IN PARTNERSHIP WITH:
CNRS

Université Denis Diderot (Paris 7)

Activity Report 2016

Project-Team GANG

Networks, Graphs and Algorithms

IN COLLABORATION WITH: Institut de Recherche en Informatique Fondamentale

RESEARCH CENTER
Paris

THEME
Networks and Telecommunications

Table of contents

Project-Team GANG

Creation of the Project-Team: 2007 July 01

Keywords:

Computer Science and Digital Science:

1.2. - Networks
1.2.3. - Routing
1.2.9. - Social Networks
1.3. - Distributed Systems
3.5. - Social networks
3.5.1. - Analysis of large graphs
6.1.3. - Discrete Modeling (multi-agent, people centered)
7.1. - Parallel and distributed algorithms
7.2. - Discrete mathematics, combinatorics
7.3. - Optimization
7.9. - Graph theory
7.10. - Network science
7.13. - Quantum algorithms

Other Research Topics and Application Domains:

1.1.8. - Evolutionnary biology
1.1.11. - Systems biology
6.3.2. - Network protocols
6.3.4. - Social Networks
7.2. - Smart travel

1. Members

Research Scientists

Laurent Viennot [Team leader, Inria, Senior Researcher, HDR]
Pierre Fraigniaud [CNRS, Senior Researcher, HDR]
Amos Korman [CNRS, Researcher, HDR]
Adrian Kosowski [Inria, Researcher, HDR]

Faculty Members

Yacine Boufkhad [Univ. Paris VII, Associate Professor]
Pierre Charbit [Univ. Paris VII, Associate Professor]
Fabien de Montgolfier [Univ. Paris VII, Associate Professor]
Hugues Fauconnier [Univ. Paris VII, Professor, HDR]
Carole Gallet Delporte [Univ. Paris VII, Professor, HDR]
Michel Habib [Univ. Paris VII, Professor, HDR]

PhD Students

Leonardo Linguaglossa [Inria, until Jun 2016, granted by Alcatel-Lucent Bell Labs]
Simon Collet [ERC BDA]
Lucas Boczkowski [ERC BDA]
Laurent Feuilloley [ENS Cachan]

Mengchuan Zou [Inria, from Oct 2016]

Administrative Assistant

Christine Anocq [Inria]

2. Overall Objectives

2.1. Overall Objectives

GANG focuses on algorithm design for large scale networks using structural properties of these networks. Application domains include the development of optimized protocols for large dynamic networks such as mobile networks or overlay networks over Internet. This includes for instance peer-to-peer applications, or the navigability of social networks. GANG tools come from recent advances in the field of graph algorithms, both in centralized and distributed settings. In particular, this includes graph decomposition and geometric properties (such as low doubling dimension, low dimension embedding, etc.). Today, the management of large networks, Internet being the reference, is best effort. However, the demand for mobility (ad hoc networks, wireless connectivity, etc.) and for dynamicity (node churn, fault tolerance, etc.) is increasing. In this distributed setting, it becomes necessary to design a new generation of algorithms and protocols to face the challenge of large scale mobility and dynamicity. In the mean time, recent and sophisticated theoretical results have emerged, offering interesting new tracks for managing large networks. These results concern centralized and decentralized algorithms for solving key problems in communication networks, including routing, but also information retrieval, localization, or load balancing. They are mainly based on structural properties observed in most of real networks: approximate topology with low dimension metric spaces, low treewidth, low doubling dimension, graph minor freeness, etc. In addition, graph decomposition techniques have recently progressed. The scientific community has now tools for optimizing network management. First striking results include designing overlay networks for peer-to-peer systems and understanding the navigability of large social networks.

3. Research Program

3.1. Graph and Combinatorial Algorithms

We focus on two approaches for designing algorithms for large graphs: decomposing the graph and relying on simple graph traversals.

3.1.1. Graph Decompositions

We study new decompositions schemes such as 2-join, skew partitions and others partition problems. These graph decompositions appeared in the structural graph theory and are the basis of some well-known theorems such as the Perfect Graph Theorem. For these decompositions there is a lack of efficient algorithms. We aim at designing algorithms working in $O(nm)$ since we think that this could be a lower bound for these decompositions.

3.1.2. Graph Search

We more deeply study multi-sweep graph searches. In this domain a graph search only yields a total ordering of the vertices which can be used by the subsequent graph searches. This technique can be used on huge graphs and do not need extra memory. We already have obtained preliminary results in this direction and many well-known graph algorithms can be put in this framework. The idea behind this approach is that each sweep discovers some structure of the graph. At the end of the process either we have found the underlying structure (for example an interval representation for an interval graph) or an approximation of it (for example in hard discrete optimization problems). We envision applications to exact computations of centers in huge graphs, to underlying combinatorial optimization problems, but also to networks arising in biology.

3.1.3. Graph Exploration

In the course of graph exploration, a mobile agent is expected to regularly visit all the nodes of an unknown network, trying to discover all its nodes as quickly as possible. Our research focuses on the design and analysis of agent-based algorithms for exploration-type problems, which operate efficiently in a dynamic network environment, and satisfy imposed constraints on local computational resources, performance, and resilience. Our recent contributions in this area concern the design of fast deterministic algorithms for teams of agents operating in parallel in a graph, with limited or no persistent state information available at nodes. We plan further studies to better understand the impact of memory constraints and of the availability of true randomness on efficiency of the graph exploration process.

3.2. Distributed Computing

The distributed community can be viewed as the union of two sub-communities. This is true even in our team. Even though they are not completely disjoint, they are disjoint enough not to leverage each others' results. At a high level, one is mostly interested in timing issues (clock drifts, link delays, crashes, etc.) while the other one is mostly interested in spatial issues (network structure, memory requirements, etc.). Indeed, one sub-community is mostly focusing on the combined impact of asynchronism and faults on distributed computation, while the other addresses the impact of network structural properties on distributed computation. Both communities address various forms of computational complexities, through the analysis of different concepts. This includes, e.g., failure detectors and wait-free hierarchy for the former community, and compact labeling schemes and computing with advice for the latter community. We have the ambitious project to achieve the reconciliation between the two communities by focusing on the same class of problems, the yes/no-problems, and establishing the scientific foundations for building up a consistent theory of computability and complexity for distributed computing. The main question addressed is therefore: is the absence of globally coherent computational complexity theories covering more than fragments of distributed computing, inherent to the field? One issue is obviously the types of problems located at the core of distributed computing. Tasks like consensus, leader election, and broadcasting are of very different nature. They are not *yes-no* problems, neither are they minimization problems. Coloring and Minimal Spanning Tree are optimization problems but we are often more interested in constructing an optimal solution than in verifying the correctness of a given solution. Still, it makes full sense to analyze the *yes-no* problems corresponding to checking the validity of the output of tasks. Another issue is the power of individual computation. The FLP impossibility result as well as Linial's lower bound hold independently from the individual computational power of the involved computing entities. For instance, the individual power of solving NP-hard problems in constant time would not help overcoming these limits which are inherent to the fact that computation is distributed. A third issue is the abundance of models for distributed computing frameworks, from shared memory to message passing, spanning all kinds of specific network structures (complete graphs, unit-disk graphs, etc.) and or timing constraints (from complete synchronism to full asynchronism). There are however models, typically the wait-free model and the LOCAL model, which, though they do not claim to reflect accurately real distributed computing systems, enable focusing on some core issues. Our research program is ongoing to carry many important notions of Distributed Computing into a *standard* computational complexity.

3.3. Network Algorithms and Analysis

Based on our scientific foundation on both graph algorithms and distributed algorithms, we plan to analyze the behavior of various networks such as future Internet, social networks, overlay networks resulting from distributed applications or online social networks.

3.3.1. Information Dissemination

One of the key aspects of networks resides in the dissemination of information among the nodes. We aim at analyzing various procedures of information propagation from dedicated algorithms to simple distributed schemes such as flooding. We also consider various models, where noise can alter information as it propagates or where memory of nodes is limited for example.

3.3.2. *Routing Paradigms*

We try to explore new routing paradigms such as greedy routing in social networks for example. We are also interested in content centric networking where routing is based on content name rather than content address. One of our target is multiple path routing: how to design forwarding tables providing multiple disjoint paths to a destination?

3.3.3. *Beyond Peer-to-Peer*

Based on our past experience of peer-to-peer application design, we would like to broaden the spectrum of distributed applications where new efficient algorithms and analysis can be performed. We especially target online social networks if we see them as collaborative tools for exchanging information. A basic question resides in making the right connections for gathering filtered and accurate information with sufficient coverage.

3.3.4. *SAT and Forwarding Information Verification*

As forwarding tables of networks grow and are sometimes manually modified, the problem of verifying forwarding information becomes critical and has recently gained in interest. Some problems that arise in network verification such as loop detection for example, may be naturally encoded as Boolean Satisfiability problems. Beside the theoretical interest of this encoding in complexity proofs, it has also a practical value for solving these problems by taking advantage of the many efficient Satisfiability testing solvers. Indeed, SAT solvers have proved to be very efficient in solving problems coming from various areas (Circuit Verification, Dependency and Conflicts in Software distributions...) and encoded in Conjunctive Normal Form. To test an approach using SAT solvers in network verification, one need to collect data sets from real network and to develop good models for generating realistic networks. The technique of encoding and the solvers themselves need to be adapted to this kind of problems. All this represent a rich experimental field of future research.

3.3.5. *Network Analysis*

Finally, we are interested in analyzing the structural properties of practical networks. This can include diameter computation or ranking of nodes. As we mostly consider large networks, we are often interested in efficient heuristics. Ideally, we target heuristics that give exact answer although fast computation time is not guaranteed for all networks. We already have designed such heuristics for diameter computation; understanding the structural properties that enable fast computation time in practice is still an open question.

4. Application Domains

4.1. Large scale networks

Application domains include evaluating Internet performances, the design of new peer-to-peer applications, enabling large scale networks, and developping tools for transportation networks.

5. New Software and Platforms

5.1. big-graph-tools

FUNCTIONAL DESCRIPTION

Gang is developing software for big graph manipulation. A preliminary library offering diameter and skeleton computation is available online [1]. This library was used to compute the diameters of the worldwide road network (200M edges), skeleton subtrees of the shortest-path trees of continental-sized road networks, as well as the largest strongly connected component of the Twitter follower-followee graph (23G edges).

- Contact: Laurent Viennot
- URL: https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/

---

[1] https://who.rocq.inria.fr/Laurent.Viennot/dev/big-graph-tools/

6. New Results

6.1. Graph and Combinatorial Algorithms

6.1.1. New Results in Multi-sweep Graph Search

A theoretical model to describe a series of successive graph searches is proposed in [7]. We apply this model to deal with cocomparability graphs (i.e., complement of comparability graphs) in [6] and in [48] or [44]. In this series of papers we provide a general algorithmic framework for many optimization problems on cocomparability graphs, such as Minimum Path Cover, Maximum Independent Set, Maximal interval subgraph, etc.

We also provide a new very simple algorithm for the recognition of cocomparability graphs. This algorithm is also based on a series of successive graph searches in [13].

We mainly use the two well-known Lexicographic graph searches: LBFS and LDFS, but not only. In [48], we also introduced a new graph search LocalMNS which seems to behave nicely on cocomparability graphs.

6.1.2. Studies of Read Networks and Laminar Graphs

In the context of biological networks, in [50] we introduce k-laminar graphs — a new class of graphs which extends the idea of Asteroidal-triple-free graphs. A graph is k-laminar if it admits a diametral path that is k-dominating. This bio-inspired class of graphs was motivated by a biological application dealing with sequence similarity networks of reads. We briefly develop the context of the biological application in which this graph class appeared and then we consider the relationships of this new graph class among known graph classes and then we study its recognition problem. For the recognition of k-laminar graphs, we develop polynomial algorithms when k is fixed. For $k = 1$, our algorithm improves a Deogun and Krastch's algorithm (1999). We finish by an NP-completeness result when k is unbounded.

6.1.3. Further Studies into Shortest Paths, Eccentricity, and Laminarity

From our recent research on diameter computations on graphs we also investigated some reductions between polynomial problems on graphs [3].

We also extend the well-known multisweep BFS to give a better polynomial-time approximation for the Maximum Eccentricity Shortest Path Problem, in relation with the k-Laminarity Problem [20].

6.1.4. Clique Colourings of Perfect Graphs

A *clique-coloring* of a graph G is an assignment of colors to the vertices of G in such a way that no inclusion-wise maximal clique of size at least two of G is monochromatic (as usual, a set of vertices is *monochromatic* if all vertices in the set received the same color). The *clique-chromatic number* of G, denoted by $\chi_C(G)$, is the smallest integer k such that G is admits a clique-coloring using at most k colors. Note that every proper coloring of G is also a clique-coloring of G, and so $\chi_C(G) \leq \chi(G)$. Furthermore, if G is triangle-free, then $\chi_C(G) = \chi(G)$ (since there are triangle-free graphs of arbitrarily large chromatic number, this implies that there are triangle-free graphs of arbitrarily large clique-chromatic number). However, if G contains triangles, $\chi_C(G)$ may be much smaller than $\chi(G)$. For instance, if G contains a dominating vertex, then $\chi_C(G) \leq 2$ (we assign the color 1 to the dominating vertex and the color 2 to all other vertices of G), while $\chi(G)$ may be arbitrarily large. Note that this implies that the clique-chromatic number is not monotone with respect to induced subgraphs, that is, there exist graphs H and G such that H is an induced subgraph of G, but $\chi_C(H) > \chi_C(G)$. (In particular, the restriction of a clique-coloring of G to an induced subgraph H of G need not be a clique-coloring of H.)

A graph G is *perfect* if all its induced subgraphs H satisfy $\chi(H) = \omega(H)$, where $\omega(H)$ denotes the size of a maximum clique. It was asked by Duffus, Sands, Sauer, and Woodrow in a paper from 1991 whether perfect graphs have a bounded clique-chromatic number and indeed it has been proven since that for many sublasses of the class of perfect graphs, this holds. Even more, until now it was not known whether there were any perfect graphs of clique-chromatic number greater than three. The main result of [4] is to prove that there exist perfect graphs of arbitrarily large clique-chromatic number, which gives a negative answer for the question of Duffus et al. mentioned above.

6.1.5. Algorithmic Aspects of Switch Cographs

Cographs are the graphs totally decomposable using series and parallel operations, in [5] we introduced an interesting generalization, namely the class of switch cographs. These are the class of graphs that are totally decomposable w.r.t involution modular decomposition — a generalization of the modular decomposition of 2-structure, which has a unique linear-sized decomposition tree. We use our new decomposition tool to design three practical algorithms for the maximum cut, vertex cover and vertex separator problems. The complexity of these problems was previously unknown for this class of graphs.

6.1.6. Shrinking Maxima, Decreasing Costs: New Online Packing and Covering Problems

In [16], we consider two new variants of online integer programs that are duals. In the packing problem we are given a set of items and a collection of knapsack constraints over these items that are revealed over time in an online fashion. Upon arrival of a constraint we may need to remove several items (irrevocably) so as to maintain feasibility of the solution. Hence, the set of packed items becomes smaller over time. The goal is to maximize the number, or value, of packed items. The problem originates from a buffer-overflow model in communication networks, where items represent information units broken into multiple packets. The other problem considered is online covering: there is a universe to be covered. Sets arrive online, and we must decide for each set whether we add it to the cover or give it up. The cost of a solution is the total cost of sets taken, plus a penalty for each uncovered element. The number of sets in the solution grows over time, but its cost goes down. This problem is motivated by team formation, where the universe consists of skills, and sets represent candidates we may hire. The packing problem was introduced in Emek et al. (SIAM J Comput 41(4):728-746, 2012) for the special case where the matrix is binary; in this paper we extend the solution to general matrices with non-negative integer entries. The covering problem is introduced in this paper; we present matching upper and lower bounds on its competitive ratio.

6.1.7. The Complexity of the Shortest-path Broadcast Problem

In [8], we study the shortest-path broadcast problem in graphs and digraphs, where a message has to be transmitted from a source node s to all the nodes along shortest paths, in the classical telephone model. For both graphs and digraphs, we show that the problem is equivalent to the broadcast problem in layered directed graphs. We then prove that this latter problem is NP-hard, and therefore that the shortest-path broadcast problem is NP-hard in graphs as well as in digraphs. Nevertheless, we prove that a simple polynomial-time algorithm, called MDST-broadcast, based on min-degree spanning trees, approximates the optimal broadcast time within a multiplicative factor $3/2$ in 3-layer digraphs, and $O(\log n/\log\log n)$ in arbitrary multi-layer digraphs. As a consequence, one can approximate the optimal shortest-path broadcast time in polynomial time within a multiplicative factor $3/2$ whenever the source has eccentricity at most 2, and within a multiplicative factor $O(\log n/\log\log n)$ in the general case, for both graphs and digraphs. The analysis of MDST-broadcast is tight, as we prove that this algorithm cannot approximate the optimal broadcast time within a factor smaller than $\Omega(\log n/\log\log n)$.

6.1.8. Setting Ports in an Anonymous Network: How to Reduce the Level of Symmetry

A fundamental question in the setting of anonymous graphs concerns the ability of nodes to spontaneously break symmetries, based on their local perception of the network. In contrast to previous work, which focuses on symmetry breaking under arbitrary port labelings, in [37] we study the following design question: Given an anonymous graph G without port labels, how to assign labels to the ports of G, in interval form at each vertex, so that symmetry breaking can be achieved using a message-passing protocol requiring as few rounds of synchronous communication as possible?

More formally, for an integer $l > 0$, the *truncated view* $\mathcal{V}_l(v)$ of a node v of a port-labeled graph is defined as a tree encoding labels encountered along all walks in the network which originate from node v and have length at most l, and we ask about an assignment of labels to the ports of G so that the views $\mathcal{V}_l(v)$ are distinct for all nodes $v \in V$, with the goal being to minimize l.

We present such efficient port labelings for any graph G, and we exhibit examples of graphs showing that the derived bounds are asymptotically optimal in general. More precisely, our results imply the following statements.

1. For any graph G with n nodes and diameter D, a uniformly random port labeling achieves $l = O(\min(D, \log n))$, w.h.p.
2. For any graph G with n nodes and diameter D, it is possible to construct in polynomial time a labeling that satisfies $l = O(\min(D, \log n))$.
3. For any integers $n \geq 2$ and $D \leq \log_2 n - \log_2 \log_2 n$, there exists a graph G with n nodes and diameter D which satisfies $l \geq \frac{1}{2}D - \frac{5}{2}$.

6.1.9. Robustness of the Rotor-Router Mechanism

The *rotor-router model*, also called the *Propp machine*, was first considered as a deterministic alternative to the random walk. The edges adjacent to each node v (or equivalently, the exit ports at v) are arranged in a fixed cyclic order, which does not change during the exploration. Each node v maintains a *port pointer* π_v which indicates the exit port to be adopted by an agent on the conclusion of the next visit to this node (the “next exit port”). The rotor-router mechanism guarantees that after each consecutive visit at the same node, the pointer at this node is moved to the next port in the cyclic order. It is known that, in an undirected graph G with m edges, the route adopted by an agent controlled by the rotor-router mechanism forms eventually an Euler tour based on arcs obtained via replacing each edge in G by two arcs with opposite direction. The process of ushering the agent to an Euler tour is referred to as the *lock-in problem*. In [Yanovski et al., Algorithmica 37(3), 165–186 (2003)], it was proved that, independently of the initial configuration of the rotor-router mechanism in G, the agent locks-in in time bounded by $2mD$, where D is the diameter of G.

In [2], we examine the dependence of the lock-in time on the initial configuration of the rotor-router mechanism. Our analysis is performed in the form of a game between a player p intending to lock-in the agent in an Euler tour as quickly as possible and its adversary a with the counter objective. We consider all cases of who decides the initial cyclic orders and the initial values π_v. We show, for example, that if a provides its own port numbering after the initial setup of pointers by p, the complexity of the lock-in problem is $O(m \cdot \min\{\log m, D\})$.

We also investigate the robustness of the rotor-router graph exploration in presence of faults in the pointers π_v or dynamic changes in the graph. We show, for example, that after the exploration establishes an Eulerian cycle, if k edges are added to the graph, then a new Eulerian cycle is established within $O(km)$ steps.

6.1.10. The Multi-Agent Rotor-Router on the Ring: A Deterministic Alternative to Parallel Random Walks

Continuing the line of research on the rotor-router model, in [18] we consider the setting in which multiple, indistinguishable agents are deployed in parallel in the nodes of the graph, and move around the graph in synchronous rounds, interacting with a single rotor-router system. We propose new techniques which allow us to perform a theoretical analysis of the multi-agent rotor-router model, and to compare it to the scenario of parallel independent random walks in a graph. Our main results concern the n-node ring, and suggest a strong similarity between the performance characteristics of this deterministic model and random walks.

We show that on the ring the rotor-router with k agents admits a cover time of between $\Theta(n^2/k^2)$ in the best case and $\Theta(n^2/\log k)$ in the worst case, depending on the initial locations of the agents, and that both these bounds are tight. The corresponding expected value of the cover time for k random walks, depending on the initial locations of the walkers, is proven to belong to a similar range, namely between $\Theta(n^2/(k^2/\log^2 k))$ and $\Theta(n^2/\log k)$.

Finally, we study the limit behavior of the rotor-router system. We show that, once the rotor-router system has stabilized, all the nodes of the ring are always visited by some agent every $\Theta(n/k)$ steps, regardless of how the system was initialized. This asymptotic bound corresponds to the expected time between successive visits to a node in the case of k random walks. All our results hold up to a polynomially large number of agents ($1 \leq k < n^{1/11}$).

6.1.11. Bounds on the Cover Time of Parallel Rotor Walks

In [12], we study the parallel rotor-router model in the case of general graphs. We consider the cover time of such a system, i.e., the number of steps after which each node has been visited by at least one walk, regardless of the initialization of the walks. We show that for any graph with m edges and diameter D, this cover time is at most $\Theta(mD/\log k)$ and at least $\Theta(mD/k)$, which corresponds to a speedup of between $\Theta(\log k)$ and $\Theta(k)$ with respect to the cover time of a single walk.

6.2. Distributed Computing

6.2.1. Local Conflict Coloring

Locally finding a solution to symmetry-breaking tasks such as vertex-coloring, edge-coloring, maximal matching, maximal independent set, etc., is a long-standing challenge in distributed network computing. More recently, it has also become a challenge in the framework of centralized local computation. In [30], we introduce conflict coloring as a general symmetry-breaking task that includes all the aforementioned tasks as specific instantiations — conflict coloring includes all locally checkable labeling tasks from [Naor&Stockmeyer, STOC 1993]. Conflict coloring is characterized by two parameters l and d, where the former measures the amount of freedom given to the nodes for selecting their colors, and the latter measures the number of constraints which colors of adjacent nodes are subject to. We show that, in the standard LOCAL model for distributed network computing, if $l/d > \Delta$, then conflict coloring can be solved in $\widetilde{O}(\sqrt{\Delta}) + \log^* n$ rounds in n-node graphs with maximum degree Δ, where $\widetilde{O}$ ignores the polylog factors in Δ. The dependency in n is optimal, as a consequence of the $\Omega(\log^* n)$ lower bound by [Linial, SIAM J. Comp. 1992] for $(\Delta + 1)$-coloring. An important special case of our result is a significant improvement over the best known algorithm for distributed $(\Delta + 1)$-coloring due to [Barenboim, PODC 2015], which required $\widetilde{O}(\Delta^{3/4}) + \log^* n$ rounds. Improvements for other variants of coloring, including $(\Delta + 1)$-list-coloring, $(2\Delta - 1)$-edge-coloring, T-coloring, etc., also follow from our general result on conflict coloring. Likewise, in the framework of centralized local computation algorithms (LCAs), our general result yields an LCA which requires a smaller number of probes than the previously best known algorithm for vertex-coloring, and works for a wide range of coloring problems.

6.2.2. A Hierarchy of Local Decision

In [29], we extend the notion of *distributed decision* in the framework of distributed network computing, inspired by recent results on so-called *distributed graph automata*. We show that, by using distributed decision mechanisms based on the interaction between a *prover* and a *disprover*, the size of the certificates distributed to the nodes for certifying a given network property can be drastically reduced. For instance, we prove that minimum spanning tree can be certified with $O(\log n)$-bit certificates in n-node graphs, with just one interaction between the prover and the disprover, while it is known that certifying MST requires $\Omega(\log^2 n)$-bit certificates if only the prover can act. The improvement can even be exponential for some simple graph properties. For instance, it is known that certifying the existence of a nontrivial automorphism requires $\Omega(n^2)$ bits if only the prover can act. We show that there is a protocol with two interactions between the prover and the disprover enabling to certify nontrivial automorphism with $O(\log n)$-bit certificates. These results are achieved by defining and analysing a *local hierarchy* of decision which generalizes the classical notions of *proof-labelling schemes* and *locally checkable proofs*.

6.2.3. Distributed Testing of Excluded Subgraphs

In [35], we study property testing in the context of distributed computing, under the classical CONGEST model. It is known that testing whether a graph is triangle-free can be done in a constant number of rounds, where the constant depends on how far the input graph is from being triangle-free. We show that, for every connected 4-node graph H, testing whether a graph is H-free can be done in a constant number of rounds too. The constant also depends on how far the input graph is from being H-free, and the dependence is identical to the one in the case of testing triangle-freeness. Hence, in particular, testing whether a graph is K_4-free, and testing whether a graph is C_4-free can be done in a constant number of rounds (where K_k denotes the k-node clique, and C_k denotes the k-node cycle). On the other hand, we show that testing K_k-freeness and C_k-freeness for $k \geq 5$ appear to be much harder. Specifically, we investigate two natural types of generic algorithms for testing H-freeness, called DFS tester and BFS tester. The latter captures the previously known algorithm to test the presence of triangles, while the former captures our generic algorithm to test the presence of a 4-node graph pattern H. We prove that both DFS and BFS testers fail to test K_k-freeness and C_k-freeness in a constant number of rounds for $k \geq 5$.

6.2.4. Asynchronous Coordination Under Preferences and Constraints

Adaptive renaming can be viewed as a coordination task involving a set of asynchronous agents, each aiming at grabbing a single resource out of a set of resources totally ordered by their desirability. Similarly, musical chairs is also defined as a coordination task involving a set of asynchronous agents, each aiming at picking one of a set of available resources, where every agent comes with an a priori preference for some resource. In [22], we foresee instances in which some combinations of resources are allowed, while others are disallowed. We model these constraints, i.e., the restrictions on the ability to use some combinations of resources, as an undirected graph whose nodes represent the resources, and an edge between two resources indicates that these two resources cannot be used simultaneously. In other words, the sets of resources that are allowed are those which form independent sets in the graph. E.g., renaming and musical chairs are specific cases where the graph is stable (i.e., it the empty graph containing no edges). As for musical chairs, we assume that each agent comes with an a priori preference for some resource. If an agent's preference is not in conflict with the preferences of the other agents, then this preference can be grabbed by the agent. Otherwise, the agents must coordinate to resolve their conflicts, and potentially choose non preferred resources. We investigate the following problem: given a graph, what is the maximum number of agents that can be accommodated subject to non-altruistic behaviors of early arriving agents? We entirely solve this problem under the restriction that agents which cannot grab their preferred resources must then choose a resource among the nodes of a predefined independent set. However, the general case, where agents which cannot grab their preferred resource are then free to choose any resource, is shown to be far more complex. In particular, just for cyclic constraints, the problem is surprisingly difficult. Indeed, we show that, intriguingly, the natural algorithm inspired from optimal solutions to adaptive renaming or musical chairs is sub-optimal for cycles, but proven to be at most 1 to the optimal. The main message of this paper is that finding optimal solutions to the coordination with constraints and preferences task requires to design " dynamic " algorithms, that is, algorithms of a completely different nature than the "static" algorithms used for, e.g., renaming.

6.2.5. Making Local Algorithms Wait-Free: The Case of Ring Coloring

When considering distributed computing, reliable message-passing synchronous systems on the one side, and asynchronous failure-prone shared-memory systems on the other side, remain two quite independently studied ends of the reliability/asynchrony spectrum. The concept of locality of a computation is central to the first one, while the concept of wait-freedom is central to the second one. This work proposes a new DECOUPLED model in an attempt to reconcile these two worlds. It consists of a synchronous and reliable communication graph of n nodes, and on top a set of asynchronous crash-prone processes, each attached to a communication node. To illustrate the DECOUPLED model, the paper [21] presents an asynchronous 3-coloring algorithm for the processes of a ring. From the processes point of view, the algorithm is wait-free. From a locality point of view, each process uses information only from processes at distance $O(\log^* n)$ from it. This local wait-free algorithm is based on an extension of the classical Cole and Vishkin vertex coloring algorithm in which the processes are not required to start simultaneously.

6.2.6. t-Resilient Immediate Snapshot Is Impossible

An immediate snapshot object is a high level communication object, built on top of a read/write distributed system in which all except one processes may crash. It allows each process to write a value and obtains a set of pairs (process id, value) such that, despite process crashes and asynchrony, the sets obtained by the processes satisfy noteworthy inclusion properties. Considering an n-process model in which up to t processes are allowed to crash (t-crash system model), the paper [25] is on the construction of t-resilient immediate snapshot objects. In the t-crash system model, a process can obtain values from at least $(n-t)$ processes, and, consequently, t-immediate snapshot is assumed to have the properties of the basic $(n-1)$-resilient immediate snapshot plus the additional property stating that each process obtains values from at least $(n-t)$ processes. The main result of the work is the following. While there is a (deterministic) $(n-1)$-resilient algorithm implementing the basic $(n-1)$-immediate snapshot in an $(n-1)$-crash read/write system, there is no t-resilient algorithm in a t-crash read/write model when $t \in [1...(n-2)]$. This means that, when $t < n-1$, the notion of t-resilience is inoperative when one has to implement t-immediate snapshot for these values of t: the model assumption "at most $t < n-1$ processes may crash" does not provide us with additional computational power allowing for the design of a genuine t-resilient algorithm (genuine meaning that such an algorithm would work in the t-crash model, but not in the $(t+1)$-crash model). To show these results, we rely on well-known distributed computing agreement problems such as consensus and k-set agreement.

6.2.7. Perfect Failure Detection with Very Few Bits

A *failure detector* is a distributed oracle that provides the processes with information about failures. The *perfect* failure detector provides accurate and eventually complete information about process failures. In [34], we show that, in asynchronous failure-prone message-passing systems, perfect failure detection can be achieved by an oracle that outputs at most $\lceil \log \alpha(n) \rceil + 1$ bits per process in n-process systems, where α denotes the inverse-Ackermann function. This result is essentially optimal, as we also show that, in the same environment, no failure detectors outputting a constant number of bit per process can achieve perfect failure detection.

6.2.8. Decentralized Asynchronous Crash-Resilient Runtime Verification

Runtime Verification (RV) is a lightweight method for monitoring the formal specification of a system during its execution. It has recently been shown that a given state predicate can be monitored consistently by a set of crash-prone asynchronous *distributed* monitors, only if sufficiently many different verdicts can be emitted by each monitor. In [27], we revisit this impossibility result in the context of LTL semantics for RV. We show that employing the four-valued logic RVLTL will result in inconsistent distributed monitoring for some formulas. Our first main contribution is a family of logics, called $\mathrm{LTL}(k)$, that refines RVLTL incorporating $2k+4$ truth values, for each $k \geq 0$. The truth values of $\mathrm{LTL}(k)$ can be effectively used by each monitor to reach a consistent global set of verdicts for each given formula, provided k is sufficiently large. Our second main contribution is an algorithm for monitor construction enabling fault-tolerant distributed monitoring based on the aggregation of the individual verdicts by each monitor.

6.2.9. Asynchronous Consensus with Bounded Memory

The paper [11] presents a bounded memory size Obstruction-Free consensus algorithm for the asynchronous shared memory model. More precisely for a set of n processes, this algorithm uses $n+2$ multi-writer multi-reader registers, each of these registers being of size $O(\log n)$ bits. From this, we get a bounded memory size space complexity consensus algorithm with single-writer multi-reader registers and a bounded memory size space complexity consensus algorithm in the asynchronous message passing model with a majority of correct processes. As it is easy to ensure the Obstruction-Free assumption with randomization (or with leader election failure detector Ω) we obtain a bounded memory size randomized consensus algorithm and a bounded memory size consensus algorithm with failure detector.

6.2.10. Implementing Snapshot Objects on Top of Crash-Prone Asynchronous Message-Passing Systems

Distributed snapshots, as introduced by Chandy and Lamport in the context of asynchronous failure-free message-passing distributed systems, are consistent global states in which the observed distributed application might have passed through. It appears that two such distributed snapshots cannot necessarily be compared (in the sense of determining which one of them is the “first”). Differently, snapshots introduced in asynchronous crash-prone read/write distributed systems are totally ordered, which greatly simplify their use by upper layer applications. In order to benefit from shared memory snapshot objects, it is possible to simulate a read/write shared memory on top of an asynchronous crash-prone message-passing system, and build then snapshot objects on top of it. This algorithm stacking is costly in both time and messages. To circumvent this drawback, the paper [24] presents algorithms building snapshot objects directly on top of asynchronous crash-prone message-passing system. “Directly” means here “without building an intermediate layer such as a read/write shared memory”. To the authors knowledge, the proposed algorithms are the first providing such constructions. Interestingly enough, these algorithms are efficient and relatively simple.

6.2.11. Set-Consensus Collections are Decidable

A natural way to measure the power of a distributed-computing model is to characterize the set of tasks that can be solved in it. In general, however, the question of whether a given task can be solved in a given model is undecidable, even if we only consider the wait-free shared-memory. In [23], we address this question for restricted classes of models and tasks. We show that the question of whether a collection C of (ℓ, j)-set consensus objects, for various ℓ (the number of processes that can invoke the object) and j (the number of distinct outputs the object returns), can be used by n processes to solve wait-free k-set consensus is decidable. Moreover, we provide a simple $O(n^2)$ decision algorithm, based on a dynamic programming solution to the Knapsack optimization problem. We then present an adaptive wait-free set-consensus algorithm that, for each set of participating processes, achieves the best level of agreement that is possible to achieve using C. Overall, this gives us a complete characterization of a read-write model defined by a collection of set-consensus objects through its set-consensus power.

6.2.12. Minimizing the Number of Opinions for Fault-Tolerant Distributed Decision Using Well-Quasi Orderings

The notion of deciding a *distributed language* L is of growing interest in various distributed computing settings. Each process p_i is given an input value x_i, and the processes should collectively decide whether their set of input values $x = (x_i)_i$ is a valid state of the system w.r.t. to some specification, i.e., if $x \in L$. In *non-deterministic* distributed decision each process p_i gets a local certificate c_i in addition to its input x_i. If the input $x \in L$then there exists a certificate $c = (c_i)_i$ such that the processes collectively accept x, and if $x \notin L$, then for every c, the processes should collectively reject x. The collective decision is expressed by the set of *opinions* emitted by the processes, and one aims at minimizing the number of possible opinions emitted by each process. In [33], we study non-deterministic distributed decision in asynchronous systems where processes may crash. In this setting, it is known that the number of opinions needed to deterministically decide a language can grow with n, the number of processes in the system.We prove that every distributed language L can be non-deterministically decided using only three opinions, with certificates of size $\lceil \log \alpha(n) \rceil + 1$ bits, where α grows at least as slowly as the inverse of the Ackerman function. The result is optimal, as we show that there are distributed languages that cannot be decided using just two opinions, even with arbitrarily large certificates. To prove our upper bound, we introduce the notion of *distributed encoding of the integers*, that provides an explicit construction of a long *bad sequence* in the *well-quasi-ordering*$(\{0,1\}^*, \leq_*)$ controlled by the successor function. Thus, we provide a new class of applications for well-quasi-orderings that lies outside logic and complexity theory. For the lower bound we use combinatorial topology techniques.

6.2.13. Collision-Free Network Exploration

In [9], we consider a network exploration setting in which mobile agents start at different nodes of an n-node network. The agents synchronously move along the network edges in a *collision-free* way, i.e., in no round two agents may occupy the same node. An agent has no knowledge of the number and initial positions of other agents. We are looking for the shortest time required to reach a configuration in which each agent has visited all nodes and returned to its starting location. In the scenario when each mobile agent knows the map of the network, we provide tight (up to a constant factor) lower and upper bounds on the collision-free exploration time in arbitrary graphs, and the exact bound for the trees. In the second scenario, where the network is unknown to the agents, we propose collision-free exploration strategies running in $O(n^2)$ rounds in tree networks and in $O(n^5 \log n)$ rounds in networks with an arbitrary topology.

6.2.14. When Patrolmen Become Corrupted: Monitoring a Graph Using Faulty Mobile Robots

In [10], we consider a setting in which a team of k mobile robots is deployed on a weighted graph whose edge weights represent distances. The robots perpetually move along the domain, represented by all points belonging to the graph edges, not exceeding their maximal speed. The robots need to patrol the graph by regularly visiting all points of the domain. We consider a team of robots (patrolmen), at most f of which may be unreliable, failing to comply with their patrolling duties. What algorithm should be followed so as to minimize the maximum time between successive visits of every edge point by a reliable patrolmen? The corresponding measure of efficiency of patrolling called *idleness* has been widely accepted in the robotics literature. We extend it to the case of untrusted patrolmen; we denote by $I_k^f(G)$ the maximum time that a point of the domain may remain unvisited by reliable patrolmen. The objective is to find patrolling strategies minimizing $I_k^f(G)$.

We investigate this problem for various classes of graphs. We design optimal algorithms for line segments, which turn out to be surprisingly different from strategies for related patrolling problems proposed in the literature. We then use these results to provide algorithms for general graphs. For Eulerian graphs G, we give an optimal patrolling strategy with idleness $I_k^f(G) = (f+1)E(G)/k$, where $E(G)$ is the sum of the lengths of the edges of G. For arbitrary graphs and given ratio r of faulty robots, $r := f/k < 1/2$, we design a strategy which is a $(1+\epsilon)$ approximation of the optimal one, for sufficiently large k. Further, we show the hardness of the problem of computing the idle time for three robots, at most one of which is faulty, by reduction from 3-edge-coloring of cubic graphs — a known NP-hard problem. A byproduct of our proof is the investigation of classes of graphs minimizing idle time (with respect to the total length of edges); an example of such a class is known in the literature under the name of Kotzig graphs.

6.2.15. Noisy Rumor Spreading and Plurality Consensus

Error-correcting codes are efficient methods for handling noisy communication channels in the context of technological networks. However, such elaborate methods differ a lot from the unsophisticated way biological entities are supposed to communicate. Yet, it has been recently shown by Feinerman, Haeupler, and Korman [PODC 2014] that complex coordination tasks such as rumor spreading and majority consensus can plausibly be achieved in biological systems subject to noisy communication channels, where every message transferred through a channel remains intact with small probability $1+\varepsilon$, without using coding techniques. This result is a considerable step towards a better understanding of the way biological entities may cooperate. It has nevertheless been established only in the case of 2-valued opinions: rumor spreading aims at broadcasting a single-bit opinion to all nodes, and majority consensus aims at leading all nodes to adopt the single-bit opinion that was initially present in the system with (relative) majority. In [32], we extend this previous work to k-valued opinions, for any constant $k \geq 2$. Our extension requires to address a series of important issues, some conceptual, others technical. We had to entirely revisit the notion of noise, for handling channels carrying k-valued messages. In fact, we precisely characterize the type of noise patterns for which plurality consensus is solvable. Also, a key result employed in the bivalued case by Feinerman et al. is an estimate of the probability of observing the most frequent opinion from observing the mode of a small sample. We generalize this result to the multivalued case by providing a new analytical proof for the bivalued case that is amenable to be extended, by induction, and that is of independent interest.

6.3. Models and Algorithms for Networks

6.3.1. Beyond Highway Dimension: Small Distance Labels Using Tree Skeletons

The goal of a hub-based distance labeling scheme for a network $G = (V, E)$ is to assign a small subset $S(u) \subseteq V$ to each node $u \in V$, in such a way that for any pair of nodes u, v, the intersection of hub sets $S(u) \cap S(v)$ contains a node on the shortest uv-path. The existence of small hub sets, and consequently efficient shortest path processing algorithms, for road networks is an empirical observation. A theoretical explanation for this phenomenon was proposed by Abraham et al. (SODA 2010) through a network parameter they called highway dimension, which captures the size of a hitting set for a collection of shortest paths of length at least r intersecting a given ball of radius $2r$. In [38], we revisit this explanation, introducing a more tractable (and directly comparable) parameter based solely on the structure of shortest-path spanning trees, which we call skeleton dimension. We show that skeleton dimension admits an intuitive definition for both directed and undirected graphs, provides a way of computing labels more efficiently than by using highway dimension, and leads to comparable or stronger theoretical bounds on hub set size.

6.3.2. Sublinear-Space Distance Labeling using Hubs

Continuing work in the previously discussed framework of hub-based distance labeling schemes, in [36], [39], we present a hub labeling which allows us to decode exact distances in sparse graphs using labels of size sublinear in the number of nodes. For graphs with at most n nodes and average degree Δ, the tradeoff between label bit size L and query decoding time T for our approach is given by $L = O(n \log \log_{\Delta} T / \log_{\Delta} T)$, for any $T \leq n$. Our simple approach is thus the first sublinear-space distance labeling for sparse graphs that simultaneously admits small decoding time (for constant Δ, we can achieve any $T = \omega(1)$ while maintaining $L = o(n)$), and it also provides an improvement in terms of label size with respect to previous slower approaches.

By using similar techniques, we then present a 2-additive labeling scheme for general graphs, i.e., one in which the decoder provides a 2-additive-approximation of the distance between any pair of nodes. We achieve almost the same label size-time tradeoff $L = O(n \log^2 \log T / \log T)$, for any $T \leq n$. To our knowledge, this is the first additive scheme with constant absolute error to use labels of sublinear size. The corresponding decoding time is then small (any $T = \omega(1)$ is sufficient).

We believe all of our techniques are of independent value and provide a desirable simplification of previous approaches.

6.3.3. Labeling Schemes for Ancestry Relation

In [17], we solve the ancestry-labeling scheme problem which aims at assigning the shortest possible labels (bit strings) to nodes of rooted trees, so that ancestry queries between any two nodes can be answered by inspecting their assigned labels only. This problem was introduced more than twenty years ago by Kannan et al. [STOC '88], and is among the most well-studied problems in the field of informative labeling schemes. We construct an ancestry-labeling scheme for n-node trees with label size $\log_2 n + O(\log \log n)$ bits, thus matching the $log_2 n + \Omega(\log \log n)$ bits lower bound given by Alstrup et al. [SODA '03]. Our scheme is based on a simplified ancestry scheme that operates extremely well on a restricted set of trees. In particular, for the set of n-node trees with depth at most d, the simplified ancestry scheme enjoys label size of $\log_2 n + 2 \log_2 d + O(1)$ bits. Since the depth of most XML trees is at most some small constant, such an ancestry scheme may be of practical use. In addition, we also obtain an adjacency-labeling scheme that labels n-node trees of depth d with labels of size $\log_2 n + 3 \log_2 d + O(1)$ bits. All our schemes assign the labels in linear time, and guarantee that any query can be answered in constant time. Finally, our ancestry scheme finds applications to the construction of small universal partially ordered sets (posets). Specifically, for any fixed integer k, it enables the construction of a universal poset of size $O(n^k)$ for the family of n-element posets with tree-dimension at most k. Up to lower order terms, this bound is tight thanks to a lower bound of $n^{k-o(1)}$ due to Alon and Scheinerman [Order '88].

6.3.4. Independent Lazy Better-Response Dynamics on Network Games

In [43], we study an independent best-response dynamics on network games in which the nodes (players) decide to revise their strategies independently with some probability. We are interested in the convergence time to the equilibrium as a function of this probability, the degree of the network, and the potential of the underlying games.

6.3.5. Forwarding Tables Verification through Representative Header Sets

Forwarding table verification consists in checking the distributed data-structure resulting from the forwarding tables of a network. A classical concern is the detection of loops. We study in [42] this problem in the context of software-defined networking (SDN) where forwarding rules can be arbitrary bitmasks (generalizing prefix matching) and where tables are updated by a centralized controller. Basic verification problems such as loop detection are NP-hard and most previous work solves them with heuristics or SAT solvers. We follow a different approach based on computing a representation of the header classes, i.e. the sets of headers that match the same rules. This representation consists in a collection of representative header sets, at least one for each class, and can be computed centrally in time which is polynomial in the number of classes. Classical verification tasks can then be trivially solved by checking each representative header set. In general, the number of header classes can increase exponentially with header length, but it remains polynomial in the number of rules in the practical case where rules are constituted with predefined fields where exact, prefix matching or range matching is applied in each field (e.g., IP/MAC addresses, TCP/UDP ports). We propose general techniques that work in polynomial time as long as the number of classes of headers is polynomial and that do not make specific assumptions about the structure of the sets associated to rules. The efficiency of our method rely on the fact that the data-structure representing rules allows efficient computation of intersection, cardinal and inclusion. Finally, we propose an algorithm to maintain such representation in presence of updates (i.e., rule insert/update/removal). We also provide a local distributed algorithm for checking the absence of black-holes and a proof labeling scheme for locally checking the absence of loops.

6.3.6. A Locally-Blazed Ant Trail Achieves Efficient Collective Navigation Despite Limited Information

This work fits into the framework of computationally-inspired analysis of biological systems. Any organism faces sensory and cognitive limitations which may result in maladaptive decisions. Such limitations are prominent in the context of groups where the relevant information at the individual level may not coincide with collective requirements. In [14], we study the navigational decisions exhibited by *Paratrechina longicornis* ants as they cooperatively transport a large food item. These decisions hinge on the perception of individuals which often restricts them from providing the group with reliable directional information. We find that, to achieve efficient navigation despite partial and even misleading information, these ants employ a locally-blazed trail. This trail significantly deviates from the classical notion of an ant trail: First, instead of systematically marking the full path, ants mark short segments originating at the load. Second, the carrying team constantly loses the guiding trail. We experimentally and theoretically show that the locally-blazed trail optimally and robustly exploits useful knowledge while avoiding the pitfalls of misleading information.

6.3.7. Parallel Exhaustive Search without Coordination

In [31], we analyze parallel algorithms in the context of *exhaustive search* over totally ordered sets. Imagine an infinite list of “boxes”, with a “treasure” hidden in one of them, where the boxes’ order reflects the importance of finding the treasure in a given box. At each time step, a search protocol executed by a searcher has the ability to peek into one box, and see whether the treasure is present or not. Clearly, the best strategy of a single searcher would be to open the boxes one by one, in increasing order. Moreover, by equally dividing the workload between them, k searchers can trivially find the treasure k times faster than one searcher. However, this straightforward strategy is very sensitive to failures (*e.g.,* crashes of processors), and overcoming this issue seems to require a large amount of communication. We therefore address the question of designing parallel search algorithms maximizing their *speed-up* and maintaining high levels of *robustness*, while minimizing the amount of resources for coordination. Based on the observation that algorithms that avoid communication are inherently robust, we focus our attention on identifying the best running time performance of *non-coordinating* algorithms. Specifically, we devise non-coordinating algorithms that achieve a speed-up of $9/8$ for two searchers, a speed-up of $4/3$ for three searchers, and in general, a speed-up of $\frac{k}{4}(1+1/k)^2$ for any $k \geq 1$ searchers. Thus, asymptotically, the speed-up is only four times worse compared to the case of full-coordination. Moreover, these bounds are tight in a strong sense as no non-coordinating search algorithm can achieve better speed-ups. Furthermore, our algorithms are surprisingly simple and hence applicable. Overall, we highlight that, in faulty contexts in which coordination between the searchers is technically difficult to implement, intrusive with respect to privacy, and/or costly in term of resources, it might well be worth giving up on coordination, and simply run our non-coordinating exhaustive search algorithms.

6.3.8. Rumor Spreading in Random Evolving Graphs

Randomized gossip is one of the most popular way of disseminating information in large scale networks. This method is appreciated for its simplicity, robustness, and efficiency. In the Push protocol, every informed node selects, at every time step (a.k.a. round), one of its neighboring node uniformly at random and forwards the information to this node. This protocol is known to complete information spreading in $O(\log n)$ time steps with high probability (w.h.p.) in several families of n-node *static* networks. The Push protocol has also been empirically shown to perform well in practice, and, specifically, toe robust against dynamic topological changes. In [15], we aim at analyzing the Push protocol in *dynamic* networks. We consider the *edge-Markovian* evolving graph model which captures natural temporal dependencies between the structure of the network at time t, and the one at time $t+1$. Precisely, a non-edge appears with probability p, while an existing edge dies with probability q. In order to fit with real-world traces, we mostly concentrate our study on the case where $p=\Omega(\frac{1}{n})$ and q is constant. We prove that, in this realistic scenario, the Push protocol does perform well, completing information spreading in $O(\log n)$ time steps w.h.p. Note that this performance holds even when the network is, w.h.p., disconnected at every time step (e.g., when $p \ll \frac{\log n}{n}$). Our result provides the first formal argument demonstrating the robustness of the Push protocol against network changes. We also address another range of parameters p and q, namely $p+q=1$ with arbitrary p and q. Although this latter range does not precisely fit with the measures performed on real-world traces, they can be of independent interest for other settings. The result in this case confirms the positive impact of dynamism.

6.3.9. Sparsifying Congested Cliques and Core-Periphery Networks

The *core-periphery* network architecture proposed by Avin et al. [ICALP 2014] was shown to support fast computation for many distributed algorithms, while being much sparser than the *congested clique*. For being efficient, the core-periphery architecture is however bounded to satisfy three axioms, among which is the capability of the core to emulate the clique, i.e., to implement the all-to-all communication pattern, in $O(1)$ rounds in the CONGEST model. In [26], we show that implementing all-to-all communication in k rounds can be done in n-node networks with roughly n^2/k edges, and this bound is tight. Hence, sparsifying the core beyond just saving a fraction of the edges requires to relax the constraint on the time to simulate the congested clique. We show that, for $p \gg \sqrt{\log n / n}$, a random graph in $\mathcal{G}_{n,p}$ can, w.h.p., perform the all-to-all communication pattern in $O(\min\{\frac{1}{p^2}, np\})$ rounds. Finally, we show that if the core can emulate the congested clique in t rounds, then there exists a distributed MST construction algorithm performing in $O(t \log n)$ rounds. Hence, for $t=O(1)$, our (deterministic) algorithm improves the best known (randomized) algorithm for constructing MST in core-periphery networks by a factor $\Theta(\log n)$.

6.3.10. Core-periphery Clustering and Collaboration Networks

In [28], we analyse the core-periphery clustering properties of collaboration networks, where the core of a network is formed by the nodes with highest degree. In particular, we first observe that, even for random graph models aiming at matching the degree-distribution and/or the clustering coefficient of real networks, these models produce synthetic graphs which have a spatial distribution of the triangles with respect to the core and to the periphery which does not match the spatial distribution of the triangles in the real networks. We therefore propose a new model, called CPCL, whose aim is to distribute the triangles in a way fitting with their real core-periphery distribution, and thus producing graphs matching the core-periphery clustering of real networks.

7. Bilateral Contracts and Grants with Industry

7.1. Collaboration with Nokia Bell Labs

Gang has a strong collaboration with Bell Labs (Nokia). We notably collaborate with Fabien Mathieu who is a former member of GANG and Nidhi Hegde. An ADR (joint research action) is dedicated to content centric networks and forwarding information verification. The PhD thesis of Leonardo Linguaglossa was funded by this contract.

This collaboration is developed inside the Alcatel-Lucent and Inria joint research lab.

8. Partnerships and Cooperations

8.1. Regional Initiatives

8.1.1. Laboratory of Information, Networking and Communication Sciences (LINCS)

Gang is participating to the LINCS, a research centre co-founded by Inria, Institut Mines-Télécom, UPMC and Alcatel-Lucent Bell Labs, dedicated to research and innovation in the domains of future information and communication networks, systems and services. Gang contributes to work on online social networks, content centric networking and forwarding information verification.

8.2. National Initiatives

8.2.1. ANR Displexity

Participants: Carole Gallet Delporte, Hugues Fauconnier, Pierre Fraigniaud, Amos Korman, Adrian Kosowski, Laurent Viennot.

Managed by University Paris Diderot, C. Delporte and H. Fauconnier lead this project that grants 1 Post-Doc.

Distributed computation keep raising new questions concerning computability and complexity. For instance, as far as fault-tolerant distributed computing is concerned, impossibility results do not depend on the computational power of the processes, demonstrating a form of undecidability which is significantly different from the one encountered in sequential computing. In the same way, as far as network computing is concerned, the impossibility of solving certain tasks locally does not depend on the computational power of the individual processes.

The main goal of DISPLEXITY (for DIStributed computing: computability and ComPLEXITY) is to establish the scientific foundations for building up a consistent theory of computability and complexity for distributed computing.

One difficulty to be faced by DISPLEXITY is to reconcile the different sub-communities corresponding to a variety of classes of distributed computing models. The current distributed computing community may indeed be viewed as two not necessarily disjoint sub-communities, one focusing on the impact of temporal issues, while the other focusing on the impact of spatial issues. The different working frameworks tackled by these two communities induce different objectives: computability is the main concern of the former, while complexity is the main concern of the latter.

Within DISPLEXITY, the reconciliation between the two communities will be achieved by focusing on the same class of problems, those for which the distributed outputs are interpreted as a single binary output: yes or no. Those are known as the yes/no-problems. The strength of DISPLEXITY is to gather specialists of the two main streams of distributed computing. Hence, DISPLEXITY will take advantage of the experience gained over the last decade by both communities concerning the challenges to be faced when building up a complexity theory encompassing more than a fragment of the field.

In order to reach its objectives, DISPLEXITY aims at achieving the following tasks:

- Formalizing yes/no-problems (decision problems) in the context of distributed computing. Such problems are expected to play an analogous role in the field of distributed computing as that played by decision problems in the context of sequential computing.
- Formalizing decision problems (yes/no-problems) in the context of distributed computing. Such problems are expected to play an analogous role in the field of distributed computing as that played by decision problems in the context of sequential computing.

- Revisiting the various explicit (e.g., failure-detectors) or implicit (e.g., a priori information) notions of oracles used in the context of distributed computing allowing us to express them in terms of decidability/complexity classes based on oracles.
- Identifying the impact of non-determinism on complexity in distributed computing. In particular, DISPLEXITY aims at a better understanding of the apparent lack of impact of non-determinism in the context of fault-tolerant computing, to be contrasted with the apparent huge impact of non-determinism in the context of network computing. Also, it is foreseen that non-determinism will enable the comparison of complexity classes defined in the context of fault-tolerance with complexity classes defined in the context of network computing.
- Last but not least, DISPLEXITY will focus on new computational paradigms and frameworks, including, but not limited to distributed quantum computing and algorithmic game theory (e.g., network formation games).

The project will have to face and solve a number of challenging problems. Hence, we have built the DISPLEXITY consortium so as to coordinate the efforts of those worldwide leaders in Distributed Computing who are working in our country. A successful execution of the project will result in a tremendous increase in the current knowledge and understanding of decentralized computing and place us in a unique position in the field.

The project has been extended until June 2016.

8.2.2. ANR DESCARTES

Participants: Carole Gallet Delporte, Hugues Fauconnier, Pierre Fraigniaud, Adrian Kosowski, Laurent Viennot.

Cyril Gavoille (U. Bordeaux) leads this project that grants 1 Post-Doc. H. Fauconnier is the local coordinator (This project began in October 2016).

Despite the practical interests of reusable frameworks for implementing specific distributed services, many of these frameworks still lack solid theoretical bases, and only provide partial solutions for a narrow range of services. We argue that this is mainly due to the lack of a generic framework that is able to unify the large body of fundamental knowledge on distributed computation that has been acquired over the last 40 years. The DESCARTES project aims at bridging this gap, by developing a systematic model of distributed computation that organizes the functionalities of a distributed computing system into reusable modular constructs assembled via well-defined mechanisms that maintain sound theoretical guarantees on the resulting system. DESCARTES arises from the strong belief that distributed computing is now mature enough to resolve the tension between the social needs for distributed computing systems, and the lack of a fundamentally sound and systematic way to realize these systems.

8.3. European Initiatives

8.3.1. FP7 & H2020 Projects

Amos Korman has an ERC Consolidator Grant entitled “Distributed Biological Algorithms (DBA)”, started in May 2015. This project proposes a new application for computational reasoning. More specifically, the purpose of this interdisciplinary project is to demonstrate the usefulness of an algorithmic perspective in studies of complex biological systems. We focus on the domain of collective behavior, and demonstrate the benefits of using techniques from the field of theoretical distributed computing in order to establish algorithmic insights regarding the behavior of biological ensembles. The project includes three related tasks, for which we have already obtained promising preliminary results. Each task contains a purely theoretical algorithmic component as well as one which integrates theoretical algorithmic studies with experiments. Most experiments are strategically designed by the PI based on computational insights, and are physically conducted by experimental biologists that have been carefully chosen by the PI. In turn, experimental outcomes will be theoretically analyzed via an algorithmic perspective. By this integration, we aim at deciphering how a biological individual (such as an ant) “thinks”, without having direct access to the neurological process within its brain, and how such limited individuals assemble into ensembles that appear to be far greater than the sum of their parts. The ultimate vision behind this project is to enable the formation of a new scientific field, called algorithmic biology, that bases biological studies on theoretical algorithmic insights.

8.3.2. LIA Struco

Pierre Charbit is director of the LIA STRUCO, which is an Associated International Laboratory of CNRS between IÚUK, Prague, and IRIF, Paris. The director on the Czech side is Pr. Jaroslav Nešetřil. The primary theme of the laboratory is graph theory, more specifically: sparsity of graphs (nowhere dense classes of graphs, bounded expansion classes of graphs), extremal graph theory, graph coloring, Ramsey theory, universality and morphism duality, graph and matroid algorithms and model checking.

STRUCO focuses on high-level study of fundamental combinatorial objects, with a particular emphasis on comprehending and disseminating the state-of-the-art theories and techniques developed. The obtained insights shall be applied to obtain new results on existing problems as well as to identify directions and questions for future work.

One of the main goals of STRUCO is to provide a sustainable and reliable structure to help Czech and French researchers cooperate on long-term projects, disseminate the results to students of both countries and create links between these students more systematically. The chosen themes of the project indeed cover timely and difficult questions, for which a stable and significant cooperation structure is needed. By gathering an important number of excellent researchers and students, the LEA will create the required environment for making advances, which shall be achieved not only by short-term exchanges of researchers, but also by a strong involvement of Ph. D students in the learning of state-of-the-art techniques and in the international collaborations.

STRUCO is a natural place to federate and organize these many isolated collaborations between our two countries. Thus, the project would ensure long-term cooperations and allow young researchers (especially PhD students) to maintain the fruitful exchanges between the two countries in the future years, in a structured and federated way.

8.4. International Initiatives

8.4.1. Inria International Partners

8.4.1.1. Informal International Partners

Ofer Feinerman (Physics department of complex systems, Weizmann Institute of Science, Rehovot, Israel), is a team member in Amos Korman's ERC project DBA. This collaboration has been formally established by signing a contract between the CNRS and the Weizmann Institute of Science, as part of the ERC project.

Rachid Guerraoui (School of Computer and Communication Sciences, EPFL, Switzerland) maintains an active research collaboration with Gang team members (Carole Delporte, Hugues Fauconnier).

Pierluigi Crescenzi (University of Florence, Italy) is a frequent visitor to the team and maintains an active research collaboration with Gang team members (Pierre Fraigniaud).

Sergio Rajsbaum (UNAM, Mexico) is a regular collaborator of the team, also involved formally in a joint French-Mexican research project (see next subsection).

Boaz Patt-Shamir (Tel Aviv University, Israel) is a regular collaborator of the team, also involved formally in a joint French-Israeli research project (see next subsection).

8.4.2. Participation in Other International Programs

Involvement in the bilateral Franco-Mexican project ECOS NORD (2013-2016) on “Distributed Verification”. Pierre Fraigniaud was the project's co-coordinator for the French side. Partners: IRIF and LaBRI (France), UNAM (Mexico).

8.5. International Research Visitors

8.5.1. Visits of International Scientists

Eli Gafni (1 month – June 2016)

Zvi Lotker, guest of Amos Korman (2 months – May, June 2016)

Thomas Sauerwald, guest of Adrian Kosowski (1 month – November 2016)

8.5.2. Visits to International Teams

Sergio Rasjbaum's Team (UNAM), C. Delporte and H. Fauconnier, 10 days (March 2016)

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific Events Organisation

9.1.1.1. General Chair, Scientific Chair

Pierre Fraigniaud has organized the 1st conference on Highlights of Algorithms (HALG 2016), Paris, June 2016. This conference has gathered more than 200 participants to attend talks presenting the most significant results on algorithms produced in the academic year 2015-2016. The 2nd issue of this conference, Highlights of Algorithms 2017, will take place in Berlin.
http://highlightsofalgorithms.org

9.1.2. Scientific Events Selection

9.1.2.1. Chair of Conference Program Committees

- Carole Delporte was PC Co-chair of NETYS 2016 — the 4th International Conference on Networked Systems, Morocco, May 18-20, 2016.
- Pierre Fraigniaud is PC chair for the Track Algorithms of the 31st IEEE International Parallel & Distributed Processing Symposium (IPDPS), to be held in Orlando, Florida, USA, May 29 - June 2, 2017. http://www.ipdps.org

9.1.2.2. Member of Conference Program Committees

- Hugues Fauconnier: NETYS 2016, PODC 2016.
- Carole Delporte: ICDCN 2016, ICDCS 2016, DISC 2016, OPODIS 2016.
- Adrian Kosowski: PODC 2016, ALGOSENSORS 2016, ICALP 2017.
- Pierre Fraigniaud: WWW 2016, ESA 2016, IPDPS 2016, OPODIS 2016, SIROCCO 2016, ALGOSENSORS 2016, FUN 2016, WWW 2017, SPAA 2017, DISC 2017.
- Amos Korman: ICALP 2017

9.1.3. Journal

9.1.3.1. Member of Editorial Boards

- Pierre Fraigniaud is a member of the Editorial Board of Distributed Computing (DC).
- Pierre Fraigniaud is a member of the Editorial Board of Theory of Computing Systems (TOCS).
- Pierre Fraigniaud is a member of the Editorial Board of Fundamenta Informaticae (FI).

9.1.4. Invited Talks

Adrian Kosowski: SIROCCO 2016

9.1.5. Scientific Expertise

Pierre Fraigniaud is member of the evaluation committee for the ERC Starting Grants (Panel 6).

9.1.6. Research Administration

Pierre Fraigniaud is director of the Institute de Recherche en Informatique Fondamentale (IRIF).

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Master: Carole Delporte and Hugues Fauconnier, Algorithmique distribuée avec mémoire partagée, 6h, M2, Université Paris Diderot

Master: Hugues Fauconnier, Cours programmation répartie, 33h, M2, Univ. Paris Diderot

Master: Carole Delporte, Cours et TP Protocoles des services internet, 44h, M2, Univ. Paris Diderot

Master: Carole Delporte, Cours Algorithme réparti, 33h, M2, Univ. Paris Diderot

Master: Carole Delporte and Hugues Fauconnier, Protocoles Réseaux, 72h, M1, Université Paris Diderot

Licence: Carole Delporte and Hugues Fauconnier, Sécurité informatique, 36h, L3, Univ. Paris Diderot

Licence: Hugues Fauconnier, Programmation objet et interfaces graphiques, 48h, L2-L3, EIDD

Licence: Boufkhad Yacine, Algorithmique et Informatique, 132h, L1, IUT de l'Université Paris Diderot

Licence: Boufkhad Yacine, Programmation Orientée Objet, 60h, L2, IUT de l'Université Paris Diderot

Licence: Boufkhad Yacine, Traitement de données, 16h, L2, IUT de l'Université Paris Diderot

Master: Pierre Fraigniaud, Algorithmique avancée, 24h, Ecole Centrale Supélec Paris, M2

Master: Pierre Fraigniaud, Algorithmique parallèle et distribuée, 24h, Ecole Centrale Supélec Paris, M2

Master: Adrian Kosowski, Randomization in Computer Science: Games, Networks, Epidemic and Evolutionary Algorithms, 18h, M1, École Polytechnique

Licence: Adrian Kosowski, Design and Analysis of Algorithms, 32h, L3, École Polytechnique

Master: Pierre Fraigniaud and Adrian Kosowski, Algorithmique distribuée pour les réseaux, 24h, M2, Master Parisien de Recherche en Informatique (MPRI)

Master: Fabien de Montgolfier and Michel Habib, Grand Réseaux d'Interaction, 44h, M2, Univ Paris Diderot

Licence: Fabien de Montgolfier, Protocoles Réseau (TP/TD), 24h, M1, Univ Paris Diderot

Licence: Fabien de Montgolfier, Programmation avancée (cours/TD/projet, bio-informatique), 52h, L3, Univ. Paris Diderot

Master: Fabien de Montgolfier, Algorithmique avancée (bio-informatique), 26h, M1, Univ Paris Diderot

Licence: Fabien de Montgolfier, Algorithmique (TD), 26h, L3, Ecole d'Ingénieurs Denis Diderot

Master : Laurent Viennot, Graph Mining, 3h, M2 MPRI, Univ. Paris Diderot

Licence: Michel Habib, Algorithmique, 45h, L, ENS Cachan

Master: Michel Habib, Algorithmique avancée, 24h, M1, Univ. Paris Diderot

Master: Michel Habib, Mobilité, 33h, M2, Univ. Paris Diderot

Master: Michel Habib, Méthodes et algorithmes pour l'accès à l'information numérique, 16h, M2, Univ. Paris Diderot

Master: Michel Habib, Algorithmique de graphes, 12h, M2, Univ. Paris Diderot

Licence: Pierre Charbit, Introduction a la Programmation, 30h, L1, Université Paris Diderot, France

Licence: Pierre Charbit, Automates finis, 52h, L2, Université Paris Diderot, France

Licence: Pierre Charbit, Types de Données et Objet, 52h, L1, Université Paris Diderot, France

Master: Pierre Charbit, Programmation, 60h, M2Pro PISE, Université Paris Diderot, France

Master: Pierre Charbit, Algorithmique de Graphes, 12h, M2 MPRI, Université Paris Diderot, France

9.2.2. Supervision

PhD: Leonardo Linguaglossa (co-advised by Laurent Viennot, Fabien Mathieu and Diego Perino, both from Nokia Bell Labs) was a PhD hired by Inria through the ADR CCN contract. He obained his PhD last September [1] at Paris Diderot University.

PhD in progress: Simon Collet (co-advised by Amos Korman and Pierre Fraigniaud). Title of thesis is: "Algorithmic Game Theory Applied to Biology". Started September 2015.

PhD in progress: Lucas Boczkowski (co-advised by Amos Korman and Iordanis Kerenidis). Title of thesis is: "Computing with Limited Resources in Uncertain Environments". Started September 2015.

PhD in progress: Brieuc Guinard (advised by Amos Korman). Title of thesis is: "Algorithmic Aspects of Random Biological Processes". Started October 2016.

PhD in progress: Laurent Feuilloley (advised by Pierre Fraigniaud). Title of thesis is: "Synchronous Distributed Computing". Started September 2015.

PhD in progress: Mengchuan Zou (co-advised by Adrian Kosowski and Michel Habib). Title of thesis is: "Local and Adaptive Algorithms for Optimization Problems in Large Networks". Started October 2016.

PhD in progress: Finn Volkel (advised by Michel Habib). Title of Thesis: "Convexity in graphs", started september 2016.

PhD in progress: Léo Planche (co-advised by Étienne Birmelé and Fabien de Montgolfier). Title if thesis is : "Classification de collections de graphes". Started October 2015.

PhD in progress: Alkida Balliu and Dennis Olivetti (PhD students from L'Aquilla University and Gran Sasso Science Institute) are supervised by Pierre Fraigniaud.

PhD in progress: Lucas Hosseini (co-advised by Pierre Charbit, Patrice Ossona de Mendez and Jaroslav Nešetřil since Sept. 2014. Title : Limits of Structures.

9.2.3. Juries

Laurent Viennot was president of the jury in the PhD defense of Claudio Imbrenda on “Analysing Traffic Cacheability in the Access Network at Line Rate” at Telecom ParisTech (November 2016).

Michel Habib was reviewer for the PhD thesis of Matteo Seminaroti, “Combinatorial Algorithms for the seriation problem”, Tilburg University, Holland, december 2016.

Laurent Viennot was reviewer for the PhD thesis of Guillaume Ducoffe on “Proprie´te´s me´triques des grands graphes” at Côte d'Azur Univ. (December 2016).

Laurent Viennot was co-advisor for the PhD thesis of Leonardo Linguaglossa on “Two challenges of Software Networking: Name-based Forwarding and Table Verification” at Paris Diderot Univ. (September 2016).

Carole Delporte was reviewer for the PhD thesis of Claire Capdevielle on "Étude de la complexité des implémentations d'objets concurrents et sans attente, abandonnables ou solo-rapides" at Bordeaux Univ (November 2016).

9.3. Popularization

- Laurent Viennot is “commissaire d’exposition” for the permanent exposition on “Informatique et sciences du numérique” at Palais de la déécouverte in Paris (opening in October 2017).
- An article centered around the eLife 2016 paper “A locally-blazed ant trail achieves efficient collective navigation despite limited information” [14], whose co-authors include Amos Korman (co-corresponding author), Lucas Boczkowski, and Adrian Kosowski, appeared on the Israeli daily newspaper “Haaretz”, Dec 2016.
- The team has made contributions to the Encyclopedia of Algorithms [44], [41].

10. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] L. LINGUAGLOSSA. *Two challenges of Software Networking: Name-based Forwarding and Table Verification*, Université Paris Diderot (Paris 7) Sorbonne Paris Cité, September 2016, https://tel.archives-ouvertes.fr/tel-01386788

Articles in International Peer-Reviewed Journals

[2] E. BAMPAS, L. GĄSIENIEC, N. HANUSSE, D. ILCINKAS, R. KLASING, A. KOSOWSKI, T. RADZIK. *Robustness of the Rotor–Router Mechanism*, in "Algorithmica", June 2016 [*DOI :* 10.1007/s00453-016-0179-Y], https://hal.inria.fr/hal-01416012

[3] M. BORASSI, P. CRESCENZI, M. HABIB. *Into the Square: On the Complexity of Some Quadratic-time Solvable Problems*, in "Electronic Notes in Theoretical Computer Science", 2016, vol. 322, pp. 51-67, https://hal.inria.fr/hal-01422030

[4] P. CHARBIT, I. PENEV, S. THOMASSÉ, N. TROTIGNON. *Perfect graphs of arbitrarily large clique-chromatic number*, in "Journal of Combinatorial Theory, Series B", January 2016, vol. 116, pp. 456-464 [*DOI :* 10.1016/J.JCTB.2015.09.008], https://hal.archives-ouvertes.fr/hal-01324052

[5] V. COHEN-ADDAD, M. HABIB, F. DE MONTGOLFIER. *Algorithmic aspects of switch cographs*, in "Discrete Applied Mathematics", 2016, vol. 200, pp. 23 - 42 [*DOI :* 10.1016/J.DAM.2015.07.008], https://hal.inria.fr/hal-01423053

[6] D. G. CORNEIL, J. DUSART, M. HABIB, E. KÖHLER. *On the power of graph searching for cocomparability graphs*, in "Siam Journal on Discrete Mathematics", 2016, https://hal.inria.fr/hal-01273687

[7] D. G. CORNEIL, J. DUSART, M. HABIB, A. MAMCARZ, F. DE MONTGOLFIER. *A tie-break model for graph search*, in "Discrete Applied Mathematics", January 2016, vol. 199, pp. 89–100 [*DOI :* 10.1016/J.DAM.2015.06.011], https://hal.inria.fr/hal-01255055

[8] P. CRESCENZI, P. FRAIGNIAUD, M. M. HALLDORSSON, H. HARUTYUNYAN, C. PIERUCCI, A. PIETRACAPRINA, G. PUCCI. *On the complexity of the shortest-path broadcast problem*, in "Discrete Applied Mathematics", 2016, vol. 199, pp. 101-109 [*DOI :* 10.1016/J.DAM.2015.05.004], https://hal.inria.fr/hal-01255198

[9] J. CZYZOWICZ, D. DERENIOWSKI, L. GĄSIENIEC, R. KLASING, A. KOSOWSKI, D. PAJĄK. *Collision-Free Network Exploration*, in "Journal of Computer and System Sciences", December 2016 [*DOI :* 10.1016/J.JCSS.2016.11.008], https://hal.inria.fr/hal-01416026

[10] J. CZYZOWICZ, L. GĄSIENIEC, A. KOSOWSKI, E. KRANAKIS, D. KRIZANC, N. TALEB. *When Patrolmen Become Corrupted: Monitoring a Graph Using Faulty Mobile Robots*, in "Algorithmica", October 2016 [*DOI :* 10.1007/S00453-016-0233-9], https://hal.inria.fr/hal-01416010

[11] C. DELPORTE-GALLET, H. FAUCONNIER. *Asynchronous Consensus with Bounded Memory*, in "Lecture notes in computer science", May 2016, vol. 9944, 15 p. [*DOI :* 10.1007/978-3-319-46140-3_12], https://hal.inria.fr/hal-01416509

[12] D. DERENIOWSKI, A. KOSOWSKI, D. PAJĄK, P. UZNANSKI. *Bounds on the cover time of parallel rotor walks*, in "Journal of Computer and System Sciences", August 2016, vol. 82, n^o 5, pp. 802 - 816 [*DOI :* 10.1016/J.JCSS.2016.01.004], https://hal.archives-ouvertes.fr/hal-01415102

[13] J. DUSART, M. HABIB. *A new LBFS-based algorithm for cocomparability graph recognition*, in "Discrete Mathematics", 2016 [*DOI :* 10.1016/J.DAM.2015.07.016], https://hal.inria.fr/hal-01274023

[14] E. FONIO, Y. HEYMAN, L. BOCZKOWSKI, A. GELBLUM, A. KOSOWSKI, A. KORMAN, O. FEINERMAN. *A locally-blazed ant trail achieves efficient collective navigation despite limited information*, in "eLife", November 2016, vol. 2016;5:e20185 [*DOI :* 10.7554/ELIFE.20185], https://hal.inria.fr/hal-01413748

[15] P. FRAIGNIAUD, A. CLEMENTI, P. CRESCENZI, C. DOERR, F. PASQUALE, R. SILVESTRI. *Rumor spreading in random evolving graphs*, in "Random Structures and Algorithms", 2016, vol. 48, n^o 2, pp. 290-312, https://hal.inria.fr/hal-01423643

[16] P. FRAIGNIAUD, M. M. HALLDORSSON, B. PATT-SHAMIR, D. RAWITZ, A. ROSÉN. *Shrinking Maxima, Decreasing Costs: New Online Packing and Covering Problems*, in "Algorithmica", 2016, vol. 74, n^o 4, pp. 1205-1223, https://hal.inria.fr/hal-01423640

[17] P. FRAIGNIAUD, A. KORMAN. *An Optimal Ancestry Labeling Scheme with Applications to XML Trees and Universal Posets*, in "Journal of the ACM", 2016, vol. 63, pp. 1 - 31 [*DOI :* 10.1145/2794076], https://hal.inria.fr/hal-01393643

[18] R. KLASING, A. KOSOWSKI, D. PAJĄK, T. SAUERWALD. *The multi-agent rotor-router on the ring: a deterministic alternative to parallel random walks*, in "Distributed Computing", September 2016 [*DOI :* 10.1007/S00446-016-0282-Y], https://hal.inria.fr/hal-01416011

Invited Conferences

[19] A. KOSOWSKI. *What Makes a Distributed Problem Truly Local?*, in "SIROCCO 2016 - 23rd International Colloquium on Structural Information and Communication Complexity", Helsinki, Finland, J. SUOMELA (editor), Lecture Notes in Computer Science, Springer, July 2016, vol. 9988, 3 p. [*DOI :* 10.1007/978-3-319-48314-6], https://hal.inria.fr/hal-01415090

International Conferences with Proceedings

[20] É. BIRMELÉ, F. DE MONTGOLFIER, L. PLANCHE. *Minimum Eccentricity Shortest Path Problem: An Approximation Algorithm and Relation with the k-Laminarity Problem*, in "COCOA 2016, Combinatorial

Optimization and Applications - 10th International Conference", Hong Kong, China, December 2016, pp. 216 - 229 [*DOI :* 10.1007/978-3-319-48749-6_16], https://hal.inria.fr/hal-01424469

[21] A. CASTAÑEDA, C. DELPORTE-GALLET, H. FAUCONNIER, S. RAJSBAUM, M. RAYNAL. *Making Local Algorithms Wait-Free: The Case of Ring Coloring*, in "SSS", Lyon, France, Stabilization, Safety, and Security of Distributed Systems, November 2016, vol. 10083, 16 p. , https://hal.inria.fr/hal-01416522

[22] A. CASTAÑEDA, P. FRAIGNIAUD, E. GAFNI, S. RAJSBAUM, M. ROY. *Asynchronous Coordination Under Preferences and Constraints*, in "23rd International Colloquium on Structural Information and Communication Complexity", Helsinki, Finland, SIROCCO 2016 pre-proceedings, July 2016, https://hal.archives-ouvertes.fr/hal-01341710

[23] C. DELPORTE-GALLET, H. FAUCONNIER, E. GAFNI, P. KUZNETSOV. *Set-Consensus Collections are Decidable*, in "OPODIS", MADRID, Spain, LNCS, December 2016, https://hal.inria.fr/hal-01416981

[24] C. DELPORTE-GALLET, H. FAUCONNIER, M. RAYNAL, S. RAJSBAUM. *Implementing Snapshot Objects on Top of Crash-Prone Asynchronous Message-Passing Systems*, in "Algorithms and Architectures for Parallel Processing (ICA3PP)", Granada, France, Algorithms and Architectures for Parallel Processing Volume 10048 of the series Lecture Notes in Computer Science pp 341-355, Springer, December 2016, vol. 10048, 15 p. , https://hal.inria.fr/hal-01416498

[25] C. DELPORTE-GALLET, H. FAUCONNIER, M. RAYNAL, S. RAJSBAUM. *t-Resilient Immediate Snapshot Is Impossible*, in "SIROCCO", Helsiinki, France, Structural Information and Communication Complexity, July 2016, vol. 9988, 15 p. , https://hal.inria.fr/hal-01416515

[26] P. FRAIGNIAUD, B. ALKIDA, Z. LOTKER, O. DENNIS. *Sparsifying Congested Cliques and Core-Periphery Networks*, in "23rd International Colloquium on Structural Information and Communication Complexity (SIROCCO)", Helsinki, Finland, 2016, https://hal.inria.fr/hal-01423688

[27] P. FRAIGNIAUD, B. BONAKDARPOUR, S. RAJSBAUM, D. ROSENBLUETH, C. TRAVERS. *Decentralized Asynchronous Crash-Resilient Runtime Verification*, in "27th International Conference on Concurrency Theory (CONCUR)", Québec, Canada, 2016, https://hal.inria.fr/hal-01423646

[28] P. FRAIGNIAUD, P. CRESCENZI, Z. LOTKER, P. PENNA. *Core-periphery clustering and collaboration networks*, in "IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)", San Francisco, United States, 2016, https://hal.inria.fr/hal-01423645

[29] P. FRAIGNIAUD, L. FEUILLOLEY, J. HIRVONEN. *A Hierarchy of Local Decision*, in "43rd International Colloquium on Automata, Languages, and Programming (ICALP)", Roma, Italy, 2016, https://hal.inria.fr/hal-01423644

[30] P. FRAIGNIAUD, M. HEINRICH, A. KOSOWSKI. *Local Conflict Coloring*, in "FOCS 2016 - 57th Annual IEEE Symposium on Foundations of Computer Science", New Brunswick, NJ, United States, October 2016, https://hal.inria.fr/hal-01223494

[31] P. FRAIGNIAUD, A. KORMAN, R. YOAV. *Parallel exhaustive search without coordination*, in "48th ACM Symposium on Theory of Computing (STOC)", Cambridge, MA, USA, France, 2016, https://hal.inria.fr/hal-01423634

[32] P. FRAIGNIAUD, E. NATALE. *Noisy Rumor Spreading and Plurality Consensus*, in "ACM Symposium on Principles of Distributed Computing (PODC)", Chicago, United States, 2016, https://hal.inria.fr/hal-01423686

[33] P. FRAIGNIAUD, S. RAJSBAUM, C. TRAVERS. *Minimizing the Number of Opinions for Fault-Tolerant Distributed Decision Using Well-Quasi Orderings*, in "12th Latin American Symposium on Theoretical Informatics (LATIN)", Ensenada, Mexico, 2016, https://hal.inria.fr/hal-01423679

[34] P. FRAIGNIAUD, S. RAJSBAUM, C. TRAVERS, P. KUZNETSOV, T. RIEUTORD. *Perfect Failure Detection with Very Few Bits*, in "18th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)", Lyon, France, 2016, https://hal.inria.fr/hal-01423637

[35] P. FRAIGNIAUD, I. RAPAPORT, V. SALO, I. TODINCA. *Distributed Testing of Excluded Subgraphs*, in "30th International Symposium on Distributed Computing (DISC 2016)", Paris, France, C. GAVOILLE, D. ILCINKAS (editors), Lecture Notes in Computer Science, Springer, September 2016, vol. LNCS 9888, pp. 342 - 356 [*DOI :* 10.1007/978-3-662-53426-7_25], https://hal.inria.fr/hal-01423633

[36] P. GAWRYCHOWSKI, A. KOSOWSKI, P. UZNANSKI. *Sublinear-Space Distance Labeling Using Hubs*, in "DISC 2016 - 30th International Symposium on DIStributed Computing", Paris, France, C. GAVOILLE, D. ILCINKAS (editors), Distributed Computing, Springer, September 2016, vol. 9888, pp. 230-242 [*DOI :* 10.1007/978-3-662-53426-7_17], https://hal.inria.fr/hal-01415064

[37] R. KLASING, A. KOSOWSKI, D. PAJĄK. *Setting Ports in an Anonymous Network: How to Reduce the Level of Symmetry?*, in "SIROCCO 2016 - 23rd International Colloquium on Structural Information and Communication Complexity", Helsinki, Finland, J. SUOMELA (editor), Structural Information and Communication Complexity, Springer, July 2016, vol. 9988, pp. 35-48 [*DOI :* 10.1007/978-3-319-48314-6_3], https://hal.inria.fr/hal-01415079

[38] A. KOSOWSKI, L. VIENNOT. *Beyond Highway Dimension: Small Distance Labels Using Tree Skeletons*, in "SODA 2017 - 28th ACM-SIAM Symposium on Discrete Algorithms", Barcelona, Spain, January 2017, https://hal.inria.fr/hal-01359084

Conferences without Proceedings

[39] P. GAWRYCHOWSKI, A. KOSOWSKI, P. UZNANSKI. *Brief Announcement: Sublinear-Space Distance Labeling Using Hubs*, in "PODC 2016 - 35th ACM Symposium on Principles of Distributed Computing", Chicago, IL, United States, July 2016, pp. 43-45 [*DOI :* 10.1145/2933057.2933077], https://hal.inria.fr/hal-01415072

Scientific Books (or Scientific Book chapters)

[40] C. DELPORTE-GALLET, P. A. ABDULLA (editors). *Networked Systems - 4th International Conference, NETYS 2016, Marrakech, Morocco, May 18-20, 2016, Revised Selected Papers*, LNCS, May 2016, vol. 9944, https://hal.inria.fr/hal-01416966

[41] P. FRAIGNIAUD. *Locality in Distributed Graph Algorithms*, in "Encyclopedia of Algorithms", Springer, 2016, pp. 1143-1148 [*DOI :* 10.1007/978-1-4939-2864-4_608], https://hal.inria.fr/hal-01423632

Research Reports

[42] Y. BOUFKHAD, R. DE LA PAZ, L. LINGUAGLOSSA, F. MATHIEU, D. PERINO, L. VIENNOT. *Forwarding Tables Verification through Representative Header Sets*, Inria Paris Rocquencourt, January 2016, https://hal.inria.fr/hal-01262130

[43] P. PENNA, L. VIENNOT. *Independent lazy better-response dynamics on network games*, ETH Zurich ; Inria ; Universite Paris Diderot-Paris VII, September 2016, https://hal.inria.fr/hal-01373411

Scientific Popularization

[44] D. G. CORNEIL, M. HABIB. *Unified View of Graph Searching and LDFS-Based Certifying Algorithms*, in "Encyclopedia of Algorithms", Springer-verlag, 2016 [*DOI :* 10.1007/978-3-642-27848-8_685-1], https://hal.inria.fr/hal-01255083

Other Publications

[45] A. CASTAÑEDA, C. DELPORTE, H. FAUCONNIER, S. RAJSBAUM, M. RAYNAL. *Wait-freedom and Locality are not Incompatible (with Distributed Ring Coloring as an Example)*, January 2016, working paper or preprint, https://hal.inria.fr/hal-01265958

[46] C. DELPORTE, H. FAUCONNIER, S. RAJSBAUM, M. RAYNAL. *Implementing Snapshot Objects on Top of Crash-Prone Asynchronous Message-Passing Systems*, May 2016, working paper or preprint, https://hal.inria.fr/hal-01313584

[47] C. DELPORTE, H. FAUCONNIER, S. RAJSBAUM, M. RAYNAL. *t-Resilient Immediate Snapshot is Impossible*, May 2016, working paper or preprint, https://hal.inria.fr/hal-01315342

[48] J. DUSART, M. HABIB, D. G. CORNEIL. *Maximal cliques structure for cocomparability graphs and applications*, November 2016, Il s'agit d'une recherche sur les relations entre les graphes d'intervalles et les graphes de cocomparabilité, https://hal.inria.fr/hal-01393927

[49] L. FEUILLOLEY, P. FRAIGNIAUD. *Survey of Distributed Decision*, June 2016, working paper or preprint, https://hal.archives-ouvertes.fr/hal-01331880

[50] F. VÖLKEL, E. BAPTESTE, M. HABIB, P. LOPEZ, C. VIGLIOTTI. *Read networks and k-laminar graphs*, March 2016, working paper or preprint, https://hal.inria.fr/hal-01282715