1}{2d+1}y_i + \sum_{j \neq i} \frac{2}{2d+1}y_j$$

We now define directly in a geometric way the *standard chromatic subdivision* of simplex S , where $S = (x_0, x_1, \dots, x_n)$.

The chromatic subdivision $Chr(S)$ for the chromatic simplex $S = (x_0, \dots, x_n)$ is a simplicial complex defined by the set of vertices $V(Chr(S)) = \{\zeta_V(x_i) \mid i \in [0, n], V \subset V(S), x_i \in V\}$. The simplices of $Chr(S)$ are the set of $d + 1$ points $\{\zeta_{V_0}(x_{i_0}), \dots, \zeta_{V_d}(x_{i_d})\}$ that can be ordered by containment.

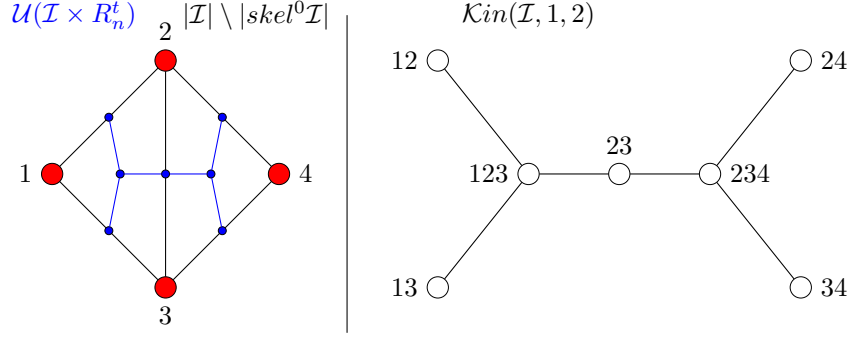
In Fig. 5, we present the construction for $Chr(S^2)$. For convenience, we associate $\circ, \bullet, \bullet$ to the processes 0, 1, 2 respectively. In Fig. 5a, we consider the triangle x_0, x_1, x_2 in $\mathbb{R}^2$, with $x_0 = (0, 0)$, $x_1 = (1, 0)$, $x_2 = (\frac{1}{2}, \frac{\sqrt{3}}{2})$. We have that $\zeta_{\{x_0, x_1\}}(x_1) = (\frac{1}{3}, 0)$, $\zeta_{\{x_0, x_2\}}(x_0) = (\frac{2}{3}, 0)$ and $\zeta_{\{x_0, x_1, x_2\}}(x_2) = (\frac{1}{2}, \frac{\sqrt{3}}{10})$. The relation between instant graph G (top) and simplex $\left\{(\frac{2}{3}, 0), (1, 0), (\frac{1}{2}, \frac{\sqrt{3}}{10})\right\}$ (gray area in Fig. 5b) is detailed in the section 5.2.

In the following, we will be interested in iterations of $Chr(S^n)$.

In [17], Kozlov showed how the standard chromatic subdivision complex relates to Schlegel diagrams (special projections of cross-polytopes), and used this relation to prove the standard chromatic subdivision was actually a subdivision. In [11, section 3.6.3], a general embedding in $\mathbb{R}^n$ parameterized by $\epsilon \in \mathbb{R}$ is given for the standard chromatic subdivision. The geometrization here is done choosing $\epsilon = \frac{d}{2d+1}$ in order to have “well balanced” drawings.

A.1 Additional figure

On the left of Figure 6, we have an input complex with 3 processes and $t = 1$; hence, we removed all the vertices of the $\mathcal{I}$ complex. Moreover we represented in blue $\mathcal{U}(\mathcal{I} \times R_n^t)$. On the right, we have every set of size 2 or 3 that can be formed by any proper simplices of $\mathcal{I}$. We can remark that in this example $\mathcal{U}_1$ is equal to $Kin(\mathcal{I}, 1, 2)$.

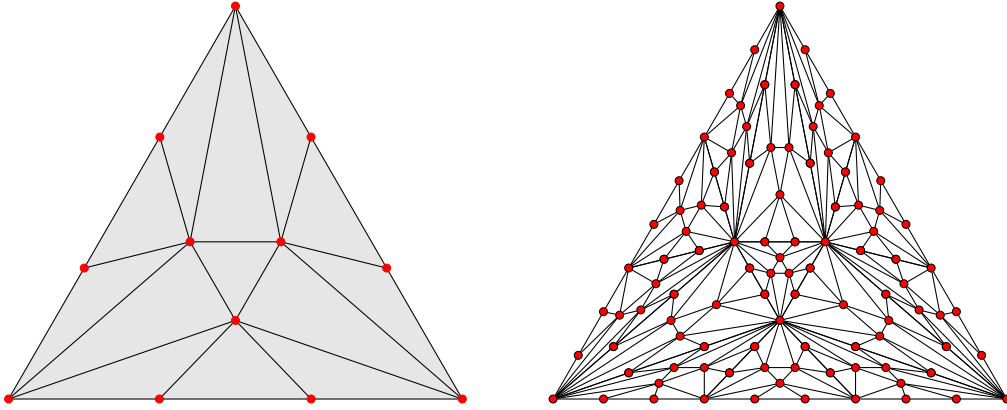


■ **Figure 6** Example of $Kin(\mathcal{I}, 1, 2)$ and $\mathcal{U}_1(\mathcal{I})$.

A.2 A complete presentation of the geometrization of t -resilient model

The fair executions on a simplex σ of $\dim(\sigma) = n - t - 1$ have $n - t$ processes that see each other an infinite amount of time (which means that they have not crashed). By definition, these executions do not belong to R_t^n , because $t + 1$ processes will not ultimately participate.

For example, with 3 processes and one crash at most, we obtain the complex in Figure 4b the set of execution $E = (G_1 + G_2 + G_3)^\omega + (G_5 + G_6 + G_7)^\omega + (G_9 + G_{10} + G_{11})^\omega$ are removed in R_1^2 and $geo(E)$ correspond to the 3 points that for this triangle, which is the skeleton of dimension $n - t - 1 = 0$ of $\mathcal{I}$.



■ **Figure 7** The geometrization of R_1^2 .

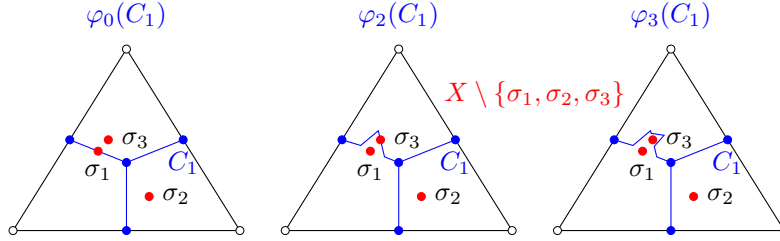
To obtain the complete geometrization of R_t^n we need to repeat this process at every step of the standard chromatic subdivision, for example: $G_{13}.E$ is a set of executions that will also be removed (and correspond to the 3 vertices of the triangle G_{13}). This process yields a fractal like structure, another partial representation can be seen in Figure 7 where all points removed are marked in red for different levels of subdivision.

B Additional Proofs

► **Lemma 37.** *Let $X \subseteq \mathbb{R}^n$, let τ a simplex of dimension t such that $|\tau| \subseteq X$. Let $Z = \{\sigma_1, \sigma_2, \dots, \sigma_{k_1}\}$ a collection of simplices of $\dim(\sigma_i) \leq n - t - 1$ and $\partial(|\tau|) \subseteq X \setminus Z$. Then $\forall k_2 \in \mathbb{N}, k_2 \leq k_1, \exists \varphi_{k_2} : |\tau| \rightarrow X \setminus \{\sigma_1, \dots, \sigma_{k_2}\}$ is a continuous function that is the identity on $\partial(|\tau|)$*

The proof of this lemma revolves around taking each intersection of $|\tau|$ with a $|\sigma_i|$ and constructing a cone to move away from $|\sigma_i|$. We then use that lemma to, iteratively, correct any up-part of the chromatic subdivision.

Proof. Let $k_2 \in \mathbb{N}$ such that $k_2 \leq k_1$. We will construct a series of function $\varphi_{k_2} : |\tau| \rightarrow X \setminus \{\sigma_1, \dots, \sigma_{k_2}\}$ a continuous property such that $\partial(|\tau|)$ is the identity. We set $\varphi_0(|\tau|)$ as the identity function. Let Y_{k_2} a connected subspace of $|\sigma_{k_2}| \cap \varphi_{k_2-1}(\tau)$. We have that $\dim(Y_{k_2}) \leq t-1$ and $\partial(|\tau|) \cap \sigma_{k_2}$ since σ_{k_2} is a simplex. Let $N_\epsilon(Y_{k_2}) = \{x \in X \mid \text{MIND}(x, Y_{k_2}) = \epsilon\}$. Then let $\epsilon_0 = +\infty$ and let $\epsilon_i \in \mathbb{R}$ positive, such that $\partial(|\tau|) \cap N_{\epsilon_{k_2}}(Y_{k_2}) = \emptyset$ and $N_{\epsilon_{k_2}}(Y_{k_2}) \cap \varphi_{k_2-1}(\tau)$ is a deformation of $\mathbb{S}^{\dim(\partial(\tau))}$ and $\epsilon_{k_2} < \epsilon_{k_2-1}/3$. Such ϵ_{k_2} exist since Y_{k_2} is a closed set and $\varphi_{k_2-1}(|\tau|) \setminus \partial(|\tau|)$ is an open set. Now, let $x \in \text{Int}(Y_{k_2})$, let $y \in N_\epsilon(Y_{k_2}) \setminus \varphi_{k_2-1}(|\tau|)$. We can construct $B_{k_2} : N_\epsilon(Y_{k_2}) \cap \varphi_{k_2-1}(|\tau|)$ completed in $\varphi_{k_2-1}(|\tau|)$. And $A_{k_2} = N_\epsilon(Y_{k_2}) \cap \varphi_{k_2-1}(|\tau|)$ completed in $N_\epsilon(Y_{k_2})$ and of dimension t going through y . Then $\varphi_{k_2}(B_{k_2})$ is the projection of B_{k_2} to $A_{k_2} \setminus B_{k_2}$. This operation is continuous, hence $\varphi_{k_2}(|\tau|)$ is a continuous function. Moreover, we have that $\forall \sigma_i \in \{\sigma_1, \dots, \sigma_{k_2}\}, \varphi_{k_2}(|\tau|) \cap |\sigma_i| = \emptyset$ since we have that $\sum_{i=1}^{k_2} \epsilon_j < \epsilon_i$. Hence, $\varphi_{k_2}(|\tau|)$ is a continuous deformation of $w\text{Int}(|\tau|)$ that does not intersect $\{\sigma_1, \dots, \sigma_{k_2}\}$. ◀



■ **Figure 8** Example of the construction from Lemma 37.

Figure 8 is a representation of a simple case from Lemma 37. We have $n = 2, t = 1$, X is the full triangle, and we remove $\{\sigma_1, \sigma_2, \sigma_3\}$. On φ_0 the blue segment isn't modified; it gets modified in φ_2 since σ_1 intersect τ . The modification is made to form a cone that avoids σ_1 , even if it can intersect another point, σ_3 in this figure. Finally, in φ_3 , σ_3 is avoided and $\varphi_3(C_1)$ doesn't need further modification.

On a core-dependent adversary, we show that restricting crashes happening before the $k - th$ round does not change the computability. Let $\mathcal{H}_k = \{w \in IIS \mid \forall p \in \Pi \setminus Q(w), \forall q \in Q(w), \forall r \in \mathbb{N}, r > k, (p, q) \notin A(w(r))\}$, with $Q(w)$ the set of processes that can see each other an infinite amount of time in w .

► **Proposition 38.** *Let $(\mathcal{I}, \mathcal{O}, \Delta)$ a colorless task, let $\mathcal{C} \subseteq \mathcal{I}$ a condition on the input and $\mathcal{H}$ a core-dependent adversary on $\mathcal{C}$, then, for any $k \in \mathbb{N}$ the task is solvable on $\mathcal{H}_k$ if and only if it is solvable on $\mathcal{H}_0$.*

Proof. We denote by $\mathcal{U}_k = \mathcal{U}(\text{Chr}^k(\mathcal{C}))$. Note that $|\mathcal{U}_{k+1}| \subset |\mathcal{U}_k|$. We prove the result by recursion.

Assume this is true for k . We work simplex by simplex from $\text{Chr}^{k+1}(\mathcal{C})$. Denote τ the current (maximal) simplex and t the size of the current core. By using Lemma 37 with $\mathcal{U}_k$ it is possible to derive a new continuous function that avoids all new simplices appearing in $\text{Chr}^{k+1}(\mathcal{C})$.

This way, it is possible to project $|\mathcal{U}_{k+1}|$ on $\mathcal{U}_0$. So, from Thm 16, we get equivalent colorless computability on $\mathcal{H}_{k+1}$ and $\mathcal{H}_0$. ◀

Using the previous proposition for all $k \in \mathbb{N}$, we have that this sequence of $\mathcal{H}_k$ forms a projective limit, a notion from category theory, where $\mathcal{H}$ is the limit of $\mathcal{H}_k$. Since the projection holds for any k , the limit mapping also exists (topological spaces form a *complete* category) and is also continuous.

In more detail, we consider the family of inclusion morphisms $f_{k,k'} : \text{geo}(\mathcal{H}_{k'}) \hookrightarrow \text{geo}(\times \mathcal{H}_k)$, with $k \leq k'$. Of course, we have $f_{k,k'} \circ f_{k',k''} = f_{k,k''}$ when $k \leq k' \leq k''$. So this sequence defines a system of morphisms that is a *diagram* and since the category of topological spaces with continuous functions is complete [1, Section 12.6], the limit exists and there is a continuous mapping from $\mathcal{U}(\mathcal{H}_0)$ to this limit. The last step is to see that the limit $\mathcal{L}$ is actually (homeomorphic to) $\text{geo}(\mathcal{H})$, since $\text{geo}(\mathcal{H}) = \bigcap_k \text{geo}(\mathcal{H}_k)$.