# Distributed Network Computing through the Lens of Combinatorial Topology (DUCAT)

**Consortium:** IRIF (CNRS & Université de Paris) — LIS (CNRS & Université Aix-Marseille)

**Summary table of persons involved in the project:**

| Partner | Name | First name | Current position | Role & responsibilities in the project | Person-Month |
|---|---|---|---|---|---|
| South team (LIS) | CHALOPIN | Jérémie | CR CNRS | Coordinator South team<br>Coordinator WP2<br>Leader Tasks 2.2, 3.2 | 50%<br>24pm |
| North team (IRIF) | DELPORTE | Carole | PR Université de Paris | Leader Task 3.3 | 25%<br>12pm |
| North team (IRIF) | FAUCONNIER | Hugues | PR Université de Paris | Leader Task 2.3 | 25%<br>12pm |
| North team (IRIF) | FRAIGNIAUD | Pierre | DR CNRS | Project coordinator<br>Coordinator WP1<br>Leader Tasks 1.3, 2.1, 3.1 | 50%<br>24pm |
| South team (LIS) | GODARD | Emmanuel | PR Université Aix-Marseille | Coordinator WP3<br>Leader Task 1.2 | 50%<br>24pm |
| South team (LIS) | IMBS | Damien | MCF Université Aix-Marseille | | 25%<br>12pm |
| North team (IRIF) | KUZNETSOV | Petr | PR Telecom Paris | Leader Task 1.4 | 25%<br>12pm |
| North team (IRIF) | TASSON | Christine | MCF Université de Paris | | 12,5%<br>6pm |
| South team (LIS) | TRAVERS | Corentin | MCF Université de Bordeaux | Leader Task 1.1 | 25%<br>12pm |

Repartition of the working load per partners:

- North team: 66 pm + 1 PhD students will be at 100% on the project (plus 8 pm master interships).
- South team: 72 pm + 1 post-doc will be at 100% on the project (plus 8 pm master interships).

It is worth to mention that DUCAT will also take benefit from direct interactions with two world-class experts in distributed computing: David Peleg (Weizmann Institute of Science, Israel), and Sergio Rajsbaum (UNAM, Mexico). David Peleg, author of the textbook [41] on distributed network computing, will stay 4 months at IRIF in Spring 2021, supported by the FSMP Chairs program. Sergio Rajsbaum, co-author of the textbook [29] on topology applied to distributed computing, is already planing to spend a couple of months at IRIF and LIX in 2021, supported by Ecole Polytechnique and Universisté de Paris.

## I Proposal's context, positioning and objective(s)

### a Objectives and research hypothesis

Distributed computing is the field of computer science that studies how a collection of autonomous computing entities (processes) can cooperate in the absence of a centralized coordinator, to efficiently achieve a common objective. Distributed computing is often referred to as *computing with uncertainty* as every process might have to take decisions with only a partial knowledge about the status of other processes, due to asynchrony, failures, spatial distribution, etc. The turn of the 21st century witnessed a significant breakthrough in our understanding of distributed shared-memory computing, thanks to the use of the elements of *combinatorial topology*. In this project, we intend to apply the topological approach to distributed *network* computing, with the objective of deriving novel conceptual insights for the design of network algorithms, and generic tools for deriving lower bounds and impossibility results.

**Indistinguishability.** In distributed computing, impossibility results and lower bounds are often obtained using indistinguishability arguments. Such arguments consist in proving that, whenever communication between the processes is somehow restrained, a given process might not be able to distinguish between system configurations in which different, possibly conflicting, decisions must be taken. For many models and problems of distributed computing, identifying the existence of such configurations is challenging, and some of the most celebrated results in distributed computing precisely addressed this challenge. The FLP theorem [20] regarding consensus in asynchronous systems subject to crash failures, and the $\Omega(\log^* n)$ lower bound[1] by Linial [36] on the number of rounds for vertex-coloring in failure-free synchronous networks, are typical examples of such outstanding results. In fact, literally hundreds of impossibility results and lower bounds have been obtained using indistinguishability arguments [19, 38]. However, these results were obtained using *ad hoc* techniques, without providing a *global picture* that would allow us to reason about impossibilities and lower bounds in distributed computing in a uniform way.

**The advent of combinatorial topology.** At the turn of the 21st century, a significant discovery has been made in the context of distributed computing, by analyzing distributed computations using elements of combinatorial topology. The celebrated results by Herlihy and Shavit[2][30] specifically connects asynchronous shared-memory computing with combinatorial topology. Roughly, all combinations of input states (resp., output states) for a distributed task can be described as combinatorial structures, called *simplicial complexes*, defined by all possible combinations (simplices) of compatible individual input (resp., output) states. A *carrier map* completes the specification of the task, by defining which output simplices are allowed for each input simplex. In essence, [30] established that the execution of a distributed asynchronous algorithm with shared memory corresponds to a specific *topological deformation* of the input complex — namely, subdivisions. The outputs of the processes induce a *simplicial map* from the subdivided input complex to the output complex. Overall, [30] show that there exists an algorithm for solving a task in the read-write shared-memory asynchronous model if and only if there exists a simplicial map from a subdivision of the input complex to the output complex, which complies with the specification of the task. The impossibility of consensus and, more generally, set agreement, is a direct consequence of this characterization. E.g., the input complex of consensus is connected, while its output complex is not. Hence, there cannot be a simplicial map from any subdivision of the input complex to the output complex that complies with the specification of consensus.

**Applications of combinatorial topology in distributed computing.** The approach discovered by Herlihy and Shavit [30] was later successfully extended to various models of distributed computing, including $t$-resilient [48], Byzantine [39], anonymous [16, 35], adversarial [34], and other models. Combinatorial topology has not only provided distributed computing with a powerful generic toolbox, but, perhaps more importantly, contributed to *deepening our conceptual understanding* of distributed computing, and enabled *bridging the gaps* between seemingly different and incomparable models. The book [29] includes numerous illustrations of these two aspects of combinatorial topology applied to distributed computing. In a big picture, combinatorial topology tells us that the task of consensus is about connectivity, the task of k-set agreement is about higher-order (k-1)-connectivity, approximate agreement is about stretching, and, in general, computability in the presence of asynchrony and failures is about subdividing complexes, creating holes, and determining simplicial maps.

**Distributed computing in networks.** Until now, however, these insights were derived for communication models in which every pair of processes can communicate directly via powerful abstractions such that shared memory or point-to-point communication channels of unbounded capacity. This was partially caused by the fact that these results are based on the use of *full-information* protocols in which every process periodically shares its complete state with the others. This approach might be challenging to apply in distributed network models, where the ability of the processes to communicate is limited by the network structure,

[1] Roughly, $\log^* n$ is the number of times the function $\log$ must be iterated to go from $n$ to a number less than 1.

[2] M. Herlihy and N. Shavit, together M. Saks and F. Zaharoglou, were awarded the prestigious Gödel Prize in 2004 for establishing the fundamental impossibility of wait-free set agreement using combinatorial topology. E. Borowsy and E. Gafni, who concurrently obtained the same result, were later awarded the Dijkstra prize.

and the channel capacity. These obstacles might explain why distributed network computing has until now ignored combinatorial topology. Yet, this important sub-domain of distributed computing has flourished during the last decade, with a blossom of outstanding results, including the design of faster algorithms or better lower bounds for practically and/or conceptually important tasks like coloring, maximal independent set (MIS), distributed Lovasz Local Lemma (LLL), etc. Nevertheless, despite the tremendous successes of the distributed computing community in tackling the power and limitation of network computing, and although a plethora of powerful techniques have been developed for designing efficient algorithms in this framework, one can still raise reservations.

- First, the big picture of distributed network computing is still missing. That is, the *conceptual* comprehension of this domain remains limited. Despite outstanding recent technical results regarding lower bounds [3] and algorithms [44], the techniques remain somehow ad hoc, and the global understanding of the impact of the network structure on the computation remains somehow limited.
- Second, major problems remain widely open, although studied for more than a quarter of a century. For instance, regarding the central task of $(d+1)$-coloring networks of maximum degree $d$, there is still an asymptotic gap between the best known lower bound of $\Omega(\log^* n)$ rounds [36], and the recent best known polylogarithmic upper bound [44], despite intensive research.

The objective of DUCAT is to remedy this situation. We intend to apply combinatorial topology to distributed network computing, with two objectives:

**Qualitative:** understanding distributed computing in static or dynamic networks the same way combinatorial topology helped to understand asynchrony and failures in shared-memory and message passing systems;

**Quantitative:** providing generic tools for analyzing the power and limitations of distributed network algorithms in terms of their ability to solve problems, under models assuming that the underlying network is incomplete, of bounded capacity, and/or evolving over time.

## b Position of the project as it relates to the state of the art

We provide a quick survey of the most recent and/or significant advances in distributed computing, by distinguishing three frameworks: (1) *asynchronous failure-prone computing*, in which combinatorial topology has played an important role since the turn of the century, (2) *synchronous failure-free network computing*, and (3) *dynamic networks*. The latter two frameworks have not yet benefited from combinatorial topology, and the aim of this project is precisely to bring them on par with asynchronous failure-prone computing, as far as applying topology to distributed computing is concerned.

### b1 Asynchronous failure-prone computing

Failures and asynchrony are two factors that render seemingly simple problems impossible to solve in a distributed system. Meanwhile, in many real systems, nontrivial synchrony assumptions are elusive and system components are prone to failures. Here comes the problem space of asynchronous failure-prone computing. One of the most fundamental results in distributed computing was establishing the fact that, in asynchronous systems, the presence of even just one potentially faulty process may prevent the remaining processes from reaching consensus on one of their input values [20], assuming the processes are able to communicate over reliable point-to-point channels or even read-write shared memory. Since then, one of the primary challenges in distributed computing is to *characterize* the class of problems that can be solved in an asynchronous and fault-tolerant way. In other words, how to demarcate the border between solvable and unsolvable tasks, depending on both the nature of the tasks, and the specification of the computing environment.

**Read-write task computability.** A lot of efforts were applied to the *read-write* shared-memory model, as it is probably the simplest model capturing the very basic functionality of modern multiprocessors. As for problems, the popular class of *tasks* [30] model *one-shot* (one-time use) synchronization primitives. A task can be seen as a distributed variant of a function from classical (centralized) computing: given a distributed input (i.e., an *input vector*, specifying one input value for every process) the processes are required to produce a distributed output (i.e., an *output vector*, specifying one output value for every process), such that the input and output vectors satisfy the given *task specification*. For example, the task of *consensus* assumes that processes start with binary inputs, and they need to agree on one of the inputs. The task of $k$*-set agreement* is a generalization of consensus where inputs are in a set of $k+1$ values and, in every execution, the set of outputs is a subset of inputs of size at most $k$.

**Combinatorics of distributed computing.** As mentioned before, fruitful approach in characterizing task solvability in shared-memory models is based on combinatorial topology [29]. Intuitively, a set of executions of a distributed algorithm is represented as a geometrical structure (a *simplicial complex*). More precisely, a local state of a process at the end of an execution in the set is modeled as a *vertex* and the local states a set of processes can reach in a given execution is then modeled as a *simplex*, i.e., a set of vertices. Naturally, two simplices intersect when at least one process reaches the same local states in the two corresponding executions. Now the properties of the model at hand can be captured through the properties of the complex. One important property of a simplicial complex is its *connectivity*, i.e., the lowest trivial homology group of the corresponding space. A simplicial complex $C$ is said to be $k$*-connected* if any continuous map from a $k$-dimensional sphere on $C$ can be extended to a continuous maps from a $(k+1)$-dimensional disk on $C$. In particular, $0$-connected complexes are graph connected, and $1$-connected ones are *simply* connected. It turns out that the simplicial complex corresponding to a set of bounded layered snapshot shared-memory executions starting from some input configuration can be represented as a *chromatic subdivision* of $(n-1)$-dimensional simplex, where $n$ is the number of processes in the system.

**Characterizing wait-free solvable tasks.** The question of whether a given task $T$ is *wait-free* solvable is equivalent to the question whether there exists a chromatic subdivision of the *input complex* of $T$ that can be continuously mapped to the *output complex* of $T$ in a *carrier-preserving* way [30]. Informally, "carrier-preserving" means here that a process only observing a set $P$ of processes in an execution obtains an output that can be obtained when only the set $P$ participates (intuitively, the output of a process cannot depend on processes it has not seen). The most noticeable application of the topological reasoning in distributed computing was showing that the task of *set agreement*, i.e., $(n-1)$-set agreement in a system of $n$ processes, has no *wait-free* solution, i.e., cannot be solved assuming that any subset of processes can be faulty [6, 30, 31, 45, 46]. Essentially, all these results reduce solving set agreement to finding Sperner coloring [50] of a manifold, which is known to be impossible.

**Adversarial computability.** Recently, interesting generalizations of this approach have been suggested for *adversarial* models [15] specifying subsets of processes that are allowed to fail in system executions. Intuitively, adversaries allow for abstracting out models in which processes failures might occur in the non-independent and non-identical manner. A run is in the corresponding (read-write) $S$*-adversarial model* if the set of processes taking infinitely many (read-write) steps in it is in $S$. Using powerful simulation tools pushing further the original ideas of [6, 21, 24], and using combinatorial topology, a topological characterization of task solvability in the presence of a large class of *fair* adversaries has recently been suggested [34]. A similar work has been done to characterize anonymous task solvability [16, 35]. This characterization implies that the anonymity does not reduce the computational power of the asynchronous shared-memory model as far as "colorless" tasks are concerned [16].

### b2 Synchronous failure-free network computing

In the context of distributed network computing, a significant fraction of the recent work has been focusing on synchronous failure-free computing. Two models have concentrated most of this attention in this framework. Both assume that the nodes are given arbitrary distinct identifiers (IDs) in a range polynomial in the number of nodes in the network. They both assume that computation proceeds in lockstep: all processes wake up simultaneously, and execute the same algorithm that performs in synchronous *rounds*. During a round, every process performs some local computation, sends messages to its neighbor in the graph $G=(V,E)$ modeling the network, and receives the messages sent by these neighbors. In the LOCAL model [36, 40], there are no restrictions on the message size exchanged during a round. Thus, this message enables the use of *full information* protocols. Instead, the CONGEST model [25, 33, 42] assumes a bound $B$ on the number of bits that can be transferred on a link during a round (it is typically assumed that

$B = O(\log n)$). The LOCAL model is therefore perfectly suited to study *locality* in distributed computing, that is, what can be computed when processes are bounded to gather information only from the processes in their vicinity. The CONGEST model is instead perfectly suited to measure *information*, that is, how much information should every process collect about the overall system for computing solutions.

**Contributions to the LOCAL and CONGEST models.** Since the 90's, there have been an enormous amount of work on both LOCAL and CONGEST models, and we are referencing here only major contributions in direct connection with our project. The paper [51], the booklet [4], and the textbook [41] provide excellent introductions to local computing. A large fraction of the literature on the LOCAL model deals with solving graph problems that are either conceptually or practically relevant, including vertex- and edge-coloring, maximal independent set (MIS), maximal matching (MM), etc. All these problems, but $(d+1)$-coloring graphs of maximum degree $d$, were known to be solvable in a polylogarithmic number of rounds, but it is only with the recent breakthrough [44] from last summer that the same kind of upper bound was established for $(d+1)$-coloring. In fact, while $(2d-1)$-edge-coloring, MIS, and MM have essentially polylogarithmic lower bounds [32], the only lower bound known for $(d+1)$-coloring is $\Omega(\log^* n)$ rounds by Linial [36], established more than a quarter-of-a-century ago. This bound is known to be tight for the ring [40], as well as for bounded-degree graphs in general [5, 22], but the gap between the best known lower and upper bounds remain large in general. The *round-reduction* technique, implicit in [36], has been formalized and generalized in [9] to arbitrary *locally checkable labeling* (LCL) task. This technique was then applied [3, 9] for establishing lower bounds for MM, and for variants of coloring. Yet, the complexity $(d+1)$-coloring remains unknown. It is worth mentioning that *randomization* helps in distributed computing in general, as it helps to break symmetry easily. However, the role of randomization has been shown [12] to be somehow limited under the LOCAL model, as a randomized algorithm in $n$-node networks for solving an LCL task $T$ cannot be faster than the fastest deterministic algorithm in networks with $\sqrt{\log n}$ nodes. Moreover, the derandomization techniques in [26, 27] shows that this gap is essentially tight. In other words, *designing faster randomized algorithms often requires to design faster deterministic algorithms*. This invited the community to carry on efforts on the design of *determinitic* algorithms.

Problems considered in the CONGEST model are more "global" in nature, including spanning tree construction, MST, all-pair shortest paths, etc. We again refer to [41] for an excellent introduction to network computing under congestion constraints. Quite interestingly, most of the aforementioned problems admit algorithms with complexity $\tilde{O}(D + \sqrt{n})$ rounds [25, 33] in networks with diameter $D$. These bounds are often tight, and the lower bounds are quasi-systematically obtained by implicit or explicit reduction from communication complexity [49]. A variant of the CONGEST model, called CONGESTED CLIQUE, and relevant to this project, has been introduced in [37]. It is essentially the CONGEST model restricted to the clique (i.e., the complete graph $K_n$). This model is very strong [11, 37]. In fact, it was proved that obtaining non-trivial lower bounds for the CONGESTED CLIQUE model would enable obtaining a breakthrough in the design of lower bounds for certain classes of circuits [17].

**Topological approach.** It is only recently that efforts have been made to approach network computing from a topological perspective, and only for the LOCAL model. Specifically, two approaches have been considered: *global* and *local*. The global approach, introduced by members of the project [10], considers the simplicial complexes formed by global states of the system. That is, a vertex of these complexes is a pair $(p, v)$, where $p$ is the *name* of a process, and $v$ is a value associated to that process, whether it be its input value (forming the input complex $\mathcal{I}$), or its output value (forming the output complex $\mathcal{O}$), or its view after $t \geq 1$ rounds, i.e., the total information acquired from the processes at distance at most $t$ (forming the protocol complex $\mathcal{P}^{(t)}$). It was observed that, even if the processes are given fixed IDs, e.g., process $p_i$ gets ID $i$, $i = 1, \ldots, n$, in $n$-node networks $G$, and even if each process is a priori aware of which vertex of $G$ every process $p_i$ occupies in the network, the protocol complexes evolves with time in a way completely different from the protocol complexes occurring in the context of asynchronous shared-memory or message passing. In particular, facets are preserved, but scissor cuts appear between the facets along with the course of computation — see Figure 1 borrowed from[10]. By analyzing the topology of the protocol complexes resulting from these cuts, as a function of the structure of the underlying network, [10] provides a structural characterization of the number of rounds required to solve consensus and $k$-set agreement in networks.

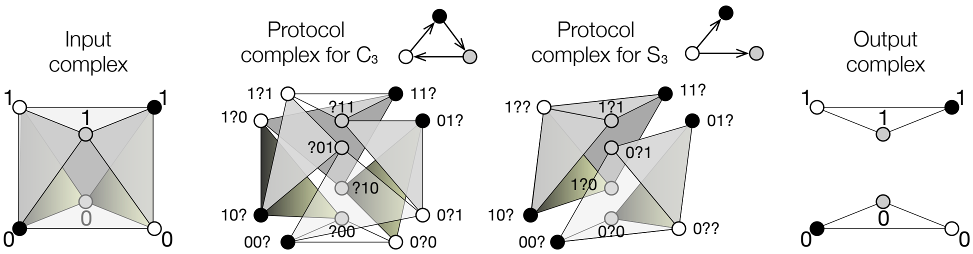


**Figure 1.** *Impact of the structure of the network on the protocol complex for binary consensus with three processes. A view "$xyz$" labeling a vertex means that the process corresponding to this vertex knows the input values $x$ from process ○, $y$ from process •, and $z$ from process •.*

Extending the results in [10] to the LOCAL model faces a difficulty caused by the presence of IDs. Indeed, with IDs, even the input complex of binary consensus as displayed on Figure 1 becomes far more complicated, with vertices of the form $(p, (\text{id}, v))$ where $p$ is the process name, id is the arbitrary ID provided to $p$, and $v \in \{0, 1\}$. Whenever the range of IDs is $\{1, \ldots, N\}$ with $N = poly(n)$ in $n$-node networks, there are $n!\binom{N}{n}$ ways of assigning IDs to the nodes, which makes even the input complex cumbersome. To overcome this explosion, [23] introduced *local* complexes, where a facet is solely composed of a process and its neighbors, together with their inputs, outputs, or views, and their IDs. Using local complexes enables to use Ramsey's Theorem for restricting IDs in a *finite* range, in a way similar to what was done in [40]. The framework of local complexes applies to all LCL task, and enabled to reprove the celebrated Linial's lower bounds for 3-coloring the ring by reduction from $k$-coloring to $2^{2^k}$-coloring.

### b3 Dynamic networks

Uncertainty is a proeminent problem in distributed computing as has already been underlined before. Dynamic networks aggregate both time and space uncertainty. They are specified by communication networks that evolve with time, for which the underlying communication structure can change dramatically from time to time. Such a framework fits with recent networks induced by the mobility of the users, including, e.g., vehicular networks. Interestingly, the mathematical model that represents dynamic networks has also been introduced to simplify shared memory models. Actually, this model is very general, and has appeared many times, motivated by different factors, and under different names. The *mobile omissions model* was introduced in [47] in 1989. It was then re-introduced later as the *iterated immediate snapshot* model in [8], with its final evolution as the *message adversary* model in [2] in 2013. It was also presented in the "Heard-of" model [13] in 2009. In the case of non-colored tasks, equivalences have been proved [7, 43] between all these synchronous specifications, and asynchronous models. In the case of dynamic networks, whenever the communication primitive is only a broadcast to the current neighbors, this (class of) model(s) can also be used.

Computability problems have been mainly investigated for consensus-like tasks. A message adversary is said to be oblivious, or iterated, when the instant graphs can be chosen from a given set of graphs. This is an important class of message adversaries, and suffix-closed message adversaries are a generalization of this model. Regarding distributed computability in oblivious message adversaries, the solvability of consensus in the context of communication networks with arbitrary topology has been introduced in [47]. The exact solvability was established in [14] for oblivious message adversaries, but it is still open in the general (non-oblivious) case. For the special case of a message adversary corresponding to a bounded number of omissions, the consensus problem is equivalent to the broadcast problem. That is, the network must be connected at each round, i.e., the number of failures must be less than the connectivity of the underlying graph, for solving 1-set agreement. The solvability of the $k$-set agreement problem has been considered in [28], and tackled with topological methods for an oblivious adversary where the dynamic network is upper-closed.

The main conclusion that can be derived from previous work in dynamic networks is that these networks model message adversaries that are more general than those considered under the wait-free shared memory model. However, topological methods have already proved to be applicable in this context of dynamic networks. In the case of agreement problems, establishing a separation between indistinguishability of dimension $k$, and partitioning into $k$ components is an important open problem.

## c Methodology and risk management

This section describes the methodology used by DUCAT for reaching its objectives, and the strategy for overcoming the obstacles to be faced.

### c1 Methodology

The project DUCAT is decomposed into three work-packages (WP):
**WP1:** Topology,
**WP2:** Computability,
**WP3:** Complexity.
The essence and role of each of these WPs can be exhibited by inspecting Figure 2, which represents the interplay between the input complex $\mathcal{I}$, the output complex $\mathcal{O}$, and the protocol complex $\mathcal{P}$ in the topological framework for distributed computing [29]. Recall that a *simplicial complex* $\mathcal{K}$ is a collection of non-empty subsets of a finite set $V$, closed under inclusion, i.e., for every $\sigma \in \mathcal{K}$, and every non-empty $\sigma' \subset \sigma$, it holds that $\sigma' \in \mathcal{K}$. Every $\sigma \in \mathcal{K}$ is called a *simplex*, and every $v \in V$ is called a *vertex*. For instance, a graph $G = (V, E)$ can be viewed as the complex $\mathcal{K} = \{\{v\} : v \in V\} \cup E$ on the set $V$ of vertices. Given a system with $n$ processes $p_1, \ldots, p_n$, a set $\{(p_i, v_i), i \in I\}$ with $I \subseteq [n]$ and $I \neq \varnothing$ is a simplex of the input complex $\mathcal{I}$ if there exists an input configuration of the system for which process $p_i$ is given input value $v_i$, for every $i \in I$. The definitions for $\mathcal{O}$, and $\mathcal{P}$ are similar, w.r.t. output configurations, and configurations resulting from the execution of a certain protocol, respectively.

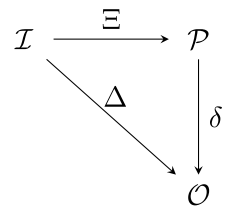


**Figure 2.** Commutative diagram of Herlihy-Shavit Theorem

A distributed computing *task* can be represented as a triple $(\mathcal{I}, \mathcal{O}, \Delta)$ where $\Delta : \mathcal{I} \to \mathcal{O}$ is a *carrier map* specifying, for each input configuration $\sigma \in \mathcal{I}$, the collection of legal output configurations $\Delta(\sigma) \subseteq \mathcal{O}$. The protocol complex $\mathcal{P}^{(t)}$ represents all possible states of the system after a certain time $t$, where the notion of "time" depends on the considered model: it is the protocol complex $\mathcal{P}$ at time $t$. The function $\Xi : \mathcal{I} \to \mathcal{P}^{(t)}$ is a carrier map describing, for every input configuration $\sigma \in \mathcal{I}$, the possible configurations of the system that can be reached at time $t$ in the protocol complex starting from $\sigma$. The fundamental theorem of topology applied to distributed computing [29, 30] states that a task is solvable is a certain time $t$ if and only if there exists a chromatic *simplicial* map[3] $\delta : \mathcal{P}^{(t)} \to \mathcal{O}$ that agrees with the specification $\Delta$ of the task, that is, such that, for every simplex $\sigma \in \mathcal{I}$,

$$\delta \circ \Xi(\sigma) \subseteq \Delta(\sigma). \tag{1}$$

Figure 2 displays the commutative diagram corresponding to Eq. (1).

**Overview of the workpackages.** WP1 is essentially focusing on the study of the topology of the protocol complex $\mathcal{P}$, and of the nature of the topological deformation $\Xi$, in the context of distributed *network* computing, where the network can be *static* or *dynamic*. WP2 is focusing on the study of the conditions that guarantee the existence of the simplicial map $\delta : \mathcal{P} \to \mathcal{O}$, or which prevent it from existing. Finally, WP3 will be dedicated to studying under which condition the diagram commutes, i.e., under which conditions on time or resources Eq. (1) is satisfied.

[3] A map $f : \mathcal{A} \to \mathcal{B}$ is simplicial if, for every simplex $\sigma \in \mathcal{A}$, $f(\sigma)$ is a simplex of $\mathcal{B}$. The map $f$ is chromatic if, in addition, for every vertex $(p, v)$ of $\mathcal{A}$, $f(p, v) = (p, v')$ for some value $v'$.

WP1 and WP2 are expected to achieve the first objective of DUCAT, that is, to provide *qualitative* understanding of distributed network computing, in the form of the properties of protocol complexes and computability characterizations, complementing our current understanding of shared-memory computing.

WP3 is expected to achieve the second objective of DUCAT, i.e., a *quantitative* analysis of the power and limitation of distributed network computing.

Each of the three workpackages of DUCAT are decomposed into a set of specific tasks, described hereafter, together with the corresponding deliverables.

### Work Package 1: Topology

WP1 focuses on the topology of the protocol complex $\mathcal{P}$ reflecting all system states reachable via a fixed number of rounds of computation in a network. Preliminary results obtained by partners of DUCAT [10, 23] show that the nature of the deformation of the input complex in static synchronous networks can be entirely different from what has been observed in asynchronous distributed shared-memory models (cf. Section b2). Interestingly, dynamic network models appear to bear topological similarities with shared memory models, as demonstrated by other partners of DUCAT [28] (cf. Section b3). WP1 aims at understanding the impact of the network structure, capacity and dynamics on the topology of the protocol complex $\mathcal{P}$. It is decomposed into four tasks.

#### *Task 1.1: Impact of the network structure*

Preliminary investigations by the members of DUCAT [10] identified the impact of the network structure on the topology of the protocol complex, at least for a simplified variant of the LOCAL model, called KNOW ALL. In this latter model, every process knows the $n$-node network $G = (V, E)$ in which it is performing, and, for every $u \in V$, it knows the ID in $\{1, \ldots, n\}$ of the process occupying $u$. The topological deformation incurred by the protocol complex during the course of computation in the KNOW ALL model corresponds to scissor cuts between the facets, as illustrated on Figure 1. The positions of these cuts heavily depend on the structure of $G$, which enables to derive necessary and sufficient conditions for solving tasks such as consensus and $k$-set agreement, in a given number of rounds, as a function of the structure of the network [10].

The objective of this task is to extend these preliminary results from the KNOW ALL model to the standard LOCAL model [41]. The LOCAL model differs significantly from the KNOW ALL model, even if the two models bear similarities. In the LOCAL model, the only information a priori given to a process is its ID, and possibly some input. Moreover, the IDs assigned to the $n$ processes in an $n$-node networks are usually taken from a range of IDs larger than $n$, typically polynomial in $n$. As a consequence, the number of facets of the input complex under the LOCAL model is considerably larger than the number of facets under the KNOW ALL model. Nevertheless, it remains that the facets are preserved during the course of computation, and, again, scissor cuts separate these facets while time proceeds.

**Deliverable:** Formalization of the topological deformations of the input complex caused by the scissor cuts as a function of the number of rounds, and of the structure of the network. Necessary and/or sufficient conditions on the network structure, and on the number of rounds guaranteeing certain core properties of the protocol complex, like path-connectivity and higher dimensional notions of connectivity. Characterization of the homotopy and homology groups of the protocol complexes, as function of structural properties of the underlying network.

#### *Task 1.2: Impact of Dynamism*

Preliminary investigations by the members of DUCAT [28] have considered the behavior of some families of dynamic networks using standard topological methods adapted to the context of these networks. The approach was proceeding in two steps. First, identifying the network components that makes the global computational behavior of dynamic networks similar to the one observed in shared memory models with the same number of processes as the number of components. Second, applying tuned topological methods that piggybacks on the standard protocol complex methods. One important outcome from the preliminary studies is that piggybacking on methods borrowed from shared-memory framework may not be sufficient, as it does not entirely capture the behavior of dynamic networks.

The objective of this task is to study one classical model of dynamic networks, which assumes a finite set $S$ of (static) networks on the same number $n$ of nodes, and an adversary who is able to pick an arbitrary network from $S$ at each round. The communications occurring at a given round are those corresponding to the network chosen by the adversary. The extreme case where $S$ contains only one network is the one studied in Task 1.1. The other extreme case in which $S$ contains all $n$-node networks correspond to what has been studied in the shared-memory model. In the general case, where $S$ is non-trivial, the protocol complex appear to be subject to both forms of deformations (subdivisions and scissor cuts). The topological approach applied to dynamic networks might thus well be a way to bridge shared-memory and (static) network computing.

**Deliverable:** Formalization of the topological deformations of the input complex caused by dynamic networks as a function of the structure of the communication patterns occurring through time. As for Task 1.1, DUCAT seeks necessary and/or sufficient conditions on the communication patterns guaranteeing certain core properties of the protocol complex, like path-connectivity and higher dimensional notions of connectivity, and characterizations of the homotopy and homology groups of the protocol complexes, as function of these communication patterns.

### *Task 1.3: Local complexes*

Very recent investigations by the members of DUCAT [23] yielded the notion of *local* complexes, for which the facets are composed of a process and its neighbors. Figure 3 displays the local complexes corresponding to MIS and 3-coloring in the ring in which nodes are provided with a consistent notion of left and right. The MIS complex is not a manifold, while the 3-coloring complex is a manifold (it is actually a Möbius strip). Local complexes offer a significant advantage over the “global” complexes considered in Tasks 1.1, and 1.2: their size is independent of the number $n$ of processes, and depend only of the maximum degree $d$ of the networks. They are therefore perfectly suited for the analysis of *locally checkable labeling* (LCL) tasks [40], whose definition is purely local (like MIS, maximal matching, coloring, minimal dominating set, etc.).

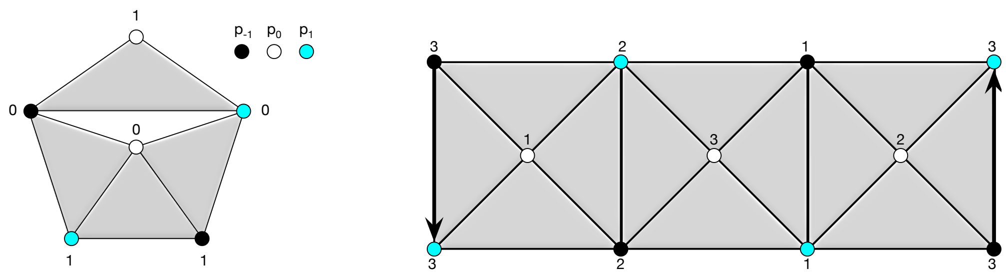


**Figure 3.** Local complex in the ring: (left) MIS — (right) 3-coloring

The topological deformations experienced by the local protocol complexes are however different from those experienced by the global protocol complexes of Task 1.1. In particular, the facets are not preserved, as processes acquire more and more information about the structure of the network, and about the inputs to the nodes in their vicinity, as computation proceeds. For instance, Figure 4 displays the protocol complex of 3-coloring after one round in the ring. In general, it is convenient to consider the framework of $d$-regular graphs with girth $\Omega(\log n)$, for which the neighborhood of every node up to distance $O(\log n)$ is a tree.

**Deliverable:** Formalization of the topological deformations of the input complex in the local topological formalization of network computing. The focus will be on the topological properties of the local protocol complex for $d$-regular networks with large girth (including connectivity, homotopy and homology).

### *Task 1.4: Ressource-bounded models*

Most of the results obtained using topology applied to distributed computing (see Section I.b) assume *full information* protocols, that is, protocols in which every process communicates its entire history whenever exchanging information with other processes. While this assumption enables to derive robust, unconditional

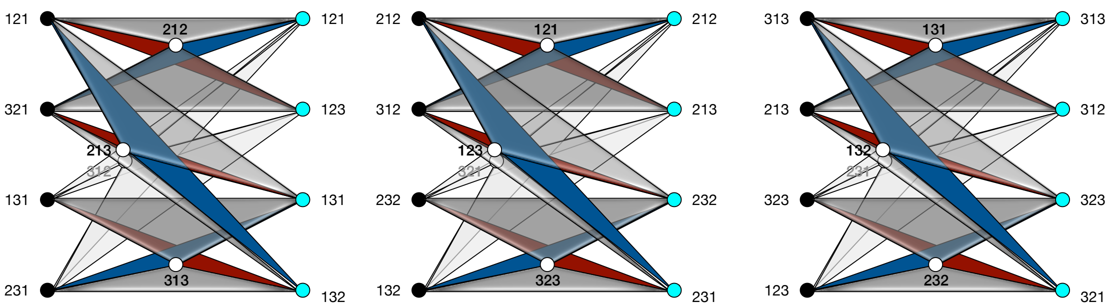


**Figure 4.** Protocol complex of 3-coloring in the ring after 1 round

lower bounds and impossibility results, it does not fit with models that specifically restrict the amount of information that can be exchanged between two processes at every round. Moreover, even in frameworks such as shared-memory, full information protocols can be inefficient due to huge amount of information transferred between the processes and the memory during the execution.

The objective of this task is to focus on topology applied to models in which ressources are bounded. As far as network computing is concerned, DUCAT is aiming for results applicable to the CONGEST model (recall that, in the CONGEST model, at most $O(\log n)$ bits can be exchanged on every edge at each round). However, DUCAT will begin by investigating shared-memory models with bounded resources: limited throughput, bounded size of registers, small number of registers, etc. In this latter framework, many questions can be raised, and in particular, is it worth considering all possible executions in the protocol complex (e.g., all choices of $O(\log n)$ bits to be written in the memory by a process at a given time)? Note that a positive answer to this question will require to modify the specification of the map $\delta : \mathcal{P} \to \mathcal{O}$ whenever the protocol complex includes simplices that correspond to executions that no algorithms actually perform. In a second step, after the constraints induced by limited resources will have been better understood in the context of shared-memory models, DUCAT will tackle the CONGEST model.

**Deliverable:** This task is among the most prospective tasks of DUCAT, and very little has been done so far in this topic, even in the context of shared-memory computing. Nevertheless, DUCAT is aiming at proposing several definitions for the protocol complexes under resource-bounded models, including CONGEST. These definitions will be confronted with the ability to be used efficiently in the context of WP2 and WP3.

## Work Package 2: Computability

WP2 intends to establish task computability characterizations in distributed network computing models. In particular, we intend to understand to which extent the methodology employed in shared-memory models is applicable to networks. For example, Sperner's Lemma (resp., Index Lemma) was instrumental in proving impossibilities of set agreement (resp., weak symmetry breaking) in read-write shared memory. Can we apply these tools in network computing? Our preliminary results [10] indicate that the answer is positive, but for a very restricted model of network computing (namely, the KNOW ALL model), and the general case is entirely open.

### *Task 2.1: Label-reduction*

As already mentioned, tasks like coloring, maximal independent set, maximal matching, etc., can be encoded by a *locally checkable labeling* (LCL). An LCL can be defined [9] as a triple of functions $(f, g, h)$ over the integers, where, for every $d \geq 1$, $f(d)$ is a finite set of labels, $g(d)$ is a finite set of pairs of labels in $f(d)$, and $h(d)$ is a finite set of $d$-tuples of labels in $f(d)$. Labeling a graph corresponds to assigning a label to each extremity of every edge. A degree-$d$ graph is then correctly labeled w.r.t. $(f, g, h)$ if (1) all labels are in $f(d)$, (2) for every edge, the pair of labels assigned to its two extremities is in $g(d)$, and, (3) for every node, the $d$-tuple of labels corresponding to the $d$ edges incident to that node is in $h(d)$. For instance, for $(d+1)$-coloring, $f(d) = \{1, \dots, d+1\}$, $g(d) = \{\{c, c'\} \in \{1, \dots, d+1\}^2 : c \neq c'\}$, and $h(d) = \{\{c, \dots, c\} : c \in \{1, \dots, d+1\}\}$.

A *label-reduction* task consists, for the nodes of a network $G$ initially labeled by an LCL $(f_1, g_1, h_1)$ to compute another labeling $(f_2, g_2, h_2)$. For instance, reduction from 3-coloring to MIS in the ring consists, for the nodes of the ring, to collectively compute a MIS starting from any initial 3-coloring of the ring. A feature of label-reduction tasks is that they can be studied in absence of IDs, as the input labeling $(f_1, g_1, h_1)$ usually provides symmetry breaking to the nodes [3]. For instance, the existence of a zero-round algorithm for reducing 3-coloring to MIS in the ring would imply the existence of a chromatic simplicial map from the complex on the right of Figure 3 to the complex on the left of Figure 3. Similarly, a 1-round algorithm would implies the existence of a chromatic simplicial map from the complex on Figure 4 to the complex on the left of Figure 3. None of these maps actually exist, and indeed, two rounds are required to reduce 3-coloring to MIS in the ring. This task is devoted to analyzing the topological properties that make label-reduction solvable quickly or not. The main framework of the task is the local complexes defined and studied in Task 1.3.

**Deliverable:** Formalization as simplicial maps of the classical algorithms for solving label-reduction tasks (e.g., iterative algorithms for vertex-coloring). Classification of the local complexes corresponding to LCLs. Impact of the homotopy/homology of these complexes on the solvability of the associated label-reduction task.

#### *Task 2.2: Topological invariants*

Since the FLP Theorem [20] about the impossibility of consensus in asynchronous systems subject to crash failures, it is known that solving consensus is about disconnecting the protocol complex. Similarly, the impossibility of $k$-set agreement by [30, 45] reveals that solving $k$-set agreement is about higher dimensional forms of connectivity — specifically, about the ability to contract $k$-dimensional spheres. A computational model, such as wait-free computing, that preserves connectivity as a topological invariant for the protocol complexes is therefore unable to solve agreement problems like consensus and $k$-set agreement.

The objective of this task is twofold. First, it is aiming at identifying topological invariants corresponding to network computing, including LOCAL and dynamic network models, with or without IDs. Namely, what are the properties of the input complex $\mathcal{I}$ that are preserved throughout computation by the protocol complexes $\mathcal{P}$? Second, this task is aiming at identifying which of these properties prevent a simplicial map $\delta : \mathcal{P} \to \mathcal{O}$ from existing, depending on the topology of the output complex $\mathcal{O}$. In both cases, in addition to classical agreement problems such as consensus, we are specifically interested in tackling these questions for natural graph problems (MIS, coloring, etc.) for which the outputs of the nodes depend on the structure of the network, and not much about inputs given to these nodes.

**Deliverable:** Characterization of the topological invariants of the (local or global) protocol complexes $\mathcal{P}$ as a function of the number of rounds, and of the structure of static networks, or of the communication patterns of dynamic networks. Identification of the tasks $(\mathcal{I}, \mathcal{O}, \Delta)$ for which these topological invariants enable or prevent a simplicial map $\delta : \mathcal{P} \to \mathcal{O}$ from existing.

#### *Task 2.3: Computing with restricted ressources*

This task is dedicated to identifying conditions under which some specific forms of mappings from the protocol complex $\mathcal{P}$ to the output complex $\mathcal{O}$ exist, in the context of models with restricted ressources, like CONGEST in the framework of static networks. Specifically, for each form of protocol complexes defined in Task 1.4, this task aims at identifying the corresponding form of mapping from $\mathcal{P}$ to $\mathcal{O}$. For instance, for a definition of protocol complexes including the simplices corresponding to all possible exchanges of $O(\log n)$-bit messages in the CONGEST model, the mapping should not necessarily provide an image for every simplex, as an algorithm chooses specifically the $O(\log n)$ bits it sends to every edge at every round. In this case, one must specify which of the simplices of the protocol complex should be mapped to the output complex. In the other extreme case in which one insists on the existence of a simplicial map $\delta : \mathcal{P} \to \mathcal{O}$, then the protocol complex must include only a subset of possible simplices, and one must identify which simplices are worth to be considered.

These two extreme cases mentioned in the previous paragraph indicate that this task may have to restrict or extend the type of algorithms under consideration. One class of algorithms of particular interest is the class of randomized algorithms. In the context of randomized algorithms, all (legal) simplices can exist in the protocol complex, each being weighted by its probability of occurring, and thus simplicial maps can be considered. Another way to achieving the objective of this task is to consider very weak models such as the BEEPING model [1], in which every node either beeps or keeps quiet at every round. In this case, one may again consider all (legal) simplices in the protocol complexes, and focus on simplicial maps.

**Deliverable:** Establishing correspondences between, on the one hand, the type of protocol complexes as defined in Task 1.4, and the type of mappings that can exist from these protocol complexes to output complexes. Identification of the models and/or classes of algorithms for which simplicial maps remain the appropriate concept for formalizing task solvability.

## Work Package 3: Complexity

WP3 is dedicated to using our understanding of the protocol complex resulting from WP1, and the characterizations obtained in WP2 for determining complexity bounds on time and space of prominent tasks in distributed network computing. In the context of static networks, we plan to focus on analyzing graph problems exhibiting some forms of locality in their specification, including the challenging coloring task. In the context of dynamic networks, WP3 will mostly focus on traditional system problems, including agreement tasks.

### *Task 3.1: Complexity in the LOCAL model*

This task will be dedicated to studying the complexity of graph problems in the LOCAL model. Part of this task will therefore consist in studying specific problems through the lens of the diagram in Figure 2, i.e., a la Herlihy-Shavit. Using the outcomes of Tasks 1.1, 1.3, 2.1, and 2.2, the objective is, given a graph problem, to identify the smallest $t$ such that the diagram of Figure 2 commutes, where $\mathcal{P}$ is the protocol complex after $t$ rounds.

Yet, Task 3.1 also proposes to tackle complexity under the LOCAL model from a different and original perspective. Indeed, preliminary investigations from the members of DUCAT [23] suggest that the powerful *round-reduction* technique for lower bounds [3, 9, 36] can be captured topologically by the diagram displayed on Figure 5. This diagram reads as follows. Let us assume the existence of a $t$-round algorithm solving the task $(\mathcal{I}, \mathcal{O}, \Delta)$. This algorithm guarantees the existence of a simplicial map $\delta : \mathcal{P}^{(t)} \to \mathcal{O}$. Now, let us assume the existence of a functor $\Phi$ that maps complexes to complexes, and the existence of a simplicial map $f : \mathcal{P}^{(t-1)} \to \Phi(\mathcal{P}^{(t)})$. Then $\widehat{\delta} = \Phi(\delta) \circ f$ is a witness that the task $(\mathcal{I}, \Phi(\mathcal{O}), \Phi \circ \Delta)$ is solvable in $t-1$ rounds.

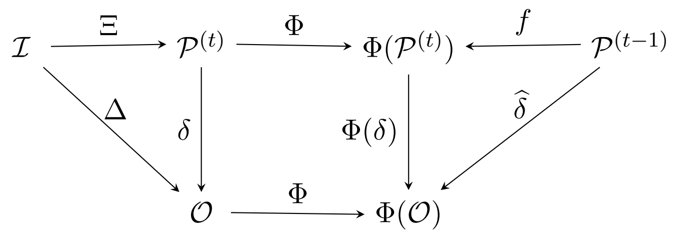


**Figure 5.** *Commutative diagrams for round reduction*

This approach is precisely the formalization of several existing lower bounds based on round-reduction (actually, most lower bounds in the LOCAL model uses the round-reduction technique, either implicitly or explicitly). This is for instance the case of Linial's $\Omega(\log^* n)$-round lower bound for 3-coloring the ring. In this case, the functor $\Phi$ maps the $k$-coloring output complex to the $2^{2^k}$-coloring output complex. In [3], the authors define an ad hoc functor for maximal matching in bi-colored graphs. The function implicitly defined in [9] is generic, i.e., applicable to all tasks $(\mathcal{I}, \mathcal{O}, \Delta)$.

**Deliverable:** Extensive exploration of the round-reduction technique from a topological perspective, according to the approach displayed on Figure 5. E.g., identification of the minimal requirements to be satisfied by the functor $\Phi$ for $f$ to exist, and identification of the tasks $(\mathcal{I}, \mathcal{O}, \Delta)$ for which $\Phi(\mathcal{O}) = \mathcal{O}$, like sinkless orientation. The ultimate objective is to be able to derive new lower bounds for $(d+1)$-coloring graphs with maximum degree $d$, better than the celebrated $\Omega(\log^* n)$ rounds lower bound by Linial [36].

#### *Task 3.2: Complexity in dynamic network models*

This task involves two main directions. First, as it is classically done in the literature (cf. Section b3), we will derive lower bounds on the round complexity for models that behave similarly to oblivious models, for which impossibility results already exist. The high significance of this task is however expected to result from exploring the second direction. Specifically, successful outcomes from Tasks 1.1, 1.2, and 2.2 are expected to help understanding how the propagation of information in dynamic networks is impacting the topology of the protocol complexes, in particular as far as the interactions between the subdivision operator and the scissor-cut operator are concerned.

The main objective of the task is to study dynamic networks that guarantee some "stable" connections, for some interval of time. This is for instance the case of dynamic networks with stable connected cores. A typical example is $T$-connected networks, where, for any $T$ consecutive rounds, there exists a spanning tree that is common to all communication graphs. So, when considering dynamic networks with stable cores, it could be doable to describe the subdivisions occurring in the simplices, by classical means as well as with the help of our understanding of scissors cuts resulting from the stable core, to describe how the global connectivity of the complexes behaves.

**Deliverable:** A description of the protocol complex as a combination of two operators: one related to the internal subdivisions, and one related to scissors cuts. The task will specifically focus on particular families of dynamic networks, e.g., those with stable cores, with the objective to derive upper and lower bounds on the number of rounds for solving classical agreement tasks in dynamic networks.

#### *Task 3.3: Complexity in ressource-bounded models*

The ultimate objective of this task is to figure out whether the topological approach of distributed computing could shed new light on the CONGEST model. To this aim, based on the outcome of Tasks 1.4 and 2.3, DUCAT will proceed gradually, starting from elementary ressource-bounded models like the BEEPING model [1], and similar very weak models [18]. For each considered model (including CONGEST), this task is aiming first at establishing theorems sharing the flavor of Eq. 1, as illustrated by the commutative diagram on Figure 2. As explained in the description of Tasks 1.4 and 2.3, such a commutative diagram may not involve simplicial maps, but mappings that ignore some of the simplices in the protocol complex (those that are cannot be reached by any "natural" algorithms).

It is worth pointing out that the BEEPING model and its variants are mostly dedicated to solving local tasks inferred from graph problems like MIS. On the other hand, the CONGEST model finds its main interest in solving non-local problems like minimum-weight spanning tree (MST), or all-pair shortest path (APSP). For this latter types of tasks, deriving lower bounds boils down to finding appropriate reductions to communication complexity. Nevertheless, some local tasks are also studied in the context of CONGEST. A typical example is pattern detection. Specifically, given a (connected) graph $H$, the decision problem consisting of checking $H$-freeness has received a lot of attention. A particular case of interest is checking triangle-freeness ($H = C_3$) whose exact complexity is still not known in the CONGEST model. Although communication complexity is a quite efficient tool for deriving lower bounds for $H$-freeness too, its application to triangle-freeness is complicated, for it would involve multi-party communication complexity. So, triangle-freeness is an appealing task for studying the potential impact of topology on the CONGEST model. Note also that, more generally, decision problems are appealing in this context, as the output-complex for these problems simply consists of an $(n-1)$-dimensional sphere.

**Deliverable:** General characterization theorems à la Herlihy-Shavit for various models, from very weak ones (e.g., BEEPING) up to the CONGEST model. Application of these theorems to local tasks such as pattern detection (including triangle-freeness), and others (e.g., checking distance-$k$ proper coloring).

### c2 Grant chart, feasibility, and risk management

Figure 6 is displaying the Gantt diagram for DUCAT. Essentially, we plan to tackle the work packages in the natural order, that is, the topological properties of protocol complexes established in WP1 will enable deriving computability characterizations in WP2, which, in turn, can bring new complexity bounds in WP3. Within a same WP, all tasks are executed in parallel, to benefit from the synergy induced by tackling

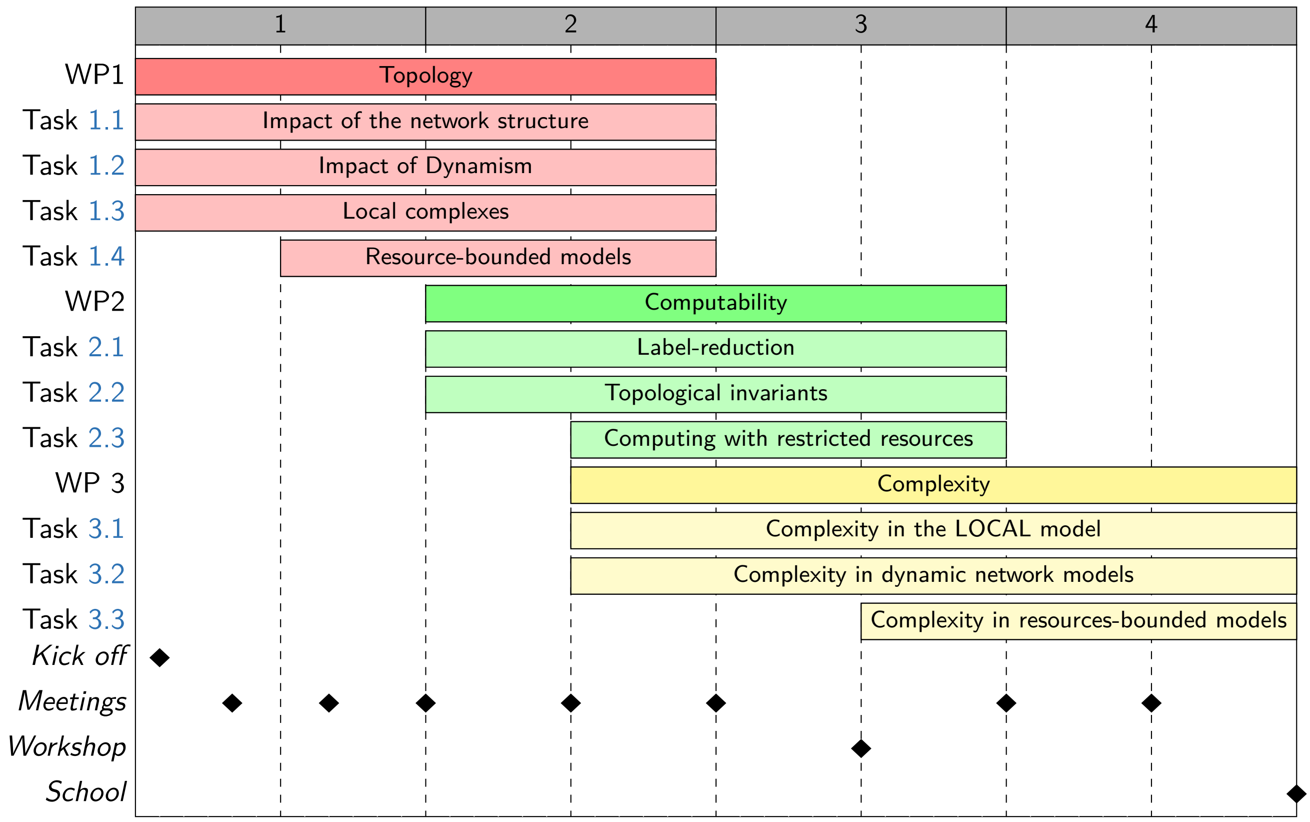


**Figure 6.** Gantt diagram of the project

problems defined under different settings but sharing the same underlying framework, and using similar technical tools. This is to the exception of all tasks related to resource-bounded models, which are delayed withing each WP, for allowing time to grasp the general perspective on each model (static and dynamic networks) under the full-information setting, before considering the challenging case of bounded resources.

**Feasibility.** In fact, the point is not so much whether WP1 and WP2 are feasible, but the point is whether their outcomes will enable WP3 to be tackled successfully. Indeed, there is little doubt that non-trivial mathematical theorems can be derived from the tasks carried out withing WP1 and WP2. However, the risk is that these theorems might be too abstract or too complicate for being applied to the more "practical" tasks of WP3. To clarify this point, let us consider the previous achievements of the community in the context of asynchronous shared-memory models.

In asynchronous shared-memory models, appropriate characterizations of task solvability are based on complex abstractions regarding the memory operations, including *iterated immediate snapshots*. The use of these complex abstractions results in expressing wait-free computing smoothly, as chromatic subdivisions of simplices. Instead, using natural and more realistic memory operations, e.g., basic atomic read and atomic write, results in cumbersome protocol complexes that are often extremely hard to manipulate. This is to say that if the characterization of tasks solvability in wait-free computing would have been expressed using read/write, then the applicability of this characterization would have been far more difficult to grasp, and therefore to apply. Instead, using the characterization of tasks solvability by using iterated immediate snapshots enable to derive non-trivial impossibility results via simple application of powerful tools such as Sperner's Lemma.

Nevertheless, we do not expect to face difficulties such as the ones described above, as far as WP1 and WP2 are concerned. Indeed, we mostly deal with *synchronous* computing that does not involve complex interleaving of the computations, especially in static networks. On the other hand, we are aware that the structure of the networks, and their potential evolution over time might incur new types of difficulties. In particular, the pace in which information propagates through the network has a huge impact, compared to shared-memory models, where communication is somehow "instantaneous". Finding relevant topological invariants is a challenging task that might require novel topological representations of network computations.

The feasibility of the project comes at this cost. So, indeed, the project is risky, but the expected gain is high.

**Risk management.** There are multiple network features that DUCAT plans to account for, in all considered network models. This includes the role of the identifiers, the importance of a priori knowledge given to the processes about the network, the various way the networks evolve over time, the various constraints considered when focusing on resource limitation, etc. One way for DUCAT to mitigate the risks is to incorporate these features progressively. The use of simple (even potentially artificial) models, such as the KNOW ALL model or the BEEPING model, before considering models like LOCAL, or CONGEST, is another way to progressively approaching the objective of the project, and thus mitigating the risks. As another example of the progressive approach of the problems, WP1 considers the study of static and dynamic networks separately, as each of these two models seem to involve different topological transformations: scissor cuts for the former, and chromatic subdivisions for the latter. These two topological transformation shall be considered simultaneously only later, in WP2 and WP3. Therefore, DUCAT addresses risk management according to two main rules:

1. a progressive approach consisting of starting with simple models, and of increasing difficulty slowly by incorporating more realistic features in the models incrementally, and
2. a synergical approach consisting of working in parallel on static and dynamic networks, with frequent mutual interactions between the two topics.

These rules should enable finding escape paths rapidly when the project faces obstacles too big for being overcame easily. As a final remark, DUCAT plans to organize frequent meetings, especially at the beginning of the project (see Figure 6). Moreover, these meetings are planed to be long enough for providing enough time for extensive discussions and exchanges, therefore enforcing the synergy among the participants. The pace of the meetings will slow down after the first year, once DUCAT will have entered its steady state.

# II Organisation and implementation of the project

## a Scientific coordinator and its consortium / its team

**Scientific coordinator (SC):** Pierre Fraigniaud (DR CNRS, IRIF). The SC is a leading researcher in Distributed Computing. He has been the PC Chair of the main conferences in the field, including ACM PODC 2011, DISC 2005, and ACM SPAA 2001, and, more recently, IEEE IPDPS 2017 (Track Algorithms), and ICALP 2014 (Track C). He is member of the Editorial Board of Distributed Computing, and of Theory of Computing Systems. He was Management Committee Chair of the European COST Action 295 DYNAMO on Algorithmic Aspects of Dynamic Networks (2005-2009), and SC of the ANR project ALADDIN (2007-2011). He is currently Chair of the national working group Complexity and Algorithms (CoA) of GdR IM.

**Consortium:** The consortium is composed of two teams, including the following faculty members:

- **IRIF — Team North:** Carole Delporte (PR, IRIF, Paris), Hugues Fauconnier (PR, IRIF, Paris), Pierre Fraigniaud (DR CNRS, IRIF, Paris), Petr Kuznetsov (PR, LTCI, Telecom Paristech), Christine Tasson (MCF, IRIF, Paris);
- **LIS — Team South:** Jérémie Chalopin (CR CNRS, LIS, Marseille), Emmanuel Godard (PR, LIS, Marseille), Damien Imbs (MCF, LIS, Marseille), Corentin Travers (MCF, LaBRI, Bordeaux).

The consortium has been set for bringing together French experts of complementary fields of distributed computing such as: network computing (J. Chalopin, P. Fraigniaud), distributed computing in dynamic networks (E. Godard), mobile computing in networks (J. Chalopin), topological methods in distributed computing (C. Delporte, H. Fauconnier, D. Imbs, P. Kuznetsov, C. Travers), graph theoretical aspects of distributed computing (J. Chalopin, P. Fraigniaud), Byzantine failures (C. Tasson) and fault-tolerant distributed computing in general (C. Delporte, H. Fauconnier, D. Imbs, P. Kuznetsov, C. Travers).

The consortium is not including mathematician experts of combinatorial topology because textbook methods are arguably sufficient for the objectives of DUCAT — two of the authors of [29] are frequent collaborators of several partners of the project. Also, DUCAT is not including experts of directed topology, as the project is not planning to study the execution of concurrent programs, but the feasibility and complexity of distributed tasks.

C. Delporte, H. Fauconnier, P. Fraigniaud (North), and C. Travers, and E. Godard (South) have already recently collaborated within the framework of the ANR project DESCARTES (2016-2020) aiming at organizing distributed computing in modules, in a way similar to the way networks are organized in layers. J. Chalopin was SC of the ANR JCJC project MACARON (2013-2017) on mobile computing in networks. P. Kuznetsov was co-SC of the ANR/DFG project DISCMAT (2014-2018) aiming at pushing further our knowledge on topological aspect of shared-memory wait-free computing. DUCAT will take benefits of the outcomes of these past projects.

Beside cooperations within the framework of institutional national or international projects, the partners of DUCAT have a long experience of informal but fruitful collaborations thanks to their frequent and regular participations to conferences, whether it be international (PODC, DISC, etc.), or national (AlgoTel), and working groups (e.g., the WG Complexity of Algorithms of GdR Informatique Mathématique). They share the same scientific language, and the same vision of distributed computing. They are complementary, as experts of different aspects of distributed computing. It follows that DUCAT is not expecting any "warming up" period at the beginning of the project, as all participants are already familiar with most (if not all) aspects of the project, even if each one is expert of only a subset of the topics included in DUCAT.

It is worth noticing that IRIF benefits from the presence of the Foundation Sciences Mathématiques de Paris (FSMP), providing financial supports for medium to long-term visits of foreign researchers. In this context, DUCAT will take benefit of the presence at IRIF of David Peleg[4] from Weizmann Institute of Science, for 4 months in Spring 2021, supported by the FSMP Chairs program. Similarly, IRIF is planning to invite Sergio Rajsbaum[5] from UNAM, for at least a couple of months in 2021 or 2022, supported by a FSMP Distinguished Professor Fellowship.

**Implication of the scientific coordinator and partner's scientific leader in on-going project(s):**

| Researcher | Person-month | Funding agency | Project's title | Scientific coordinator | Start-end |
|---|---|---|---|---|---|
| Pierre FRAIGNIAUD | 50% 24pm | ANR PRC | DESCARTES<br>Abstraction Layers for for Distributed Computing | Cyril GAVOILLE | 2016-20 |
| Pierre FRAIGNIAUD | 33% 20pm | ANR PRC | FREDDA<br>Formal methods for the design of distributed algorithms | Arnaud SANGNIER | 2017-22 |
| Pierre FRAIGNIAUD | 25% 12pm | ANR PRC | QuDATA<br>Quantum algorithms for massive data | Frédéric MAGNIEZ | 2018-22 |
| Jérémie CHALOPIN | 30% 15 pm | ANR PRC | DISTANCIA<br>Metric Graph Theory | Victor CHEPOI | 2017-21 |

## b Implemented and requested resources to reach the objectives

Mission expenses and travel costs result roughly a third of the total funds requested to ANR for DUCAT, but only roughly 6% of the total budget. This part of the budget is indeed absolutely necessary for the success of the project, which requires a combination of intensive exchanges among the participants, and regular confrontation of the results obtained by DUCAT with those concurrently produced by other teams throughout the world. The former will be achieved via frequent short-term visits of the participants between the two sites of DUCAT, and regular meetings gathering all participants of DUCAT. The latter will be achieved via the frequent participation to international workshops and conferences.

### b1 Partner 1: IRIF (North Team)

**Staff expenses:** One PhD Student will be assigned to the North Team, working 100% on the project (on Tasks 1.1, 1.3, 2.2, and 3.1), amounting to 102 960€. An additional amount of 4 728€ will be

[4] David Peleg is one of the worldwide leading experts in distributed network computing, author of [41].
[5] Sergio Rajsbaum is expert in combinatorial topology applied to distributed computing, co-author of [29].

dedicated to cover 8pm of Master internships.

**Instruments and material costs:** 1 workstation + 1 laptops, or other consumables for IRIF = 5K€.

**Building and ground costs:** IRIF will not be subject to building or ground costs related to the execution of DUCAT.

**Outsourcing / subcontracting:** IRIF will not be subject to outsourcing or subcontracting costs related to the execution of DUCAT.

**Mission and Travel costs:** The total person-month of permanent and temporary staff from the North Team affected to the project amounts to 102, including 36pm corresponding to the PhD student financed by the project. DUCAT allocates roughly 5 500€ per 12pm (i.e., per year at 100%) to all the participants, including PhDs and post-docs, for mission expenses and travel costs (project meetings, presentations at international workshops and conferences, Dagstuhl Seminar, etc.). An additional amount of 6 000€ is allocated to each team for the whole duration of the project, dedicated to short-term invitations of foreign experts. This results to 55 000€ for the North Team.

**General and administrative costs & other operating expenses:** For the administrative management and structure costs, IRIF applies a package of 8% of the eligible expenses.

### b2 Partner 2: LIS (South Team)

**Staff expenses:** One Post-doctoral student will be assigned to the South Team, working 100% on the project (on Task 3.2), amounting to 54 684€. An additional amount of 4 728€ will be dedicated to cover 8 pm of Master internships.

**Instruments and material costs:** 3 laptops, or other consumables for LIS = 5 K€.

**Building and ground costs:** LIS will not be subject to building or ground costs related to the execution of DUCAT.

**Outsourcing / subcontracting:** LIS will not be subject to outsourcing or subcontracting costs related to the execution of DUCAT.

**Mission and Travel costs:** The total person-month of permanent and temporary staff from the South Team affected to the project amounts to 84, including 12pm corresponding to the post-doc financed by the project. DUCAT allocates roughly 5 500€ per 12pm (i.e., per year at 100%) to all the participants, including PhDs and post-docs, for mission expenses and travel costs (project meetings, presentations at international workshops and conferences, Dagstuhl Seminar, etc.). An additional amount of 6 000€ is allocated to each team for the whole duration of the project, dedicated to short-term invitations of foreign experts. This results to 45 000€ for the South Team.

**General and administrative costs & other operating expenses:** For the administrative management and structure costs, LIS applies a package of 8% of the eligible expenses.

### b3 Requested means by item of expenditure and by partner

| | **IRIF** (North team) | **LIS** (South team) |
|---|---|---|
| Staff expenses | 107 688 | 59 412 |
| Instruments and consumables | 5 000 | 5 000 |
| Building and ground costs | 0 | 0 |
| Outsourcing / subcontracting | 0 | 0 |
| Travel costs | 55 000 | 45 000 |
| Administrative management & structure costs | 13 415 | 8 753 |
| **Sub-total** | 181 103 | 118 165 |
| **Requested funding: 299 268 €** | | |

## III Impact and benefits of the project

**General impact and benefits.** The project DUCAT is part of a fundamental approach of computer science, and it is not expected to have immediate social impacts. However, DUCAT is positioned at the core of modern applications of computer science, for which distributed computing already plays a major role, a role that is expected to be rapidly growing in the near future. Indeed, essentially all the chain of computation already contains some distributed computing aspects, whether it be at the micro-level of multicore architectures, or at the macro-level of cloud computing. The advent of the next generation of wireless networks is expected to make the Internet of things (IoT) a reality throughout most aspects of our life, and will therefore increase again the practical impact of distributed computing. It can thus be foreseen that the vast majority of systems, applications, and usages will be distributed somehow in the near future. On the other hand, our knowledge and understanding of the nature of distributed computing, although deep and elaborated, is very much scattered. The main reason for that is the absence of an equivalent of the Turing machine for distributed computing, that is, the lack of an abstract model capturing the essence of distributed computing, and applicable at all scales, from multicores to the cloud, and from computing platforms to IoT. DUCAT is ambitiously aiming at helping remedying to this lack of a universal model.

The ambition of DUCAT is grounded on the claim that the most fundamental and important aspects of distributed computing, that is, asynchrony, failures, indistinguishability, etc., which make distributed computing the science of *computing with uncertainty* can be consistently captured by topological objects modeling both the problems, and the algorithms for solving these problems, in the same framework. This claim is definitely a fact regarding asynchronous fault-tolerant distributed computing, whether it be shared-memory or message-passing, or even including various forms of faults, from crashes to adversarial behaviors, as demonstrated by 20 years of outstanding results obtained since the seminal aforementioned work of Herlihy, Saks, Shavit, and Zaharoglou. DUCAT is aiming at testing whether the claim holds for networking aspects, that is, when the structure of a (static or dynamic) network impacts distributed computing. If, as expected, DUCAT succeeds to translate networking notions such as locality, and congestion into topological concepts, then the project will contribute to put distributed computing under a single conceptual umbrella allowing us to reason about distributed computing in a consistent way despite the huge disparity between the various forms of distributed computations. Beyond the impact on algorithm design and analysis of unifying distributed computing under the umbrella of algebraic topology, such a unification may benefit to program verification, proof and certification. Indeed, the framework of algebraic topology is perfectly suited for the use of tools from formal methods, enabling to reason about well defined combinatorial objects rather than about partially unspecified models, and potentially ambiguous algorithms.

**Dissemination and exploitation.** Obviously, it is expected that the results of DUCAT will be presented at the most visible colloquiums gathering the members of the distributed computing community in particular, and of the TCS community in general, and will be eventually published in archival journals. However, DUCAT is also planing to promote the topological framework for distributed computing via publications in mainstream scientific journals such as the magazines La Recherche or Pour la Science in France.

DUCAT is also planing to organize a 1-week *International School on the Topological Aspects of Distributed Computing* at the end of its term, mostly dedicated to PhD's and post-docs for promoting the use of topology within distributed computing. Indeed, the members of DUCAT believe that the knowledge of the computer scientists about the topological aspects of distributed computing must become into par with the knowledge about other central concepts in computer science, such as sequential complexity, and sequential algorithm design and analysis. This school might be organized under the scientific umbrella of the Working Group *Complexité et Algorithmes* (CoA) of the CNRS GdR *Informatique Mathématique* (IM). It is planed to be organized at a well established center for international schools (e.g., Bertinoro international Center for informatics in Italy). DUCAT is not asking financial support for such a school to the ANR, as the organizers of events at, say, Bertinoro, do not have costs related to the organization.

Last but not least, DUCAT is planing to organize an *International Workshop on the Topological Aspects of Distributed Computing* during its second term, dedicated to PhD, post-docs, and academics, with the objective of cross-fertilization between the different fields of distributed computing impacted by combinatorial topology. This workshop is expected to take the form of a Dagstuhl Seminar. DUCAT is not asking financial support for such a workshop to the ANR, as the organizers of, say, Dagstuhl Seminars, do not have costs related to the organization.

# IV References related to the project

As required by ANR, authors members of the project are marked in bold face and underlined.

## References

[1] Y. Afek, N. Alon, Z. Bar-Joseph, A. Cornejo, B. Haeupler, and F. Kuhn. "Beeping a maximal independent set". In: *Distributed Computing* 26.4 (2013), pp. 195–208.

[2] Y. Afek and E. Gafni. "Asynchrony from synchrony". In: *14th International Conference on Distributed Computing and Networking (ICDCN)*. 2013, pp. 225–239.

[3] A. Balliu, S. Brandt, J. Hirvonen, D. Olivetti, M. Rabie, and J. Suomela. "Lower bounds for maximal matchings and maximal independent sets". In: *60th IEEE Symposium on Foundations of Computer Science (FOCS)*. 2020.

[4] L. Barenboim and M. Elkin. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool, 2013.

[5] L. Barenboim, M. Elkin, and U. Goldenberg. "Locally-Iterative Distributed $(\Delta+1)$-Coloring below Szegedy-Vishwanathan Barrier, and Applications to Self-Stabilization and to Restricted-Bandwidth Models". In: *37th ACM Symposium on Principles of Distributed Computing (PODC)*. 2018, pp. 437–446.

[6] E. Borowsky and E. Gafni. "Generalized FLP impossibility result for $t$-resilient asynchronous computations". In: *25th ACM Symposium on Theory of Computing (STOC)*. 1993, pp. 91–100.

[7] E. Borowsky and E. Gafni. "A simple algorithmically reasoned characterization of wait-free computations". In: *16th ACM Symposium on Principles of Distributed Computing (PODC)*. 1997, pp. 189–198.

[8] E. Borowsky and E. Gafni. "Immediate atomic snapshots and fast renaming". In: *12th ACM Symposium on Principles of Distributed Computing (PODC)*. 1993, pp. 41–51.

[9] S. Brandt. "An Automatic Speedup Theorem for Distributed Problems". In: *38th ACM Symposium on Principles of Distributed Computing (PODC)*. 2019, pp. 379–388.

[10] A. Castañeda, **P. Fraigniaud**, A. Paz, S. Rajsbaum, M. Roy, and **C. Travers**. "A Topological Perspective on Distributed Network Algorithms". In: *26th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*. LNCS 11639. Springer, 2019, pp. 3–18.

[11] K. Censor-Hillel, P. Kaski, J. H. Korhonen, C. Lenzen, A. Paz, and J. Suomela. "Algebraic methods in the congested clique". In: *Distributed Computing* 32.6 (2019), pp. 461–478.

[12] Y.-J. Chang, T. Kopelowitz, and S. Pettie. "An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model". In: *SIAM J. Comput.* 48.1 (2019), pp. 122–143.

[13] B. Charron-Bost and A. Schiper. "The heard-of model: computing in distributed systems with benign faults". In: *Distributed Computing* 22.1 (2009), pp. 49–71.

[14] E. Coulouma, **E. Godard**, and J. G. Peters. "A characterization of oblivious message adversaries for which consensus is solvable". In: *Theor. Comput. Sci.* 584 (2015), pp. 80–90.

[15] **Delporte-Gallet C.**, **H. Fauconnier**, R. Guerraoui, and A. Tielmann. "The disagreement power of an adversary". In: *Distributed Computing* 24.3-4 (2011), pp. 137–147.

[16] **Delporte-Gallet C.**, **H. Fauconnier**, S. Rajsbaum, and N. Yanagisawa. "A Characterization of $t$-Resilient Colorless Task Anonymous Solvability". In: *25th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*. LNCS 11085. Springer, 2018, pp. 178–192.

[17] A. Drucker, F. Kuhn, and R. Oshman. "On the power of the congested clique model". In: *33rd ACM Symposium on Principles of Distributed Computing (PODC)*. 2014, pp. 367–376.

[18] Y. Emek and R. Wattenhofer. "Stone age distributed computing". In: *32nd ACM Symposium on Principles of Distributed Computing (PODC)*. 2013, pp. 137–146.

[19] F. E. Fich and E. Ruppert. "Hundreds of impossibility results for distributed computing". In: *Distributed Computing* 16.2-3 (2003), pp. 121–163.

[20] M. J. Fischer, N. A. Lynch, and M. Paterson. "Impossibility of distributed consensus with one faulty process". In: *J. ACM* 32.2 (1985), pp. 374–382.

[21] **Fraigniaud P.**, E. Gafni, S. Rajsbaum, and M. Roy. "Automatically adjusting concurrency to the level of synchrony". In: *28th International Symposium on Distributed Computing (DISC)*. 2014, pp. 1–15.

[22] **Fraigniaud P.**, M. Heinrich, and A. Kosowski. "Local Conflict Coloring". In: *57th IEEE Symposium on Foundations of Computer Science (FOCS)*. 2016, pp. 625–634.

[23] **Fraigniaud P.** and A. Paz. "The Topology of Local Computing in Networks". In: *47th International Colloquium on Automata, Languages and Programming (ICALP)*. 2020.

[24] E. Gafni. "The extended BG-simulation and the characterization of $t$-resiliency". In: *41st ACM Symposium on Theory of Computing (STOC)*. Vol. 85–92. 2009.

[25] J. A. Garay, S. Kutten, and D. Peleg. "A Sublinear Time Distributed Algorithm for Minimum-Weight Spanning Trees". In: *SIAM J. Comput.* 27.1 (1998), pp. 302–316.

[26] M. Ghaffari, D. G. Harris, and F. Kuhn. "On Derandomizing Local Distributed Algorithms". In: *59th IEEE Symposium on Foundations of Computer Science (FOCS)*. 2018, pp. 662–673.

[27] M. Ghaffari, F. Kuhn, and Y. Maus. "On the complexity of local distributed graph problems". In: *49th ACM Symposium on Theory of Computing (STOC)*. 2017, pp. 784–797.

[28] **Godard E.** and E. Perdereau. "$k$-Set Agreement in Communication Networks with Omission Faults". In: *20th International Conference on Principles of Distributed Systems (OPODIS)*. LIPIcs 70. Schloss Dagstuhl, 2016, 8:1–8:17.

[29] M. Herlihy, D. Kozlov, and S. Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 2013.

[30] M. Herlihy and N. Shavit. "The topological structure of asynchronous computability". In: *J. ACM* 46.6 (1999), pp. 858–923.

[31] M. Herlihy and N. Shavit. "The asynchronous computability theorem for $t$-resilient tasks". In: *25th ACM Symposium on Theory of Computing (STOC)*. 1993, pp. 111–120.

[32] F. Kuhn, T. Moscibroda, and R. Wattenhofer. "Local Computation: Lower and Upper Bounds". In: *J. ACM* 63.2 (2016), 17:1–17:44.

[33] S. Kutten and D. Peleg. "Fast Distributed Construction of Small k-Dominating Sets and Applications". In: *J. Algorithms* 28.1 (1998), pp. 40–66.

[34] **Kuznetsov P.**, T. Rieutord, and Yuan He. "An asynchronous computability theorem for fair adversaries". In: *ACM Symposium on Principles of Distributed Computing (PODC)*. 2018, pp. 1–10.

[35] **Kuznetsov P.** and N. Yanagisawa. "Task Computability in Unreliable Anonymous Networks". In: *22nd Int. Conference on Principles of Distributed Systems (OPODIS)*. 2018, 23:1–23–13.

[36] N. Linial. "Locality in distributed graph algorithms". In: *SIAM J. Comput.* 21.1 (1992), pp. 193–201.

[37] Z. Lotker, B. Patt-Shamir, E. Pavlov, and D. Peleg. "Minimum-weight spanning tree construction in $O(\log\log n)$ communication rounds". In: *SIAM Journal on Computing* 35.1 (2005), pp. 120–131.

[38] N. A. Lynch. "A Hundred Impossibility Proofs for Distributed Computing". In: *8th ACM Symposium on Principles of Distributed Computing (PODC)*. 1989, pp. 1–28.

[39] H. Mendes, **C. Tasson**, and M. Herlihy. "Distributed computability in Byzantine asynchronous systems". In: *46th ACM Symposium on Theory of Computing (STOC)*. 2014, pp. 704–713.

[40] Moni Naor and Larry J. Stockmeyer. "What Can be Computed Locally?" In: *SIAM J. Comput.* 24.6 (1995), pp. 1259–1277.

[41] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[42] D. Peleg and V. Rubinovich. "A Near-Tight Lower Bound on the Time Complexity of Distributed Minimum-Weight Spanning Tree Construction". In: *SIAM J. Comput.* 30.5 (2000), pp. 1427–1442.

[43] M. Raynal and J. Stainer. "Synchrony weakened by message adversaries vs asynchrony restricted by failure detectors". In: *ACM Symposium on Principles of Distributed Computing (PODC)*. 2013, pp. 166–175.

[44] V. Rozhoň and M. Ghaffari. "Polylogarithmic-Time Deterministic Network Decomposition and Distributed Derandomization". In: *52nd ACM Symposium on Theory of Computing (STOC)*. 2020.

[45] M. E. Saks and F. Zaharoglou: "Wait-Free $k$-Set Agreement is Impossible: The Topology of Public Knowledge". In: *SIAM J. Comput.* 29.5 (2000), pp. 1449–1483.

[46] M. Saks and F. Zaharoglou. "Wait-free $k$-set agreement is impossible: The topology of public knowledge". In: *25th ACM Symposium on Theory of Computing (STOC)*. 1993, pp. 101–110.

[47] N. Santoro and P. Widmayer. "Time is not a healer". In: *6th Symposium on Theoretical Aspects of Computer Science (STACS)*. 1989, pp. 304–313.

[48] V. Saraph, M. Herlihy, and E. Gafni. "Asynchronous computability theorems for $t$-resilient systems". In: *30th International Symposium on Distributed Computing (DISC)*. LNCS 9888. Springer, 2016, pp. 428–441.

[49] A. Das Sarma, S. Holzer, A. Korman L. Kor, D. Nanongkai, G. Pandurangan, D. Peleg, and R. Wattenhofer. "Distributed Verification and Hardness of Distributed Approximation". In: *SIAM J. Comput.* 41.5 (2012), pp. 1235–1265.

[50] E. Sperner. *Beweis für die invarianz der dimensionszahl und des gebietes*. Abh. Math. Semin. Hamburg. Univ. Bd. 6:265-272. 1928.

[51] J. Suomela. "Survey of local algorithms". In: *ACM Comput. Surv.* 45.2 (2013), 24:1–24:40.